PLC IMPLEMENTATION OF A DES SUPERVISOR FOR A MANUFACTURING TESTBED: AN IMPLEMENTATION PERSPECTIVE

by

Ryan James Leduc

A thesis submitted in conformity with the requirements
for the Degree of Master of Applied Science
Graduate Department of Computer and Electrical Engineering
University of Toronto

Title: *PLC Implementation of A DES Supervisor for a Manufacturing Testbed: An Implementation Perspective*

Name: Ryan James Leduc

Degree: Master of Applied Science

Year of Convocation: 1996

Department: Electrical and Computer Engineering, University of Toronto

Abstract

The goal of this work is to investigate issues involved in modeling and designing supervisors for large, real systems, and issues involved in implementing DES supervisors on programmable logic controllers (PLC). A PLC based manufacturing testbed has been created to help investigate these issues.

A detailed plant model for the testbed was created (size on the order of 10^{16} states) and 29 modular supervisors were designed to implement the control specifications. Several model reduction theorems were created to handle a plant this large.

A generic implementation method was created, translating supervisors into clocked Moore synchronous state machines (CMSSM). The CMSSM were then implemented as relay ladder logic programs on the PLC. The testbed supervisors were implemented using the algorithm, and the testbed is now operational. This demonstrates that the algorithm works.

Acknowledgments

I would like to gratefully acknowledge the support of all my friends and family. Without their help, support and interest, I would never have made it through this project sane. I would like to especially thank my Mother, Sheila Das, Sharon Shew, Alireza Langari, Barry and Elizabeth Rael, Steven Postma, and Alan Lynch. I also want to thank my friends Dr. Kai Wong, Mark Lawford, Tom Chun, and Shahin Hashtrudi-Zad for patiently listening and offering criticism about my ideas and theories. Special thanks to my friend Christos Derventzis for his help in improving my work environment. Without his aid, I would never have finished in time.

Many thanks to Dr. Bertil Brandin for his support and guidance. I would like to acknowledge that the original idea of using model trains to simulate a manufacturing cell was Dr. Brandin's, as well as the layout of the tracks. Also, it was Dr. Brandin's suggestion to find an easy way to implement supervisors modularly.

I would also like to gratefully thank Prof W.M. Wonham for his guidance, support, and patience; for believing in this project and making it possible.

Many thanks to Joyce Larcher and everyone in the purchasing and shipping departments for their help in acquiring the material to build the testbed. Thanks also to the staff of John's Photo and Hobby for their advice on Märklin digital trains. Thanks to Allen Bradley (Canada) for providing a generous educational discount on the purchase of the testbed's PLC. Thanks also to the Department of Electrical and Computer Engineering for their aide in acquiring the PLC. Finally, thanks to Henry Ching for his help in entering the circuit schematics for the testbed, as well as other tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goal of this work is to investigate the following two matters. The first area of investigation is the issues involved in modeling and designing supervisors for large, real systems. The second area to be examined, and also the primary focus of the work, is the issues involved in converting a theoretical discrete event system (DES) supervisor to a physical implementation on programmable logic controllers (PLC). These two areas of investigation are discussed in the following sections. Some of the information presented in this thesis is also available in conference papers [16] and [17].

1.1 Issues for Large, Real Systems

When working with large, real systems several problems quickly arise. The first is "How does the DES plant model actually correspond to the real plant?" "How do events map to actual occurrences in the plant, and in particular, how do controllable events and their disablement actually translate to the real plant?" The second problem is the inhibiting effect of the combinatorial explosion for complex plants. For realistic plants, the plant model can quickly become quite large (easily greater than 10^{16} states) and thus exceed the capability of current software. New methods to handle large models must be developed.

1.2 Implementation Issues

Using supervisory control, controllers can be designed for a given system that are mathematically guaranteed always to be controllable and never deadlock. The question that quickly arises is, “What does a physical implementation of a theoretical supervisor look like?” An even more important question is, “How can one easily translate a supervisor into a physical controller such that the above qualities are not lost?” The above issues have been investigated for specific applications by Brandin ([8], [10]); here we propose a general approach.

To aid in the understanding of implementation issues, a PLC based testbed that physically simulates a manufacturing workcell was built. The design of the testbed will be discussed in detail along with the modeling of the testbed, and the design and implementation of its controllers. One of the goals of this work is to fully document the testbed so that it can be easily understood, used, repaired or re-created by others.

Chapter 2

Problem Definition and Background

2.1 Problem Definition

The primary focus of this thesis is the implementation of DES supervisors (designed using RW theory) on programmable logic controllers (PLC); to demonstrate that RW-theory can be used as a design tool for real complex systems, and to create a general implementation algorithm. To further this purpose, a manufacturing testbed was built. The objective the testbed fulfills as well as an overview of the proposed implementation method will be discussed in the following sections.

2.1.1 Purpose of Testbed

The purpose of the testbed was to provide a complex, real, PLC-based plant for testing RW-theory on. The testbed was designed to simulate a manufacturing plant, providing similar control issues as an actual plant but at a much smaller price and occupying less space. The testbed was also designed to be modular, flexible, and easily expandable. By modeling the testbed, designing DES supervisors, and implementing the supervisors on a PLC, many of the issues necessary for implementation will be explored. This information and insight will be used to create a general implementation algorithm.

The testbed is also part of the *Ontario/Rhône-Alpes Project on the Supervisory Control of Automated Manufacturing Systems* ([9]). With respect to this project, the testbed's purpose is to investigate supervisory control of transfer systems in automated manufacturing systems. The Ontario/Rhône-Alpes Project was a collaborative effort between the Ontario Government and the Rhône-Alpes Laboratories, and their associated industrial partners. The goal of the project was to investigate the implementation of supervisory control applications in industry, and to apply performance evaluation methods to automated manufacturing systems that are controlled by DES supervisors. The goals of this project emphasizes the importance and relevance of a general implementation method for DES supervisors.

The testbed has been named Prometheus[1] ([20]) to symbolize supervisory control's potential benefit to humanity. The author believes that RW theory contains the inherent capacity to be able to allow people to easily and safely control and use very large complex systems, with the subsequent inherent advantages. The testbed symbolizes one of many steps towards this goal.

2.1.2 Overview of Implementation Method

The implementation process consists of three main steps, shown in the flow diagram in Figure 2.1. In the first step, the control specifications for the plant are implemented as DES supervisors, using RW-theory. The supervisors are then translated into clocked Moore synchronous state machines. The CMSSM are used to represent the supervisors because they are a natural, intuitive representation of a RW supervisor, and because a CMSSM can easily be implemented on most digital logic devices. Thus, the implementation method can easily be modified to implement supervisors on other digital logic devices, besides programmable logic controllers.

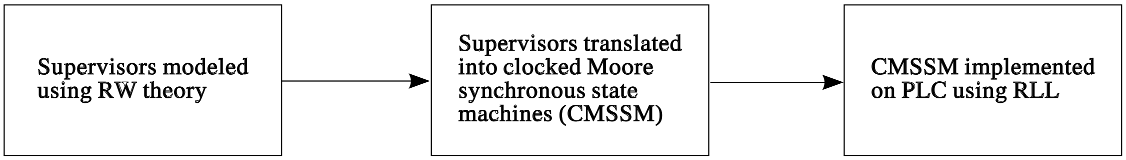

Supervisors modeled using RW theory
Supervisors translated into clocked Moore synchronous state machines (CMSSM)
CMSSM implemented on PLC using RLL

Figure 2.1: Implementation Overview

In the final step of the process, the boolean logic that defines the CMSSM will be

[1] A Titan who stole fire from Olympus to give to mankind.

implemented on the PLC as Relay Ladder Logic (RLL) programs. The PLC was chosen as the implementation device because of its robustness and its wide use in industry.

2.2 Supervisory Control Theory

Ramadge-Wonham supervisory control (RW theory:[28], [32], and [33]) provides a theoretical framework for the control of systems that are discrete in space and time. These systems are modeled as automata that generate a formal language of discrete events. These systems are customarily referred to as discrete event systems (DES). The DES are event-driven and may be non-deterministic[2]. The DES do not model when or why an event occurs, just the possible strings of events that the plant can generate. The events are considered to occur in an interleaving fashion.

2.2.1 Generators

The DES automaton is represented as a 5-tuple as shown below.

$$\mathbf{G} = (Y, \Sigma, \eta, y_o, Y_m), \quad \Sigma = \Sigma_u \cup \Sigma_c$$

where Y is the non-empty state set; Σ is the set of events (also referred to as the alphabet) containing both uncontrollable and controllable (can be disabled and thus prevented from occurring) events; η is the transition function; y_o is the initial state ($y_o \in Y$) and Y_m is the set of marked states ($Y_m \subseteq Y$).

The transition function $\eta : Y \,\mathrm{x}\, \Sigma^* \rightarrow Y$ is a partial function and is only defined for a subset of Σ at a given $y \in Y$. The notation $\eta(y, \sigma)!$ indicates that η is defined for σ at state y. In the above definition, $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$ where ϵ is the empty string and Σ^+ is the set of all sequences of symbols $\sigma_1\sigma_2\sigma_3 \ldots \sigma_k$, $k \geq 1$ and $\sigma_i \in \Sigma$, $i = 1, 2, \ldots, k$.

For the plant $\mathbf{G}$, the language generated by the automaton is denoted by $L(\mathbf{G})$ and is called the closed behavior of $\mathbf{G}$. For the plant $\mathbf{G}$,

$$s \in L(\mathbf{G}) \leftrightarrow s \in \Sigma^* \text{ and } \eta(y_o, s)!$$

[2] Capable of choosing between two possible next states by chance or unmodeled system dynamics.

The marked behavior of $\mathbf{G}$, $L_m(\mathbf{G})$, is defined:

$$L_m(\mathbf{G}) \subseteq L(\mathbf{G}) \text{ where } s \in L_m(\mathbf{G}) \leftrightarrow \eta(y_o, s) \in Y_m$$

A DES $\mathbf{G}$ is said to be "non-blocking" or "trim" if

$$\overline{L_m}(\mathbf{G}) = L(\mathbf{G})$$

where $\overline{L_m}(\mathbf{G})$ is the prefix closure (defined below) of $L_m(\mathbf{G})$. Non-blocking means that every string in $L(\mathbf{G})$ can be completed to a valid marked string, thus the DES can always return to a marked state.

A string $t \in \Sigma^*$ is a prefix of $s \in \Sigma^*$ if $s = tu$, for some $u \in \Sigma^*$. The relation "t is a prefix of s" is expressed as $t \leq s$. The prefix closure of a language $L \subseteq \Sigma^*$ is defined as $\overline{L} = \{t \in \Sigma^* \mid t \leq s \text{ for some } s \in L\}$.

An example of DES plant is given in Figure 2.2. Here, the plant is composed of two automata, **mach1** and **mach2**. The composite plant model is obtained by taking the synchronous product (defined below) of **mach1** and **mach2**. In the diagram, the double arrow attached to state 0 indicates that this is the initial state and that the state is marked. Events with a bar across them (such as α_1 and α_2) indicate that they are controllable events.

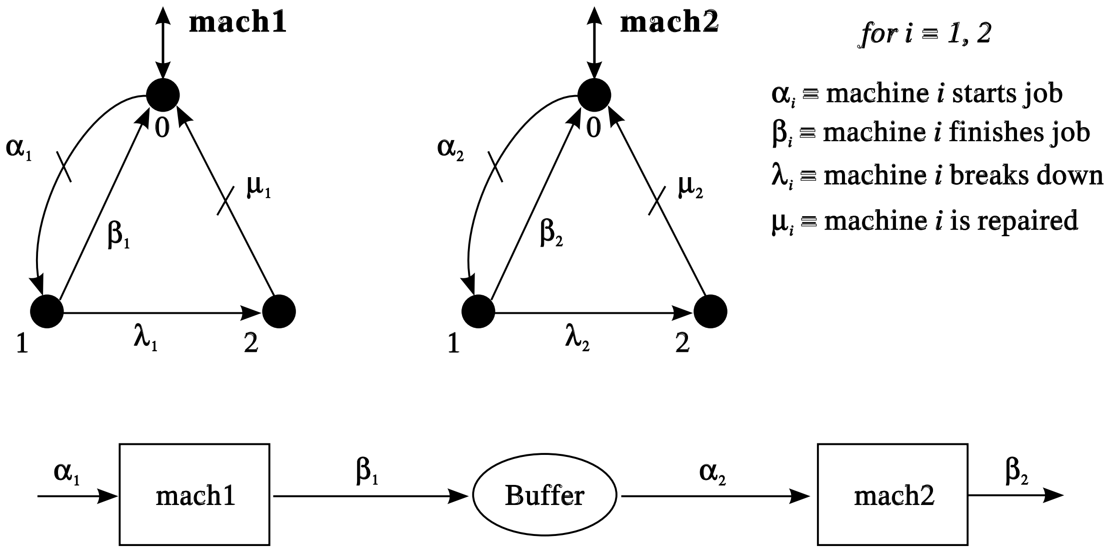

mach1
0
1
2
α1
β1
λ1
μ1
mach2
0
1
2
α2
β2
λ2
μ2
for i ≡ 1, 2
αi ≡ machine i starts job
βi ≡ machine i finishes job
λi ≡ machine i breaks down
μi ≡ machine i is repaired
α1
mach1
β1
Buffer
α2
mach2
β2

Figure 2.2: Simple Factory Example

2.2.2 Operations

Four useful operations for languages and automata are the catenation of strings (**cat**), the natural projection (**P**), the synchronous product of two languages, and the meet of two DES (**meet**). The catenation of strings is defined as follows: **cat** : Σ^* x $\Sigma^* \rightarrow \Sigma^*$ where:

$$\mathbf{cat}(\epsilon, s) = \mathbf{cat}(s, \epsilon) = s, \quad s \in \Sigma^*$$

$$\mathbf{cat}(s,t) = st \quad s, t \in \Sigma^+$$

The natural projection is defined with respect to the subset of a larger alphabet. Let $\Sigma_o \subseteq \Sigma^*$. Define the natural projection of Σ^* onto Σ_o^* as follows:

$$\mathbf{P}_o : \Sigma^* \rightarrow \Sigma_o^* \quad \text{where,}$$

$$\begin{aligned} \mathbf{P}_o(\epsilon) &= \epsilon && (2.1) \\ \mathbf{P}_o(\sigma) &= \begin{cases} \epsilon \text{ if } \sigma \notin \Sigma_o \\ \sigma \text{ if } \sigma \in \Sigma_o \end{cases} && (2.2) \\ \mathbf{P}_o(s\sigma) &= \mathbf{P}_o(s)\mathbf{P}_o(\sigma) \quad \text{where } s \in \Sigma^*, \sigma \in \Sigma && (2.3) \end{aligned}$$

The synchronous product of two languages $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ ($\Sigma = \Sigma_1 \cup \Sigma_2$) is defined using the natural projection. Let $\mathbf{P}_i$ be the natural projection of Σ^* onto Σ_i^*, $i = 1, 2$. The synchronous product $L_1||_s L_2$ is defined to be:

$$L_1||_s L_2 = \mathbf{P}_1^{-1} L_1 \cap \mathbf{P}_2^{-1} L_2$$

where $\mathbf{P}_i^{-1}$, $i = 1, 2$ is the inverse image map of $\mathbf{P}_i$. To perform the **sync** operation on two DES $\mathbf{G_1}$ and $\mathbf{G_2}$, gives $\mathbf{G} = \mathbf{sync}(\mathbf{G_1}, \mathbf{G_2})$ where

$$L_m(\mathbf{G}) = L_m(\mathbf{G_1})||_s L_m(\mathbf{G_2}), \;\; L(\mathbf{G}) = L(\mathbf{G_1})||_s L(\mathbf{G_2})$$

The meet of two DES $\mathbf{G_1}$ and $\mathbf{G_2}$ gives the reachable DES, $\mathbf{G} = \mathbf{meet}(\mathbf{G_1}, \mathbf{G_2})$, where

$$L_m(\mathbf{G}) = L_m(\mathbf{G_1}) \cap L_m(\mathbf{G_2}), \;\; L(\mathbf{G}) = L(\mathbf{G_1}) \cap L(\mathbf{G_2})$$

A DES $\mathbf{G}$ is reachable if every state is reachable from the initial state, by a string in $L(\mathbf{G})$.

2.2.3 Supervisors

Supervisors monitor the events generated by the plant, and disable events according to some control law. The supervisors are represented as automata and defined as below:

$$\mathbf{S} = (X, \Sigma, \xi, x_o, X_m)$$

For $\sigma \in \Sigma_c$, an event is enabled (and thus can occur) at any given $x \in X$ for $\mathbf{S}$, only if $\xi(x, \sigma)!$. We denote the closed loop behavior of $\mathbf{G}$ under $\mathbf{S}$ as:

$$\mathbf{S/G} = (X \ \times \ Y, \Sigma, \xi \ \times \ \eta, (x_o, yo), X_m \ \times \ Y_m) = \mathbf{meet}(\mathbf{S}, \mathbf{G})$$

A supervisor $\mathbf{S}$ is controllable for plant $\mathbf{G}$ if it does not require the disablement of an uncontrollable event to prevent $\mathbf{G}$ from generating an undesired event. Formally, the language $L(\mathbf{S})$ is controllable for $\mathbf{G}$ if the following is true:

$$\overline{L}(\mathbf{S})\Sigma_u \cap L(\mathbf{G}) \subseteq \overline{L}(\mathbf{S})$$

Modular supervisors are implemented by taking the conjunction of two or more supervisors. The conjunction of two supervisors $\mathbf{S_1}$ and $\mathbf{S_2}$ (expressed as $\mathbf{S_1} \wedge \mathbf{S_2}$) is defined to be:

$$\mathbf{S_1} \wedge \mathbf{S_2} = \text{meet}(\mathbf{S_1}, \mathbf{S_2})$$

The supervisor $\mathbf{S_1} \wedge \mathbf{S_2}$ is taken to be the supervisor as above and everything else is the same.

2.3 TCT

TCT is a software program that has been developed by Professor W.M. Wonham for the analysis, verification, and synthesis of supervisory control. To explain the functionality of TCT, the file tctinfo.txt is reproduced with permission of the author. The file *tctinfo* is taken from [33]. The license to include the file is given below.

I, **Professor W.M. Wonham**, grant permission for the textfile *tctinfo* to be included in the M.A.Sc thesis entitled *"PLC Implementation of a DES Supervisor for a Manufacturing Testbed: An Implementation Perspective,"* and grant the National Library of Canada a non-exclusive liscense to reproduce, loan, distribute or sell copies of the text of the file, as included in the thesis, by any means and in any format.

Signature	Print Name	Date

2.3.1 General Information

TCT is a program for the synthesis of supervisory controls for discrete-event systems. Generators and recognizers are represented as standard DES in the form of a 5-tuple

[Size, Init, Mark, Voc, Tran]

Size is the number of states (the standard state set is 0,...,Size-1), Init is the initial state (always taken to be 0), Mark lists the marker states, Voc the vocal states, and Tran the transitions. A vocal state is a pair [I,V] representing positive integer output V at state I. A transition is a triple [I,E,J] representing a transition from the exit state I to the entrance state J and having event label E. E is an odd or even nonnegative integer, depending on whether the corresponding event is controllable or uncontrollable.

```
                event E
exit I o ------------------->o J entrance
```

All DES transition structures must be deterministic: distinct transitions from the same exit state must carry distinct labels.

2.3.2 Synthesis Procedures

Synthesis procedures currently available are the following:

CREATE: prompts the user to define a new discrete-event system (DES).

SELFLOOP: augments an existing DES by adjoining selfloops at each state with event labels in a LIST provided by the user.

TRIM: for **DES1** constructs the trim (reachable and coreachable) substructure **DES2**.

SYNC: forms the reachable synchronous product of **DES1** and **DES2** to create **DES3**. Not for use with vocalized DES.

MEET: forms the meet (reachable cartesian product) of **DES1** and **DES2** to create **DES3**. **DES3** need not be coreachable. Not for use with vocalized DES.

SUPCON: for a controlled generator **DES1**, forms a trim recognizer for the supremal controllable sublanguage of the marked ("legal") language generated by **DES2** to create **DES3**. This structure provides a proper supervisor for **DES1**. Not for use with vocalized DES.

MUTEX: forms **DES3** from the shuffle of **DES1** and **DES2**, by excluding state pairs specified in LIST = [I1,J1], [I2,J2],..., plus all state pairs from which LIST is reachable along an uncontrollable path; then taking the reachable substructure of the result. **DES3** is thus reachable and controllable, but need not be coreachable. For the corresponding control data, compute **DES** = **SYNC**(**DES1**, **DES2**), then DAT = **CONDAT**(**DES**, **DES3**). If **DES3** is trim, it provides a proper supervisor for the mutual exclusion problem; if not, a solution is **SUP** = **SUPCON**(**DES**,**DES3**). Not for use with vocalized DES.

CONDAT: returns control data DAT for the supervisor **DES2** of the controlled system **DES1**. If **DES2** represents a controllable language (with respect to **DES1**), as when **DES2** has been previously computed with **SUPCON**, then **CONDAT** will display the events that are to be disabled at each state of **DES2**. In general **CONDAT** can be used to test whether a given language **DES2** is controllable: just check that the disabled events tabled by **CONDAT** are themselves controllable (have odd-numbered labels). To SHOW DAT call **SA**. **CONDAT** is not for use with vocalized DES.

MINSTATE: reduces **DES1** to a minimal state transition structure **DES2** that generates the same closed and marked languages, and the same string mapping induced by vocalization (if any). **DES2** is reachable but not necessarily coreachable.

COMPLEMENT: for a generator **DES1** and a LIST of event labels, forms a generator **DES2** of the marked language complementary to the marked language of **DES1**, with respect to the extended alphabet comprising the event labels of **DES1** plus those in the auxiliary LIST. The closed behavior of **DES2** is all strings over the extended alphabet. The string mapping induced by vocalization (if any) is unchanged.

PROJECT: for a generator **DES1** and a LIST of event labels, forms a generator **DES2** of the closed and marked languages of **DES1** with the LISTed events erased. In decentralized control, **DES2** could be an observer's local model of **DES1**. Not for use with vocalized DES.

CONVERT: returns **DES2** corresponding to a specified mapping of event labels in **DES1**; unmapped labels are unchanged. Can be used with **PROJECT** to construct an arbitrary zero-memory output map having as domain the language represented by **DES1**. Not for use with vocalized DES.

VOCALIZE: returns a transition structure **DES2** having the same closed and marked behaviors as **DES1**, but with specified state outputs corresponding to selected state/event input pairs.

OUTCONSIS: returns a transition structure **DES2** having the same closed and marked behaviors as **DES1**, but which is output-consistent in the sense that nonzero state outputs are unambiguously controllable or uncontrollable. A vocal state with output V in 10...99 may be split into siblings with outputs respectively V1 or V0.

HICONSIS: returns a transition structure **DES2** having the same closed and marked behaviors as **DES1**, but hierarchically consistent in the sense that high-level controllable events may be disabled without side effects. This may require additional vocalization together with change in the control status of existing state outputs. This procedure incorporates and extends **OUTCONSIS**.

HIGEN: returns a transition structure **DES2** over the state-output alphabet of **DES1**, representing the closed and marked state-output (or 'high-level') behaviors of DES1. For instance, starting with a 'low-level' vocalized model **GLO**, the sequence

$$\mathbf{OCGLO} = \mathbf{outconsis}(\mathbf{GLO})$$
$$\mathbf{HCGLO} = \mathbf{hiconsis}(\mathbf{OCGLO})$$
$$\mathbf{HCGHI} = \mathbf{higen}(\mathbf{HCGLO})$$

returns a DES pair (**HCGLO**, **HCGHI**) that is hierarchically consistent: controllable languages in **HCGHI** can be synthesized, via the state-output map, as controllable languages in **HCGLO**.

SUPNORM: returns a trim transition structure **DES3** which represents the supremal sublanguage of the legal language **DES1**, that is normal with respect to the marked behavior of the plant generator **DES2** and the projection specified by NULL_EVENT_LIST (the list of unobservable events). Not for use with vocalized DES.

For supervisor synthesis, **project** **DES2** to get **PDES2**, and **DES3** to get **PDES3**, with respect to NULL_EVENT_LIST. The local supervisor is **DES4** = **supcon**(**PDES2**, **PDES3**). The global supervised behavior is represented by

$$\mathbf{DES5} = \mathbf{meet}(\mathbf{DES2}, \mathbf{selfloop}(\mathbf{DES4}, \mathit{NULL\_EVENT\_LIST}))$$

In general **DES5** may fail to be nonblocking: trim to check.

NONCONFLICT: tests whether **DES1**, **DES2** are nonconflicting, namely whether all reachable states of the product DES are coreachable. Not for use with vocalized DES.

ISOMORPH: tests whether **DES1** and **DES2** are identical up to renumbering of states; if so, their state correspondence is displayed.

2.3.3 Utilities

EDIT: allows the user to modify an existing DES.

SHOW: **SE** displays an existing DES, **SA** a DAT (condat) table, **SX** a TXT (text) file. Tables can be browsed with Page keys. MAKEIT.TXT keeps a record of user files as they are generated.

PRINT: **OE** (resp. **OA**, **OX**) directs DES (resp. DAT, TXT) files to the printer, in 80-column format. Epson printers are supported. Optionally, output can be sent to the current user subdirectory as an ASCII file with suffix PDS (resp. PDT, PXT).

USER FILE DIRECTORY: lists the current user subdirectory.

Chapter 3

Modeling Methods and Supervisor Design

This chapter deals with issues involved in modeling a real plant, including developing and interpreting plant models. The validity of the plant model is also analyzed. Finally, supervisor design is discussed.

3.1 Developing the Model

Important issues in developing plant models are how to reduce the modeling of a large plant into several small plant models, and how to avoid unnecessarily complicating the model.

3.1.1 Model Decomposition

It is important to model large plants as a group of smaller sub-models for the following reasons:

- For large systems like the testbed (plant model on the order of 10^{16} states) it would be extremely difficult to come up with a single model.
- It is more accurate to design several small models as opposed to keeping track of all the intricacies of one large model.
- It is quicker to design several small models than one large model.

The difficulty arises in deciding how to break down a large plant model into component models. A useful approach is to break the sub-models into two categories: fundamental sub-models, and interaction sub-models. The fundamental models are the models of the basic elements of the plant by themselves, ignoring how these elements interact with each other. For the testbed, the fundamental models are the trains, cranes, sensors and switches[1].

The interaction models describe how the fundamental models interact with each other. For the testbed, the interaction models are the switch request handlers, the sensor interdependencies, and the sensor dependencies on the trains, and switches. Breaking the plant model down into these categories is intuitive and clearly shows the components of the plant and how they relate to each other.

Often there will still be large models within these categories that require further decomposition into component models. For the testbed, this was the case with the interaction models for the sensors, trains, and switches. These three elements are intimately related. For a given train, sensors can only be reached in a certain order, starting from the train's initial position. Depending on the current location of the train, certain sensors may be unreachable if a switch is in the wrong position. To model this for the six switches and one train would require a plant model of over 300 states. Unfortunately, it was not immediately obvious how to reduce this model.

The trick is to remember that not every detail of the plant's behavior is required in every model, but every detail must be in at least one model. When the models are combined using the **sync** operation, the various components of the plants behavior are combined. This means that the order in which the sensors can be reached for a given train can be modeled while ignoring the effect of the switches. Then, a model can be created for each switch for a given train. This model only needs to show how the particular switch affects access to the sensors immediately before and after the switches. It is possible to compartmentalize the plant behavior this way, because the effect of the **sync** operation is to restrict the occurrence of common events until all sub-models that contain the event declare that the event can occur.

[1] Refer to Chapter 6 for a detailed description of the testbed plant models.

3.1.2 Avoiding Complicating the Model

It is important to avoid unnecessarily complicating the plant model. This is important to maintain clarity and to keep the model from getting too large. To keep the model simple, it is important to remember that not every detail needs to be shown. The occurrence of an event can often be broken down into several sub-events that represent various steps of the event taking place. If these individual steps are important for making decisions or for controlling the plant more accurately, then leave them in. If they are well represented by one event, then there is no sense in retaining the extra details. A possible exception is to allow for future requirements.

Another way to avoid complication is to remember that the plant model does not have to do everything. If you need to know a certain condition has been reached, the plant model doesn't necessarily have to track the events leading up to the event, especially if the determination of the condition is not well suited to DES modeling. The determination of the condition can be left to outside agents, and then the agents can generate the event signifying the condition has occurred. Possible agents are sensors or low level computer programs that simply monitor the plant to determine if the condition has been met.

A simple example of this situation is the need to determine when a tank being filled with water reaches a specific level. If the events *"1 unit added"* or *"1 unit removed"* were available, then a plant model could determine when the correct level was reached from knowledge of the initial level of the tank. This would be extremely inefficient. It would be better to add a sensor set to the correct level, or have a computer program monitor a digital readout of the level of the tank. The point is not to rely unnecessarily on the events you already have to supply you with information when outside sources can do so in a more compact form.

3.2 Model Validity

The most important thing about developing a model is that the model is valid for the physical plant. If this isn't the case, there is no point in verifying if a supervisor is controllable and non-blocking for the plant. The problem is that you have no guarantee. You could apply a formal method to verify the accuracy of your plant model, but this would require that you start with some assumptions about the plant. To verify these assumptions would require simpler assumptions. The point is that there is no guarantee that your plant model will be accurate, but this will be the case however you tackle the problem. When you formalize the problem with a plant model, you at least know that you can guarantee the accuracy of your work from this point. By modeling the plant as a group of smaller sub-models as described above, you are able to work at a level that you can reasonably come up with an accurate model.

The question that arises is, "What conditions does a model need to meet to make it valid for a given plant?" For a specific plant, there is an infinite number of possible models that could be created for the plant. For instance, Figure 3.1 shows two valid models for the trains used in the testbed. The first model shows the necessary information for the testbed. The second model is an expansion of the first model giving more details of the inner working of the train. This process could be continued indefinitely by expanding events. The question is, "Why is the first model valid for the train even though it lacks information?"

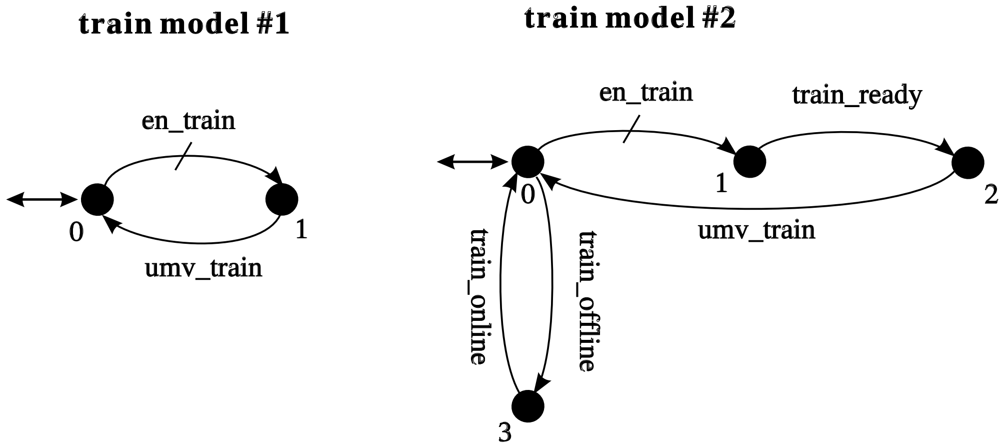


Figure 3.1: Example of an Expanded Model

When modeling a plant, the designer decides on the set of events (alphabet) that will be included in the model. The validity of the model is with respect to these events. Obviously, it is important to include the events required to properly observe and control the plant. With respect to these events, the plant model is valid if it correctly generates the sequence of events (strings) that the plant would. This means that events only occur in the plant when the plant model says that the event could occur. This means that the modeled order of events is correct and that all dependencies between events are included. By the latter statement, it is meant that if the model says event α can occur, then it can occur without having to wait for the occurrence of any other event in the alphabet.

3.3 Interpretation of the Model

When designing a model for a plant, the interpretation of what the model means is important. For plants that are fixed in their operation, the plant model simply describes their possible behavior. The machine modeled as **mach** in Figure 3.2 is an example of a fixed plant.

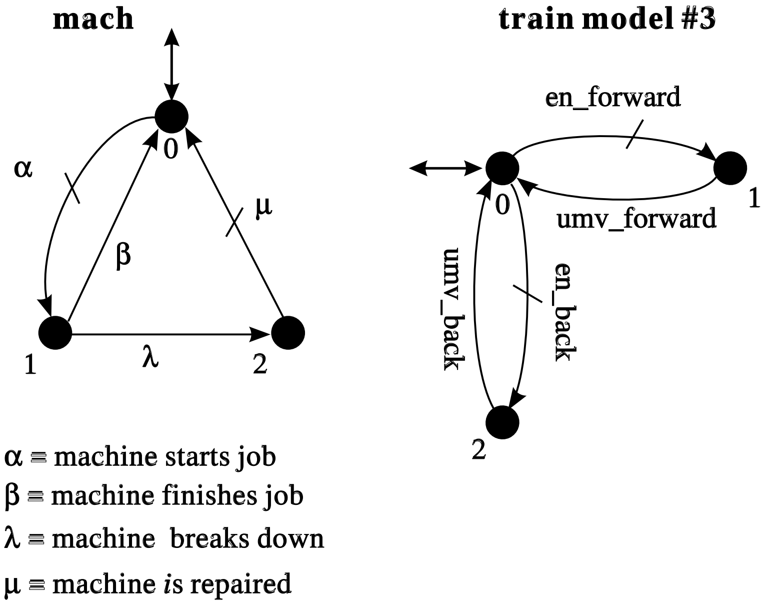


Figure 3.2: Interface Example

For programmable plants like the testbed, the plant model becomes an interface specification. The plant model for the programmable portion of the testbed specifies what events the supervisor can see and what events the supervisor can control. This creates an interface specification for the underlying software that physically controls the plant. For the plant model to be valid, the underlying software must provide a discrete interface that correctly matches the plant model so that the validity criteria in the above section are met for the plant under the control of the software. The DES supervisor is then designed to manipulate the discrete interface so that it meets its control specifications. Using the plant model as an interface specification effectively creates an abstraction layer that isolates the high level

DES supervisors from the low level hardware. The abstraction layer permits changes in the hardware or software without requiring changes in the DES supervisor. It is important to note that the interface must be physically implementable by the plant.

As an example, consider the first train model in Figure 3.1. This model allows the supervisor to start or stop the train (the forward direction is implied). An equally correct model (in the sense that the testbed could implement it) for the train is the third model in Figure 3.2. This model permits the supervisor to start or stop the train, and to move it in the forward or reverse direction. Again, the plant model simply specifies the interface for the low level software.

For a given programmable plant, there is in effect one large plant model that contains all the possible events that the plant could execute and how the events interact with one another. What a given interface does is select a portion of this large model to create the desired plant model. The interface essentially disables all these other possibilities, leaving only the desired model.

3.4 Designing Supervisors

When designing supervisors, it is imperative to remember that a supervisor enforces the specification given, not necessarily the specification intended. There is no guarantee that the specification used actually solves the intended problem. Designers must ensure that they understand the problem well and that they cast their solution correctly in terms of a DES control specification. This problem will exist in any other design method. At least when the solution is formally specified, you can guarantee that the supervisor you design meets the specification.

When designing DES supervisors, it is necessary to think in terms of event sequences. Since designers live in a world of space and time, they must remember that time is not represented in standard RW theory. The plant model doesn't say when an event can occur, only if it is able to occur at that point. Supervisors cannot be based on an event occurring within a given time, but on events occurring signifying that something about the plant has changed. Designers must find an event (or sequence of events) that signifies that a desired condition regarding the plant is true.

As an example, consider the track switches of the testbed. When a train is traversing a switch, the switch must not change position or the train will be derailed. Once the train starts crossing the switch, an obvious condition is that the train should be safely across the switch in 3 seconds. Besides not being expressible in standard RW, this condition would not take into account the speed of the train and whether or not the train stops in the middle of the switch. A better solution is to use the event that the train has reached the sensor on the other side of the switch. The fact that the event has occurred guarantees that the train has traversed the switch.

Another important issue in the design of supervisors is to resist the urge to self-loop uncontrollable events unnecessarily. If certain uncontrollable events should not occur at a given state in a supervisor, then do not self-loop them at this state. If they can actually occur, then the supervisor will fail the controllability test, warning you that there is a flaw in your logic (or for that matter, a flaw in your plant model). Obviously, uncontrollable events that are unimportant to the supervisor should be self-looped at every state.

Chapter 4

Description of Testbed

The testbed is designed to simulate a manufacturing workcell, in particular, the problems of routing and collision. The configuration of the testbed is similar to a portion of the product routing design of the Motorola Fusion Factory [30]. The testbed is described in detail in the following sections. A full listing of all components used is given in Appendix B.

4.1 Overview

The testbed is composed primarily of model railroad components. The tracks are laid out to resemble a set of three interacting work units. The trains simulate Automated Guided Vehicles (AGV) that provide and remove material to/from each manufacturing unit. Each unit has a small electric crane to simulate robots loading and unloading the AGV. Finally, the system contains several remote track switches to control the paths of the trains, plus sensors that detect the presence and identity of each train.

The testbed is controlled by two embedded MC68332 processors and an Allen-Bradley PLC. The microprocessors control the trains and cranes directly, plus they have a discrete interface for communication with the PLC. The PLC is responsible for the control of the overall system. The system also contains a Compaq PC that functions as a programming station for the embedded processors and the PLC.

The testbed contains several custom circuit boards. These include expansion boards (one for each processor) that provide interface circuitry, as well as sensor boards, and control circuitry for the remote track switches.

The overall layout of the testbed is shown in Figure 4.1. A more detailed look at the hardware present in each loop of the manufacturing cell is given in Figures 4.2, 4.3, and 4.4. The individual hardware elements will be discussed in the following sections.

4.2 Model Railroad Components

A large portion of the testbed consists of Märklin digital train equipment ([11] and [14]), HO scale. Model train equipment was chosen as the "hardware" to be controlled because it would be less expensive (compared to equipment for a real manufacturing plant) and would take up less space, yet would offer similar functionality and control problems as a real manufacturing plant. The equipment used will be discussed below, with the exception of the remote track switches which will be discussed in Section 4.6.

4.2.1 Electric Trains

The testbed contains two electric trains that consist of one engine (model 3665) and one gondola car (model 4465). At present, the gondola cars are not being used. The trains are AC powered, and they contain lights and a remote coupler each to release the gondola cars upon request.

The Märklin digital trains were chosen because they allow for remote commands to be sent individually to each train. Each train contains a digital circuit that accepts commands bearing the unique ID number of the train. Each train's speed and direction can be controlled without affecting the other train. Also, each train can be individually commanded to release its gondola car. The train can be commanded to go forward or backwards, and can select from speeds ranging from 0 (full stop) to 14. The actual speed of the train varies depending on track position and the individual train.

Individual remote control of the trains is accomplished via the design of the Märklin digital tracks. Normally, DC powered tracks have one track rail at ground potential, and the other rail at a varying DC value that controls the speed and direction of the train. Instead, the Märklin digital tracks have both rails at ground potential, plus they have a centre rail that carries a constant AC voltage. Each train internally regulates its speed depending on the remote commands addressed to it. These commands are digital signals superimposed on the AC powered centre rail.

Each train can be set to any ID number between 1 and 80. Currently, train one is set

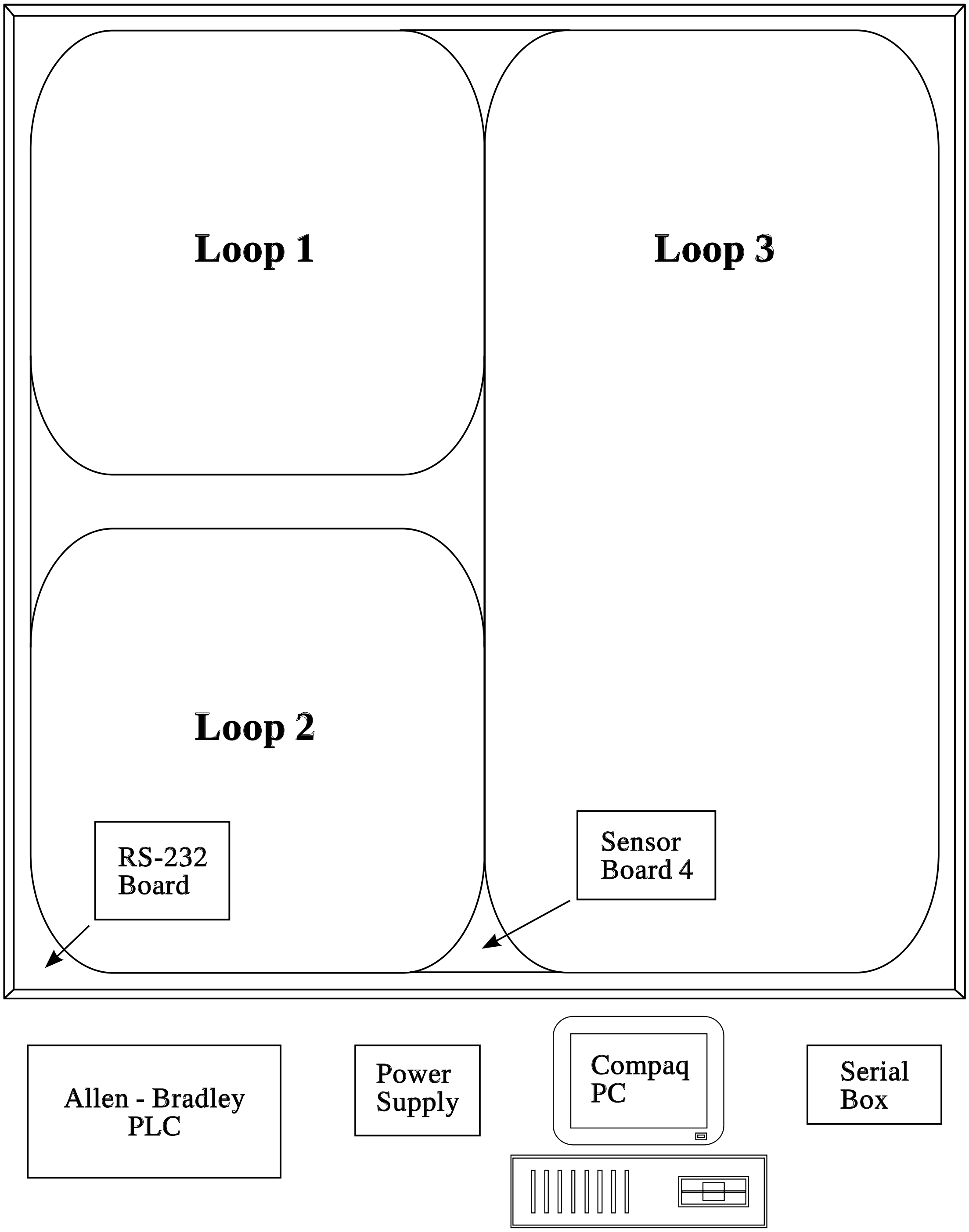


Figure 4.1: Layout of Testbed

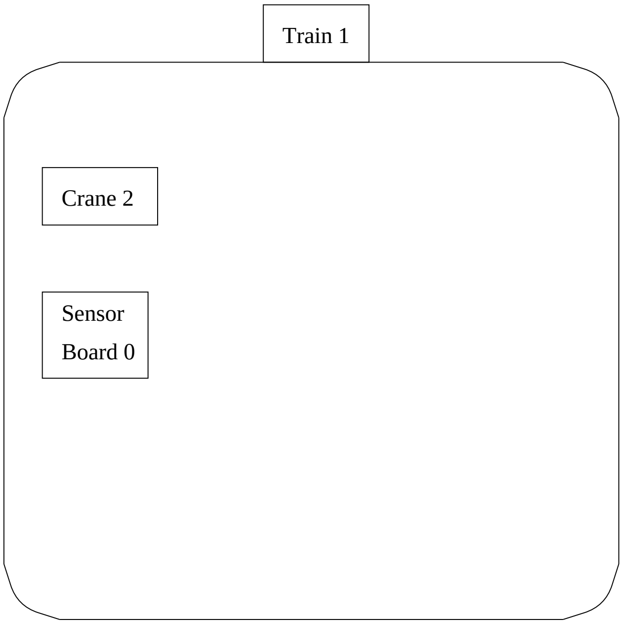


Figure 4.2: Loop 1 of Testbed Layout

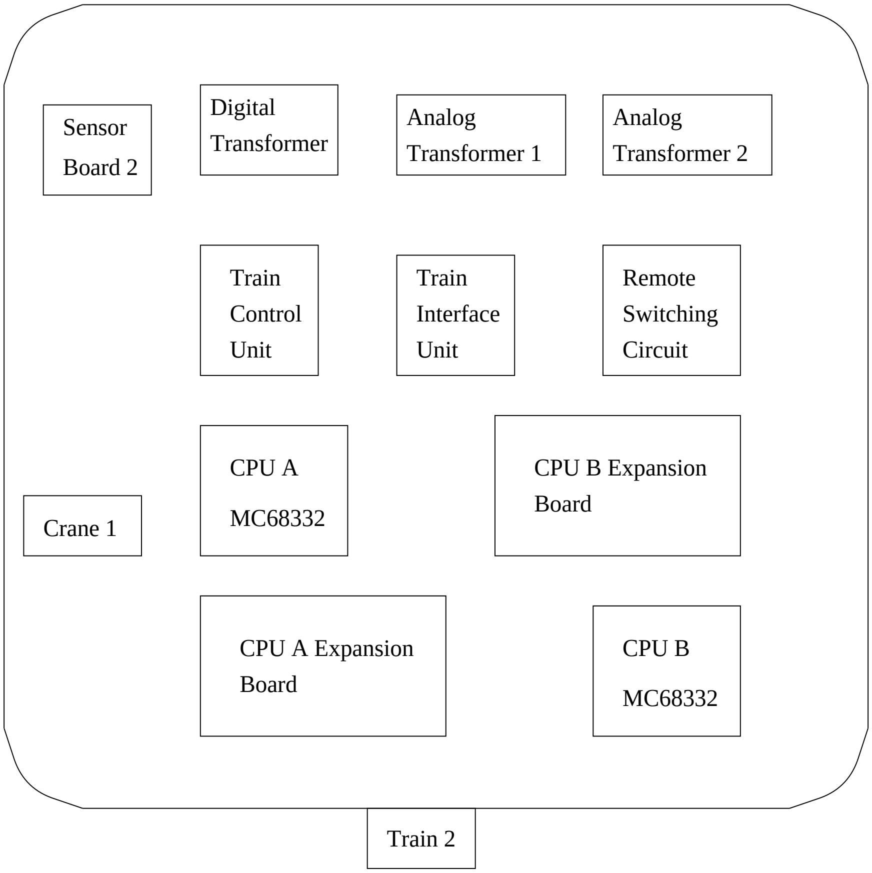


Figure 4.3: Loop 2 of Testbed Layout

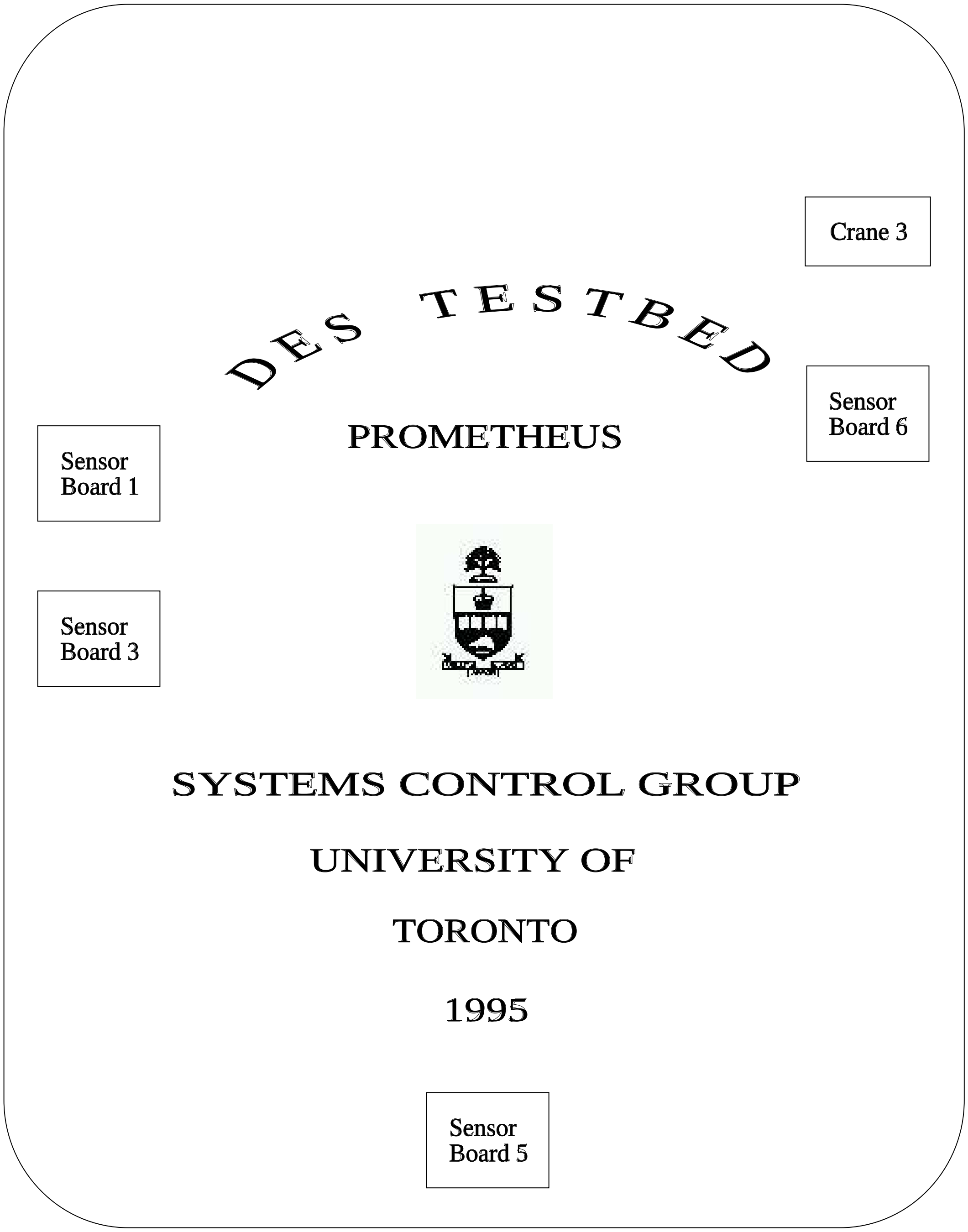


Figure 4.4: Loop 3 of Testbed Layout

to ID = 1, and train two is set to ID = 2. To set the train to a new ID, remove the top of the train, and set the dipswitches according to Table 4.1, where all indicated switches are set to ON, and all others are set to OFF.

Address	*Switch to ON*								*Address*	*Switch to ON*							
01	-	2	3	-	5	-	7	-	41	1	-	-	4	-	6	-	8
02	-	-	3	-	5	-	7	-	42	1	-	-	-	-	6	-	8
03	1	-	-	4	5	-	7	-	43	-	2	-	-	-	6	-	8
04	-	2	-	4	5	-	7	-	44	-	-	-	-	-	6	-	8
05	-	-	-	4	5	-	7	-	45	1	-	3	-	-	-	-	8
06	1	-	-	-	5	-	7	-	46	-	2	3	-	-	-	-	8
07	-	2	-	-	5	-	7	-	47	-	-	3	-	-	-	-	8
08	-	-	-	-	5	-	7	-	48	1	-	-	4	-	-	-	8
09	1	-	3	-	-	6	7	-	49	-	2	-	4	-	-	-	8
10	-	2	3	-	-	6	7	-	50	-	-	-	4	-	-	-	8
11	-	-	3	-	-	6	7	-	51	1	-	-	-	-	-	-	8
12	1	-	-	4	-	6	7	-	52	-	2	-	-	-	-	-	8
13	-	2	-	4	-	6	7	-	53	-	-	-	-	-	-	-	8
14	-	-	-	4	-	6	7	-	54	1	-	3	-	5	-	-	-
15	1	-	-	-	-	6	7	-	55	-	2	3	-	5	-	-	-
16	-	2	-	-	-	6	7	-	56	-	-	3	-	5	-	-	-
17	-	-	-	-	-	6	7	-	57	1	-	-	4	5	-	-	-
18	1	-	3	-	-	-	7	-	58	-	2	-	4	5	-	-	-
19	-	2	3	-	-	-	7	-	59	-	-	-	4	5	-	-	-
20	-	-	3	-	-	-	7	-	60	1	-	-	-	5	-	-	-
21	1	-	-	4	-	-	7	-	61	-	2	-	-	5	-	-	-
22	-	2	-	4	-	-	7	-	62	-	-	-	-	5	-	-	-
23	-	-	-	4	-	-	7	-	63	1	-	3	-	-	6	-	-
24	1	-	-	-	-	-	7	-	64	-	2	3	-	-	6	-	-
25	-	2	-	-	-	-	7	-	65	-	-	3	-	-	6	-	-
26	-	-	-	-	-	-	7	-	66	1	-	-	4	-	6	-	-
27	1	-	3	-	5	-	-	8	67	-	2	-	4	-	6	-	-
28	-	2	3	-	5	-	-	8	68	-	-	-	4	-	6	-	-
29	-	-	3	-	5	-	-	8	69	1	-	-	-	-	6	-	-
30	1	-	-	4	5	-	-	8	70	-	2	-	-	-	6	-	-
31	-	2	-	4	5	-	-	8	71	-	-	-	-	-	6	-	-
32	-	-	-	4	5	-	-	8	72	1	-	3	4	-	-	-	-
33	1	-	-	-	5	-	-	8	73	-	2	3	-	-	-	-	-
34	-	2	-	-	5	-	-	8	74	-	-	3	4	-	-	-	-
35	-	-	-	-	5	-	-	8	75	1	-	-	4	-	-	-	-
36	1	-	3	-	-	6	-	8	76	-	2	-	4	-	-	-	-
37	-	2	3	-	-	6	-	8	77	-	-	-	4	-	-	-	-
38	-	-	3	-	-	6	-	8	78	1	-	-	-	-	-	-	-
39	1	-	-	4	-	6	-	8	79	-	2	-	4	-	-	-	-
40	-	2	-	4	-	6	-	8	80	1	-	3	-	5	-	7	-

Table 4.1: Locomotive Address Decoder Settings

The Märklin system also contains an interface unit that allows a computer to send commands to the trains via an RS-232 interface. The RS-232 interface is accessed through port A of the expansion board of CPU A. The train interface unit is discussed in Section 4.2.3.

4.2.2 Electric Cranes

The testbed contains three Märklin model electric cranes (model 7051) which are used to load and unload the trains. A crane is capable of rotating left or right, and raising or lowering its electromagnet. The crane also has a small light in its cab that activates when the electromagnet is turned on. Each crane has two containers in front of it for selecting materials for loading onto trains.

Each crane is connected to one of the CPUs expansion boards via a six signal cable. Crane 1 is connected to expansion board A, while cranes 2 and 3 are connected to expansion board B. The pinout of the cables is discussed in Section 4.7.2. The cable consists of an AC ground (from one of the analog train transformers), a signal to the electromagnet and two paired signals to the crane's two AC motors for rotating the crane and raising/lowering the electromagnet. The electromagnet is activated when power is applied to its signal. The rotating motor is activated when power is applied to one of its signals. One signal corresponds to rotating to the left, and the other to the right. The raising/lowering motor is activated similarly. Each signal is attached to a relay on the appropriate expansion board. Refer to Section 5.3 for information on software control of the crane.

4.2.3 Support Hardware and Tracks

This section discusses the train tracks used and the hardware that supports the trains and cranes. They are discussed in the following sections.

Train Transformers

The testbed contains three Märklin AC transformers for powering the trains, cranes, and track switches. The digital transformer (model 6001) supplies AC power to the central control unit to power the trains. The first analog transformer (model 6667A) supplies power to crane 1 (via expansion board A) and the track switches, while the second transformer supplies power to cranes 2 and 3 (via expansion board B).

Central Control Unit

The Märklin central control unit (model 6021) is the device that controls the trains. The central unit supplies power and digital control signals to the tracks. It is capable of controlling up to 80 trains simultaneously, either from the controls on its face or through an interface unit connected to it. The central unit also contains "STOP" and "GO" buttons on its face. The "STOP" button cuts power to the tracks, and the "GO" button restores the power.

NOTE: If a train is selected by means of the controls on the face of the central unit, then it cannot be selected and controlled by the interface unit.

Interface Unit

The Märklin interface unit (model 6050) permits computer control of the trains through an RS-232 interface. The interface unit is connected to the right side of the central control unit, and its RS-232 interface is connected to port A of expansion board A. The interface is configured to run at 2400 baud, 8 bits, 1 start bit, 2 stop bits, and no parity.

The interface is operated by sending a two byte command to port A of Expansion board A. The first byte is the desired speed , and the second byte is the train's ID number. The allowable speeds are 0 (full stop) and 1 to 14. Writing a speed of 15 causes the train to reverse direction. For examples of how to operate the train interface from software, refer to Section 5.3.

NOTE: Each set of transmitted commands must be separated by approximately 120ms, or the interface will loose data.

Train Tracks

The testbed uses Märklin digital K tracks. Unlike normal DC tracks, the Märklin digital tracks have both rails at ground potential, plus they have a centre rail that carries a constant AC voltage. This allows for digital signals to be superimposed on the AC powered centre rail.

The track layout and measurements are as shown in Figure 4.5. The tracks are mounted on a 7.5 foot by 7.5 foot wooden table (plywood mounted on banquet table legs) constructed for that purpose.

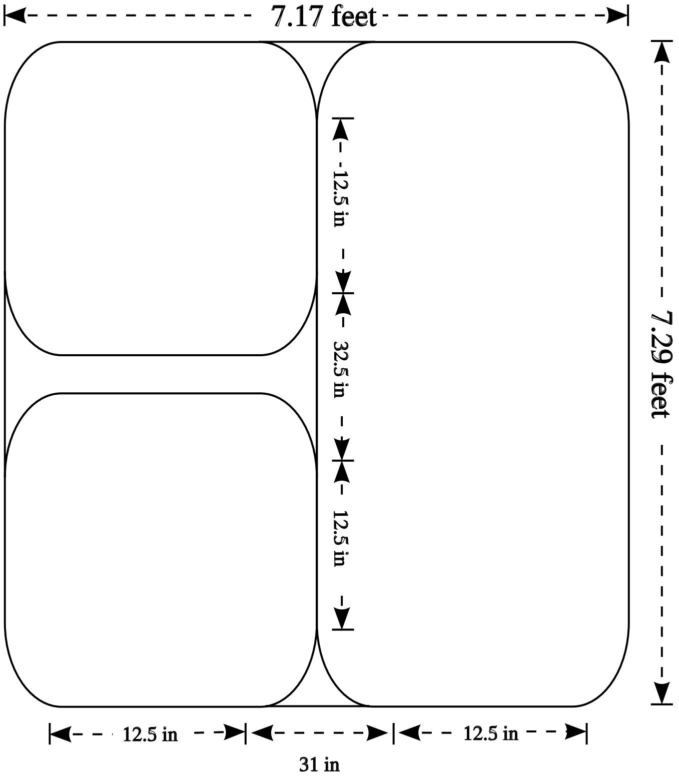


Figure 4.5: Track Measurements

4.3 Processors

The testbed consists of four separate processors: the Compaq PC, two MC68332 evaluation boards (CPU A and CPU B), and an Allen-Bradley programmable logic controller (PLC). They are discussed in the following sections.

4.3.1 Compaq PC

The Compaq 486SX PC is used as a programming station for the MC68332 processors and the PLC. The PLC is attached to communication (comm) port 1 of the PC, and the MC68332 boards are connected through a serial switch box to comm port 2 of the PC. CPU A is connected to position A of the switch box, and CPU B is connected to position B.

To access the PLC, Allen-Bradley's *6200 Series* programming software is used from the *Microsoft Windows* environment. To access the MC68332 processors, *Procomm* is used. The interface is set to 9600 baud, 8 bits, 1 stop bit, no parity, and full duplex. For information about downloading programs (in Motorola S-record format) to the MC68332 boards, refer to [25].

4.3.2 MC68332 Evaluation Boards

The testbed contains two MC68332 evaluation boards ([23], [24], [25], and [26]). They are labeled CPU A and CPU B. CPU A controls the trains and crane 1, while CPU B controls cranes 2 and 3. Both processors have almost identical expansion boards attached to them that provide each processor with two RS-232 ports, memory mapped relays to control the cranes, and a 32-bit input and a 32-bit output register to the PLC. The PLC produces a clock pulse to synchronize the transfer of data.

The MC68332 processors were added to the testbed to provide a more powerful, flexible system and to allow for an easy method to handle tasks that cannot be programmed easily into a PLC. By having the CPUs as intermediaries to portions of the testbed, one can control how the PLC sees the plant. By changing the interface to the PLC, the configuration of the apparent plant changes. The plant can be made very simple by only allowing the PLC to control and observe a few events. Alternatively, the plant can be made very complex by allowing the PLC to control and observe every aspect of the plant.

The two MC68332 processors provide the testbed with the capability of having three independent DES supervisors. This permits the investigation of issues pertaining to distributed systems. Also, since the CPUs are capable of running ‘C’ programs, complex programming tasks that do not lend themselves to discrete event modeling, can be written in a high level language.

4.3.3 Allen-Bradley PLC

The testbed contains an Allen-Bradley 1785 PLC-5/30 programmable logic controller ([2], [3], [4], [5], [15], [19], and [29]) which is responsible for the overall control of the testbed. The PLC contains an 1785 processor mounted in a 16 slot chassis. Also mounted in the chassis, are eight 16 point TTL input cards (model 1771-IGD) and six 16 point TTL output cards (model 1771-OGD). The PLC communicates with the testbed through the input/output cards. The PLC communicates with its programming station via a RS-232 port connected to comm port 1 of the Compaq PC.

The following information is important to note.

- All TTL input and output cards are set to be active low (low=TRUE). This is done by setting a jumper, accessible from the slot at the top of the card, so that it covers the two rear pins.

- To remove an I/O card, pull its terminal block forward (from top) and then slide card out carefully.

Also, the following warnings must be heeded.

- Do not remove PLC processor or I/O cards while chassis is powered. Do not handle unless properly grounded. Static charge can cause damage.

- When power is turned off to the PLC, all TTL outputs become shorts to digital ground.

The PLC has a 32-bit input and 32-bit output discrete interface to each MC68332 processor. The PLC is directly connected to all sensors and track switches. The PLC can only control the trains and cranes through the MC68332 processors.

4.4 Expansion Boards

The testbed contains two expansion boards (expansion board A and expansion board B), one for each MC68332 evaluation board. The expansion boards provide each processor with two RS-232 ports, 1Mbyte of expansion memory, memory mapped relays to control the cranes, and a 32-bit input and a 32-bit output register to the PLC. Since the two boards are almost identical, board B will be discussed in detail below followed by a discussion of how board A differs.

4.4.1 Description of CPU B Board

Expansion board B is connected to CPU B by four 34-pin ribbon cables. These cables are attached to the 64-pin expansion connectors (P1 and P2) on the MC68332 evaluation board. Due to lack of availability of 64 pin connectors, two 34 pin connectors (model 3431-5202, made by 3M Inc.) were used. Since they couldn't fit side by side, some pins on the CPU's expansion connector are not connected. Figure 4.6 shows how connectors P1A, P1B, P2A, and P2B on the expansion board are mapped to connectors P1 and P2 on the MC68332 evaluation board.

The main components of expansion board B are shown in Figure 4.7. The expansion board contains 5 primary components that sit on the CPU's bus. They are the expansion memory, crane control registers, MC68332 DUART, 32-bit PLC output register, and the 32-bit PLC input register. They are discussed below. Detailed circuit schematics as well as a placement diagram of expansion board B can be found in Appendix A. For information on how to access these components from the processor, refer to Chapter 5.

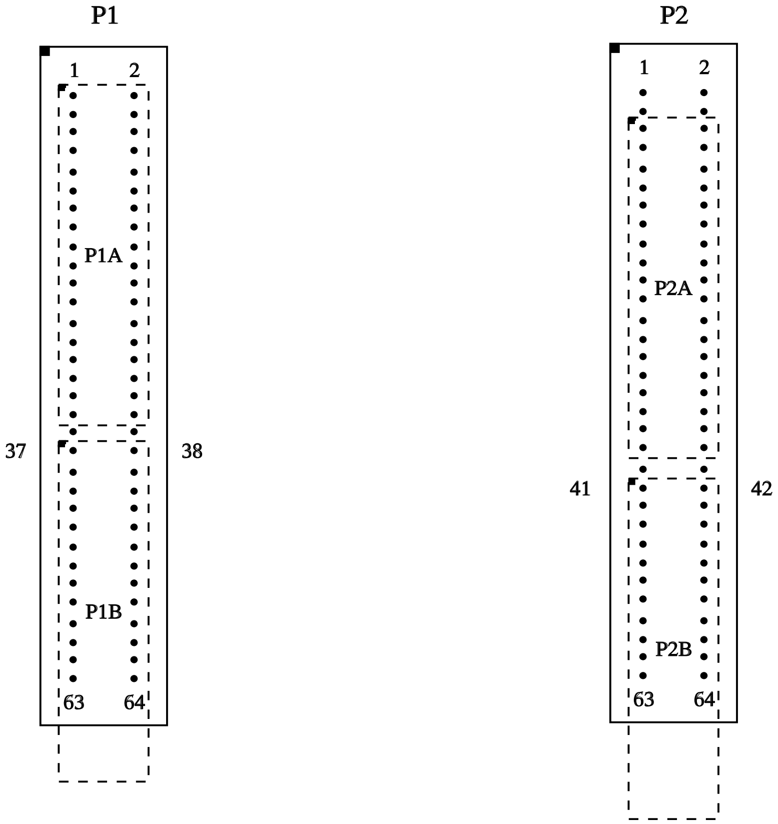


Figure 4.6: Connector Map from Expansion Board to MC68332 Board

Expansion Memory

The expansion board provides the CPU with an additional 1 MByte of memory (eight 128kByte MCM6226-30 RAM chips) to allow the CPU to download and run large programs.

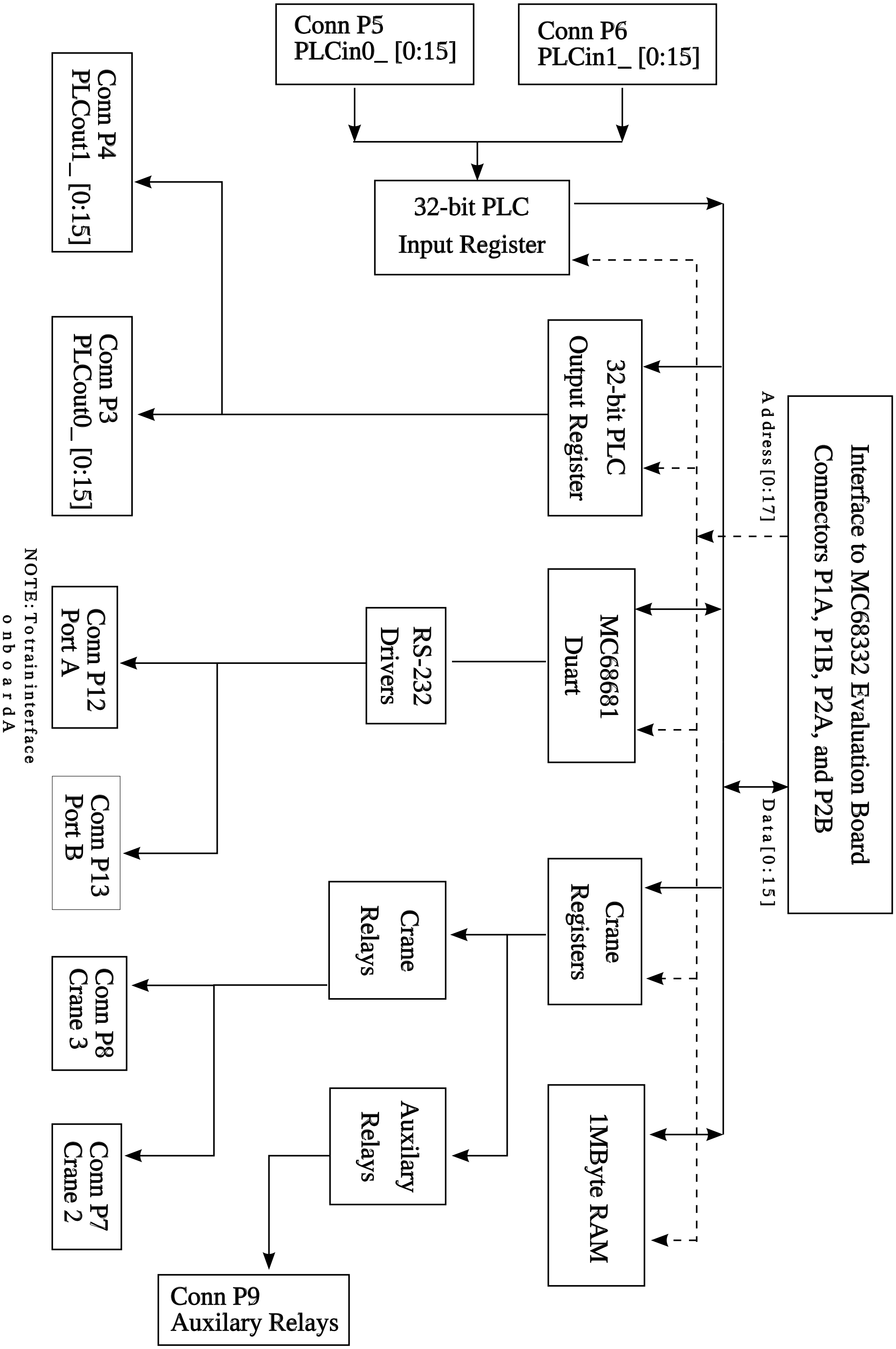


Figure 4.7: Block Diagram of Expansion Board B

Crane Control Registers

The expansion board provides two 8-bit data registers to store control data for controlling cranes 2 and 3, along with six auxiliary relays. The output of the registers drive the energizing inputs of the 10 crane relays and the six auxiliary relays. The auxiliary relays are unused relays that can be completely accessed via connector P9.

MC68681 DUART

The MC68681 dual universal asynchronous receiver/transceiver (DUART) provides the CPU with two serial ports, input capture, plus several programmable inputs and outputs. For information on how to program the MC68681 DUART, refer to [22].

PLC Registers

The expansion board provides the CPU with a 32-bit input (from the PLC) register and a 32-bit output (to the PLC) register. This gives the CPU a 64-bit discrete interface to the PLC. The PLC provides a clock pulse, labeled PLC_clock on the expansion board, to provide synchronization and to facilitate data transfer. Data is latched by the expansion board on the rising edge of the clock pulse, and expected[1] to be sampled by the PLC on the falling edge.

Note: All data signals to/from the PLC are active low signals.

4.4.2 Differences for CPU A

Expansion board A is almost identical to expansion board B. The main difference is that board A only has one 8-bit crane register. The crane register controls 5 crane relays and 3 auxiliary relays. Also, board A does not contain connector P8. An important note is that on board A, connector P12 (Port A of the DUART) is connected to the Märklin train interface unit.

[1] The PLC program is currently written to sample data on the falling edge of the clock pulse, but the PLC is not hard-wired to do so.

4.5 Remote Sensors

The testbed contains 28 remote sensors that detect the presence and identity of each train. These sensors are augmented by seven sensor boards that amplify the sensor signals for transmission to the PLC.

4.5.1 Description of Sensors

Each sensor consists of two reed switches, one mounted on each side of the track, with one end of the reed switch attached to digital ground and the other end to a sensor board. A reed switch is made of two almost touching magnetic strips, held in place by a glass support. When a correctly oriented magnet passes over a switch, the two strips are attracted together and create a short circuit (logical TRUE). Each train has a magnet mounted underneath at the front of the train. Train 1 has a magnet mounted on the left side, and train two has a magnet mounted on the right. The location of each sensor is shown in Figure 4.8.

The two reed switches are used in conjunction to determine the identity of each train, as illustrated in Figure 4.9. Each sensor generates an 'A' and a 'B' signal (RS1A and RS1B in the diagram). The switch to the left of the front of the train is always defined to be bit 0, and the switch to the right is bit 1. Thus, the two signals generate a binary signal identifying the train. In Figure 4.9, RS1A TRUE and RS1B FALSE would signify train 1, while RS1A FALSE and RS1B TRUE would signal train 2. Finally, both RS1A and RS1B FALSE would signify no trains present.

There are two main difficulties with the above method. The first is that the method is dependent on the direction the train traverses the tracks. If train 1 traverses the tracks in the direction opposite the one shown in Figure 4.9, then it will appear to be train 2. This is the main reason trains will be restricted to traverse a given portion of track in only one direction[2].

The second reason is illustrated in Figure 4.10. Experimental results show that the sensors generate a pattern similar to the one shown. The pattern is highly dependent on the speed of the train, the given sensor, and the strength of the train's magnets. Some sensors generate very small pulses (as small as 20ms), or they generate no bounce or only one. The speed of the train changes the width of each portion of the signal, plus the

[2] This restriction is part of the plant modeling assumptions. Refer to Section 6.1 for more information.

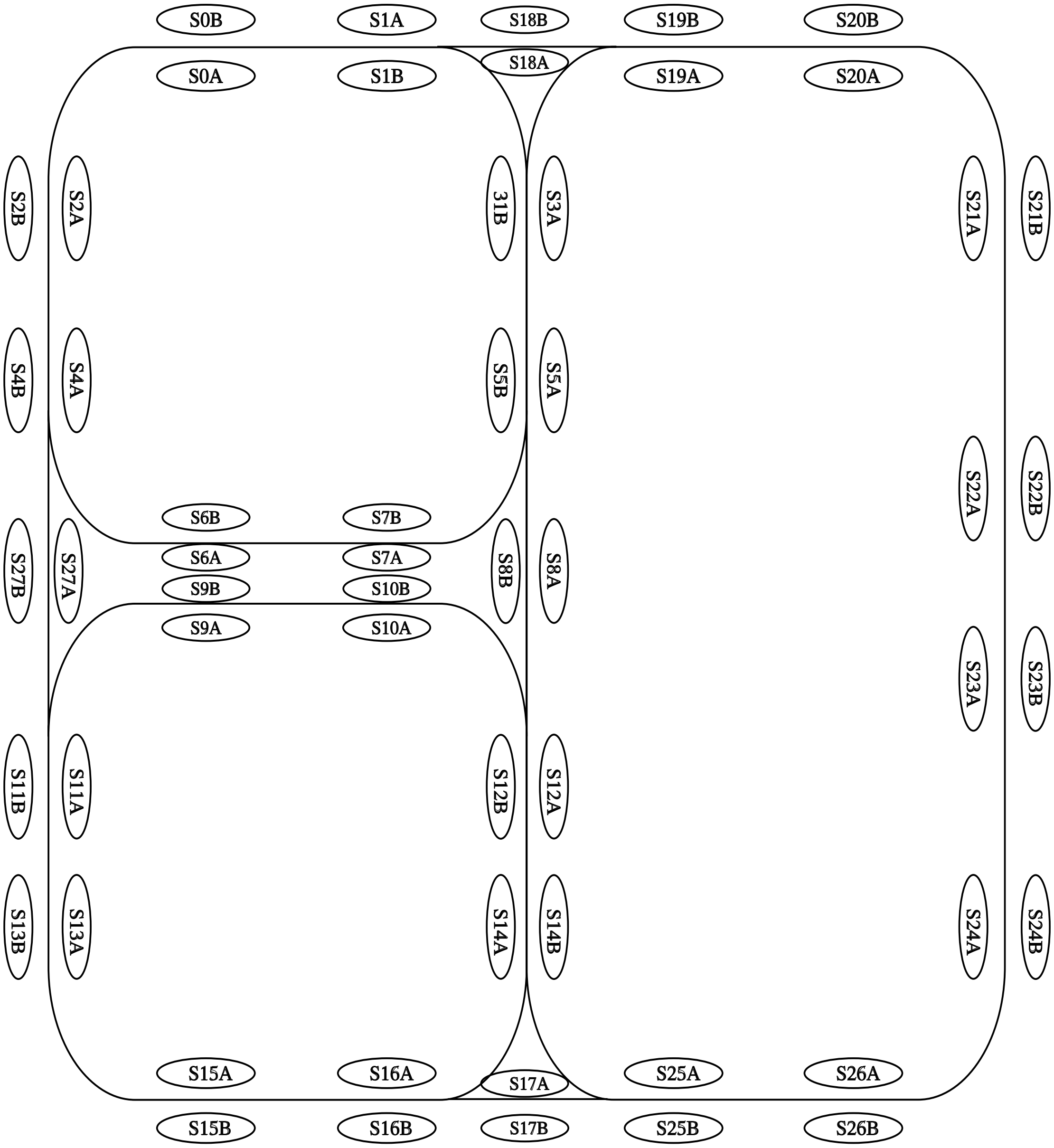


Figure 4.8: Position of Remote Sensors

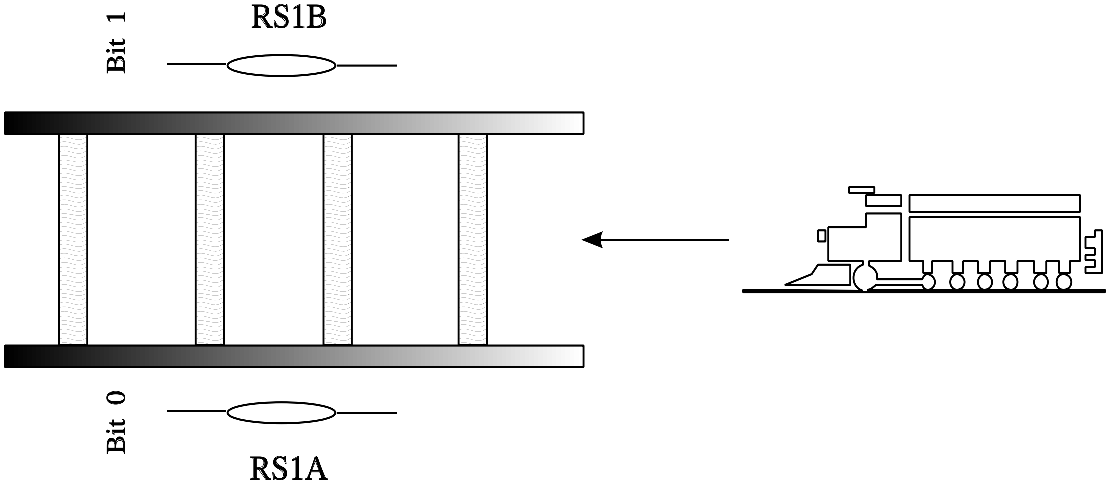


Figure 4.9: Sensor Orientation

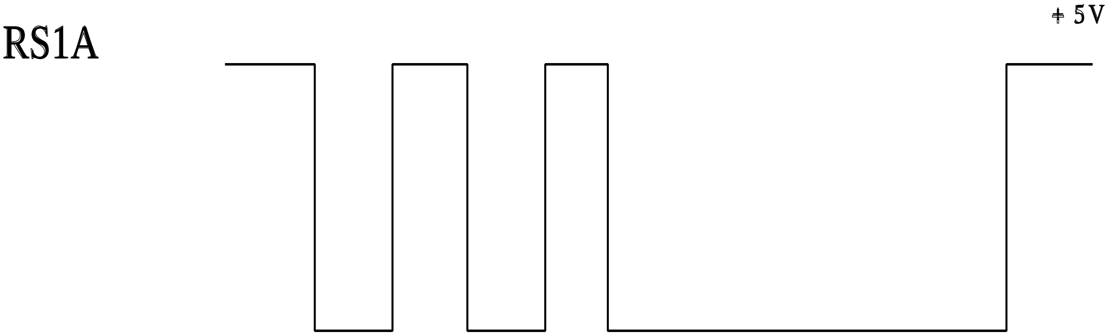


Figure 4.10: Sensor Waveform

number of bounces. The strength of the magnet creates similar effects. Clearly, it is hard to determine if only one train, more than one, or no train at all is present.

4.5.2 Description of Sensor Boards

The remote sensors are accompanied by seven sensor boards that amplify the sensor signals for transmission to the PLC. The location of each board is shown in Figures 4.1 through 4.4. Each board contains a CMOS Octal line driver chip (74HC245A) with 5 $k\Omega$ pullup resistors at the chip's inputs. When a reed switch is activated, the input to the driver chip is shorted to digital ground, creating a logical TRUE.

All seven sensor boards are the same, except for board 6. Since board 6 has a larger transmission distance than the other boards, it contains RS-232 drivers to buffer its sensor signals. Thus it requires a second board, the RS-232 buffer board, to return the RS-232 signals to TTL levels for the PLC. The location of the RS-232 buffer board is shown in Figure 4.1. Detailed circuit schematics as well as placement diagrams of all sensor boards can be found in Appendix A. Table 4.2 shows which sensor is buffered by which board. For information on how to access the sensor boards from software, refer to Chapter 5.

Note: All sensor signals going to the PLC are active low.

Board 0	*Board 1*	*Board 2*	*Board 3*	*Board 4*	*Board 5*	*Board 6*
sensor 0	sensor 3	sensor 9	sensor 6	sensor 14	sensor 23	sensor 19
sensor 1	sensor 5	sensor 11	sensor 7	sensor 15	sensor 24	sensor 20
sensor 2	sensor 8	sensor 13	sensor 10	sensor 16	sensor 25	sensor 21
sensor 4	sensor 18	sensor 27	sensor 12	sensor 17	sensor 26	sensor 22

Table 4.2: Sensor Board Sensor Assignments

4.6 Track Switches

The testbed contains 10 track switches for routing the trains. The track switches are operated by the remote switching board. Both are discussed in the following sections.

4.6.1 Description of Switches

The testbed contains four straight-right curved switches (model 2263), four straight-left curved switches (model 2262), and two three-way (left and right curve, and straight) switches (model 2270). The location of each switch is shown in Figure 4.11. Each switch contains a solenoid (three-way switches contain two) to allow for remote operation. Figure 4.12 shows how to interpret the switch control signals. For the single curve switches, the yellow wire is tied to AC power, and the appropriate blue wire is pulsed (continuous power would burn out the solenoid) to AC ground to cause the switch to change to the desired position.

For the three-way switches, the yellow wires are also tied to AC power while the appropriate blue wire is pulsed to cause the switch to change position. For the switch to go to the

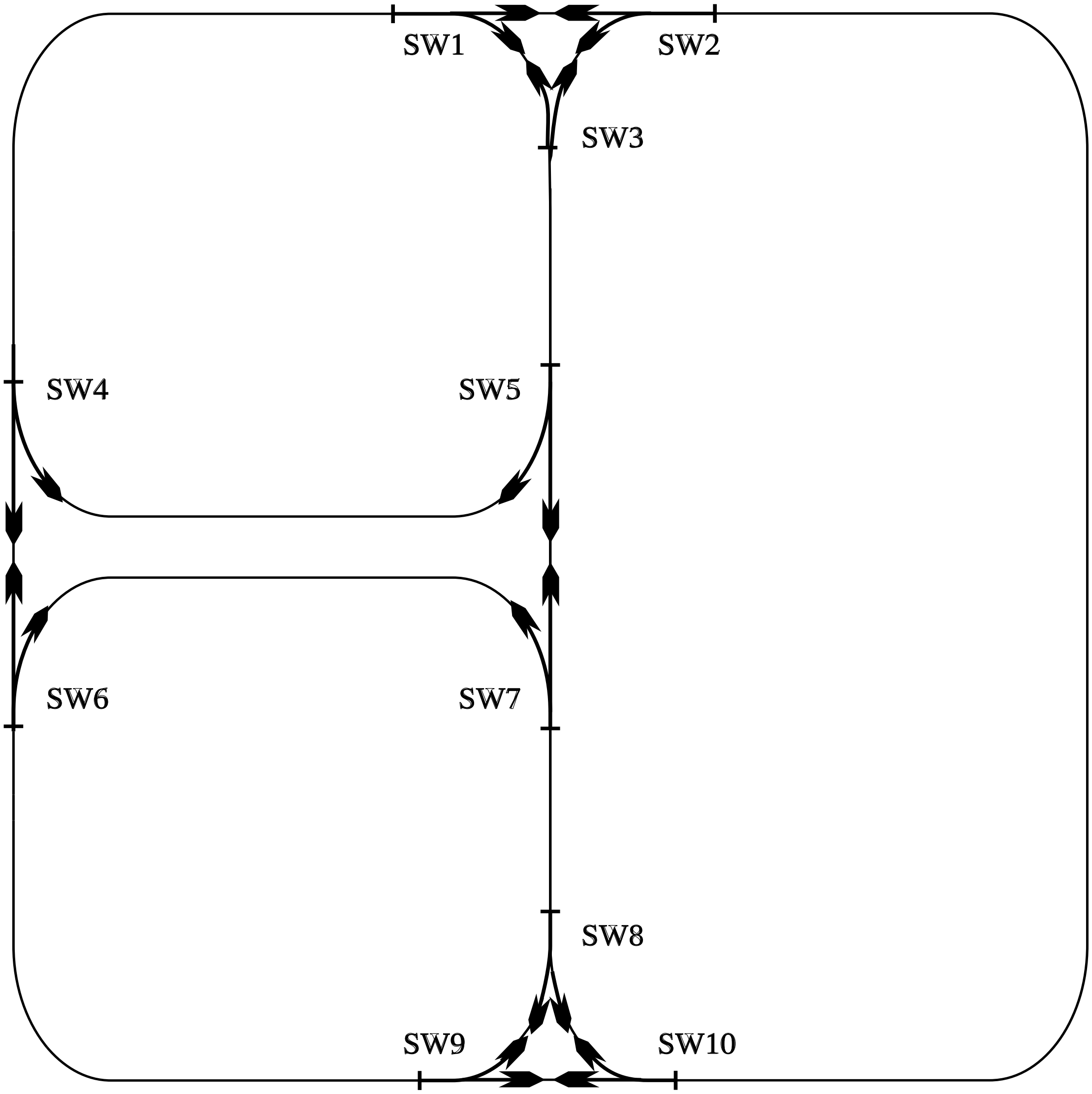


Figure 4.11: Position of Track Switches

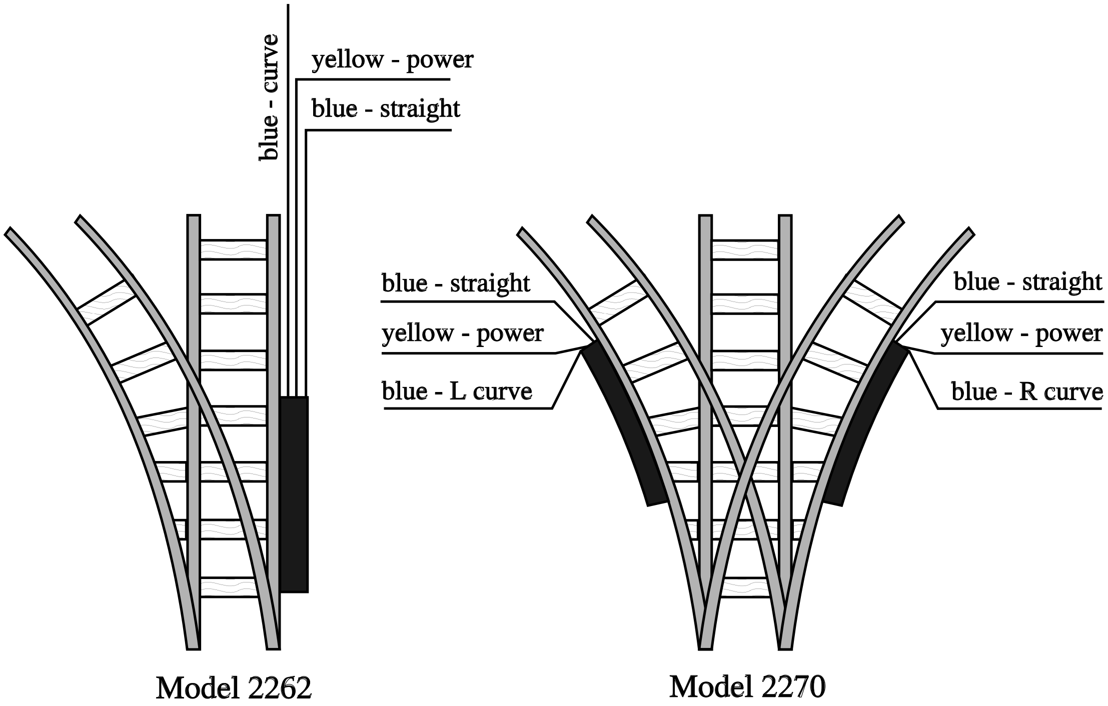


Figure 4.12: Control Signals for Switches

straight position, both wires labeled “straight” must be pulsed. To switch from one curved position to another, first the switch must be moved to the straight position, and then the appropriate blue wire must be pulsed.

Note: The pulse must be of sufficient duration to activate the switch, and that the required pulse width will vary depending upon the switch and the load on the power supply.

4.6.2 Description of Switch Circuit Board

The remote switching board controls the operation of the track switches. The board consists of three inverting line driver chips (74HC240A) (pullup resistors on their inputs) and 22 relays that, when activated, connect their attached switch control line to AC ground. Since the line driver chips are inverting, a low applied to their inputs activate the relays. The board also provides power to the switches. The board is connected to the first analog transformer. Detailed circuit schematics as well as a placement diagram of circuit board can be found in Appendix A. For information on how to access the remote switching board from software, refer to Chapter 5.

WARNING: Connectors P1 and P2 must be left disconnected when PLC is turned off. The TTL outputs on the PLC are shorted to ground when the PLC is turned off. This causes all relays (thus all switch solenoids) to be activated on the remote switching board at once since the relays are activated by an active low signal. This problem will be corrected in the future by changing the hardware to accept active high signals.

4.7 Power Supply, and Cables and Connectors

In this section, the digital power supply that powers the digital portion of the testbed, is described. The cables and their attached connectors are also discussed.

4.7.1 Digital Power Supply

The digital power supply used for the testbed is the T0-51/10 model, purchased from Tectrol Inc. This power supply is capable of providing DC voltage at 5V (1-6A), 12V (0.3-2A), -5V (0-0.1A), and -12V (0-0.5A). The power supply is mounted inside a metal casing to prevent electric shock, and the DC power connections are brought to a labeled terminal block on its face. For reference, the pinout of the internal connectors on the circuit board are given in Table 4.3.

Note: If the minimal current rating for a given voltage source (1A for the 5V source) is not met, the supply will not function correctly.

Connector P1		*Connector P2*					
Pin No		*Pin No*		*Pin No*		*Pin No*	
1	Neutral	1	-5V	4	Ground	7	+5V
3	Line	2	-12V	5	Ground		
5	AC Ground	3	+12V	6	+5V		

Table 4.3: Internal Power Supply Connector Pinout

4.7.2 Cables and Connectors

To connect the various components of the testbed 26 cables were used, not including power supply connections.

Power Connections

The PLC I/O cards, the remote switching board, and each MC68332 evaluation board, expansion board, and sensor board have a connection to +5V and ground on the digital power supply. In most cases, the large boards are directly connected to the power supply, and the smaller boards are connected to the larger boards. This is done to reduce clutter at the power supply.

Expansion board A and the remote switching board are connected (AC power and AC ground) to the first analog transformer. Expansion board B is connected to the second transformer. The Märklin central control unit is connected to the digital transformer. The control unit is then connected to the tracks by a feeder track (model 2290).

Note: The feeder track has two connections. The connection that shorts to the track's train rails should be connected to the AC ground provided by the control unit.

Signal Cables and Connectors

Twenty-six cables are used to provide interconnection for the testbed. In the case of cables that go to the PLC (excluding the cable connecting the PLC to its programming station), they contain a connector only on the end opposite of the PLC. Each cable has been labeled, including information on which board its connector is for, when necessary.

Sensor boards 0-5 each have a cable connecting them to one of the PLC's input cards. Sensor board 6 has a cable connecting it to the RS-232 buffer board, which has a cable connecting it to the PLC. The remote switching board has two cables connecting it to some of the PLC's output cards. Each expansion board has four cables connecting their PLC input and output registers to the PLC's I/O cards. Refer to section 5.4 to determine where each signal is connected to the PLC.

There are four cables connecting the testbed's hardware to the expansion boards. There is a cable connecting the Märklin train interface unit to expansion board A, as well as a cable connecting each crane to their respective expansion board. The three remaining cables are the terminal cables for the MC68332 evaluation boards and the PLC. They connect the indicated processor to the Compaq PC (the programming station). Appendix C details all signal cables and their attached connectors.

4.8 Unresolved Problems

Although the testbed is operational, there are still several minor hardware problems that need to be resolved. These problems have not yet been corrected due to lack of time, even though their solutions are known. The problems are listed below.

1. Some sensors have a pulse width that is less than the sampling period (approx 40ms) of the state machines. This causes the signal to be missed from time to time. To solve this problem, the PLC code will be modified to latch the sensor signals outside of the main control loop[3] and then unlatch them once the data has been read by the state machines.

2. A major problem is noise from the AC motors and solenoids, and from the relays switching. The AC noise is being superimposed on the DC circuits, sometimes corrupting data received from the PLC. This shouldn't have been a problem since generally the MC68332 processors latch data from the PLC at different times than when the relays switch. Unfortunately, the clock signal from the PLC is tied directly to the clock input of the registers that latch data from the PLC. Thus, when there is noise on the clock, the registers become corrupted. The clock of the PLC registers should be tied to a control output of the MC68332 processor. When the PLC clock signal generates an interrupt (this process already debounces the signal to avoid false triggering), then the processor can latch and read the data from the PLC several times to ensure that the data transfered is reliable. Also, the contacts of the relays should be filtered to reduce noise.

3. Additional sensors need to be added so that a closed loop controller for the cranes can be designed.

4. The placement of some of the sensors needs to be adjusted. In particular, the placement of the sensors in front of the cranes needs to be modified so that the trains will stop in the correct position to allow the trains to be loaded by the cranes. This will be necessary for a closed loop crane controller to be able to function correctly.

[3] Refer to Chapter 10 for information on the main control loop of the PLC software.

Chapter 5

Programming Interface

This chapter explains how to access and control the testbed's hardware, from software. Specifically, this chapter discusses how to access the processors, and program them to monitor the sensors, and to control the switches, cranes and trains. Also, this chapter will discuss how the PLC communicates with the MC68332 evaluation board, and the structure of the interrupt-driven interface program (running on the MC68332 processors) that allows the PLC to communicate with the MC68332 processors.

5.1 Accessing The Processors

To access the Allen-Bradley PLC5-L30B processor, start *Microsoft Windows* on the Compaq PC. Open the Allen-Bradley program group, and double click on *6200 PLC-5* icon. To access the processor, press function key *F1* for online programming. Ensure that you first turn the PLC processor on[1]. For detailed instruction on how to use the *6200 PLC-5* software, refer to [3], [4], and [5].

To access the MC68332 boards, start the program `c:\testbed\procomm\procomm` on the Compaq PC. The MC68332 boards are connected through a serial switch box to communication port 2 of the PC. CPU A is connected to position A of the switch box, and CPU B is connected to position B. Ensure that power to the testbed[2] has been turned on. *Procomm* should be configured to 9600 baud, 8 bits, 1 stop bit, no parity, and full

[1] To start the PLC, flip on the *POWER* switch located on the front of the PLC's power supply. The power supply is mounted to the left of the chassis.

[2] Check the power bar that the digital power supply is plugged into.

duplex. For information about down-loading programs (in Motorola S-record format) to the MC68332 boards, refer to [25]. To learn to use the board's built in debugger program (the M68CPU32Bug Debug Monitor), refer to [24].

Currently, software for the MC68332 boards is written in 68000 assembly language and assembled using *AS32.exe*, the assembler provided with the evaluation board. The assembler is located in directory `c:\CC68K` on the Compaq PC. Refer to [21] for information on how to use the assembler. A copy of this information is available in the file *as32ref.man* in `c:\CC68K`. Refer to [13] for more information on 68000 assembly language. Eventually, a 'C' cross compiler will be available for programming.

5.2 MC68332 Programming Model

Since the main components of the testbed's two expansion boards are directly on the processors' address and data buses, they can be accessed simply by reading/writing to the correct memory location. The processors must first be configured to access this memory space. Figure 5.1 shows the processor's memory map, particularly where the expansion board's components are located. The memory map is identical for both expansion boards. The location of individual registers will be discussed in Section 5.3.

Each section of the expansion board is accessed by a specific chip select signal, as shown in Figure 5.1. Chip selects are programmable active low signals, generated by the processor. They can be programmed to generate a synchronous signal when a memory access occurs in their programmed memory space[3]. They also provide data transfer acknowledges, telling the processor that the data cycle is complete. Before the expansion board can be accessed, the indicated chip selects must be configured appropriately. This means the chip selects must be configured every time the processor is reset.

The devices shown in Figure 5.1 use the chip selects to tell them when they should latch/drive the data bus. The expansion boards also use Programmable Array Logic (PAL) chips (22V10BCN) to combine the chip select information with low order address bits to generate individual chip selects for registers that share the chip selects' address range. The ABEL ([12]) programs that define the PAL's operation are given in Appendix E.

Program *csinit.s* configures the chip selects appropriately. Refer to Appendix E for a

[3] The chip selects can also be programmed to generate an autovector for an external device.

listing of the file. Refer to [23] for more information on using chip selects.

Note: Chip select six is set to create an autovector for the DUART. Chip selects 0-2, 4, and 6 are used by the MC68332 debug monitor.

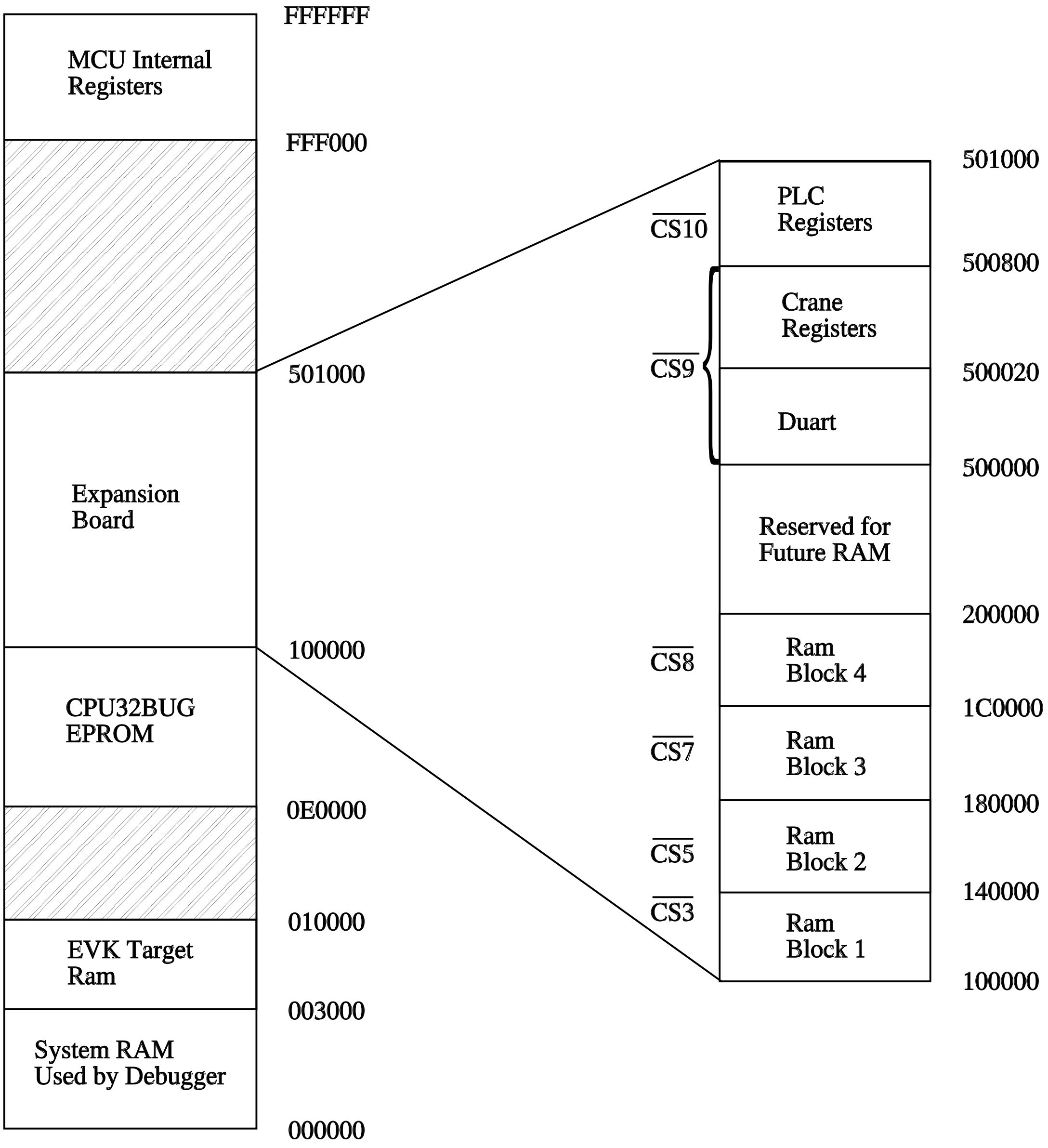


Figure 5.1: MC68332 Memory Map

5.3 Hardware Interface

In this section, the specific details of accessing the components of the expansion boards are discussed.

5.3.1 Expansion memory

The expansion memory occupies the continuous memory range of $100000[4] to $200000 and can be accessed as a byte, word, or long word just like regular memory.

5.3.2 Train DUART

This section discusses how to use the DUART on expansion board A to control the trains. This information can be used as a starting place for configuring the DUARTs on either board for other tasks.

The internal registers of the MC68681 DUART are mapped to the memory range $500000 to $50001E. The DUART sits on the upper half of the processor's data bus (data bits 8 to 15) since it has only an 8 bit data bus. This means that the DUART can only be accessed on even memory addresses. The addresses of the internal registers are given in Table 5.1. Refer to [22] for information on configuring these registers.

To access the Märklin interface unit, the DUART's port A must be configured to run at 2400 baud, 8 bits, 1 start bit, 2 stop bits, and no parity. The appropriate initialization code is given below:

```
* Registers for port A of the DUART (for the train)

MR1A    equ     $500000
MR2A    equ     $500000
CSRA    equ     $500002
CRA     equ     $500004
ACR     equ     $500008
OPCR    equ     $50001A
SRA     equ     $500002
RBA     equ     $500006
TBA     equ     $500006
IMR     equ     $50000A
IPCR    equ     $500008
```

[4] The "$ " signifies that the number is in hexadecimal notation.

Address	*Read Cycle*	*Write Cycle*
$500000	Mode Register A (MR1A,MR2A)	Mode Register A (MR1A,MR2A)
$500002	Status Register A (SRA)	Clock-Select Register A (CSRA)
$500004	Factory testing only	Command Register A (CRA)
$500006	Receiver Buffer A (RBA)	Transmitter Buffer A (TBA)
$500008	Input Port Change Register (IPCR)	Auxilary Control Register (ACR)
$50000A	Interrupt Status Register (ISR)	Interrupt Mask Register (IMR)
$50000C	Counter Mode: MSB of Counter (CUR)	Counter/Timer Upper Register (CTUR)
$50000E	Counter Mode: LSB of Counter (CLR)	Counter/Timer Lower Register (CTLR)
$500010	Mode Register B (MR1B,MR2B)	Mode Register B (MR1B,MR2B)
$500012	Status Register B (SRB)	Clock-Select Register B (CSRB)
$500014	Factory testing only	Command Register B (CRB)
$500016	Receiver Buffer B (RBB)	Transmitter Buffer B (TBB)
$500018	Interrupt-Vector Register (IVR)	Interrupt-Vector Register (IVR)
$50001A	Input Port (unlatched)	Output Port Configuration Register (OPCR)
$50001C	Start-Counter Command	Bit Set Command
$50001E	Stop-Counter Command	Bit Reset Command

Table 5.1: DUART Register Addressing and Address Triggered Commands

```
* configure DUART for use with train
        move.b  #$1a,CRA
        move.b  #$30,CRA
        move.b  #$20,CRA
        move.b  #$13,MR1A
        move.b  #$0f,MR2A
        move.b  #$88,CSRA
        move.b  #$60,ACR
        move.b  #$05,CRA
* Disable all interrupts in DUART!
        move.b  #$00,IMR
```

The train interface is operated by sending a two byte command to port A. The first byte is the desired speed , and the second byte is the train’s ID number. The allowable speeds are 0 (full stop) to 14. Writing a speed of 15 causes the train to reverse direction. The code to initialize train 1 to a full stop is given below. The ID number of train one is *”1”*, and the ID number of train two is *”2”*.

Note: Each set of commands must be separated by approximately 120ms or the interface will loose data.

```
* Initialize train 1 to stop
        move.b  #1,d2      store train ID
        move.b  #0,d4      store speed
        jsr     TRANS

************************************************************************
* Subroutine: TRANS
************************************************************************

TRANS:  movem.l d0,-(sp)

*transmit speed to train
TSTTX1  btst.b  #2,SRA
        beq     TSTTX1
        move.b  d4,TBA
        move.w  #I_BYTEW,d0

*transmit train ID
TSTTX2  btst.b  #2,SRA
        beq     TSTTX2
        move.b  d2,TBA

        movem.l (sp)+,d0
        rts
```

5.3.3 Crane Registers

The crane register on expansion board A is accessed the same as the register on board B, except that only the upper 8 bits are used. Both are accessed by writing a 16-bit word to $500020. Figures 5.2 and 5.3 show the interpretation of each bit of the crane registers. Writing a *"1"* to a bit activates its corresponding relay, which activates the appropriate motor in the appropriate crane. Writing a *"0"* deactivates the relay.

Note: Crane registers are write only. Reading registers will return junk. If you require the current value of the registers, keep a copy in software.

Before the crane registers can be used, some configuration must be performed since the register's outputs are floating after the processor is reset. The crane register's output enable is driven by bit 1 of port F on the processor. The bit must be configured as an output and set to *"0"* to enable the registers outputs. The code required is given below. For more information about port F of the CPU, refer to [23].

UPPER BYTE

15	14	13	12	11	10	9	8
AUX 2	AUX 1	AUX 0	Crane 1 Energize Magnet	Crane 1 Lower Magnet	Crane 1 Raise Magnet	Crane 1 Turn Left	Crane 1 Turn Right

LOWER BYTE

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1

Figure 5.2: Crane Register - Board A

UPPER BYTE

15	14	13	12	11	10	9	8
Crane 3 Raise Magnet	Crane 3 Turn Left	Crane 3 Turn Right	Crane 2 Energize Magnet	Crane 2 Lower Magnet	Crane 2 Raise Magnet	Crane 2 Turn Left	Crane 2 Turn Right

LOWER BYTE

7	6	5	4	3	2	1	0
AUX 5	AUX 4	AUX 3	AUX 2	AUX 1	AUX 0	Crane 3 Energize Magnet	Crane 3 Lower Magnet

Figure 5.3: Crane Register - Board B

```
* Configure crane registers

* port F control registers
PFPAR   equ     $fffa1f
DDRF    equ     $fffa1D
PORTF   equ     $fffa19

* Crane register
CREG    equ     $500020

* Clear crane registers to disable relays
        move.w  #0,CREG

* set bit 1 of port F to I/O bit
        move.b  #$fd,PFPAR
* set data direction
        move.b  #$02,DDRF
* set bit 1 to zero to enable registers outputs
        move.b  #$0,PORTF
```

5.3.4 PLC Registers

Each expansion board contains two 16-bit PLC input (from the PLC) registers and two 16-bit PLC output registers. Table 5.2 shows the addresses these registers occupy. The PLC provides a clock pulse, labelled PLC_clock on the expansion boards, to provide synchronization and to facilitate data transfer. Data is latched by the expansion board on the rising edge of the clock pulse, and expected[5] to be sampled by the PLC on the falling edge.

Note: PLC input and output registers share the same address space. They can only be read or written to, as indicated. If you require the current value of an output registers, keep a copy in software.

5.4 PLC Interface

The PLC interacts with the testbed through its eight 16-point TTL input cards and six 16-point TTL output cards. The sensor and remote switch boards, as well as the expansion board's PLC registers are all connected to these I/O cards. The clock pulse that the PLC

[5] The PLC program is currently written to sample data on the falling edge of the clock pulse, but the PLC is not hard-wired to do so.

Address	*Read Cycle*	*Write Cycle*
$500800	PLC Input Register 0 (PLCin0_[15:0])	PLC Output Register 0 (PLCout0_[15:0])
$500802	PLC Input Register 1 (PLCin1_[15:0])	PLC Output Register 1 (PLCout1_[15:0])

Table 5.2: PLC Register Addressing

generates to synchronize the transfer of data with the expansion boards, also comes from an output card.

On a PLC, data is organized as files. There are input image files, output image files, integer files, and binary files, among others. PLC programs do not directly interact with I/O cards. They read from the input image files and write to the output image files. Once per cycle, the input file is updated from the inputs, and the output file is written to the output cards by the PLC's operating system. A typical cycle is shown in Figure 5.4.

On a PLC, data is normally accessed as a bit. Figure 5.5 shows how a bit address is determined for several common data file types. Refer to [5] for more information about PLC addressing. The testbed's PLC is configured with 16-point I/O modules using 2-slot I/O groups (refer to [2]). This means that the first slot is occupied by a 16-point input module, and the second slot by a 16-point output module, but they both share the same group number. In Section 10.1, Figure 10.1 shows a 16 slot chassis, holding a PLC-5/L30B processor and 16 I/O cards. The testbed PLC does not have the last two output cards. The labelling shows which I/O group each card belongs to.

Note: The chassis is set to use 2-slot I/O groups by configuring the chassis backplane dip switches. Switches four and five should be set to the off position. Refer to [1] for more information.

Figure 5.6 shows how the PLC's I/O cards are assigned. Table 5.3 and Table 5.4 show how the signals from the sensor boards and the remote switching board correspond to the actual sensors and switches.

5.5 Structure of MC68332 Interface Programs

The interface programs are designed to provide an interrupt based interface to the PLC. The rising edge of the clock pulse (labelled PLC_clock) the PLC generates, creates an interrupt

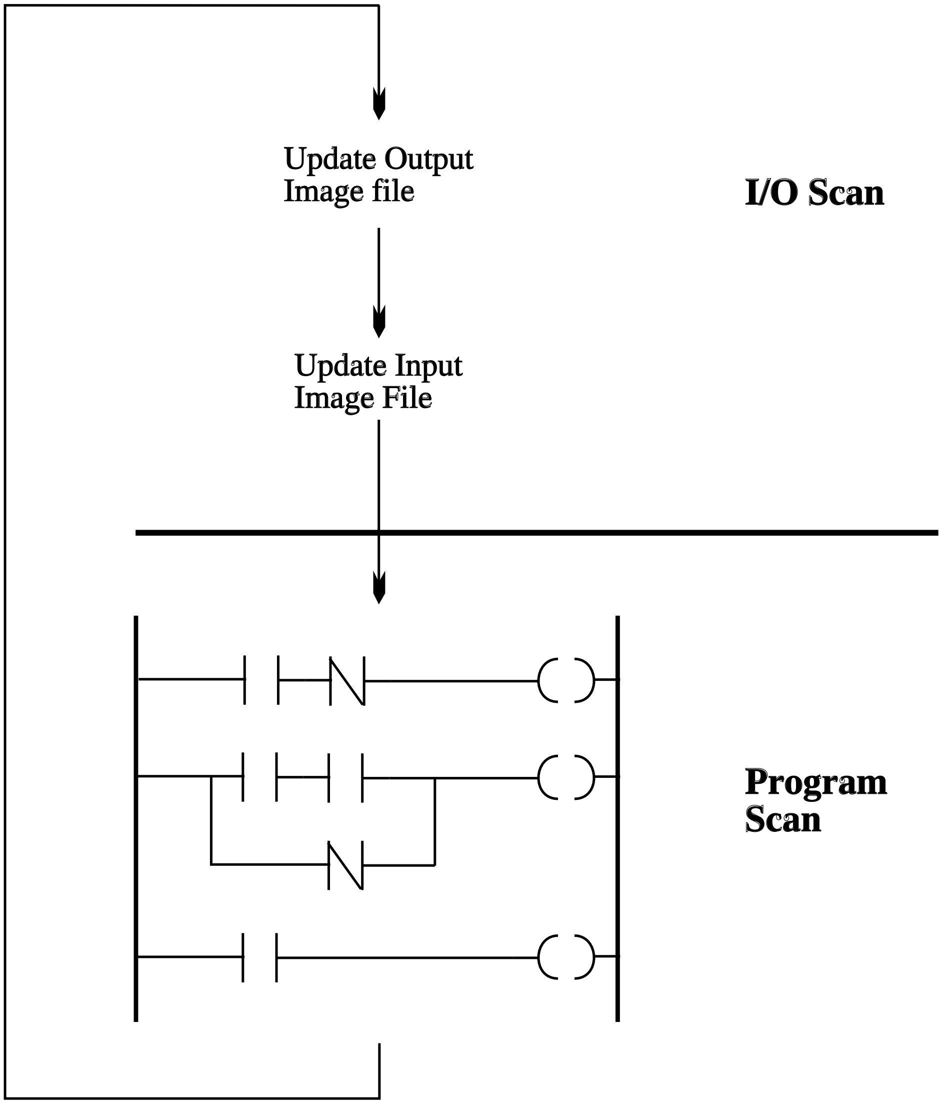


Figure 5.4: PLC Scan Cycle

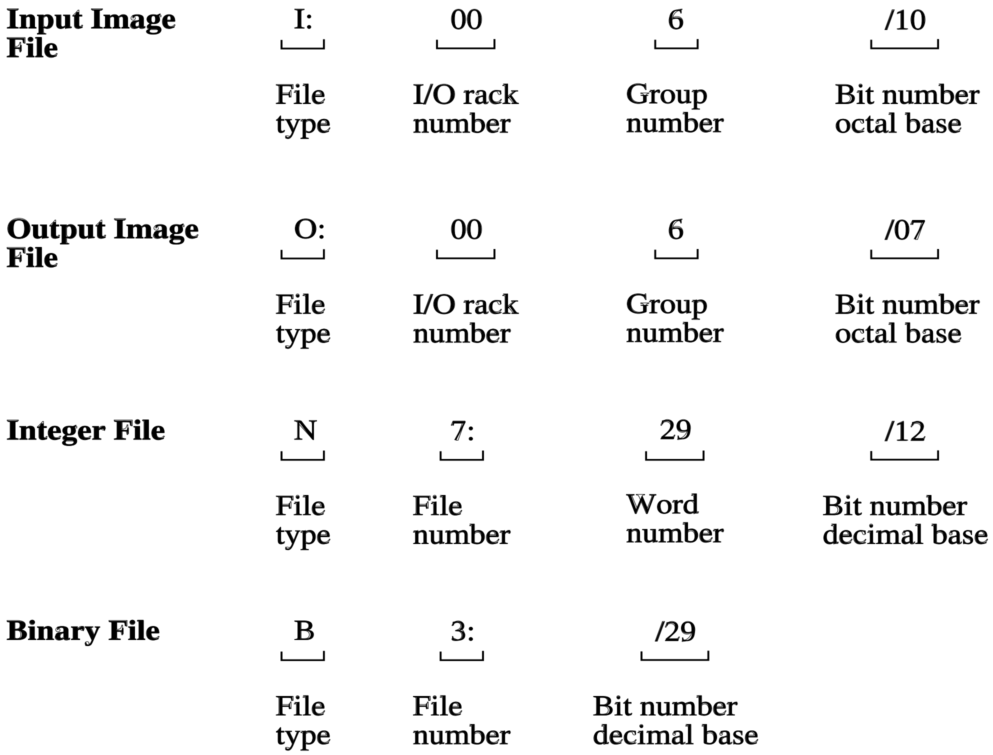


Figure 5.5: PLC Addressing

	Board 0	*Board 1*	*Board 2*	*Board 3*	*Board 4*	*Board 5*	*Board 6*
Bit	From	From	From	From	From	From	From
0	RS4A	RS8A	RS27A	RS12A	RS17A	RS25A	RS19B
1	RS4B	RS8B	RS27B	RS12B	RS17B	RS25B	RS19A
2	RS2A	RS5A	RS9A	RS10A	RS14A	RS26A	RS20B
3	RS2B	RS5B	RS9B	RS10B	RS14B	RS26B	RS20A
4	RS0A	RS3A	RS11A	RS7A	RS16A	RS24A	RS21B
5	RS0B	RS3B	RS11B	RS7B	RS16B	RS24B	RS21A
6	RS1A	RS18A	RS13A	RS6A	RS15A	RS23A	RS22B
7	RS1B	RS18B	RS13B	RS6B	RS15B	RS23B	RS22A

Table 5.3: Sensor Board Assignments

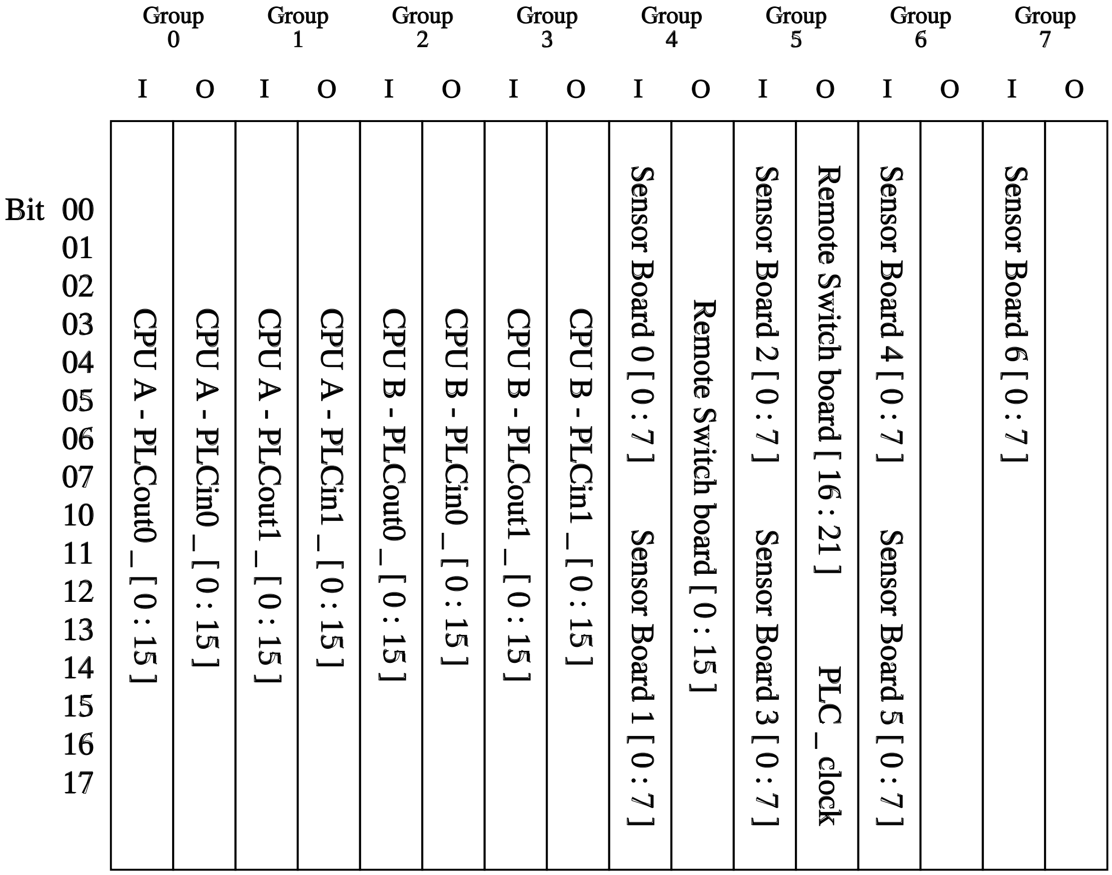

Group 0
Group 1
Group 2
Group 3
Group 4
Group 5
Group 6
Group 7
I
O
Bit 00
01
02
03
04
05
06
07
10
11
12
13
14
15
16
17
CPU A - PLCout0_ [0 : 15]
CPU A - PLCin0_ [0 : 15]
CPU A - PLCout1_ [0 : 15]
CPU A - PLCin1_ [0 : 15]
CPU B - PLCout0_ [0 : 15]
CPU B - PLCin0_ [0 : 15]
CPU B - PLCout1_ [0 : 15]
CPU B - PLCin1_ [0 : 15]
Sensor Board 0 [0 : 7]
Sensor Board 1 [0 : 7]
Remote Switch board [0 : 15]
Sensor Board 2 [0 : 7]
Sensor Board 3 [0 : 7]
Remote Switch board [16 : 21]
PLC _ clock
Sensor Board 4 [0 : 7]
Sensor Board 5 [0 : 7]
Sensor Board 6 [0 : 7]

Figure 5.6: PLC I/O Card Bit Assignments

Bit	Signal	Bit	Signal	Bit	Signal	Bit	Signal
0	PLC_turn1	6	PLC_turn4	12	PLC_turn7	18	PLC_turn10
1	PLC_str1	7	PLC_str4	13	PLC_str7	19	PLC_str10
2	PLC_turn2	8	PLC_turn5	14	PLC_turnR8	20	PLC_str3
3	PLC_str2	9	PLC_str5	15	PLC_turnL8	21	PLC_str8
4	PLC_turnR3	10	PLC_turn6	16	PLC_turn9		
5	PLC_turnL3	11	PLC_str6	17	PLC_str9		

Table 5.4: Remote Switching Board Assignments

for each MC68332 processor. The processors read the PLC input registers, update the PLC output registers, and perform any actions the PLC requested. The main program loop operates the cranes. The following sections discuss the interface programs for each of the processors.

5.5.1 The CPU A Interface Program

The CPU A interface program allows the PLC remote access to the trains and crane 1. The pseudo code for the program is given below. Refer to Appendix E for a listing of the interface program, labelled *desinta.s*.

```
************************************************************
*    Main Program Loop CPU A:
************************************************************

        Build_exception_table();
        Initialize();

MLOOP:  if (crane_request = TRUE) then
                { clear crane_request flag;
                  crane_load1();
                  set crane_finish flag;
                }

        goto MLOOP;

************************************************************
*    Subroutine: Build_exception_table
************************************************************

        copy old exception table;
        set exception \#67 to address of Timer_handler();
        set exception \#65 to address of Input_capture_handler();

        return;

************************************************************
*    Subroutine: Initialize
************************************************************

        configure periodic timer;
        initialize flags;
        initialize crane register;
        initialize DUART for use with trains;
        initialize trains;
        configure time processor unit (TPU) to create interrupt
                  on rising edge of PLC_clock;
        enable upto level 5 interupts;

        return;

************************************************************
*    Interrupt service routine: Timer_handler
*
*    Routine gets called every time periodic timer generates
*    an interrupt.
************************************************************

        increment Counter 1;

        return;

************************************************************
*    Interrupt service routine: Input_capture_handler
*
*    Routine gets called every time TPU generates an
*    an interrupt.
************************************************************

        read PLC input registers;
        if (PLC requested crane) then
                { if (crane_busy = FALSE) then;
                    { clear crane_finish flag;
                      set crane_request flag;
                      set crane_busy flag}
                  else
                    { if (crane_finish = TRUE) then;
                        { clear crane_finish flag;
                          clear crane_busy flag;
                          set flag to tell PLC crane finished}}}
        else
                { if (crane_busy = TRUE) then
                    { if (crane_finish = TRUE) then
                        { clear crane_finish flag;
                          clear crane_busy flag;
                          set flag to tell PLC crane finished}}}

        update PLC output registers
        if (PLC request train 1 to stop)
                stop train 1
        else
                start train 1;

        if (PLC request train 2 to stop)
                stop train 2
        else
                start train 2;


        return;

************************************************************
*    Subroutine: Crane_load1
************************************************************

        start periodic timer;
        move crane 1 left until Counter 1 = set value;

        stop timer;
        clear Counter 1;
        start periodic timer;
        move crane 1 down until Counter 1 = set value;

        stop timer;
        clear Counter 1;
        energize magnet;

        start periodic timer;
        move crane 1 up until Counter 1 = set value;

        stop timer;
        clear Counter 1;
        start periodic timer;
        move crane 1 right until Counter 1 = set value;

        stop timer;
        clear Counter 1;
        start periodic timer;
        move crane 1 down until Counter 1 = set value;

stop timer;
        clear Counter 1;
        de-energize magnet;

        stop timer;
        clear Counter 1;
        start periodic timer;
        move crane 1 up until Counter 1 = set value;

        return
```

5.5.2 The CPU B Interface Program

The CPU B interface program allows the PLC remote access to crane 2 and crane 3. The pseudo code for the program is given below. Refer to Appendix E for a listing of the interface program, labelled *desintb.s*.

```
************************************************************
*    Main Program Loop CPU B:
************************************************************

        Build_exception_table();
        Initialize();

MLOOP:  if (crane2_request = TRUE) then
                { clear crane2_request flag;
                  crane_load2();
                  set crane2_finish flag;
                }

        if (crane3_request = TRUE) then
                { clear crane3_request flag;
                  crane_load3();
                  set crane3_finish flag;
                }
        goto MLOOP;

************************************************************
*    Subroutine: Build_exception_table
************************************************************

        copy old exception table;
        set exception \#67 to address of Timer_handler();
        set exception \#65 to address of Input_capture_handler();

        return;
```

```
************************************************************
*    Subroutine: Initialize
************************************************************

        configure periodic timer;
        initialize flags;
        initialize crane register;
        configure time processor unit (TPU) to create interrupt
                  on rising edge of PLC_clock;
        enable upto level 6 interupts;

        return;

************************************************************
*    Interrupt service routine: Timer_handler
*
*    Routine gets called every time periodic timer generates
*    an interrupt.
************************************************************

        increment Counter 1;

        return;

************************************************************
*    Interrupt service routine: Input_capture_handler
*
*    Routine gets called every time TPU generates an
*    an interrupt.
************************************************************

        read PLC input registers;
        if (PLC requested crane 2) then
                { if (crane2_busy = FALSE) then;
                    { clear crane2_finish flag;
                      set crane2_request flag;
                      set crane2_busy flag}
                  else
                    { if (crane2_finish = TRUE) then;
                     { clear crane2_finish flag;
                   clear crane2_busy flag;
                   set flag to tell PLC crane 2 finished}}}
        else
                { if (crane2_busy = TRUE) then
                    { if (crane2_finish = TRUE) then
```

```
                     { clear crane2_finish flag;
                   clear crane2_busy flag;
                   set flag to tell PLC crane 2 finished}}}

        if (PLC requested crane 3) then
                { if (crane3_busy = FALSE) then;
                    { clear crane3_finish flag;
                      set crane3_request flag;
                      set crane3_busy flag}
                  else
                    { if (crane3_finish = TRUE) then;
                     { clear crane3_finish flag;
                   clear crane3_busy flag;
                   set flag to tell PLC crane 3 finished}}}
        else
                { if (crane3_busy = TRUE) then
                    { if (crane3_finish = TRUE) then
                     { clear crane3_finish flag;
                   clear crane3_busy flag;
                   set flag to tell PLC crane 3 finished}}}

        update PLC output registers

        return;

************************************************************
*    Subroutine: Crane_load2
************************************************************

*  same as Crane_load1 for CPU A, except affects crane 2

************************************************************
*    Subroutine: Crane_load3
************************************************************

*  same as Crane_load1 for CPU A, except affects crane 3
```

5.5.3 Future Modifications

There are several improvements to the MC68332 interface programs that would be desirable. The first would be to replace the open loop controller for the cranes by a closed loop controller.

The second change would be to alter the train interface. Due to hardware limitations, the MC68332 can not send commands to the train at a rate faster than 2 bytes every 120 ms, or data may be lost. Unfortunately this limitation was not known when the software was designed, thus commands to the trains are sometimes lost. To compensate, the software reiterates train commands every 6 clock cycles for train 1, and every 5 clock cycles for train 2. The solution works but is far from optimal. The software should be rewritten to enforce the maximum transfer rate while ensuring that a train command is sent within a given maximum time.

The third software change concerns the crane interface on CPU B. Currently, the software checks if there is a loading request for crane 2. If there is a request, then crane 2 “loads” the waiting train. Only at this point does the software check for crane 3. This causes crane 3 to wait until crane 2 is serviced. A more multitasking approach is needed. When a closed loop controller for the cranes is implemented, the interface can be modified to run the crane controllers concurrently.

Chapter 6

Plant Model

This chapter discusses the DES plant model used for the testbed, as well as related issues. The modeling assumptions used in creating the plant model are discussed, followed by controllable events that originate with the controller. The basic plant elements are discussed as additional specifications that define how the fundamental elements interact with one another. In total, there are 59 modular specifications giving a composite plant model on the order of 10^{16} states. Finally, the plant model's TCT representation is discussed.

6.1 Modeling Assumptions

In designing the testbed's plant model, two assumptions were made to simplify the plant model. The primary assumption was that all trains are only allowed to traverse a given track section in one direction. The permitted direction for each track section is shown in Figure 6.1. This also assumes that trains do not travel in reverse. As mentioned in Section 4.5.1, this assumption prevents trains from being mistaken for the other, plus eliminates the possibilities of more complex collisions. Also, the assumption makes sensors 17 and 18 unreachable, while requiring remote switches 1, 2, 9, and 10 to be fixed; thus these sensors and switches can be excluded from the plant model.

The second assumption is that the electric cranes only load the trains. They do not unload. This simplifies the crane models and the crane supervisors.

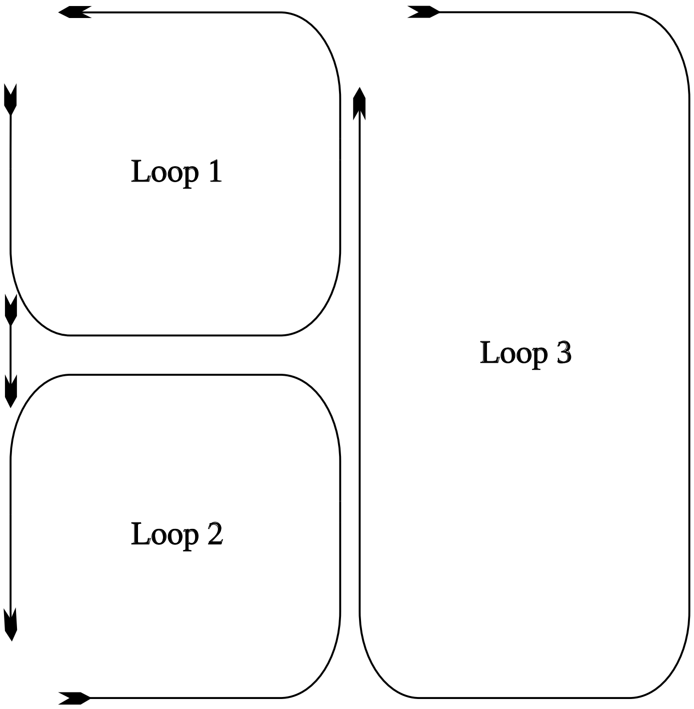


Figure 6.1: Permitted Track Traversals

6.2 Supervisor Generated Controllable Events

The standard RW theory model consists of a plant which spontaneously generates events, some controllable (can be disabled) and some uncontrollable, and a supervisor that tracks the plant's events, disabling controllable events according to some control law. This feedback loop interpretation is shown in Figure 6.2.

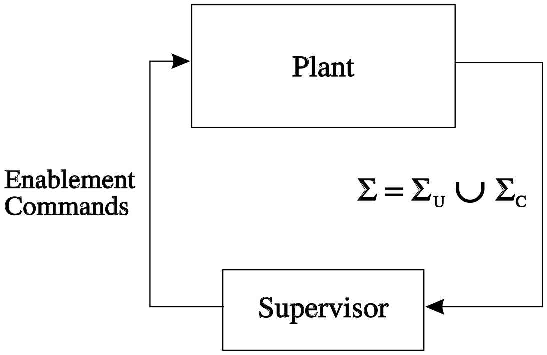


Figure 6.2: RW Feedback Loop

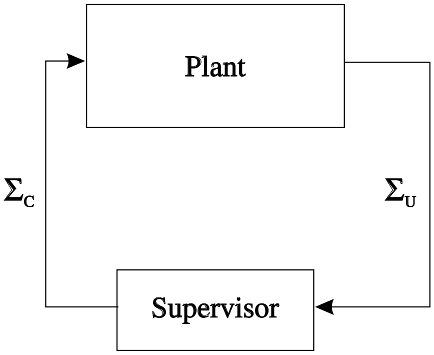


Figure 6.3: Balemi Feedback Loop

In [6], Balemi claims that this does not accurately represent most plants. He proposes the interpretation that the plant generates outputs (uncontrollable events) called "responses", and accepts inputs (controllable events) called "commands". Thus, the controllable events ("commands") are generated (forced) by the supervisor, when it wants a desired response from the plant. This interpretation is shown in Figure 6.3.

There are several problems with this. First, Balemi's interpretation unnecessarily complicates matters, in most cases. By introducing forcing, one now has to consider issues of temporal logic at the model and control synthesis stage. If the control specifications require temporal specifications, then they are unavoidable.

Secondarily, Balemi's interpretation is unnecessarily restrictive and leaves the standard RW theory. This requires the existing RW theory to be modified, where necessary to accommodate his new formalism. This is re-inventing the wheel. There is no need to leave current RW.

A third problem is that Balemi's formalism would make modular implementation extremely difficult. Several modular controllers might generate the same event at different times. It would be very difficult for them to come to a consensus on when, as a group, they should generate the event. Since most real systems are very complex, the only current feasible method of design is modular control. If Balemi's method cannot be implemented modularly, it is useless for most real systems of interest.

The following interpretation is proposed. The plant is considered to be that which generates the events in the plant models. This includes the normal plant generator and the supervisor. As discussed in Chapter 3, the important thing about a plant model is that it contains all the necessary events (for proper observability and controllability) and that the model is correct. By correct, it is meant that the plant model produces all possible strings that the plant can generate, and that it correctly shows all dependencies between events. Who generates these events (plant generator or supervisor) doesn't alter the theory as long as the events occur according to the model.

The set of controllable events is subdivided into two sets: Controllable events generated by the plant (Σ_{CP}), and controllable events generated by the supervisor (Σ_{CS}). For $\sigma \in \Sigma_{CP}$, the supervisor generates enablements as in the standard RW theory. For $\sigma \in \Sigma_{CS}$, the supervisor generates the event when it is enabled. The generated events are also fed back into the supervisor so it knows "when" the event occurs. Once the event is enabled, it will be generated by the supervisor "eventually", but the "when" is determined by the implementation method. The generation of the event is a function of the implementation, not part of the RW description of the supervisor. All the supervisor is concerned about is that events occur eventually in the desired order, and that undesirable event sequences do not occur. This interpretation is shown in Figure 6.4. Clearly, Balemi's and the traditional RW interpretation are special cases of the interpretation proposed in Figure 6.4. Refer to Section 9.3.4 to see how the proposed implementation method generates events, and how they are defined.

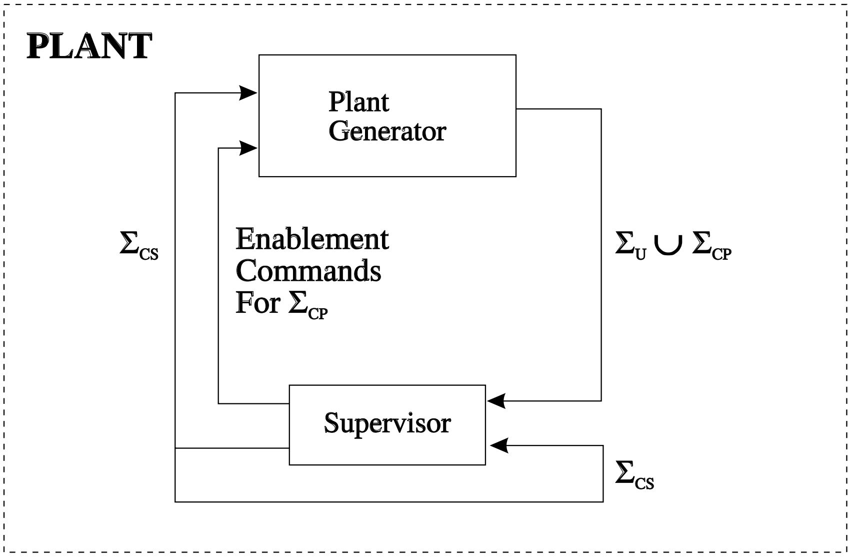

PLANT
Plant Generator
Σ_{CS}
Enablement Commands For Σ_{CP}
$\Sigma_U \cup \Sigma_{CP}$
Supervisor
Σ_{CS}

Figure 6.4: Proposed Feedback Loop Interpretation

To send commands to the plant, the supervisor disables the events it wants to generate, until the proper sequence of events has occurred, then it enables the desired event (command) and waits for it too occur. The supervisor then disables the event (unless it wants it to occur more than once). Since the supervisor monitors the events it generates, it can easily be designed and implemented as modular supervisors using the method described in Section 9.3.2.

An important point Balemi makes is that if the supervisor can generate events, then it is important that it does not do so when the plant generator is not ready to accept the events. For the new interpretation, this issue is not addressed for the proposed implementation method due to time constraints. In particular, it is not an important concern for the testbed since the plant model was designed to be flexible enough to allow for any contingency of the supervisor generating an event.

6.3 Fundamental Models

The plant model for the testbed consists of the following basic elements: trains, cranes, sensors and switches. They are shown in Figure 6.5. The models are discussed in the following sections.

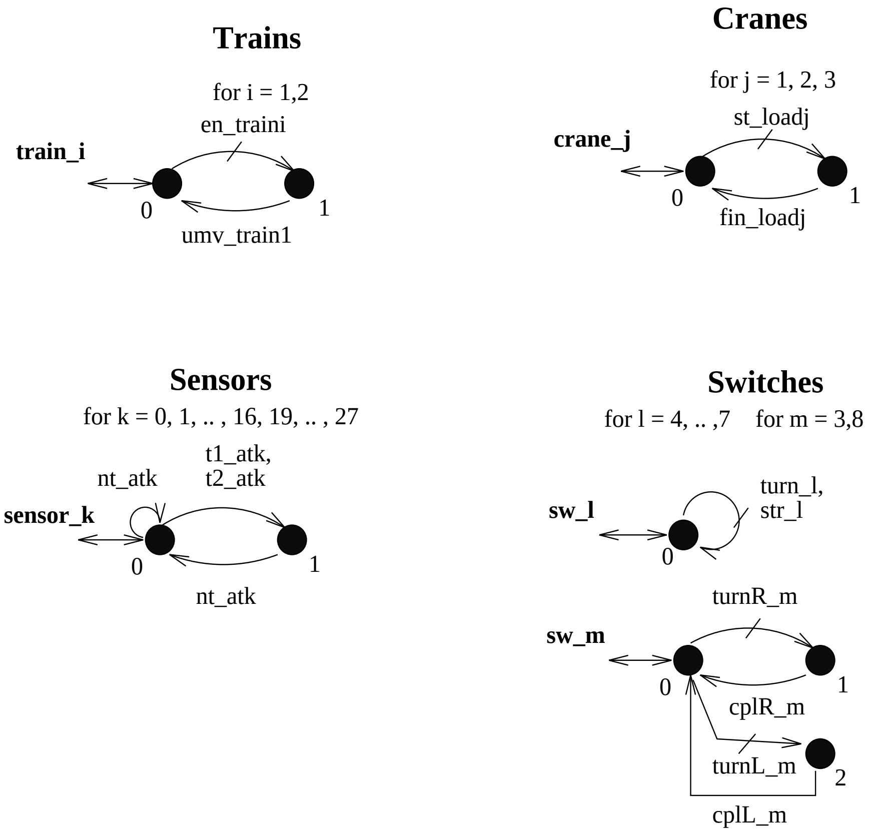


Figure 6.5: Fundamental Plant Models

6.3.1 Train Models

The train model shows how each train behaves. The model is designed to capture the idea that train i ($i = 1, 2$) starts moving forward when event *"en_traini"* occurs[1]. The train moves forward a unit distance and then checks to see if event *"en_traini"* has occurred again.

[1] Event *"en_traini"* occurring is referred to as the train being "enabled".

If it has, the train keeps moving. When the event is disabled, the train stops as soon as it has moved the unit distance permitted by the last occurrence of *"en_traini"*. This means the train can effectively be started or stopped by enabling/disabling event *"en_traini"*.

6.3.2 Crane Models

The crane models are designed to show basic loading functionality. When enabled, the crane begins its loading sequence, and then sends feedback when the task is complete.

6.3.3 Sensor Models

The sensor models tell us when a given train is present, and when no trains are present[2]. Also, they express the fact that only one train can activate a given sensor at one time, since only one train can physically occupy a stretch of track at one time.

6.3.4 Switch Models

As described in Section 4.6.1, the testbed contains two types of track switches. There are the two-way (straight or turn) switches (switches 4 to 7) and the three-way (straight, turn left, and turn right) switches (switches 3 and 8). The two way switches simply model the switch changing position. It is important to note that the turn and straight event should never be enabled at the same time since this could cause contention in the switches' solenoids.

For the three-way switches, only the turn mechanisms are modeled since part of the PLC code described in Section 10.3.3 prevents the straight position from being selected. Since it is more complicated to change the position of the three way switches, their model contains a completion event as well as an initialization event.

6.4 Interaction Models

The interaction models show the interdependencies of the fundamental models.

[2] As noted in Section 4.5.1, the event "no trains are present" for a given sensor, cannot be reliably observed.

6.4.1 Sensor Interdependencies

This series of models show the sensor's interdependencies. With respect to the starting position of a particular train, sensors can only be reached in a particular order, dictated by their physical location on the testbed. This is shown in Figure 6.6 and Figure 6.7.

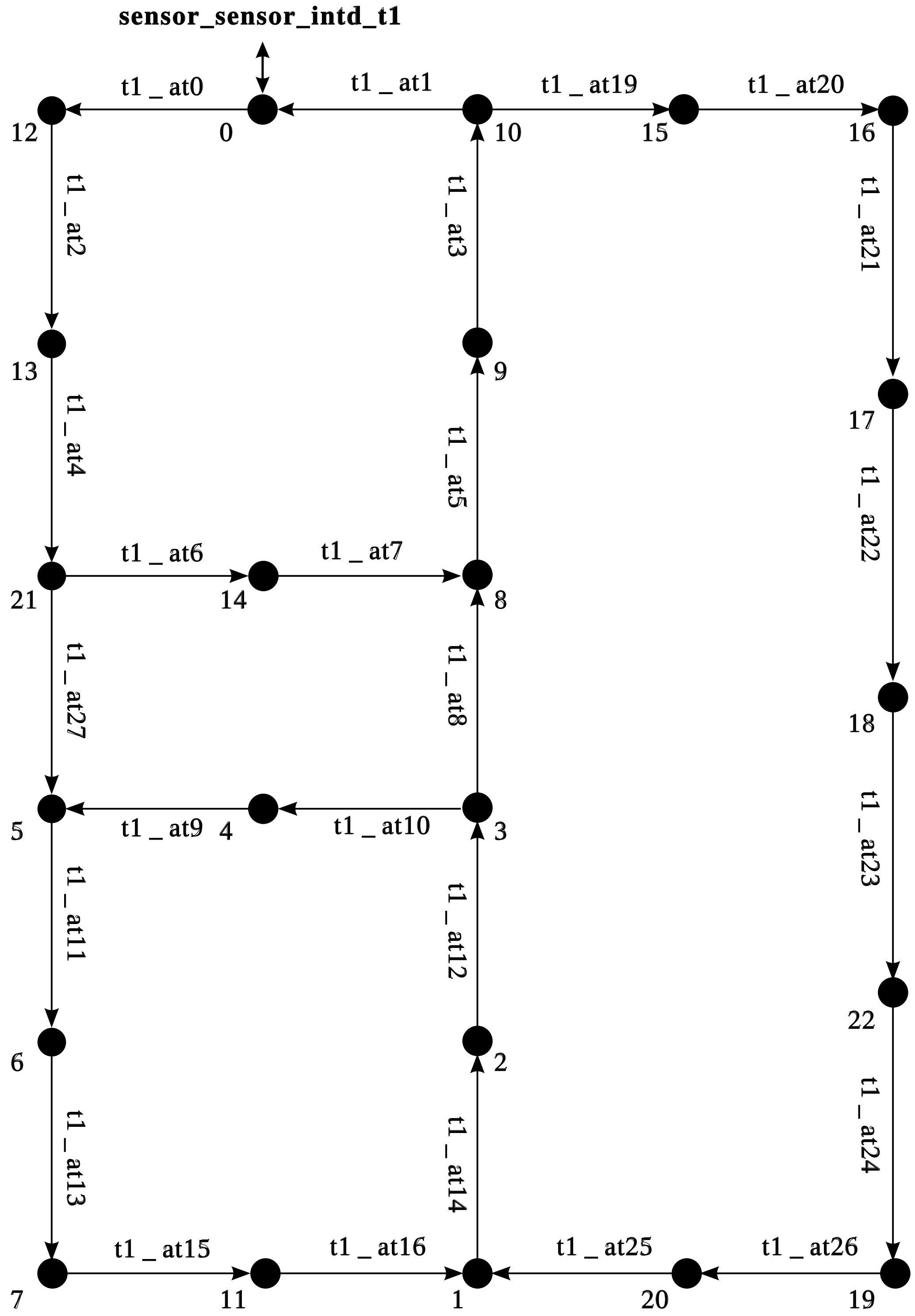


Figure 6.6: Sensor Interdependencies with Respect to Train 1

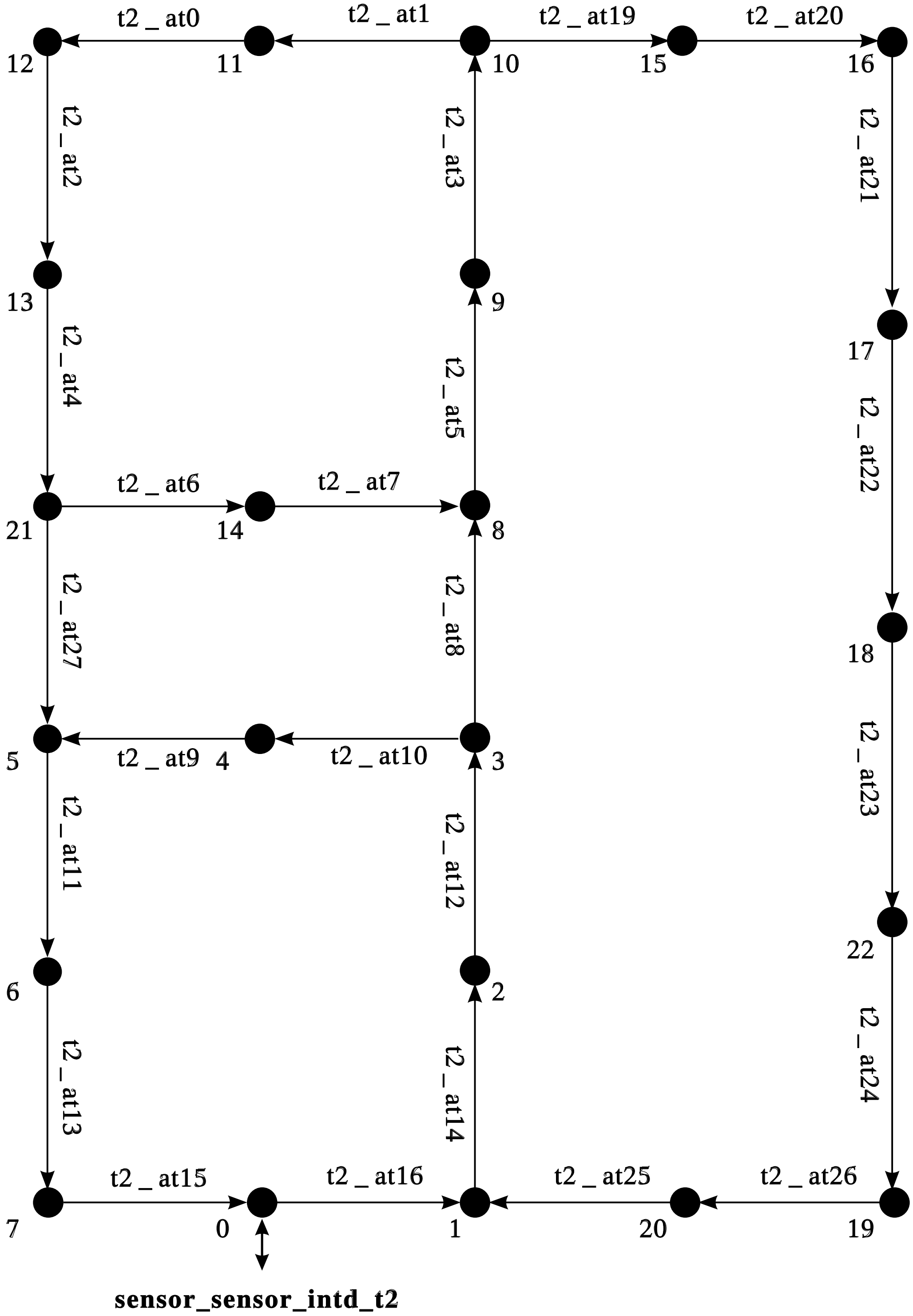
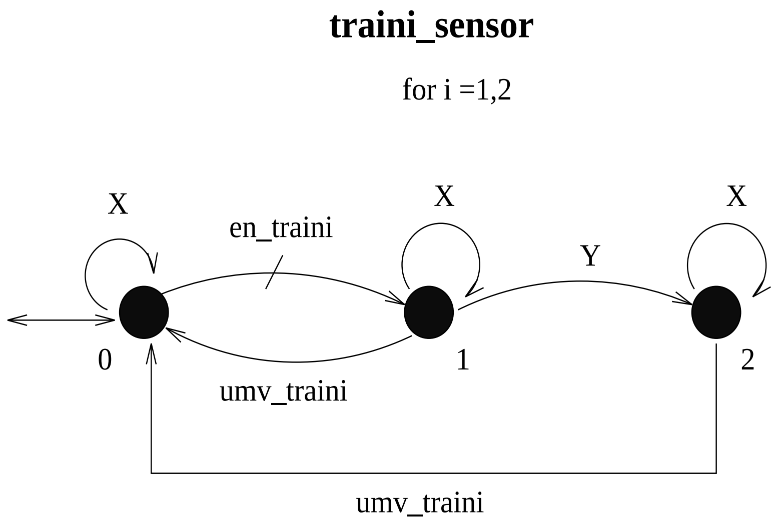


Figure 6.7: Sensor Interdependencies with Respect to Train 2

6.4.2 Sensor Dependencies on Trains

This series of models show how the sensors are dependent on train movement. A sensor is activated by a train passing the sensor. If all trains are stationary, then no sensor can be activated. Once a train starts moving, then it is possible that a sensor might be activated. When a train stops moving, then sensors can no longer be activated. As explained in Section 6.3.1, once a train has been "enabled", it moves a unit distance before it needs to be "enabled" again. This means that sensors can only be activated during the time between the train being "enabled" and the train moving a unit distance. During this time, at most one sensor can be activated. Figure 6.8 shows the plant sub-model that captures these ideas.



X = {nt_at0, . . . ,nt_at16, nt_at19, . . . ,nt_at27}

Y = {traini_at0, . . . , traini_at16, traini_at19, . . . , traini_27}

Figure 6.8: Sensor Dependency on Trains

6.4.3 Sensor Dependencies on Switches

The next series of models detail the sensor's dependency on the switches. Depending on the current location of a given train, certain sensors are inaccessible if the switch prevents the train's approach. Figures 6.9, 6.10, 6.11, 6.12, 6.13, 6.14, 6.15, 6.16, 6.17, 6.18, 6.19 and 6.20 show how the sensors depend on each switch, with respect to each train.

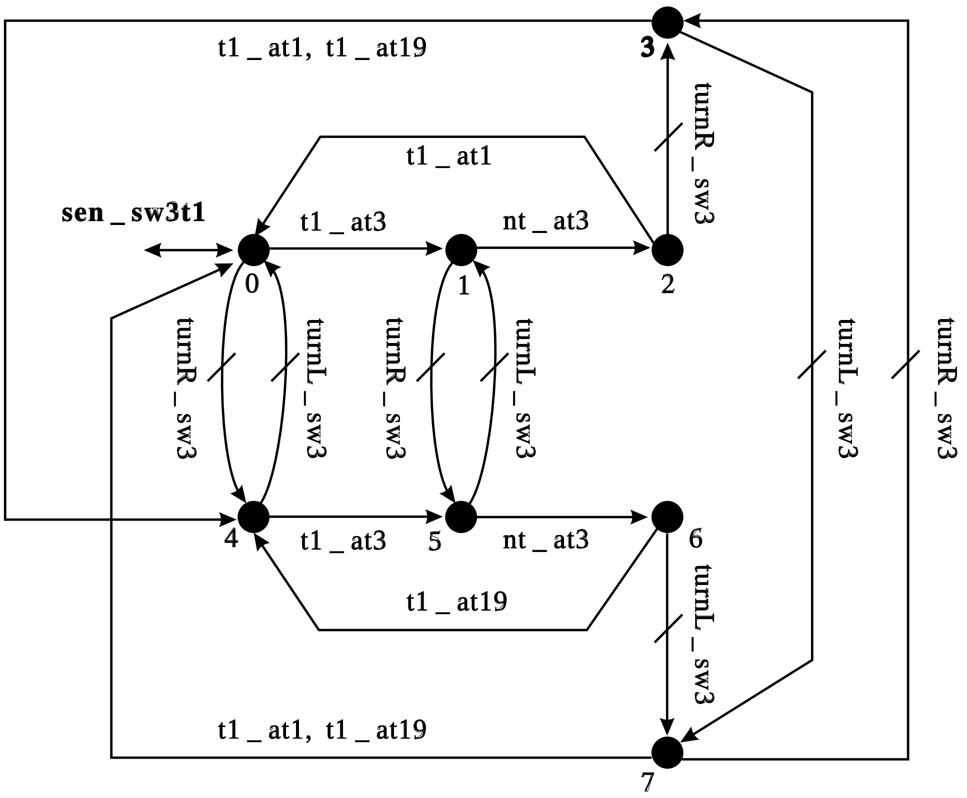


Figure 6.9: Sensor Dependency for Switch 3 with Respect to Train 1

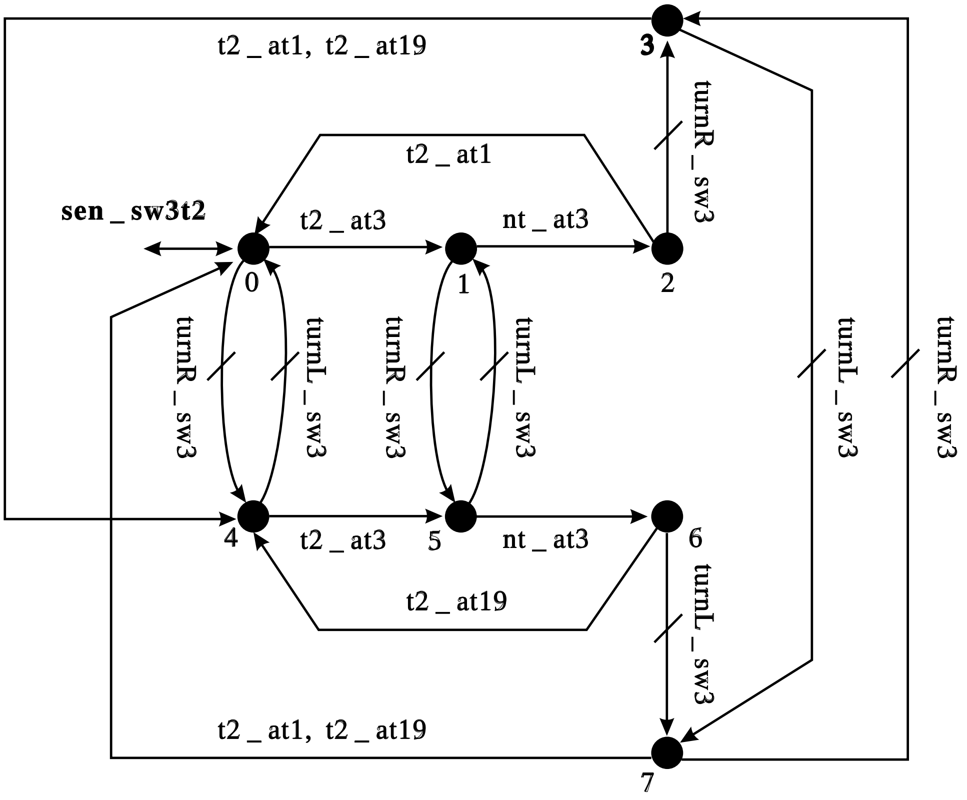


Figure 6.10: Sensor Dependency for Switch 3 with Respect to Train 2

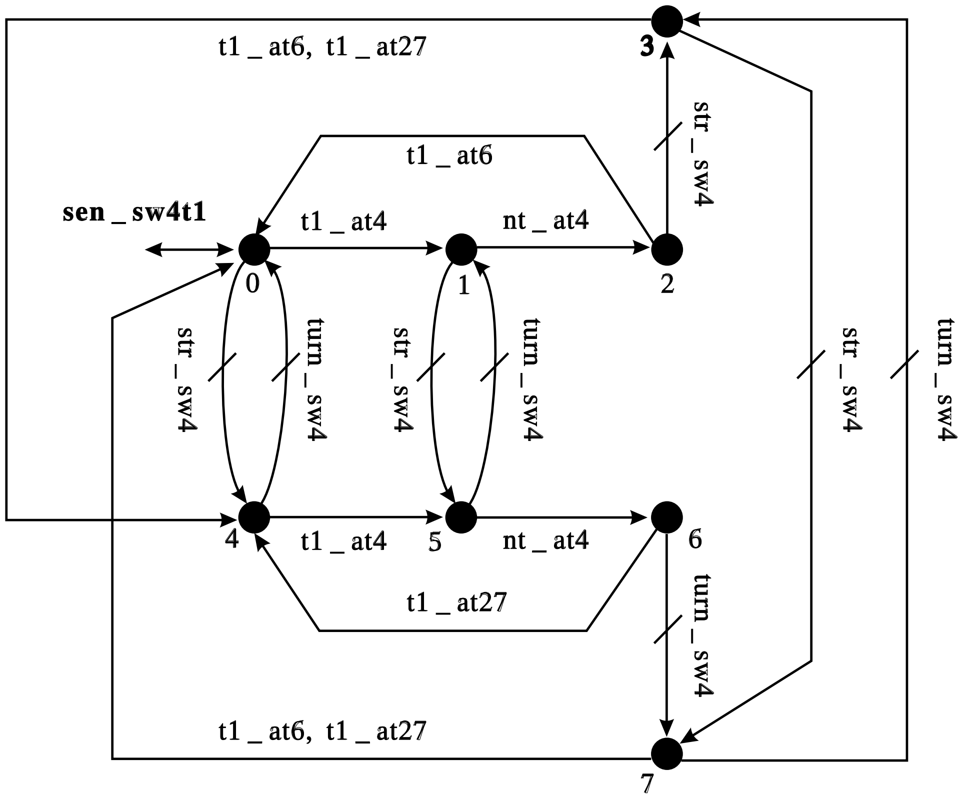


Figure 6.11: Sensor Dependency for Switch 4 with Respect to Train 1

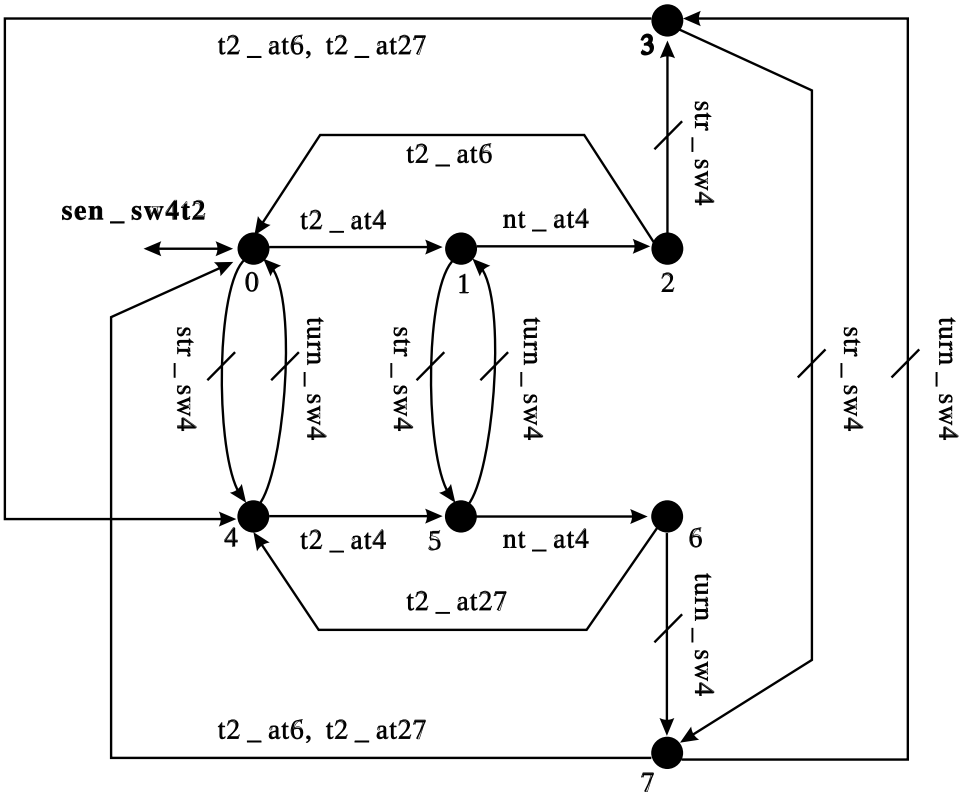


Figure 6.12: Sensor Dependency for Switch 4 with Respect to Train 2

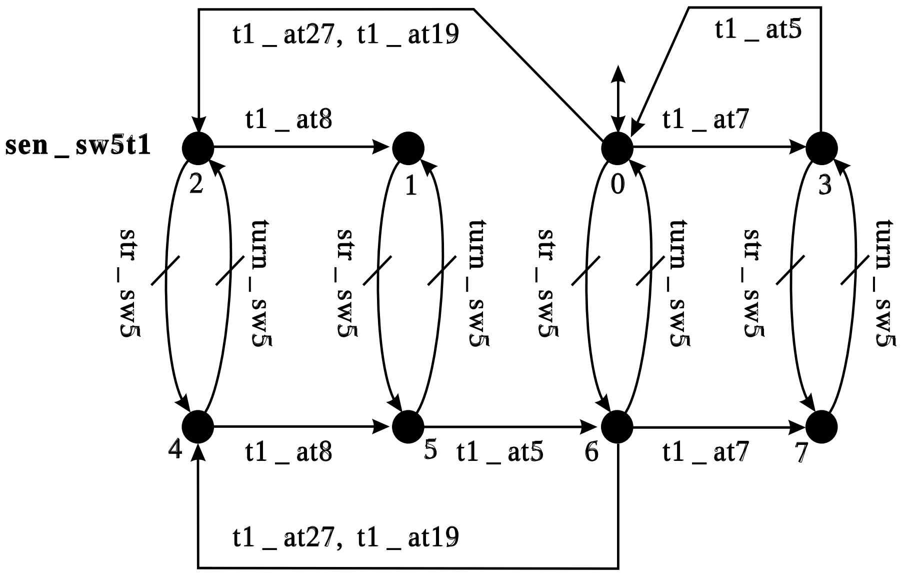


Figure 6.13: Sensor Dependency for Switch 5 with Respect to Train 1

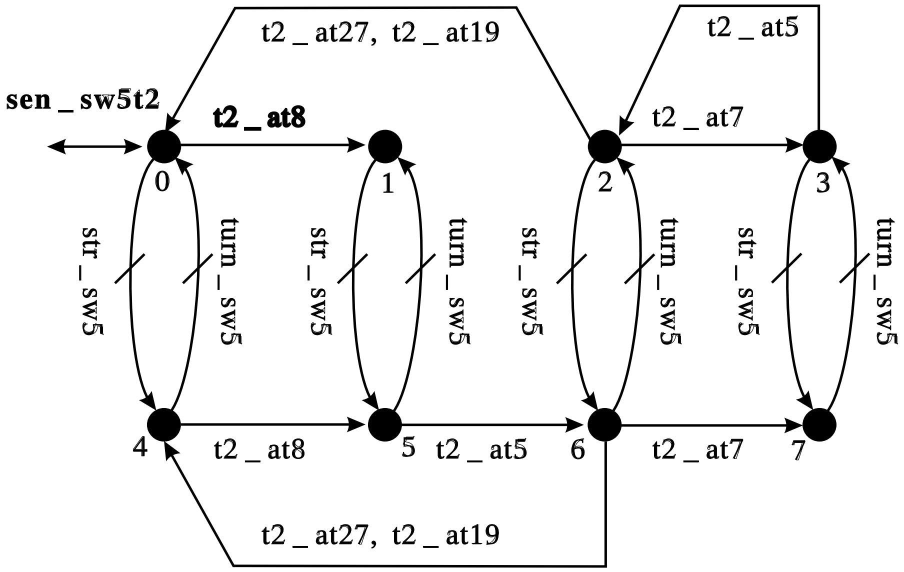


Figure 6.14: Sensor Dependency for Switch 5 with Respect to Train 2

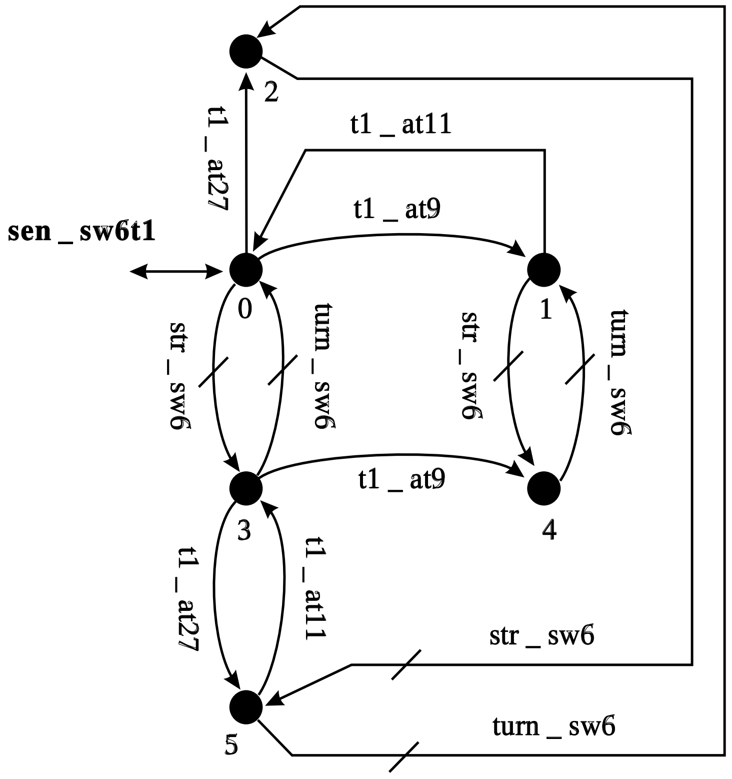


Figure 6.15: Sensor Dependency for Switch 6 with Respect to Train 1

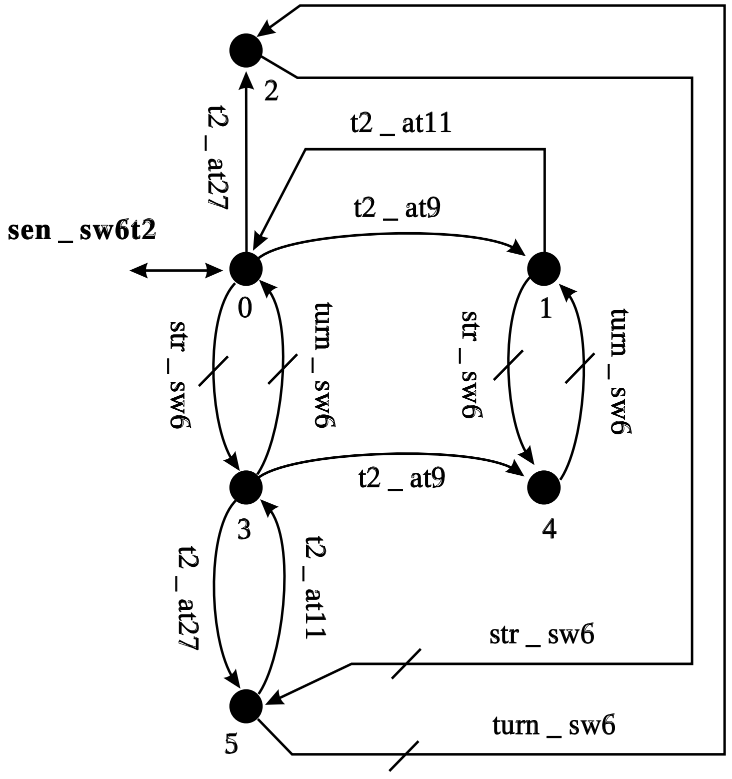


Figure 6.16: Sensor Dependency for Switch 6 with Respect to Train 2

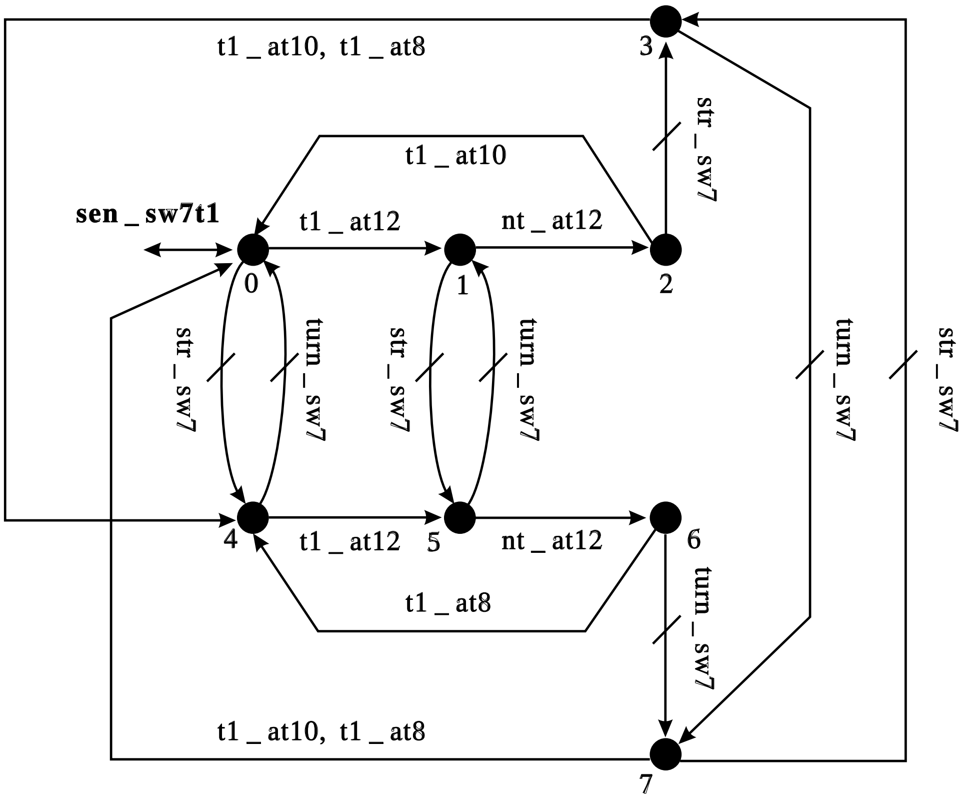


Figure 6.17: Sensor Dependency for Switch 7 with Respect to Train 1

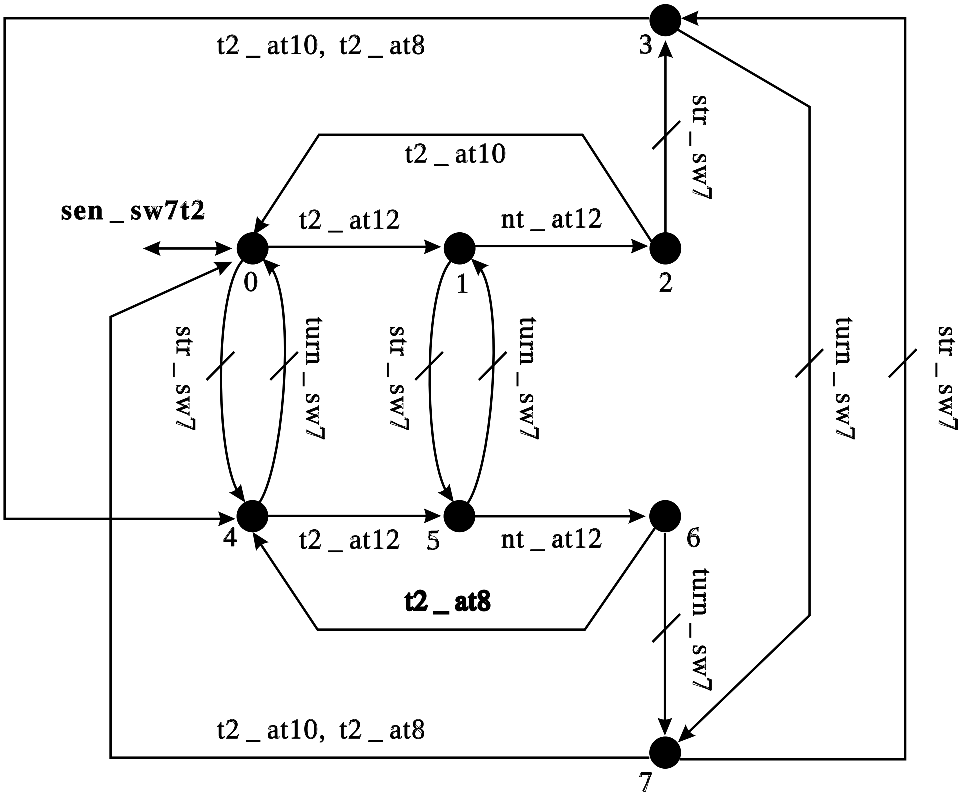


Figure 6.18: Sensor Dependency for Switch 7 with Respect to Train 2

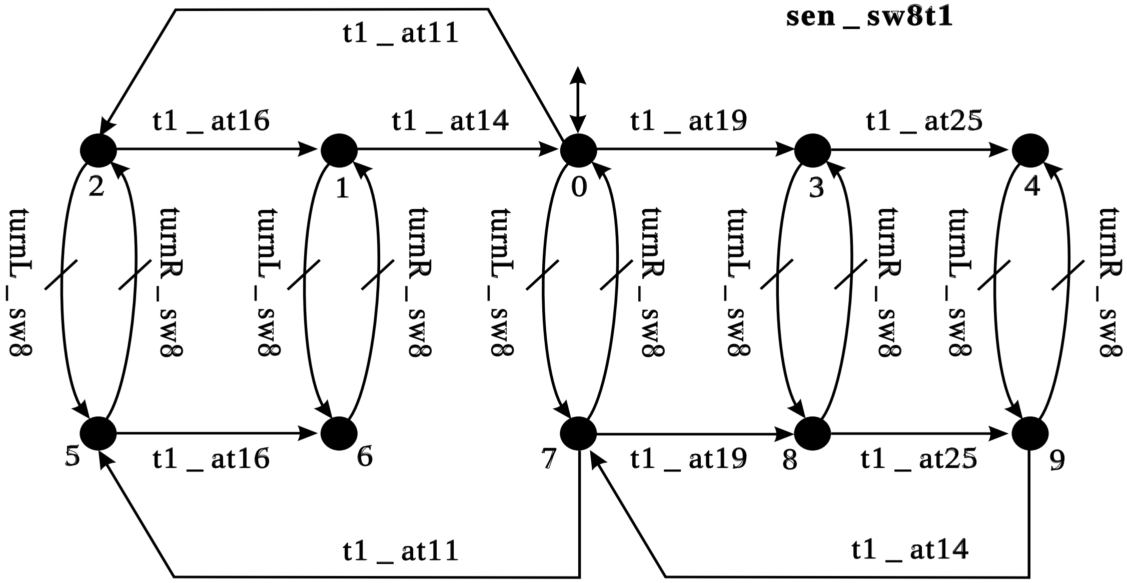


Figure 6.19: Sensor Dependency for Switch 8 with Respect to Train 1

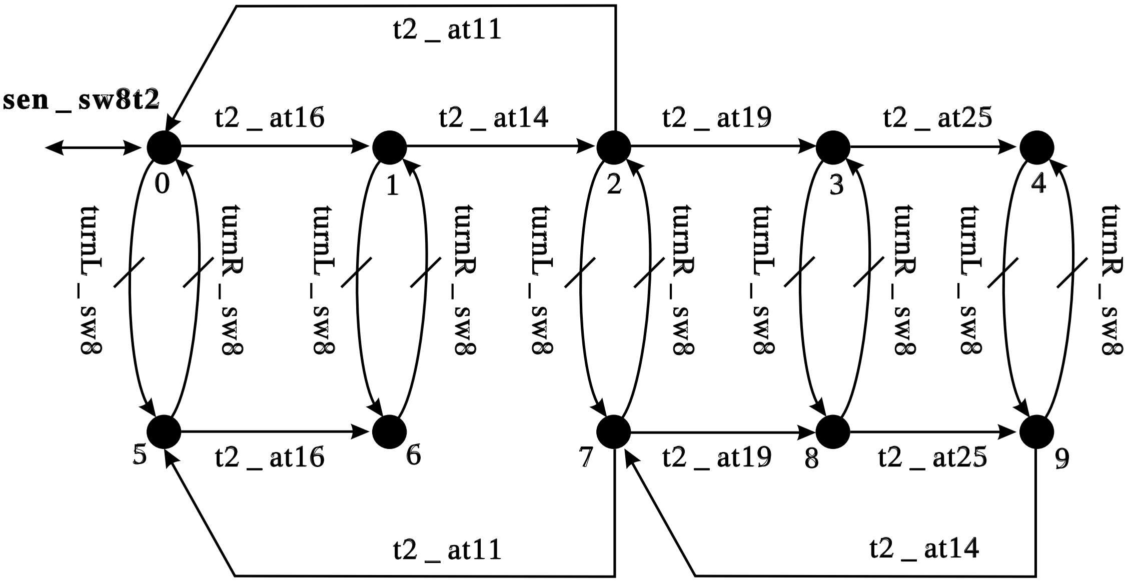


Figure 6.20: Sensor Dependency for Switch 8 with Respect to Train 2

6.4.4 Switch Request Handlers

The final series of models is slightly different from the previous series. This series, the switch request handlers, manages how the controllers interact with the switches. For the switches, it is desired that the switch "change position" events should be disabled until needed. This is currently extremely difficult to express as a modular DES supervisor.

The switch handlers were designed as an ad-hoc solution. They disable the switches' activation events until a supervisor specifically "requests" (by enabling the request event for the desired switch) one of the events. The request handlers are shown in Figure 6.21.

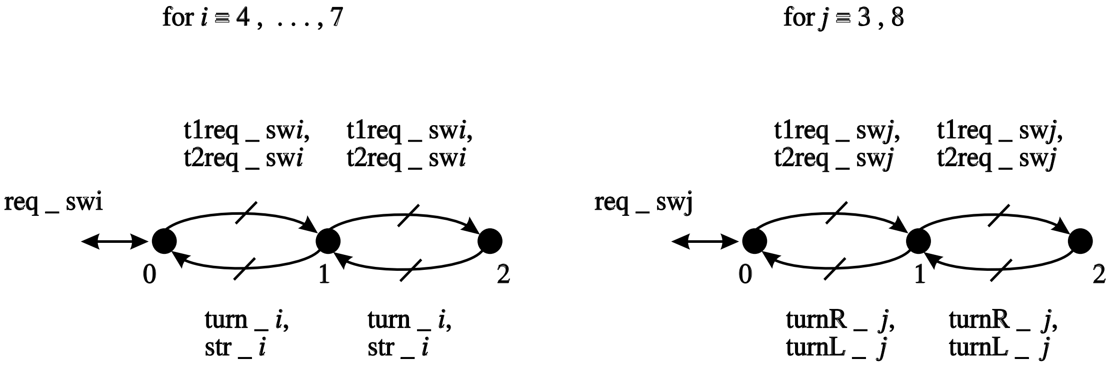


Figure 6.21: Switch Request Handlers

6.5 Event Map for TCT

For evaluation purposes, all plant models were entered into DES structures for use with TCT. The state labeling shown in the diagrams was used, while Table 6.1 shows the TCT label assigned to each event label.

Event Label	*TCT Label*	*Event Label*	*TCT Label*	*Event Label*	*TCT Label*	*Event Label*	*TCT Label*
en_train1	11	fin_load2	20	str_6	67	t2req_sw4	43
umv_train1	16	st_load3	35	turn_6	69	t1req_sw5	51
en_train2	21	fin_load3	30	str_7	77	t2req_sw5	53
umv_train2	26	turnR_3	307	turn_7	79	t1req_sw6	61
for $i = 0, \ldots, 16, 19, \ldots, 27$		turnL_3	309	turnR_8	807	t2req_sw6	63
t1_ati	i2	cplR_3	300	turnL_8	809	t1req_sw7	71
t2_ati	i4	cplL_3	302	cplR_8	800	t2req_sw7	73
nt_ati	i8	str_4	47	cplL_8	802	t1req_sw8	81
st_load1	15	turn_4	49	t1req_sw3	31	t2req_sw8	83
fin_load1	10	str_5	57	t2req_sw3	33		
st_load2	25	turn_5	59	t1req_sw4	41		

Table 6.1: TCT Event Map

Chapter 7

DES Supervisors

After the plant model was developed, 29 modular supervisors were designed to enforce the given control specifications. There are three types of supervisors: crane supervisors, routing supervisors, and collision protection supervisors. The supervisors as well as the control specifications are discussed below.

The figures shown, depicting the individual supervisors, are complete definitions of the supervisors except for the omission of unimportant self-looped events required for logical completeness. All events that are part of the plant model defined in Chapter 6, but are not shown in the diagram of the supervisor, are self-looped at every state in the indicated supervisor. Additional events that are self looped only at certain states in the supervisor, will be specified in tables.

7.1 Control Specifications

The testbed's safety and operation specifications are given below. They are discussed in detail in the following subsections.

1. Prevent trains from colliding.

2. Ensure switches are set correctly while train traverses them.

3. Enforce specified route.

4. Synchronize trains to permit loading by cranes.

7.1.1 Collision Prevention

The first specification is obvious. As the trains traverse the track along their assigned routes, it is desired to prevent rear collisions as well as collisions in intersections. Since trains only traverse a section of tracks in one direction, head-on collisions are not an issue.

7.1.2 Ensuring Correct Switch Positions

The second specification has two purposes. The first purpose is to ensure that switches are set correctly for routing the trains. The second purpose is to prevent a switch from changing position while a train is traversing the switch. This requires allowing only one train to control a switch at one time.

There are no supervisors specifically dedicated to this specification, but instead the routing, and collision protection supervisors cleverly subsume this specification into their primary mandates. This means that the routing, and collision protection supervisors already perform most of the necessary tasks to meet the above specifications. The supervisors only required slight modifications and then they were able to fully meet the above specifications.

The routing, and collision protection supervisors implement the switch position specifications in the following manner. The routing supervisors are the only supervisors that control the switches. They only change the switch's position when a train is just in front of the switch. Then, the collision protection supervisors prevent more than one train from occupying a given track section. For a switch that can be accessed from only one track section, this means that the switch moves to the correct position when the train arrives and stays there until the train has traversed the switch. Since no other train can enter the track section until the first train has crossed the switch, there will be no contention for the switch.

For switches that can be reached from two different track sections simultaneously (such as switch 6: see Figure 4.11), the situation is more complicated. The collision protection supervisor for the track section[1] that two trains are about to enter, disables the switch request event for the second train that arrives at the switch. When the first train has cleared the switch, the second train is released, and its switch request event can now occur. By disabling the second train's switch request event, the collision protection supervisor

[1] For switch 6, this is the section containing sensors 11 and 13.

prevents the train from changing the switch's position.

7.1.3 Routing Specifications

Each train is assigned a set route through the three work units of the cell. They repeat the route indefinitely[2]. The route for train 1 is as follows: Traverse left side of loop 1. Continue straight to left side of loop 2. Traverse loop 2 until right side then continue straight and into loop 3. Traverse loop 3 and then continue into loop 1. Repeat.

The route for train 2 is as follows: Traverse loop 2, loop 1, loop 3 and then repeat.

7.1.4 Crane Specifications

Specifically, the crane specifications say that, whenever a train arrives at a crane, the train is to be stopped. The crane then loads the train. When the crane is finished loading and returns to its starting position, the train is released.

7.2 Modular Supervisors

The supervisor for the testbed is implemented as several modular supervisors. Modular supervisors were chosen because centralized methods are not feasible for a system of this size. Besides the fact that it would be very difficult to conceive of the design of one large supervisor that implements all of the above specification (not to mention that the accuracy of such a design would be highly questionable), it would exceed our current software capacity to verify controllability for such a supervisor since we would be unable to break the plant into sections as is done in the model reduction theorems in Chapter 8.

In additional to feasibility issues, modular supervisors are easier (and more accurate in general) to design, modify, and implement. Implementing modular supervisors using the method given in Section 9.3, permits one to easily convert the 25 supervisors discussed in this section into PLC code by hand. One single centralized supervisor would require computer software to do the conversion and the resulting PLC code would be far more massive than the modular code.

[2] Refer to Figure 4.1 to determine labeling of track loops.

7.3 Crane Supervisors

A crane supervisor continually monitors the sensor just in front of it for a train. When a train arrives, the supervisor stops the train and activates the crane, which then proceeds to load the train. When the crane finishes loading the train, the supervisor releases the train. The crane supervisors for crane 1, crane 2, and crane 3 are shown in Figure 7.1. Table 7.1 shows additional self-looped events.

	sup_crane1	**sup_crane2**	**sup_crane3**
State	*Event*	*Event*	*Event*
0	fin_load1	fin_load2	fin_load3
1	t1_at13	t1_at2	t1_at21
1	t2_at13	t2_at2	t2_at21
2	t1_at13	t1_at2	t1_at21
2	t2_at13	t2_at2	t2_at21

Table 7.1: Self-looped Events for Crane Supervisors

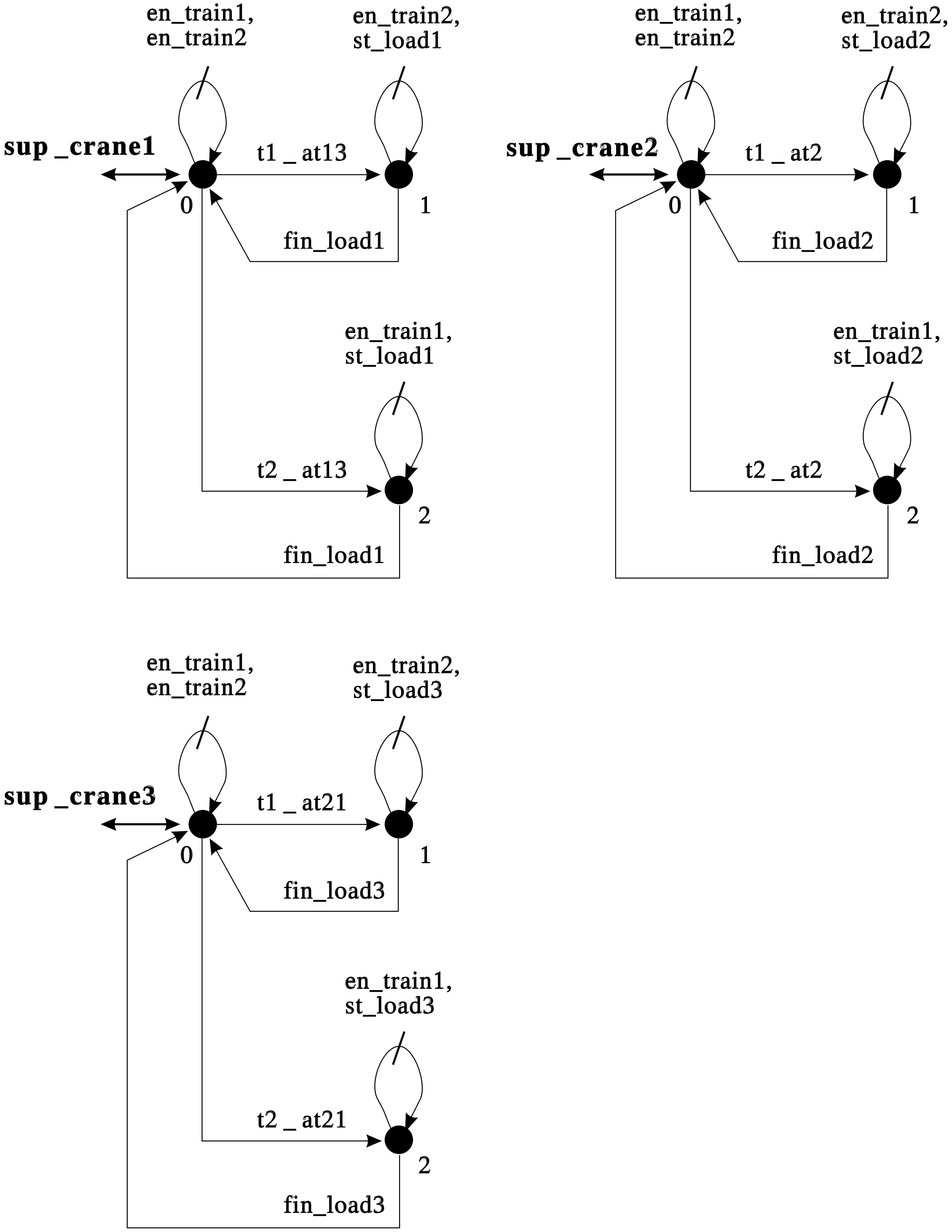


Figure 7.1: Crane Supervisors

7.4 Routing Supervisors

The routing supervisors enforce the specified route of each train. Each train has its own set of supervisors, one for each switch used (Switches 1, 2, 9, and 10 are ignored since they are not used - refer to section 6.1).

The path of each train is controlled by the position of the switches as the train passes through them. The required pattern of switch settings for each train's route was found by tracing the path of each train. This overall pattern was then further reduced by observing the individual pattern of each switch. Finally, a supervisor was written for each switch to generate the prescribed pattern as the given train traverses the tracks. When all six supervisors for a given train are put together, the overall pattern creates the proper route for the train.

7.4.1 Routing Supervisors for Train 1

The routing supervisors for train 1 are given in Figures 7.2, 7.3, 7.4, 7.5, 7.6, and 7.7. They are for switches 3 to 8. Table 7.2 shows additional self-looped events.

sup_rt1_sw3			**sup_rt1_sw4**	**sup_rt1_sw5**	**sup_rt1_sw6**	**sup_rt1_sw7**
Event	*States*	*States*	*Event*	*Event*	*Event*	*Event*
t1_at3	1, 2, 4, 5	1, 2	t1_at4	t1_at8	t1_at27	t1_at12
cplR_3	0, 1, 3, 4, 5					
cplL_3	0, 1, 2, 3, 4					

sup_rt1_sw8	
Event	*States*
t1_at16	1, 2, 3, 4, 5
t1_at25	0, 1, 2, 4, 5
cplR_8	0, 1, 3, 4, 5
cplL_8	0, 1, 2, 3, 4

Table 7.2: Self-looped Events for Routing Supervisors, Train 1

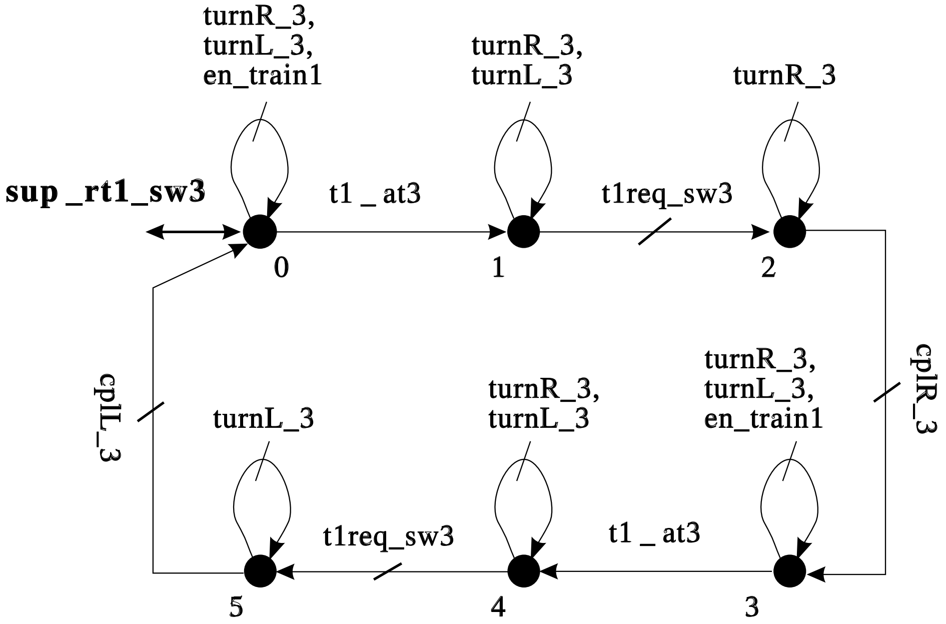


Figure 7.2: Routing Supervisor for Train 1, Switch 3

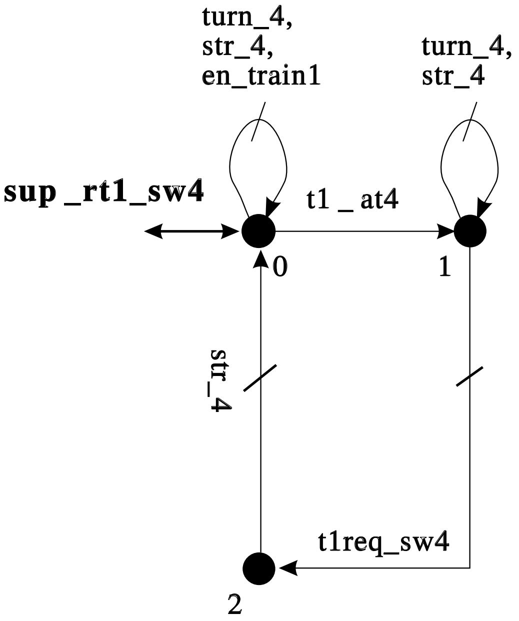


Figure 7.3: Routing Supervisor for Train 1, Switch 4

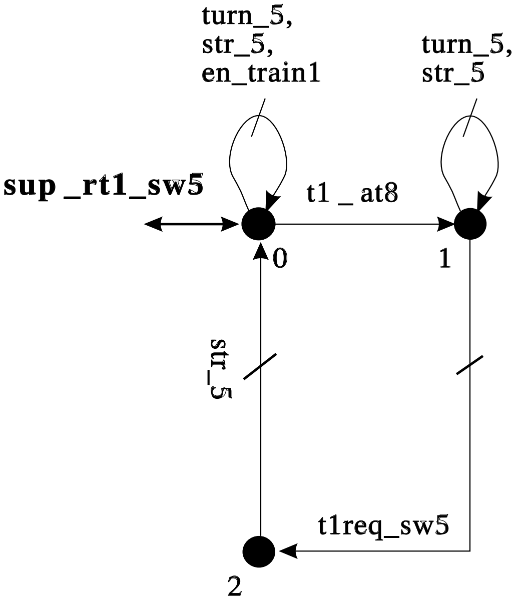


Figure 7.4: Routing Supervisor for Train 1, Switch 5

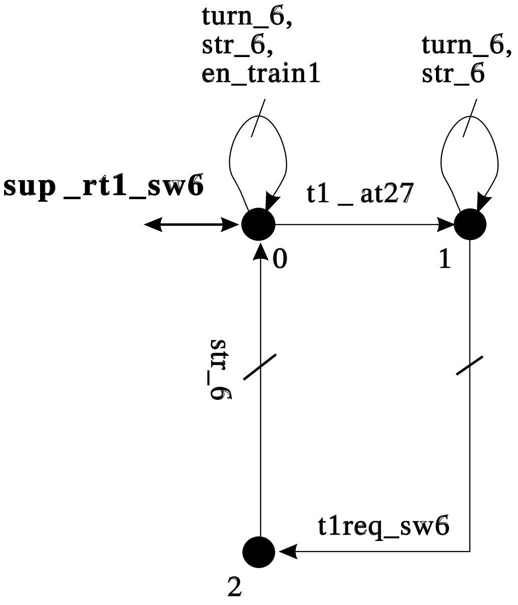


Figure 7.5: Routing Supervisor for Train 1, Switch 6

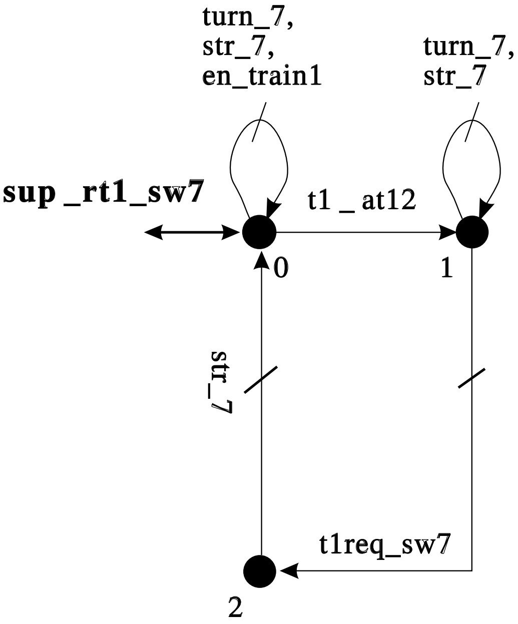


Figure 7.6: Routing Supervisor for Train 1, Switch 7

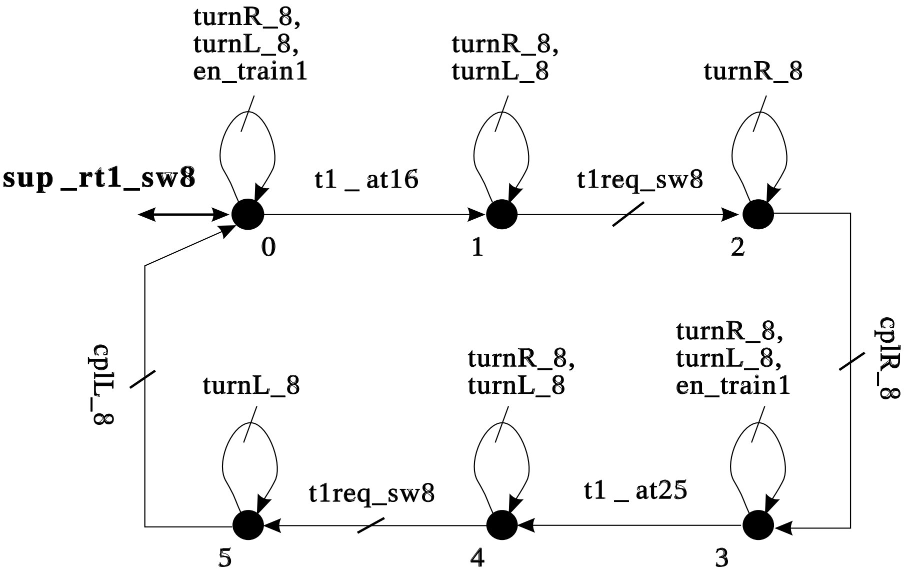


Figure 7.7: Routing Supervisor for Train 1, Switch 8

7.4.2 Routing Supervisors for Train 2

The routing supervisors for train 2 are given in Figures 7.8, 7.9, 7.11, 7.10, 7.12, and 7.13. They are for switches 3 to 8. Table 7.3 shows additional self-looped events.

sup_rt2_sw3		**sup_rt2_sw5**		**sup_rt2_sw7**	
Event	*States*	*Event*	*States*	*Event*	*States*
t2_at3	1, 2, 4, 5	t2_at8	1, 2, 3, 4, 5	t2_at12	1, 2, 3, 5, 6, 7
cplR_3	0, 1, 2, 3, 4	t2_at7	0, 1, 2, 4, 5	t2_at14	0, 2, 3, 4, 6, 7
cplL_3	0, 1, 3, 4, 5				
sup_rt2_sw8			**sup_rt2_sw4**	**sup_rt2_sw6**	
Event	*States*	*States*	*Event*	*Event*	
t2_at16	1, 2, 3, 4	1, 2	t2_at4	t2_at9	
t2_at25	1, 2, 3, 4				
cplR_8	0, 1, 3, 4				
cplL_8	0, 1, 2, 3				

Table 7.3: Self-looped Events for Routing Supervisors, Train 2

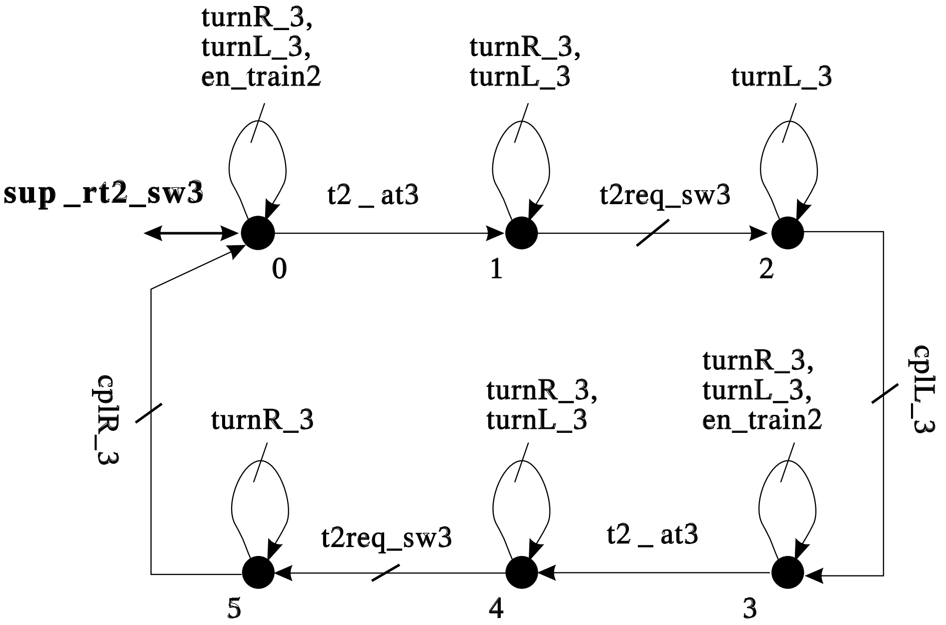


Figure 7.8: Routing Supervisor for Train 2, Switch 3

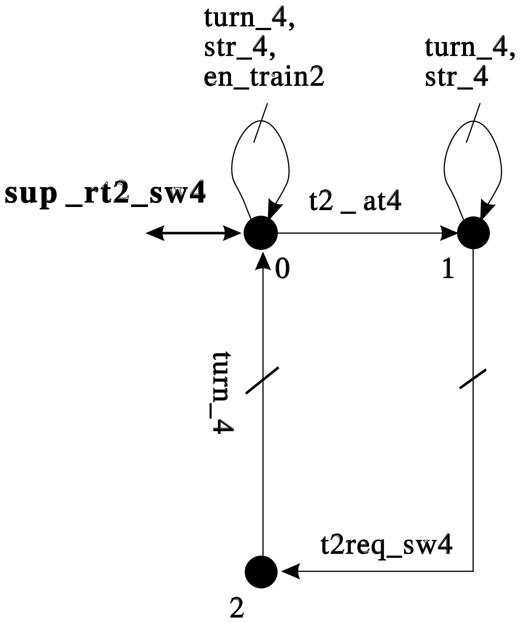


Figure 7.9: Routing Supervisor for Train 2, Switch 4

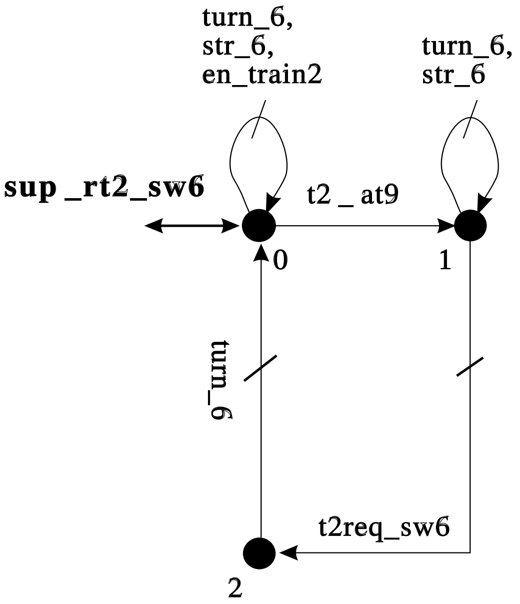


Figure 7.10: Routing Supervisor for Train 2, Switch 6

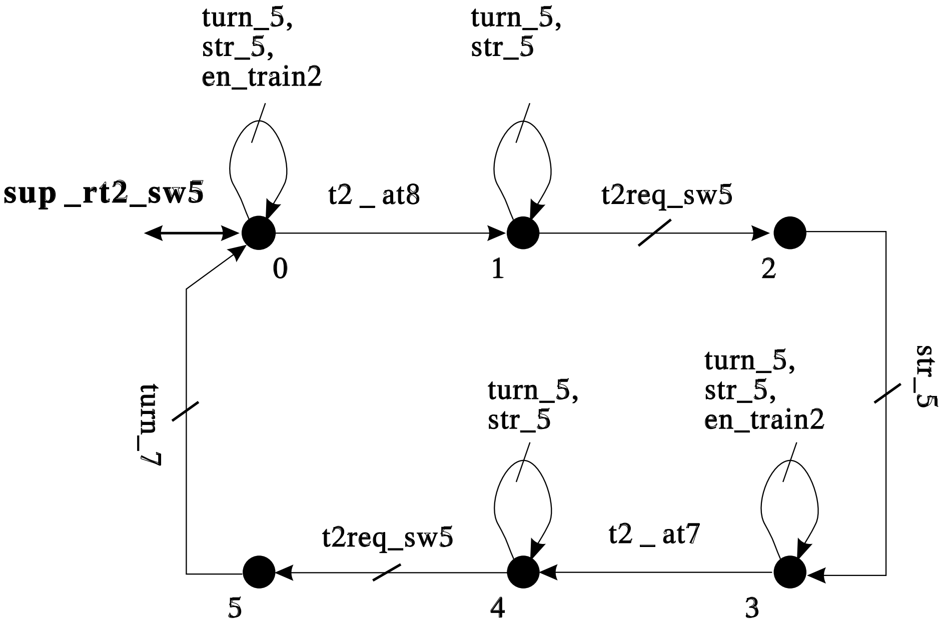


Figure 7.11: Routing Supervisor for Train 2, Switch 5

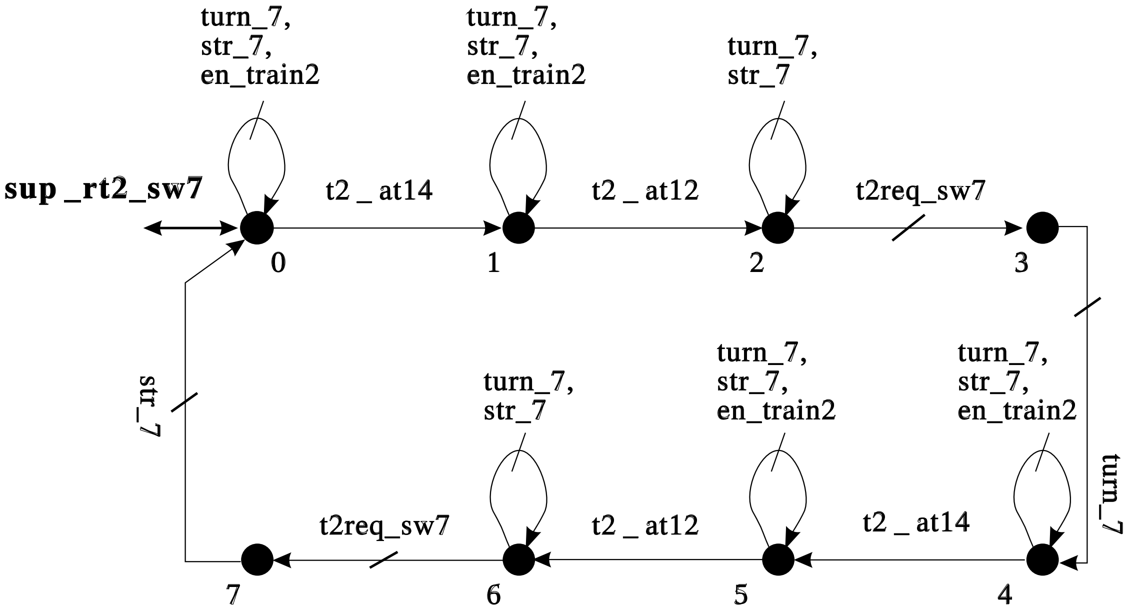


Figure 7.12: Routing Supervisor for Train 2, Switch 7

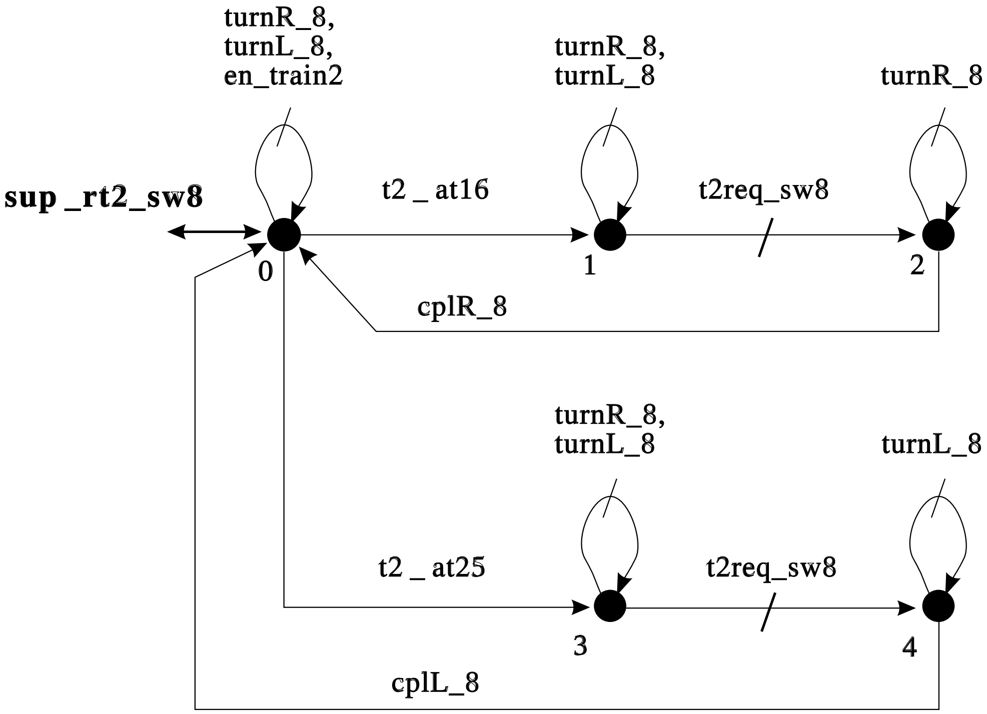


Figure 7.13: Routing Supervisor for Train 2, Switch 8

7.5 Collision Protection Supervisors

The final class of supervisors are the collision protection supervisors. Each track section (denoted by the two sensors it contains, such as sensors RS23 and RS24 in Figure 7.14) has its own supervisor that essentially permits only one train to occupy that portion of the track at any given time. Thus each train is handed off from one supervisor to the other, and together, the supervisors provided a net of safety for the trains as they travel.

The supervisors are broken down into three classes of supervisors of similar structure. This is done purely as a descriptive aid. Each class is described below.

7.5.1 Class I Supervisors

Class I supervisors are the simplest form of supervisors, and are all very similar to the supervisor for track section RS23 and RS24. The layout of track section RS23 and RS24 is shown in Figure 7.14. Sensor RS23 marks the start of the section, sensor RS26 marks the start of the adjoining section, and sensor RS22 marks the end of the preceding section.

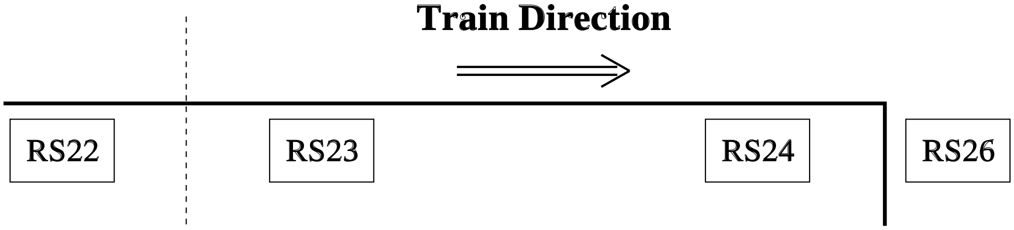


Figure 7.14: Track Section Containing RS23 and RS24

The supervisor **(sup_cprot_23_24)** monitors RS23 for a train to enter its area. When a train does (say train 1) it starts monitoring RS26 for train 1 to leave and RS22 for train 2. When train 1 reaches RS26, the supervisor starts over. If train 2 reaches RS22 before train 1 leaves the track section (RS26), then train 2 is stopped. When train 1 reaches RS26, train 2 is released. Supervisor **sup_cprot_23_24** is shown in Figure 7.15. The remaining class I supervisors are shown in Figures 7.16, 7.17, 7.18, 7.19, 7.20, and 7.21. Table 7.4 shows additional self-looped events.

sup_cprot_23_24		**sup_cprot_25_26**		**sup_cprot_21_22**		**sup_cprot_15_16**	
Event	*States*	*Event*	*States*	*Event*	*States*	*Event*	*States*
t1_at22	0, 1, 2, 4	t1_at24	0, 1, 2, 4	t1_at20	0, 1, 2, 4	t1_at15	0, 1, 3, 4
t2_at22	0, 2, 3, 4	t2_at24	0, 2, 3, 4	t2_at20	0, 2, 3, 4	t2_at15	0, 1, 3, 4
t1_at23	1, 2, 3, 4	t1_at26	1, 2, 3, 4	t1_at21	1, 2, 3, 4	t1_at13	1, 2, 3, 4
t2_at23	1, 2, 3, 4	t2_at26	1, 2, 3, 4	t2_at21	1, 2, 3, 4	t2_at13	0, 1, 2, 4
t1_at26	0, 3, 4	t1_at14	0, 3, 4	t1_at23	0, 3, 4	t1_at14	0, 1, 2
t2_at26	0, 1, 2	t2_at14	0, 1, 2	t2_at23	0, 1, 2	t2_at14	2, 3, 4
sup_cprot_2_4		**sup_cprot_rs27**		**sup_cprot_rs8**			
Event	*States*	*Event*	*States*	*Event*	*States*		
t1_at0	0, 1, 2, 4	t1_at4	0, 1, 2, 4	t1_at12	0, 1, 2, 4		
t2_at0	0, 2, 3, 4	t2_at4	0, 2, 3, 4	t2_at12	0, 2, 3, 4		
t1_at2	1, 2, 3, 4	t1_at27	1, 2, 3, 4	t1_at8	1, 2, 3, 4		
t2_at2	1, 2, 3, 4	t2_at27	1, 2, 3, 4	t2_at8	1, 2, 3, 4		
t1_at27	0, 3, 4	t1_at11	0, 3, 4	t1_at5	0, 3, 4		
t2_at27	0, 1, 2	t2_at11	0, 1, 2	t2_at5	0, 1, 2		
t1_at6	0, 3, 4						
t2_at6	0, 1, 2						

Table 7.4: Self-looped Events for Collision Protection Supervisors, Class I

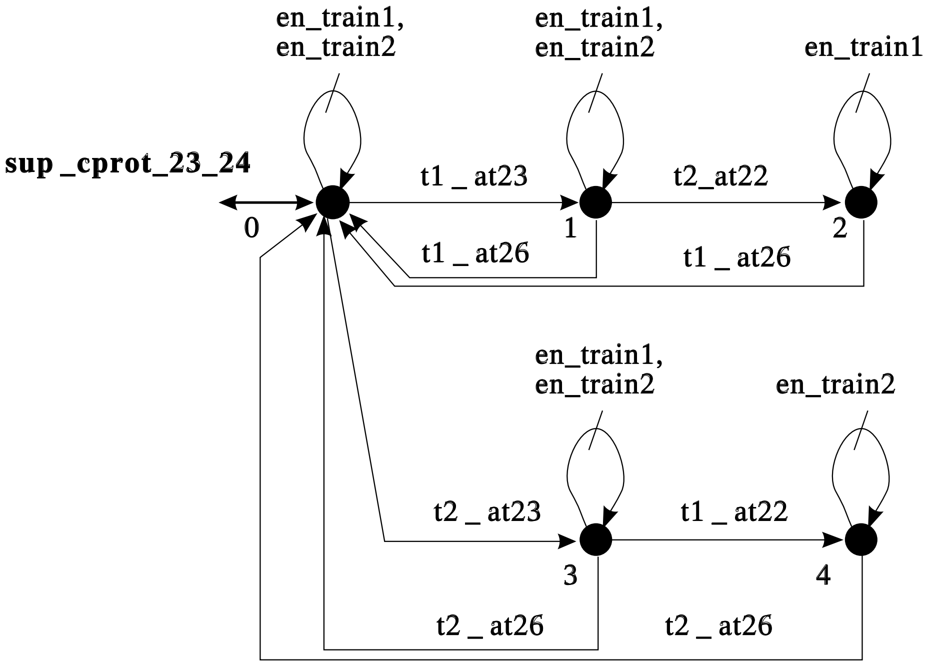


Figure 7.15: Collision Protection Supervisor for Section RS23 and RS24

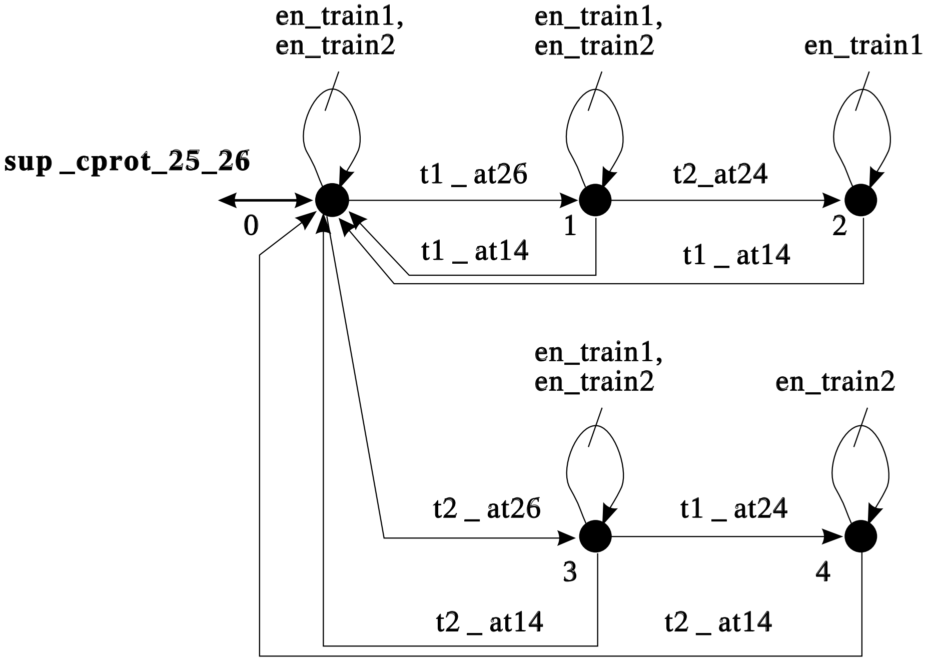


Figure 7.16: Collision Protection Supervisor for Section RS25 and RS26

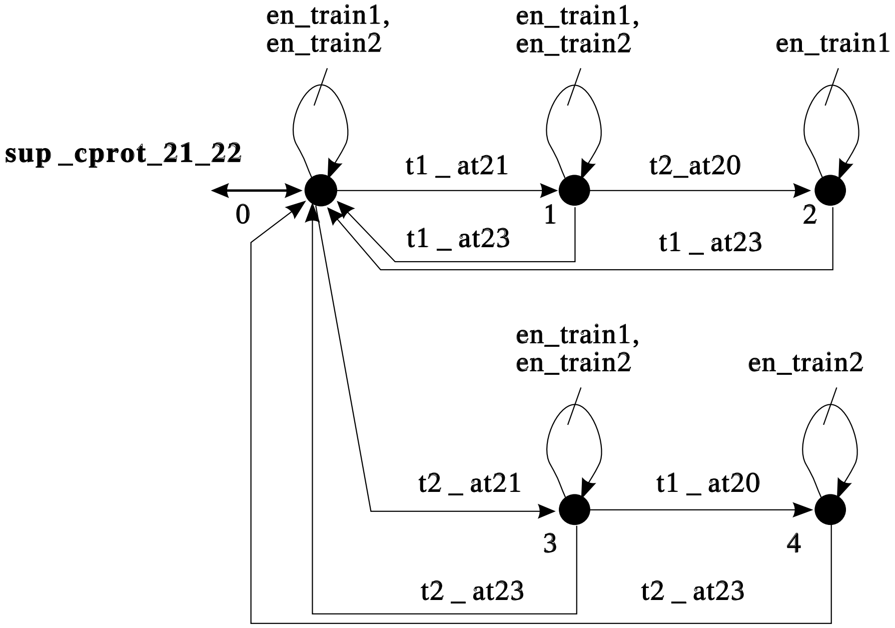


Figure 7.17: Collision Protection Supervisor for Section RS21 and RS22

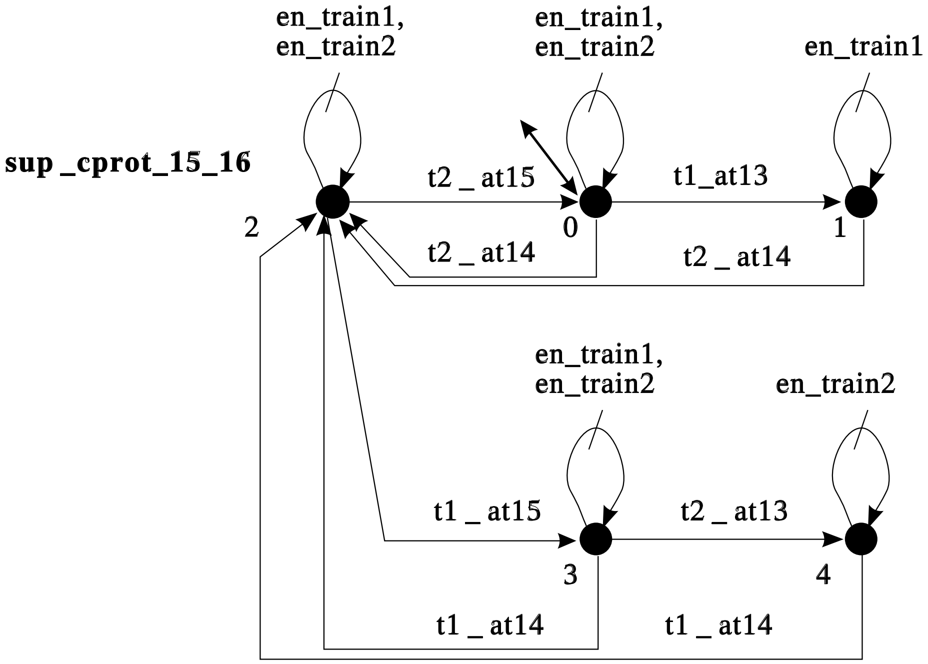


Figure 7.18: Collision Protection Supervisor for Section RS15 and RS16

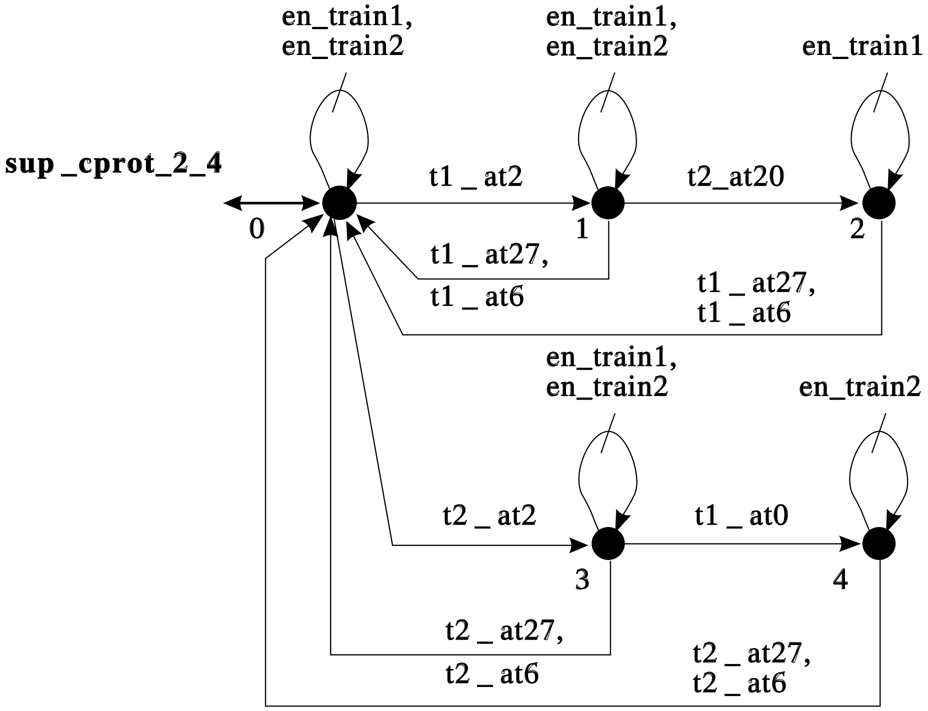


Figure 7.19: Collision Protection Supervisor for Section RS2 and RS4

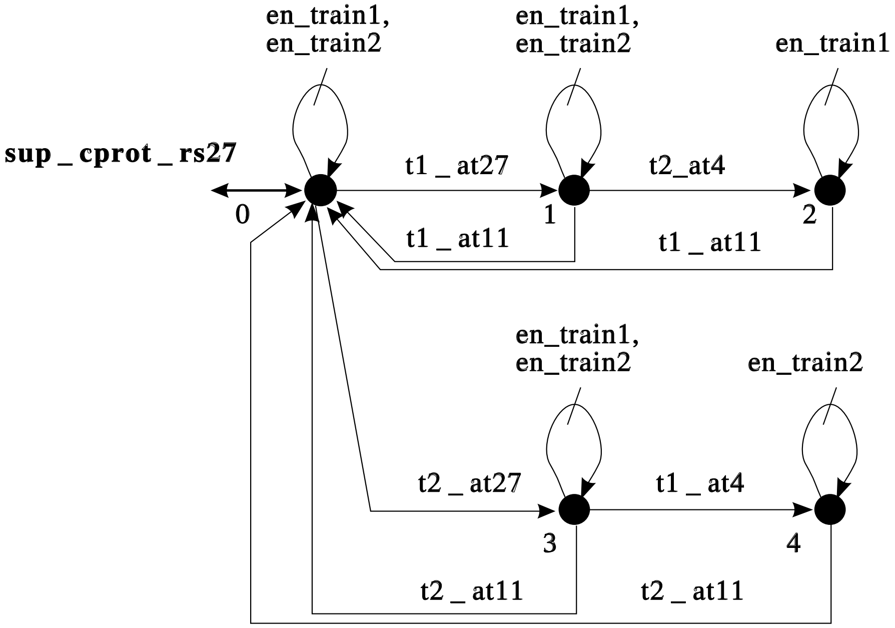


Figure 7.20: Collision Protection Supervisor for Section RS27

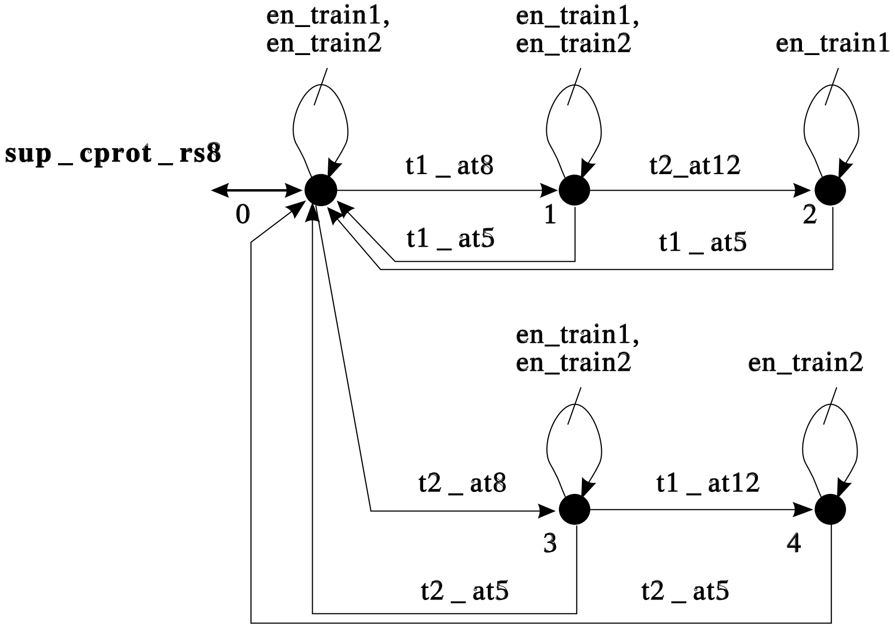


Figure 7.21: Collision Protection Supervisor for Section RS8

7.5.2 Class II Supervisors

Class II supervisors are similar in structure to class I supervisors, except they deal with track sections that can be accessed by two separate track sections. All class II supervisors are similar to the supervisor for track section RS11 and RS13. The layout of track section RS23 and RS24 is shown in Figure 7.22. The operation of the supervisor **(sup_cprot_11_13)** is similar to the operation of **sup_cprot_23_24,** except that **sup_cprot_11_13** monitors both incoming track sections for trains. Supervisor **sup_cprot_11_13** is shown in Figure 7.23. The remaining class II supervisors are shown in Figures 7.24 and 7.25. Table 7.5 shows additional self-looped events.

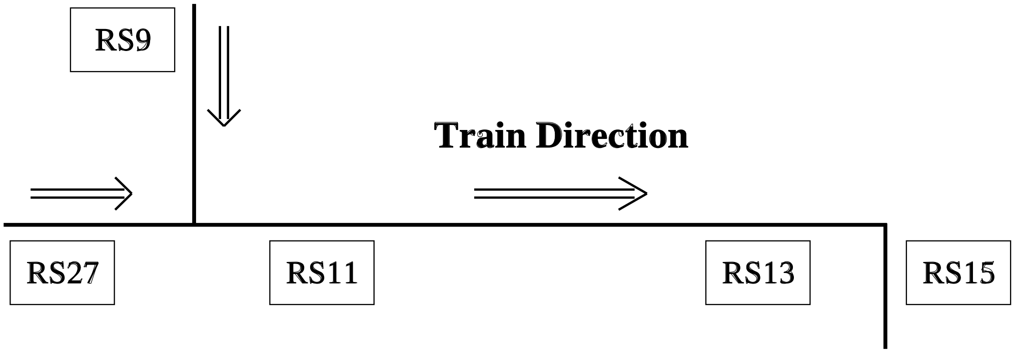


Figure 7.22: Track Section Containing RS11 and RS13

sup_cprot_11_13		**sup_cprot_12_14**		**sup_cprot_3_5**	
Event	*States*	*Event*	*States*	*Event*	*States*
t1_at9	1, 2, 4	t1_at16	1, 2, 4	t1_at7	1, 2, 4
t2_at9	3, 4, 2	t2_at16	3, 4, 2	t2_at7	3, 4, 2
t1_at27	1, 2, 4	t1_at25	1, 2, 4	t1_at8	1, 2, 4
t2_at27	2, 3, 4	t2_at25	2, 3, 4	t2_at8	2, 3, 4
t1_at15	0, 3, 4	t1_at8	0, 3, 4	t1_at19	0, 3, 4
t2_at15	0, 1, 2	t2_at8	0, 1, 2	t2_at19	0, 1, 2
		t1_at10	0, 3, 4	t1_at1	0, 3, 4
		t2_at10	0, 1, 2	t2_at1	0, 1, 2

Table 7.5: Self-looped Events for Collision Protection Supervisors, Class II

Also, as discussed in Section 7.1.2, class II supervisors help prevent switch contention. When the supervisor disables one of the trains to prevent a collision, it also disables (and later re-enables) the train's switch request event.

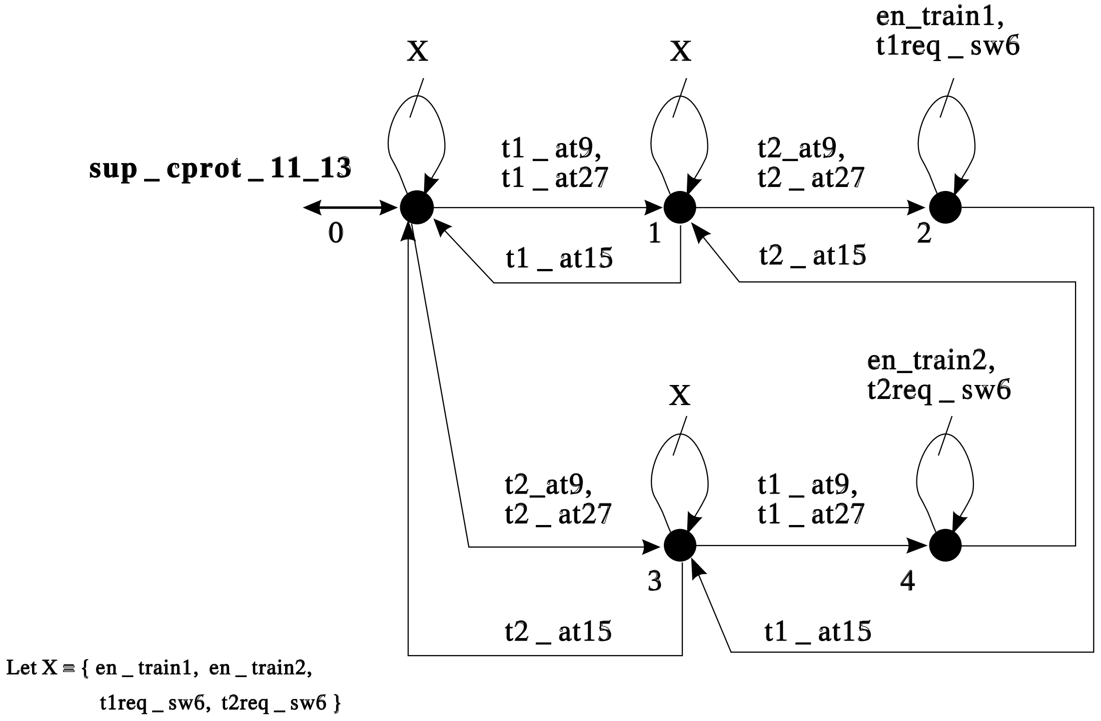


Figure 7.23: Collision Protection Supervisor for Section RS11 and RS13

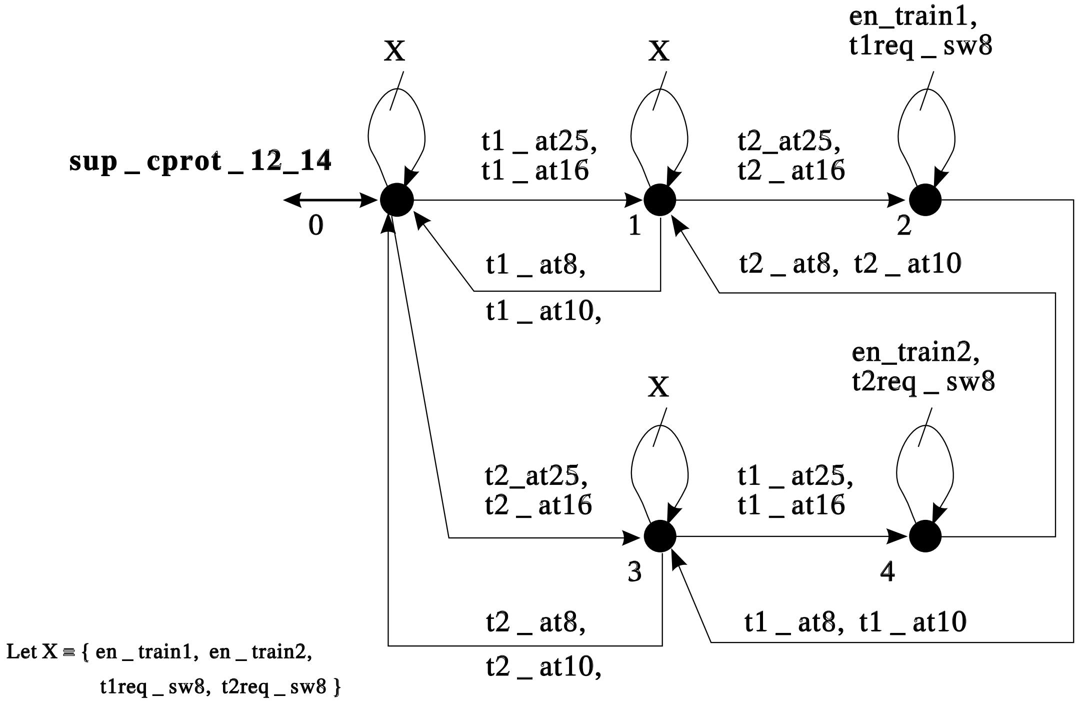


Figure 7.24: Collision Protection Supervisor for Section RS12 and RS14

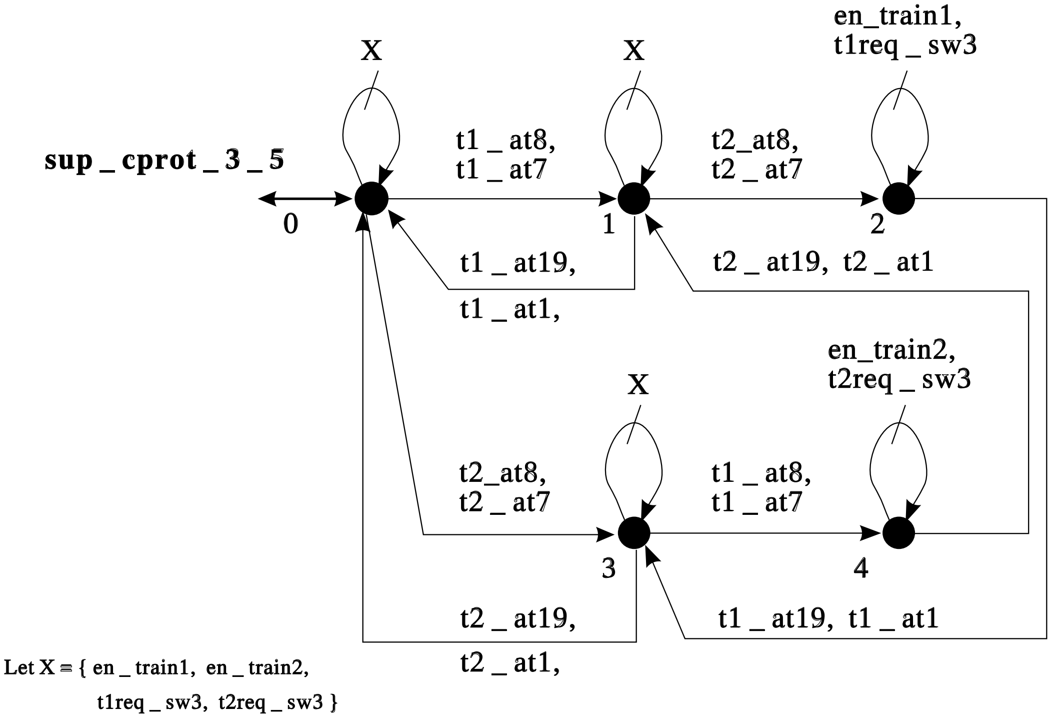


Figure 7.25: Collision Protection Supervisor for Section RS3 and RS5

7.5.3 Class III Supervisors

Class III supervisors are the most complicated of the collision protection supervisors. They deal with track sections like RS19 and RS20 (shown in Figure 7.26) that have one access section, but the train has a choice of two tracks to enter. This causes complications, since a supervisor wants to interfere as little as possible with a train that plans to enter the other track, but the supervisor has no knowledge of the train's intended path.

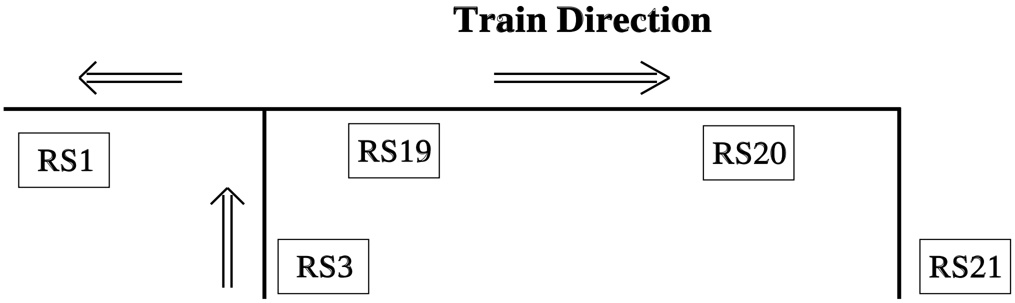


Figure 7.26: Track Section Containing RS19 and RS20

All class III supervisors are similar to supervisor **sup_cprot_19_20.** This supervisor operates similarly to **sup_cprot_23_24,** except that it releases a train waiting at RS3 when the first train reaches RS20 (an interior sensor), instead of at RS21 (the next section marker). This permits the waiting train to enter the other track section. If the train enters the section containing section RS19 instead, it is stopped at RS19 and forced to wait until the first train has left the section. Supervisor **sup_cprot_19_20** is shown in Figure 7.27. The remaining class III supervisors are shown in Figures 7.28, 7.29, and 7.30. Table 7.6 shows additional self-looped events.

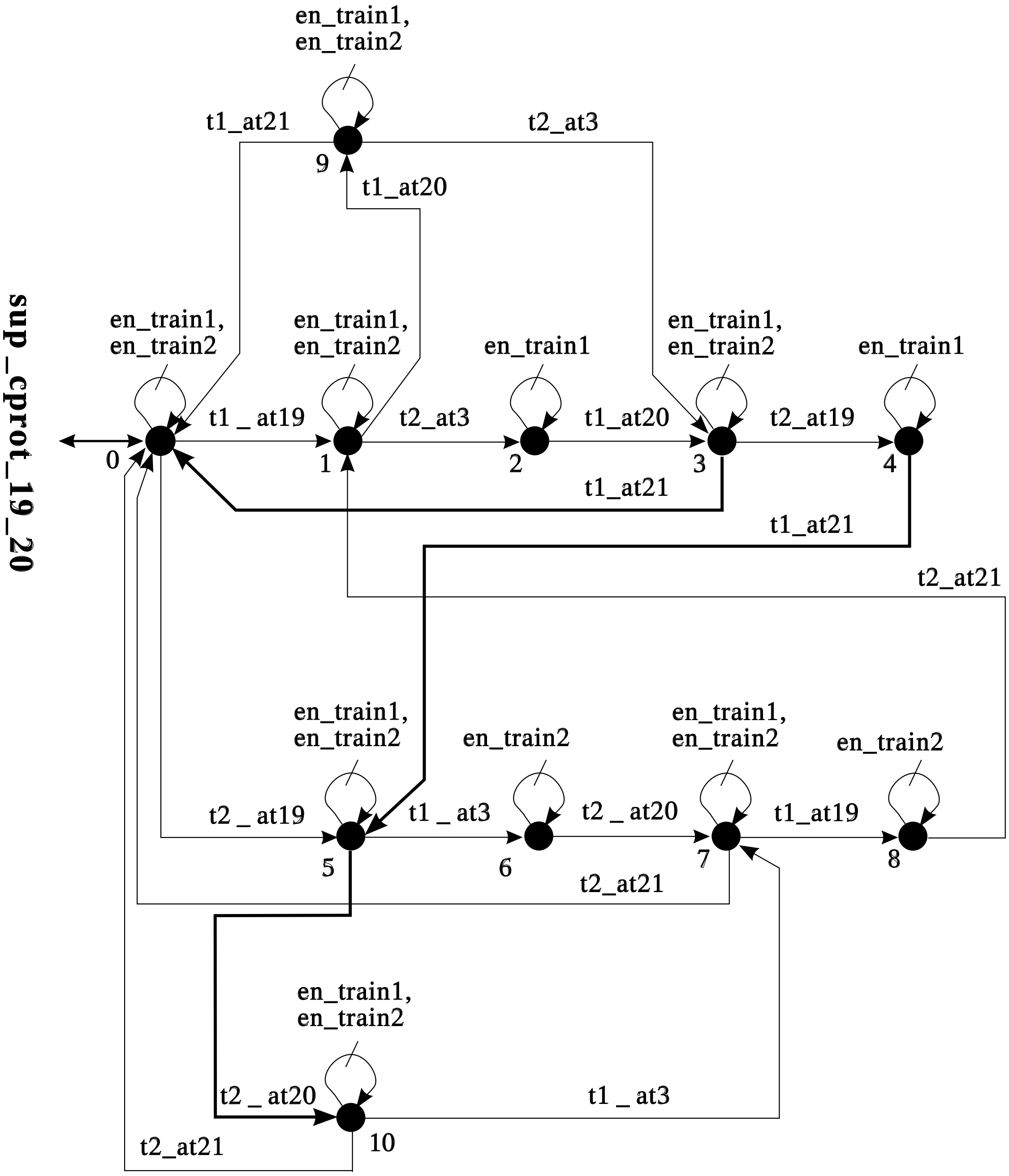


Figure 7.27: Collision Protection Supervisor for Section RS19 and RS20

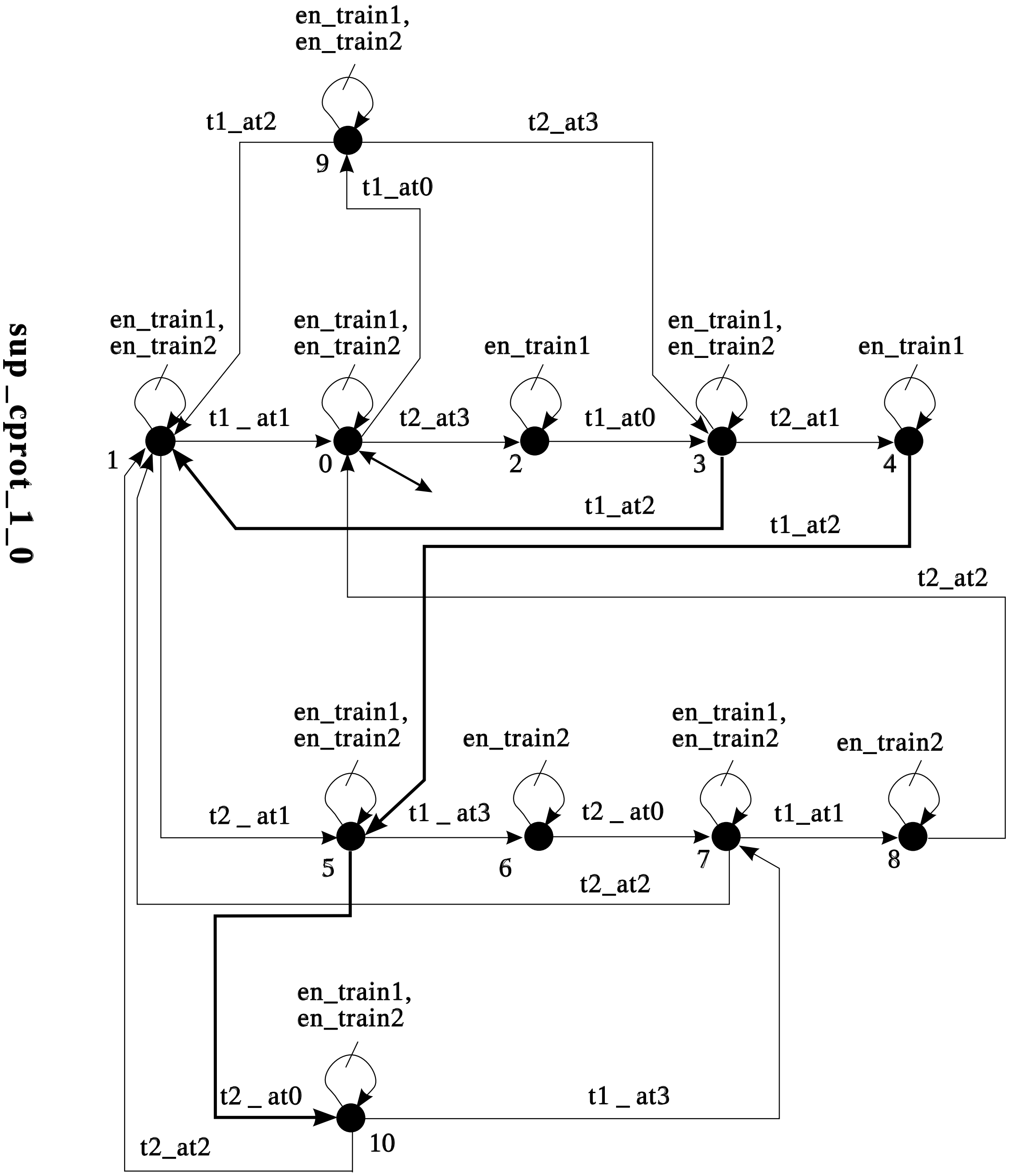


Figure 7.28: Collision Protection Supervisor for Section RS1 and RS0

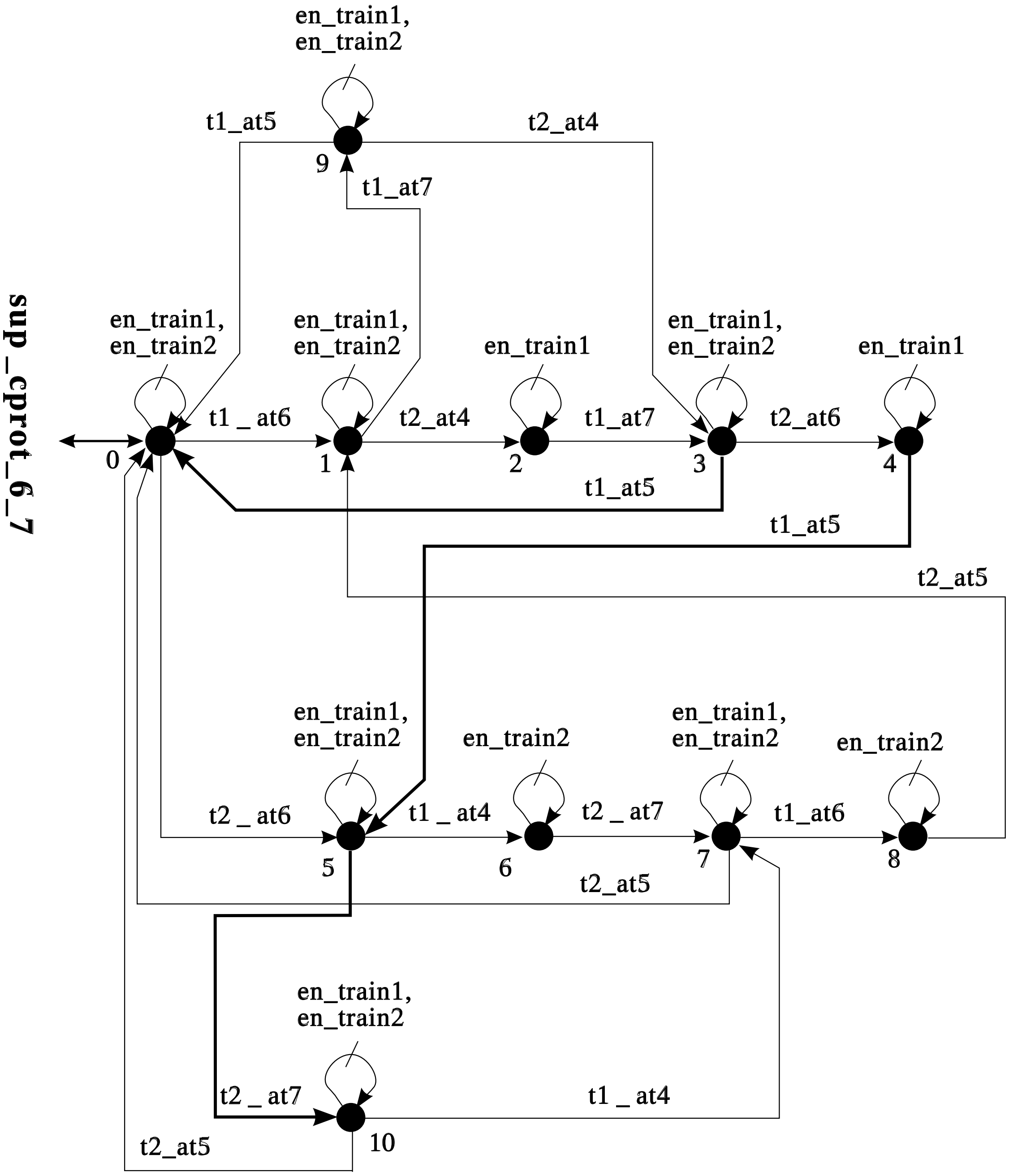


Figure 7.29: Collision Protection Supervisor for Section RS6 and RS7

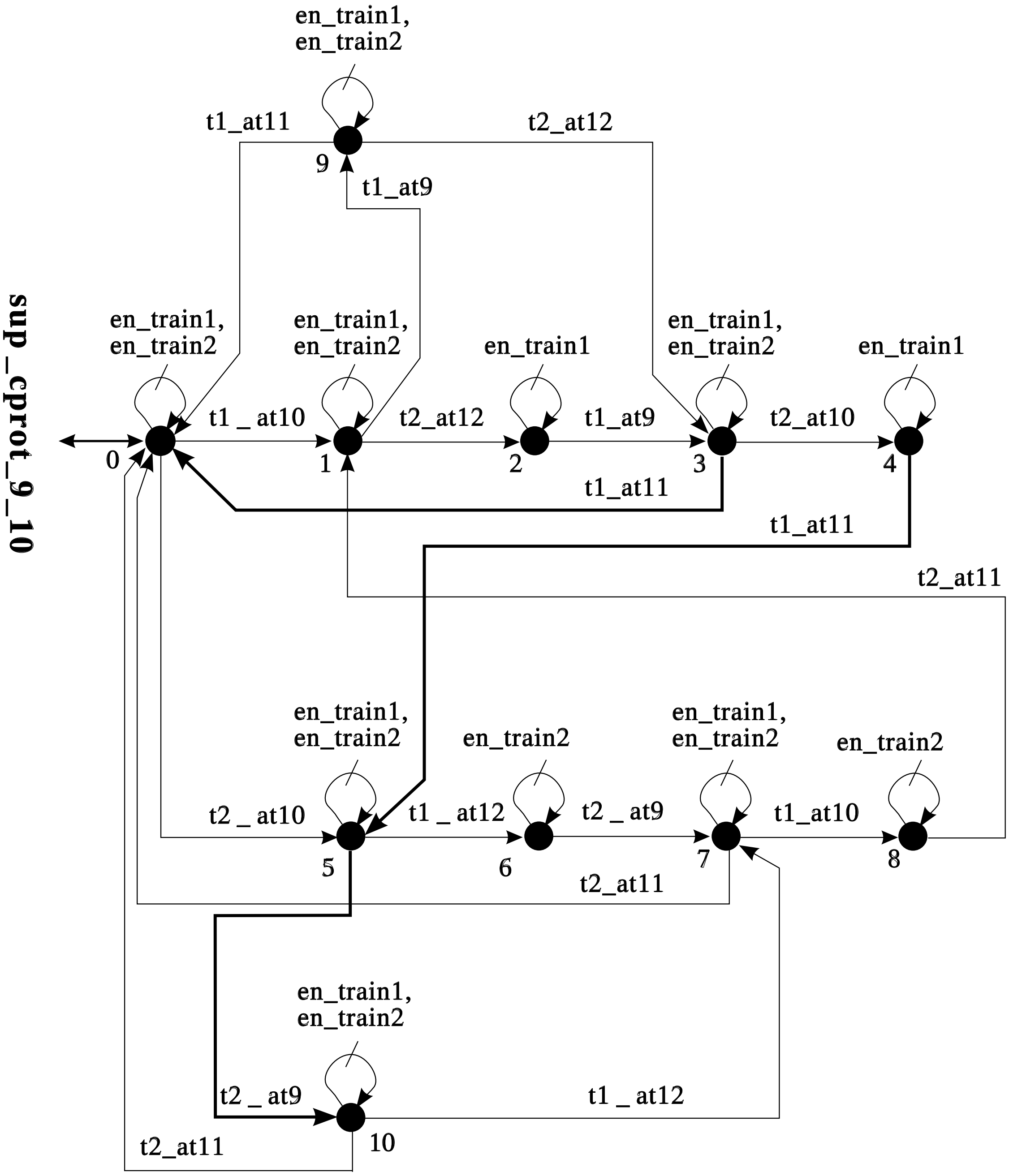


Figure 7.30: Collision Protection Supervisor for Section RS9 and RS10

sup_cprot_19_20		**sup_cprot_1_0**		**sup_cprot_6_7**	
Event	*States*	*Event*	*States*	*Event*	*States*
t1_at19	1, 2, 3, 4, 5, 6, 8, 9, 10	t1_at1	0, 2, 3, 4, 5, 6, 8, 9, 10	t1_at6	1, 2, 3, 4, 5, 6, 8, 9, 10
t2_at19	1, 2, 4, 5, 6, 7, 8, 9, 10	t2_at1	0, 2, 4, 5, 6, 7, 8, 9, 10	t2_at6	1, 2, 4, 5, 6, 7, 8, 9, 10
t1_at21	0, 5, 6, 7, 8, 10	t1_at2	1, 5, 6, 7, 8, 10	t1_at5	0, 5, 6, 7, 8, 10
t2_at21	0, 1, 2, 3, 4, 9	t2_at2	0, 1, 2, 3, 4, 9	t2_at5	0, 1, 2, 3, 4, 9
t1_at3	0, 1, 2, 3, 4, 6, 7, 8, 9	t1_at3	0, 1, 2, 3, 4, 6, 7, 8, 9	t1_at4	0, 1, 2, 3, 4, 6, 7, 8, 9
t2_at3	0, 2, 3, 4, 5, 6, 7, 8, 10	t2_at3	1, 2, 3, 4, 5, 6, 7, 8, 10	t2_at4	0, 2, 3, 4, 5, 6, 7, 8, 10
t1_at20	0, 3, 4, 5, 6, 7, 8, 9, 10	t1_at0	1, 3, 4, 5, 6, 7, 8, 9, 10	t1_at7	0, 3, 4, 5, 6, 7, 8, 9, 10
t2_at20	0, 1, 2, 3, 4, 7, 8, 9, 10	t2_at0	0, 1, 2, 3, 4, 7, 8, 9, 10	t2_at7	0, 1, 2, 3, 4, 7, 8, 9, 10
sup_cprot_9_10					
Event	*States*				
t1_at10	1, 2, 3, 4, 5, 6, 8, 9, 10				
t2_at10	1, 2, 4, 5, 6, 7, 8, 9, 10				
t1_at11	0, 5, 6, 7, 8, 10				
t2_at11	0, 1, 2, 3, 4, 9				
t1_at12	0, 1, 2, 3, 4, 6, 7, 8, 9				
t2_at12	0, 2, 3, 4, 5, 6, 7, 8, 10				
t1_at9	0, 3, 4, 5, 6, 7, 8, 9, 10				
t2_at9	0, 1, 2, 3, 4, 7, 8, 9, 10				

Table 7.6: Self-looped Events for Collision Protection Supervisors, Class III

Chapter 8

Model Reduction Theorems

After modeling the testbed, it was quickly apparent that conventional methods would be ineffective due to the large size of the composite model (on the order of 10^{16} states). To handle this, several theorems were developed for verifying controllability and nonblocking on reduced models of the plant.

8.1 Definitions

Let the plant be composed of $n \geq 1$ plant models.

Let $I = \{1, 2, \ldots, n\}$ be an index set for the n plant models. Define the n plant models as follows:

$\mathbf{G}_i = (Y_i, \Sigma_i, \eta_i, y_{o_i}, Y_{m_i}), i \in I$

$\mathbf{Plant} = \mathbf{sync}(\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_n) = (Y, \Sigma, \eta, y_o, Y_m)$

Let $J = \{1, 2, \ldots, m\}$ be an index set for the $m \geq 1$ DES supervisors for the plant. Define the m supervisors as follows:

$\mathbf{S}_j = (X_j, \Sigma, \xi_j, x_{o_j}, X_{m_j}), j \in J$

Definition: A sub-plant, $\mathbf{G}_{sub}$, is any group formed from the n models that define Plant.

Let $I_{sub} \subseteq I$ be the index set that represents the sub-plant.

Formally, $\mathbf{G}_{sub} = \mathbf{sync}(\mathbf{G}_{i_1}, \mathbf{G}_{i_2}, \ldots, \mathbf{G}_{i_k})$, $I_{sub} = \{i_1, i_2, \ldots, i_k\}$, $k \leq n$.

Definition: Arbitrary events for supervisor $\mathbf{S}_j$, $j \in J$, are events self-looped at every state in the supervisor. Let $\Sigma_{arb} \subseteq \Sigma$ represent these events.

Formally,

$$\Sigma_{arb_j} = \{\sigma \in \Sigma \mid (\forall x \in X_j)\, \xi_j(x,\, \sigma)! \wedge \xi_j(x,\, \sigma) = x\}$$

Definition: Essential events for supervisor $\mathbf{S}_j$, $j \in J$, are those events in Σ not in Σ_{arb_j}. Let $\Sigma_{ess_j} = \Sigma - \Sigma_{arb_j}$ represent these events.

Definition: Plants $\mathbf{G}_{i_1}$ and $\mathbf{G}_{i_2}$, i_1 and $i_2 \in I$, are decoupled if they have no common events ($\Sigma_{i_1} \cap \Sigma_{i_2} = \emptyset$).

Definition: A fundamental plant for supervisor $\mathbf{S}_j$, $j \in J$, is a minimal (in terms of index set size) sub-plant that contains every event in Σ_{ess_j} ($\Sigma_{ess_j} \subseteq \Sigma_{fund_j}$) and is decoupled from the remaining sub-plant. Let $\mathbf{G}_{fund_j}$ represent such a sub-plant and let $I_{fund_j} \subseteq I$ be its index set. Let $\mathbf{G}_{rem_j}$ represent the corresponding remaining sub-plant and $I_{rem_j} = I - I_{fund_j}$ be its index set.

Definition: A candidate plant for supervisor $\mathbf{S}_j$, $j \in J$, is any sub-plant that contains every event in Σ_{ess_j} ($\Sigma_{ess_j} \subseteq \Sigma_{cand_j}$). Let $\mathbf{G}_{cand_j}$ denote such a sub-plant and let $I_{cand_j} \subseteq I$ be its index set.

8.2 Controllability Theorems

The following two theorems identify sub-plants that can be used for verifying controllability of the entire plant. Theorem 1 presents necessary and sufficient conditions for the controllability of the plant. The theorem utilizes the principle that, if the portion of the plant that the given controller affects has no interaction with the rest of the plant, then the unaffected portion of the plant will have no effect on controllability.

Theorem 1 $\mathbf{S}_1$ *is controllable for* **Plant** *iff it is controllable for* $\mathbf{G}_{fund1}$.

Proof:

Define the natural projection $\mathbf{P}_{fund1} : \Sigma^* \rightarrow \Sigma^*_{fund1}$

I) "If" portion:

Let $\mathbf{S}_1$ be controllable for $\mathbf{G}_{fund1}$. (†)

Must show $\mathbf{S}_1$ is controllable for **Plant**. Sufficient to show:

$$\overline{L}(\mathbf{S}_1)\Sigma_u \cap L(\mathbf{Plant}) \subseteq \overline{L}(\mathbf{S}_1)$$

Let $s \in \overline{L}(\mathbf{S}_1)$, $\sigma \in \Sigma_u$, $s\sigma \in L(\mathbf{Plant})$. Must show $s\sigma \in \overline{L}(\mathbf{S}_1)$.

$$s\sigma \in L(\mathbf{Plant}) \Rightarrow s \in L(\mathbf{Plant}) \Rightarrow \mathbf{P}_{fund1}(s) \in L(\mathbf{G}_{fund1})$$

Since $\Sigma_{ess1} \subseteq \Sigma_{fund1}$, $\mathbf{P}_{fund1}$ only removes events in Σ_{arb1}. Thus,

$$s \in \overline{L}(\mathbf{S}_1) \Rightarrow \mathbf{P}_{fund1}(s) \in \overline{L}(\mathbf{S}_1) \text{ by definition of arbitrary events.}$$

Two cases to consider: **i)** $\sigma \in \Sigma_{arb1_u}$ and **ii)** $\sigma \in \Sigma_{ess1_u}$.

Case i) For $\sigma \in \Sigma_{arb1_u}$.
Automatic by definition of arbitrary events.

Case ii) For $\sigma \in \Sigma_{ess1_u}$.

$$\sigma \in \Sigma_{ess1_u} \Rightarrow \sigma \in \Sigma_{fund1} \text{ by definition of } \mathbf{G}_{fund}$$

Thus,

$$\mathbf{P}_{fund1}(s\sigma) = \mathbf{P}_{fund1}(s)\sigma \in L(\mathbf{G}_{fund1})$$

Also,

$$\mathbf{P}_{fund1}(s)\sigma \in \overline{L}(\mathbf{S}_1) \Rightarrow s\sigma \in \overline{L}(\mathbf{S}_1) \text{ by } ^{\dagger}$$

Thus by cases **i)** and **ii)**, $s\sigma \in \overline{L}(\mathbf{S}_1)$ and thus S_1 is controllable for **Plant**.

II) "Only If" portion:
Let $\mathbf{S}_1$ be controllable for **Plant**. $(^{\ddagger})$
Must show $\mathbf{S}_1$ is controllable for $\mathbf{G}_{fund1}$. Sufficient to show:

$$\overline{L}(\mathbf{S}_1)\Sigma_u \cap L(\mathbf{G}_{fund1}) \subseteq \overline{L}(\mathbf{S}_1)$$

Let $s \in \overline{L}(\mathbf{S}_1)$, $\sigma \in \Sigma_u$, $s\sigma \in L(\mathbf{G}_{fund1})$. Must show $s\sigma \in \overline{L}(\mathbf{S}_1)$.

Two cases to consider: **i)** $\sigma \in \Sigma_{arb1_u}$ and **ii)** $\sigma \in \Sigma_{ess1_u}$.

Case i) For $\sigma \in \Sigma_{arb1_u}$.
Automatic since:

$s \in \overline{L}(\mathbf{S}_1) \Rightarrow \xi_1(\xi_1(x_o, s), \sigma)!$ and $\xi_1(\xi_1(x_o, s), \sigma) = \xi_1(x_o, s)$ by definition of arbitrary events.

Case ii) For $\sigma \in \Sigma_{ess1_u}$.
Since **Plant** = **Sync**($\mathbf{G}_{fund1}, \mathbf{G}_{rem1}$) and by definition, $\mathbf{G}_{fund1}$ and $\mathbf{G}_{rem1}$ are decoupled,

$$s\sigma \in L(\mathbf{G}_{fund1}) \Rightarrow s\sigma \in L(\mathbf{Plant})$$

which by ‡, implies $s\sigma \in \overline{L}(\mathbf{S}_1)$

Thus, by cases **i)** and **ii)**, $s\sigma \in \overline{L}(\mathbf{S}_1)$ and thus $\mathbf{S}_1$ is controllable for $\mathbf{G}_{fund1}$.

□

Unfortunately, $\mathbf{G}_{fund}$ for the testbed is approximately the same size as the entire plant model, for every supervisor. This is due to the high degree of sub-model interaction.

To contend with this problem, Theorem 2 was developed. Theorem 2 requires a candidate plant that contains at least one instance of every event in Σ_{ess} for the given supervisor. The theorem utilizes the fact that the **sync** of two sub-models means that any event common to both sub-models can only occur when both sub-models permit it. Thus, if the supervisor handles every "situation" in the candidate plant, then it will handle every situation in the real plant since the **sync** operation will not permit any new "situations." This is because new, possibly less restrictive sub-models will be restricted (for $\sigma \in \Sigma_{ess}$) by the old sub-models.

The limitation of Theorem 2 is that several possible sub-plants qualify as $\mathbf{G}_{cand}$, many of which may fail to contain enough information to determine controllability. Designers should easily be able to use their knowledge of the plant's interactions to select a candidate plant of minimal size, yet still retain the necessary information to determine controllability.

Theorem 2 *If* $\mathbf{S}_1$ *is controllable for* $\mathbf{G}_{cand1}$ *then* $\mathbf{S}_1$ *is controllable for* **Plant***.*

Proof:

Proof is identical to **"if"** portion of proof for theorem 2, after substituting $\mathbf{G}_{cand1}$ for $\mathbf{G}_{fund1}$.

□

Example 1: Let the supervisor be **sup_cprot_19_20** (see Figure 7.27).

Let $\mathbf{G}_{cand1} = \mathbf{sync}(\mathbf{train1}, \mathbf{train2}, \mathbf{train1\_sensor}, \mathbf{train2\_sensor})$ [9 states, 408 transitions]

Supervisor **sup_cprot_19_20** is NOT controllable for $\mathbf{G}_{cand1}$.

Example 2: Let the supervisor be **sup_cprot_19_20** (see Figure 7.27).

Let $\mathbf{G}_{cand2} = \mathbf{sync}(\mathbf{G}_{cand1}$, **sensor_sensor_intd_t1**, **sensor_sensor_intd_t2**) [4761 states, 136 896 transitions]

Supervisor **sup_cprot_19_20** is controllable for $\mathbf{G}_{cand2}$ and thus it is controllable for **Plant**.

8.3 Non-Blocking Theorems

The next two theorems enable the verification of non-blocking on reduced plant models. The first theorem, Theorem 3, identifies sub-plants that can be used for testing non-blocking for individual controllers. The theorem utilizes the fact that, if the section of the plant that the supervisor affects has no interaction with the rest of the plant (which is trim), then it is only important that the supervisor be non-blocking for this section.

Theorem 3 *If* $\mathbf{S}_1$*,* $\mathbf{G}_{fund1}$*, and* $\mathbf{G}_{rem1}$ *are trim then* $\mathbf{S}_1$ *is non-blocking for* **Plant** *iff* $\mathbf{S}_1$ *is non-blocking for* $\mathbf{G}_{fund1}$*.*

Proof:

Define the natural projections $\mathbf{P}_{F1} : \Sigma^* \rightarrow \Sigma^*_{fund1}$ and $\mathbf{P}_{R1} : \Sigma^* \rightarrow \Sigma^*_{rem1}$.

Let $\mathbf{S}_1$, $\mathbf{G}_{fund_1}$, and $\mathbf{G}_{rem_1}$ be trim. (†)

I) "If" portion:

Let $\mathbf{S}_1$ be non-blocking for $\mathbf{G}_{fund1}$ (‡)

Must show $\mathbf{S}_1$ is non-blocking for **Plant**. Sufficient to show:

$$\overline{L_m}(\mathbf{S}_1/\mathbf{Plant}) = L(\mathbf{S}_1/\mathbf{Plant})$$

Requires showing: **i)** $\overline{L_m}(\mathbf{S}_1/\mathbf{Plant}) \subseteq L(\mathbf{S}_1/\mathbf{Plant})$

ii) $L(\mathbf{S}_1/\mathbf{Plant}) \subseteq \overline{L_m}(\mathbf{S}_1/\mathbf{Plant})$

i) Show $\overline{L_m}(\mathbf{S}_1/\mathbf{Plant}) \subseteq L(\mathbf{S}_1/\mathbf{Plant})$.

Automatic.

ii) Show $L(\mathbf{S}_1/\mathbf{Plant}) \subseteq \overline{L_m}(\mathbf{S}_1/\mathbf{Plant})$.

Let $s \in L(\mathbf{S}_1/\mathbf{Plant})$. Must show $s \in \overline{L_m}(\mathbf{S}_1/\mathbf{Plant})$.

$$\begin{aligned} s \in L(\mathbf{S}_1/\mathbf{Plant}) \quad &\Rightarrow \quad s \in L(\mathbf{S}_1) \cap \mathbf{P}^{-1}_{F1}L(\mathbf{G}_{fund1}) \cap \mathbf{P}^{-1}_{R1}L(\mathbf{G}_{rem1}) \\ &\Rightarrow \quad \mathbf{P}_{F1}(s) \in L(\mathbf{G}_{fund1}) \text{ and } \mathbf{P}_{R1}(s) \in L(\mathbf{G}_{rem1}) \\ &\Rightarrow \quad \mathbf{P}_{F1}(s) \in \overline{L_m}(\mathbf{G}_{fund1}) \text{ and } \mathbf{P}_{R1}(s) \in \overline{L_m}(\mathbf{G}_{rem1}) \text{ by }^\dagger. \\ &\Rightarrow \quad (\exists s_1 \in \Sigma^*_{fund1}) \text{ such that } \mathbf{P}_{F1}(s)s_1 \in L_m(\mathbf{G}_{fund1}) \text{ and} \qquad (^\diamond) \\ &\qquad\ \ (\exists s_2 \in \Sigma^*_{rem1}) \text{ such that } \mathbf{P}_{R1}(s)s_2 \in L_m(\mathbf{G}_{rem1}) \end{aligned}$$

By definition, $\Sigma_{fund1} \cap \Sigma_{rem1} = \emptyset$. This means:

$$\begin{aligned} \mathbf{P}_{F1}(ss_1s_2) \quad &= \quad \mathbf{P}_{F1}(s)s_1 \in L_m(\mathbf{G}_{fund1}) \\ \mathbf{P}_{R1}(ss_1s_2) \quad &= \quad \mathbf{P}_{R1}(s)s_2 \in L_m(\mathbf{G}_{rem1}) \end{aligned}$$

Thus,

$$ss_1s_2 \in \mathbf{P}^{-1}_{F1}L_m(\mathbf{G}_{fund1}) \cap \mathbf{P}^{-1}_{R1}L_m(\mathbf{G}_{rem1})$$

<u>*Examine relationship of* $\mathbf{S}_1$ and $\mathbf{G}_{fund1}$:</u>

$$\begin{aligned}\mathbf{P}_{F1}(s) \in L(\mathbf{S}_1) \text{ and } \mathbf{P}_{F1}(s) \in L(\mathbf{G}_{fund1}) &\Rightarrow \mathbf{P}_{F1}(s) \in L(\mathbf{S}_1) \cap L(\mathbf{G}_{fund1}) \\ &\Rightarrow \mathbf{P}_{F1}(s) \in \overline{L_m(\mathbf{S}_1) \cap L_m(\mathbf{G}_{fund1})} \text{ by } ^{\ddagger}. \\ &\Rightarrow (\exists s' \in \Sigma^*_{fund1})\ \mathbf{P}_{F1}(s)s' \in L_m(\mathbf{S}_1) \cap L_m(\mathbf{G}_{fund1})\end{aligned}$$

Clearly, s' qualifies for s_1 in $^\diamond$ so let $s_1 = s'$.

$$\mathbf{P}_{F1}(s)s_1 \in L_m(\mathbf{S}_1) \Rightarrow ss_1 \in L_m(\mathbf{S}_1) \text{ by definition of arbitrary events.}$$

Also, $ss_1s_2 \in L_m(\mathbf{S}_1)$ since $s_2 \in \Sigma^*_{rem1} \subseteq \Sigma^*_{arb1}$. Thus

$$\begin{aligned}ss_1s_2 &\in L_m(\mathbf{S}_1) \cap \mathbf{P}^{-1}_{F1}L_m(\mathbf{G}_{fund1}) \cap \mathbf{P}^{-1}_{R1}L_m(\mathbf{G}_{rem1}) \\ \Rightarrow s &\in \overline{L_m(\mathbf{S}_1) \cap L_m(\mathbf{Plant})} \text{ as required.}\end{aligned}$$

Thus by parts **i)** and **ii)**, $\mathbf{S}_1$ is non-blocking for **Plant**.

II) "Only If" portion:

Let $\mathbf{S}_1$ be non-blocking for **Plant**. $(^\triangle)$

Must show $\mathbf{S}_1$ is non-blocking for $\mathbf{G}_{fund1}$. Sufficient to show:

$$\overline{L_m}(\mathbf{S}_1/\mathbf{G}_{fund1}) = L(\mathbf{S}_1/\mathbf{G}_{fund1})$$

Requires showing: **i)** $\overline{L_m}(\mathbf{S}_1/\mathbf{G}_{fund1}) \subseteq L(\mathbf{S}_1/\mathbf{G}_{fund1})$

ii) $L(\mathbf{S}_1/\mathbf{G}_{fund1}) \subseteq \overline{L_m}(\mathbf{S}_1/\mathbf{G}_{fund1})$

i) Show $\overline{L_m}(\mathbf{S}_1/\mathbf{G}_{fund1}) \subseteq L(\mathbf{S}_1/\mathbf{G}_{fund1}$.

Automatic.

ii) Show $L(\mathbf{S}_1/\mathbf{G}_{fund1}) \subseteq \overline{L_m}(\mathbf{S}_1/\mathbf{G}_{fund1})$.

Let $s \in L(\mathbf{S}_1/\mathbf{G}_{fund1})$. Must show $s \in \overline{L_m}(\mathbf{S}_1/\mathbf{G}_{fund1})$.

$$\begin{aligned}s \in L(\mathbf{S}_1/\mathbf{G}_{fund1}) &\Rightarrow s \in L(\mathbf{G}_{fund1}) \\ &\Rightarrow s \in L(\mathbf{Plant}) \text{ since } \mathbf{G}_{fund1} \text{ and } \mathbf{G}_{rem1} \text{ are decoupled.} \\ &\Rightarrow s \in L(\mathbf{S}_1) \cap L(\mathbf{Plant}) \Rightarrow s \in \overline{L_m(\mathbf{S}_1) \cap L_m(\mathbf{Plant})} \text{ by } ^\triangle.\end{aligned}$$

This implies $(\exists s'' \in \Sigma^*)$ such that $ss'' \in L_m(\mathbf{S}_1) \cap L_m(\mathbf{Plant})$

$$\begin{aligned} ss'' \in L_m(Plant) &\Rightarrow ss'' \in \mathbf{P}_{F1}^{-1}L_m(\mathbf{G}_{fund1}) \cap \mathbf{P}_{R1}^{-1}L_m(\mathbf{G}_{rem1}) \\ &\Rightarrow ss'' \in L_m(\mathbf{S}_1) \cap \mathbf{P}_{F1}^{-1}L_m(\mathbf{G}_{fund1}) \\ ss'' \in \mathbf{P}_{F1}^{-1}L_m(\mathbf{G}_{fund1}) &\Rightarrow s\mathbf{P}_{F1}(s'') \in L_m(\mathbf{G}_{fund1}) \end{aligned}$$

Also,

$$ss'' \in L_m(\mathbf{S}_1) \Rightarrow \mathbf{P}_{F1}(ss'') = s\mathbf{P}_{F1}(s'') \in L_m(\mathbf{S}_1)$$

since $\mathbf{P}_{F1}$ only removes $\sigma \in \Sigma_{arb1}$.
Thus, $s\mathbf{P}_{F1}(s'') \in L_m(\mathbf{S}_1) \cap L_m(\mathbf{G}_{fund1})$ implies

$$s \in \overline{L_m(\mathbf{S}_1) \cap L_m(\mathbf{G}_{fund1})} \text{ as required.}$$

Thus by parts **i)** and **ii)**, $\mathbf{S}_1$ is non-blocking for $\mathbf{G}_{fund1}$.

□

Once Theorem 3 has been used to verify that each supervisor is non-blocking for **Plant**, then Theorem 4 can be applied iteratively to determine that $((\mathbf{S}_1 \wedge \mathbf{S}_2 \wedge \ldots \wedge \mathbf{S}_m) \,/\, \mathbf{Plant})$ is non-blocking. Theorem 4 is intended to be used on plants with the structure shown in Figure 8.1. As can be seen from the diagram, the fundamental plant for supervisor $\mathbf{S}_1$ has no interaction with the rest of the plant. Also, the fundamental plant for supervisor $\mathbf{S}_2$ is contained in $\mathbf{G}_{rem1}$.

Theorem 4 works on the principle that, since the fundamental sub-plants of the two supervisors do not interact, then they will not affect whether the other supervisor is non-blocking for **Plant**. Both Theorems 3 and 4 only operate well on plants with limited sub-model interaction.

Theorem 4 *If* $(\mathbf{S}_1 \,/\, \mathbf{G}_{fund1})$ *and* $(\mathbf{S}_2 \,/\, \mathbf{G}_{rem1})$ *are non-blocking and* $\mathbf{G}_{fund1}$ *is decoupled from* $\mathbf{G}_{fund2}$ *then* $(\mathbf{S}_1 \wedge \mathbf{S}_2 \,/\, \mathbf{Plant})$ *is non-blocking.*

Proof:

Define the natural projections $\mathbf{P}_{F1} : \Sigma^* \rightarrow \Sigma^*_{fund1}$ and $\mathbf{P}_{R1} : \Sigma^* \rightarrow \Sigma^*_{rem1}$.

Let $(\mathbf{S}_1/\mathbf{G}_{fund1})$ be non-blocking $(^\dagger)$

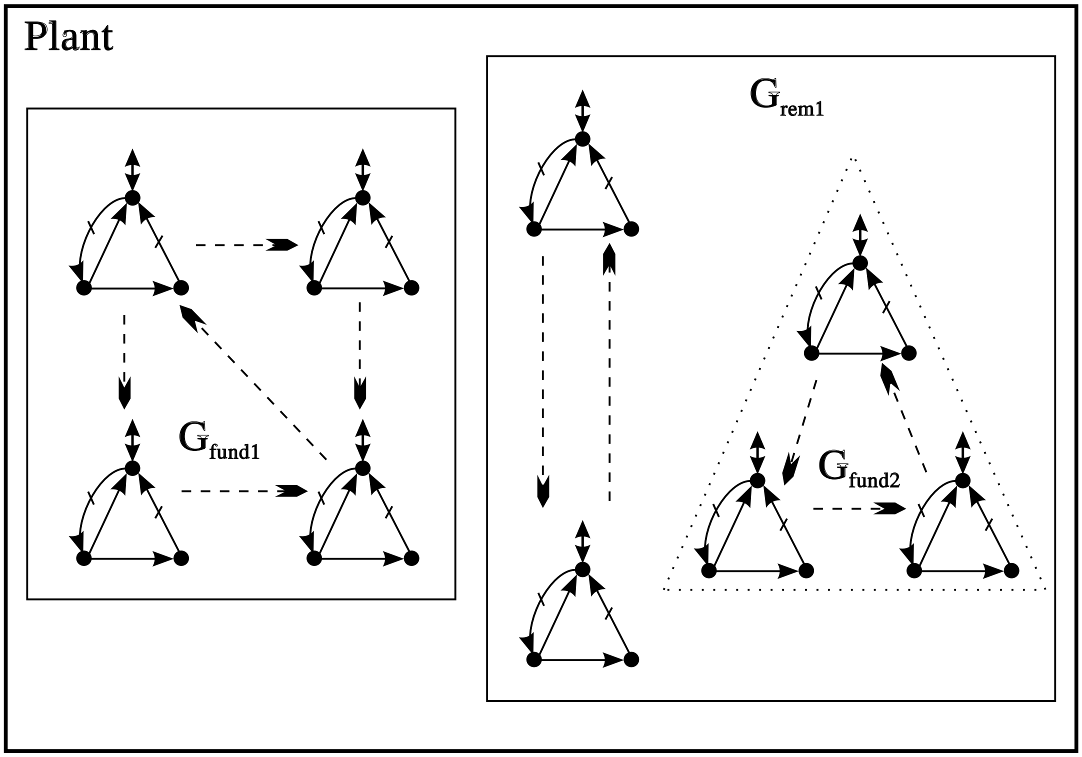

Plant
$\mathbf{G}_{\text{rem1}}$
$\mathbf{G}_{\text{fund1}}$
$\mathbf{G}_{\text{fund2}}$

Figure 8.1: Example Plant For Theorem 4

$(\mathbf{S}_1/\mathbf{G}_{rem1})$ be non-blocking $\quad (^{\ddagger})$

$\mathbf{G}_{fund1}$ and $\mathbf{G}_{fund2}$ be decoupled $\quad (^{\diamond})$

Must show: $\overline{L_m}(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant}) = L(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant})$

This requires showing: **I)** $\overline{L_m}(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant}) \subseteq L(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant})$

II) $L(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant}) \subseteq \overline{L_m}(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant})$

I) Show $\overline{L_m}(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant}) \subseteq L(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant})$.

Automatic.

II) Show $L(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant}) \subseteq \overline{L_m}(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant})$.

Let $s \in L(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant})$. Must show $s \in \overline{L_m}(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant})$

$$
\begin{aligned}
s \in L(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant}) \quad &\Rightarrow \quad s \in L(\mathbf{S}_1) \cap L(\mathbf{S}_2) \cap L(\mathbf{Plant}) \\
&\Rightarrow \quad s \in L(\mathbf{S}_1) \cap L(\mathbf{S}_2) \cap \mathbf{P}_{F1}^{-1} L(\mathbf{G}_{fund1}) \cap \mathbf{P}_{R1}^{-1} L(\mathbf{G}_{rem1})
\end{aligned}
$$

Look at Problem in three steps:

Step i) Examine $s \in L(\mathbf{S}_1) \cap \mathbf{P}_{F1}^{-1} L(\mathbf{G}_{fund1})$.

$$s \in L(\mathbf{S}_1) \cap \mathbf{P}_{F1}^{-1} L(\mathbf{G}_{fund1}) \Rightarrow \mathbf{P}_{F1}(s) \in L(\mathbf{G}_{fund1})$$

By definition of $\mathbf{G}_{fund1}$, $\Sigma_{ess1} \subseteq \Sigma_{fund1}$, thus $\mathbf{P}_{F1}$ only removes $\sigma \in \Sigma_{arb1}$. This means

$$\begin{aligned}
s \in L(\mathbf{S}_1) &\Rightarrow \mathbf{P}_{F1}(s) \in L(\mathbf{S}_1) \text{ by definition of arbitrary events.}\\
&\Rightarrow \mathbf{P}_{F1}(s) \in L(\mathbf{S}_1) \cap L(\mathbf{G}_{fund1})\\
&\Rightarrow \mathbf{P}_{F1}(s) \in \overline{L_m(\mathbf{S}_1) \cap L_m(\mathbf{G}_{fund1})} \text{ by } ^{\dagger}\\
&\Rightarrow (\exists s_1 \in \Sigma_{fund1}^*) \text{ such that } \mathbf{P}_{F1}(s)s_1 \in L_m(\mathbf{S}_1) \cap L_m(\mathbf{G}_{fund1})
\end{aligned}$$

Thus,

$$ss_1 \in L_m(\mathbf{S}_1) \cap \mathbf{P}_{F1}^{-1} L_m(\mathbf{G}_{fund1})$$

Step ii) Examine $s \in L(\mathbf{S}_2) \cap \mathbf{P}_{R1}^{-1} L(\mathbf{G}_{rem1})$.

$$s \in L(\mathbf{S}_2) \cap \mathbf{P}_{R1}^{-1} L(\mathbf{G}_{rem1}) \Rightarrow \mathbf{P}_{R1}(s) \in L(\mathbf{G}_{rem1})$$

By definition of $\mathbf{G}_{fund}$, $\Sigma_{ess2} \subseteq \Sigma_{fund2}$. Also, $\Sigma_{fund1} \cap \Sigma_{fund2} = \emptyset$ by $^{\diamond}$ and $\Sigma_{fund1} \cap \Sigma_{rem1} = \emptyset$ by definition of $\mathbf{G}_{fund}$. This means:

$$\Sigma_{fund2} \subseteq \Sigma_{rem1}$$

Since $\Sigma_{ess2} \subseteq \Sigma_{fund2}$, $\mathbf{P}_{R1}$ only removes $\sigma \in \Sigma_{arb2}$. Thus,

$$\begin{aligned}
s \in L(\mathbf{S}_2) &\Rightarrow \mathbf{P}_{R1}(s) \in L(\mathbf{S}_2) \text{ by definition of arbitrary events}\\
&\Rightarrow \mathbf{P}_{R1}(s) \in L(\mathbf{S}_2) \cap L(\mathbf{G}_{rem1})\\
&\Rightarrow \mathbf{P}_{R1}(s) \in \overline{L_m(\mathbf{S}_2) \cap L_m(\mathbf{G}_{rem1})} \text{ by } ^{\ddagger}\\
&\Rightarrow (\exists s_2 \in \Sigma_{rem1}^*) \text{ such that } \mathbf{P}_{R1}(s)s_2 \in L_m(\mathbf{S}_2) \cap L_m(\mathbf{G}_{rem1})
\end{aligned}$$

Thus,

$$ss_2 \in L_m(\mathbf{S}_2) \cap \mathbf{P}_{R1}^{-1} L_m(\mathbf{G}_{rem1})$$

Step iii) Put steps **i)** and **ii)** together.

From **i)** $ss_1 \in L_m(\mathbf{S}_1) \cap \mathbf{P}_{F1}^{-1} L_m(\mathbf{G}_{fund1}),\ s_1 \in \Sigma_{fund1}^* \Rightarrow s_1 \in \Sigma_{arb2}^*$ by $^\diamond$.

From **ii)** $ss_2 \in L_m(\mathbf{S}_2) \cap \mathbf{P}_{R1}^{-1} L_m(\mathbf{G}_{rem1}),\quad s_2 \in \Sigma_{rem1}^* \Rightarrow s_2 \in \Sigma_{arb1}^*$ by definition of $\mathbf{G}_{fund}$.

Thus,

$$ss_1s_2 \in L_m(\mathbf{S}_1) \text{ since } s_2 \in \Sigma_{arb1}^*$$
$$ss_1s_2 \in L_m(\mathbf{S}_2) \text{ since } \ s_1 \in \Sigma_{arb2}^*$$

Finally,

$$ss_1s_2 \in \mathbf{P}_{F1}^{-1} L_m(\mathbf{G}_{fund1}) \text{ since } \mathbf{P}_{F1}(ss_1s_2) = \mathbf{P}_{F1}(s)s_1$$
$$ss_1s_2 \in \mathbf{P}_{R1}^{-1} L_m(\mathbf{G}_{rem1}) \text{ since } \mathbf{P}_{R1}(ss_1s_2) = \mathbf{P}_{R1}(s)s_2$$

Thus,

$$ss_1s_2 \in L_m(\mathbf{S}_1) \cap L_m(\mathbf{S}_2) \cap \mathbf{P}_{F1}^{-1} L_m(\mathbf{G}_{fund1}) \cap \mathbf{P}_{R1}^{-1} L_m(\mathbf{G}_{rem1})$$

which implies

$$s \in \overline{L_m}(\,(\mathbf{S}_1 \wedge \mathbf{S}_2)/\mathbf{Plant}),\ \text{as required.}$$

Thus by parts **I)** and **II)**, $(\mathbf{S}_1 \wedge \mathbf{S}_2 \,/\, \mathbf{Plant})$ is non-blocking.

□

As discussed above, $\mathbf{G}_{fund}$ for the testbed is approximately the same size as the entire plant model, for every supervisor. Thus by themselves, the non-blocking theorems are not enough to verify non-blocking for the testbed. However, they can greatly reduce the necessary work.

For the testbed, the plant sub-models **train1_sensor**, **train2_sensor**, **sensor_sensor_intd_t1**, and **sensor_sensor_intd_t2** cause the non-blocking theorems to fail. For ease of reference, let's label these sub-models $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{G}_3$, and $\mathbf{G}_4$ respectively.

Since non-blocking is not concerned about controllability, all that is important is the languages generated by the plant. Thus, we could replace $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{G}_3$, and $\mathbf{G}_4$ with supervisors that would generate the same languages when applied to the plant. Let's call these new supervisors $\mathbf{S}_{G1}$, $\mathbf{S}_{G2}$, $\mathbf{S}_{G3}$, and $\mathbf{S}_{G4}$.

Note: Supervisor $\mathbf{S}_{Gi},\ i = 1, 2, 3, 4$ would be $\mathbf{G}_i$ with events $\sigma \in (\Sigma - \Sigma_i)$ self-looped at every state.

Now, the remaining plant (still very large) has a low degree of sub-model interaction. Thus Theorems 3 and 4 can now be applied to the original 29 supervisors to verify that they are non-blocking individually and as a group, for the new plant. All that would remain would be to verify that $\mathbf{S}_{G1}$, $\mathbf{S}_{G2}$, $\mathbf{S}_{G3}$, and $\mathbf{S}_{G4}$ are non-blocking individually and as a group (including original 29 supervisors) for the new plant.

8.4 Application to Testbed

Theorem 2 was successfully used to verify controllability[1] for the testbed's 29 modular supervisors. Candidate plants as small as (16 states, 72 transitions) and as large as (4761 states, 136 896 transitions) were used. Since $L(\mathbf{S}_j)$, $j \in J_{\text{tb}}$, is a closed language[2], the intersection of these languages is controllable. Thus, the conjunction of the testbed's 29 modular supervisors is controllable.

As yet there has been no success in verifying non-blocking for the testbed due to the high interaction among the sub-plants. The reduced plant was only slightly smaller than the testbed itself.

[1] The software package TCT was used to verify controllability.

[2] The set J_{tb} is the index set for the testbed's 29 modular supervisors.

Chapter 9

Synchronous State Machines

The proposed method is to implement the DES supervisors as clocked Moore synchronous state machines (CMSSM: [7], [31]) on a PLC. The supervisors will be first translated into CMSSM and then the boolean logic defining the CMSSM will be expressed in Relay Ladder Logic and executed on the PLC.

The supervisors are expressed as CMSSM as an intermediate step because CMSSM are a natural intuitive representation of a DES supervisor, they specify the controller exactly, and they can easily be implemented on most digital logic devices such as a microcomputer (ie. as a 'C' program), a programmable array logic chip (PAL), or a Xilinx chip. Thus, the implementation method proposed in this chapter can easily be modified to implement supervisors on other digital logic devices, besides programmable logic controllers.

In this chapter, CMSSM will be described, and a formal model will be presented. Then, a translation method to convert supervisors into CMSSM will be introduced, followed by a discussion on the equivalence of the supervisor and CMSSM representation.

9.1 Introduction to CMSSM

A clocked synchronous state machine is a standard way to implement control functions in digital hardware. State machines only change states on the rising or falling edge of a clock pulse. For now, state machines will be assumed to only change states on the falling edge. A Moore machine is characterized by the fact that the outputs are functions only of the present state.

In a CMSSM, all variables represent a binary digit, and the next state vector is determined by a boolean logic equation of the input variables, and the current state variables. A simple example of a CMSSM is given in Figure 9.1. The state machine is defined by an initial or reset state, a set of next state equations (one for each state variable), and a set of state to output map equations (one for each output). These are given below for the example in Figure 9.1. Thus, on the falling edge of the CMSSM's driving clock pulse, the state machine changes to the next state (starting from the initial state) dictated by its next state equations. The state to output map is then evaluated to determine the new output values.

$$\begin{aligned} q0_{new} &= \neg q_1 \wedge \neg q_0 \wedge \neg i_0 \\ q1_{new} &= (q_0 \wedge \neg i_0) \vee (\neg q_1 \wedge \neg i_0) \\ z_0 &= \neg q_0 \end{aligned}$$

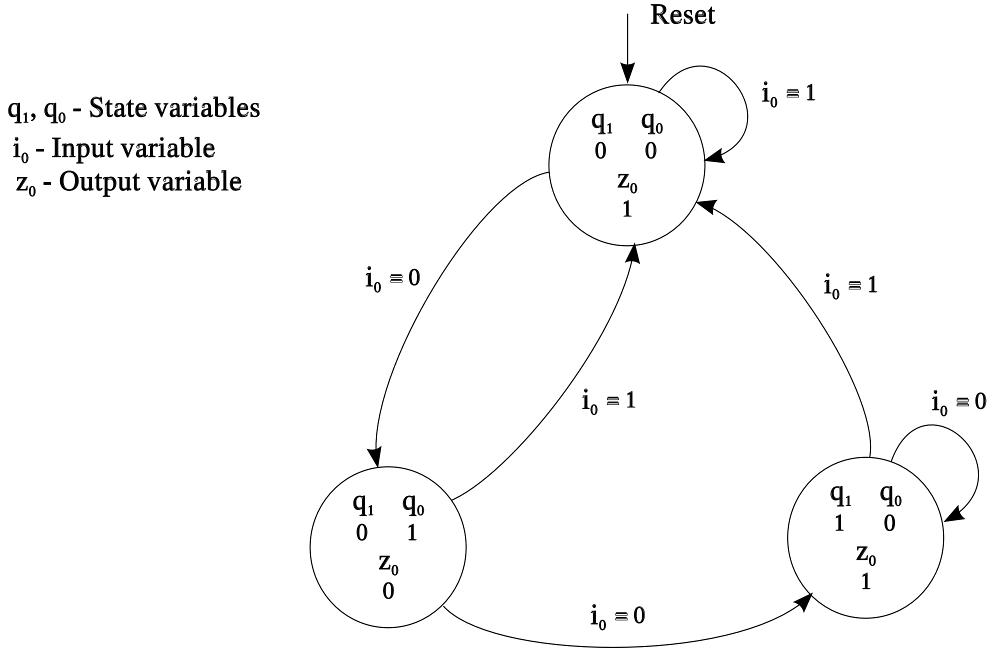


Figure 9.1: An example CMSSM

9.2 Formal Model

To prove that a CMSSM is an equivalent control representation, we first present a formal model.

Let the CMSSM be represented by the 7-tuple

$$H = (I, Z, Q, \Omega, \Phi, q_{Res}, T)$$

where I represents inputs to the state machine, Z represents outputs from the state machine, Q represents the state vector, Ω is the next state function, Φ is the state to output map, q_{Res} represents the initial or reset state, and T is the period of the clock pulse that drives the state machine.

In detail, let the states of the CMSSM be represented by:

$$Q = [q_0, q_1, \, \ldots \, , q_{l-1}], \;\; q_j \in \{0, 1\}, \;\; j = 0, 1, \, \ldots \, , l - 1;$$

the input by:

$$I = [i_0, i_1, \, \ldots \, , i_{v-1}], \;\; i_j \in \{0, 1\}, \;\; j = 0, 1, \, \ldots \, , v - 1;$$

and the output by:

$$Z = [z_0, z_1, \, \ldots \, , z_{r-1}], \;\; z_j \in \{0, 1\}, \;\; j = 0, 1, \, \ldots \, , r - 1$$

Since inputs, outputs, and states only change on the rising (falling) edge of the clock pulse, they each form a discrete series. Let k represent the index for the series, where $k = 0$ corresponds to when the state machine starts, k = 1 to the state machine $1 \cdot T$ seconds later, and so on. Thus, the state variables form a series:

$$Q(k), \;\; k = 0, 1, 2, 3, \, \ldots \quad \text{where} \;\; Q(k) = [q_0(k), q_1(k), \, \ldots \, , q_{l-1}(k)].$$

with $Q(0) = q_{Res}$. The input and output vectors form similar series.

The new input, $I(k + 1)$ is simply the sampled value of the input at the $(k + 1)^{th}$ clock pulse. The CMSSM doesn't model how the input changes, but simply makes decisions based on its last state and the new input.

The next value of Q is determined by the next state function, Ω, defined as follows:

$$\Omega :\; Q \,\mathrm{x}\, I \rightarrow Q; \quad Q(k+1) = \Omega(Q(k),\, I(k+1))$$

The current output is determined by the output map, Φ, defined as follows:

$$\Phi :\; Q \rightarrow Z; \;\; Z(k+1) = \Phi(Q(k+1))$$

9.3 Translation Method

In this section, a translation method is defined to convert a deterministic DES supervisor into a CMSSM. We present both a centralized and modular implementation. First, we state a few definitions.

Let the plant model be defined as:

$$\mathbf{G} = (Y, \Sigma, \eta, y_o, Y_m), \;\; \Sigma = \Sigma_u \cup \Sigma_c$$

Let $\mathbf{S}_1$, $\mathbf{S}_2$, ... , $\mathbf{S}_n$ be n supervisors for $\mathbf{G}$, with

$$\mathbf{S}_j = (X_j, \Sigma, \xi_j, x_{o_j}, X_{m_j}), \;\; j = 1, 2, \ldots n$$

Let $\mathbf{S} = \mathbf{meet}(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n) = (X, \Sigma, \xi, x_o, X_m)$

9.3.1 Centralized Controller

For the centralized implementation, the CMSSM will be defined with respect to the composite supervisor, $\mathbf{S}$.

Define the state size of Q to satisfy $2^l \geq \mid X \mid$. Define an arbitrary injective map, $\lambda :\; X \rightarrow Q$. The initial state is defined to be $q_{Res} = \lambda(x_0)$.

Define the state size of I to be $v = \mid \Sigma \mid$. Define an arbitrary bijective map, $\gamma :\; \Sigma \rightarrow \{i_0, i_1, \ldots, i_{v-1}\}$. Then define an injective map, $\Gamma :\; \Sigma \rightarrow I$ where:

$$\Gamma(\sigma) = [0 \;\cdots\; 0 \;\; \underset{j}{1} \;\; 0 \;\cdots\; 0] \;\text{ for }\; \sigma \in \Sigma \;\text{and}\; \gamma(\sigma) = i_j$$

Define the state size of Z to be $r = \mid \Sigma_c \mid$. Define an arbitrary bijective map, $\delta : \Sigma_c \rightarrow \{z_0, z_1, \ldots, z_{r-1}\}$.

The next state function, Ω, is defined in accordance with the supervisor. For $x \in X$ and $\sigma \in \Sigma$,

$$\text{if} \quad \xi(x,\sigma)! \quad \text{then} \quad \Omega(\lambda(x), \Gamma(\sigma)) = \lambda(\xi(x,\sigma)) \quad \text{else} \quad \Omega(\lambda(x), \Gamma(\sigma)) = \lambda(x)$$

All remaining values of Ω are assigned arbitrarily.

The output map, Φ, is similarly based on the supervisor. For $x \in X$ and $\sigma \in \Sigma_c$,

$$\text{if} \quad \xi(x,\sigma)! \quad \text{then} \quad \delta(\sigma) = 1 \text{ for state } \lambda(x). \text{ Thus, } \Phi(\lambda(x)) = [\cdots \underset{j}{1} \cdots] \text{ for } \delta(\sigma) = z_j$$

$$\text{otherwise} \quad \delta(\sigma) = 0 \text{ for state } \lambda(x) \text{ and } \Phi(\lambda(x)) = [\cdots \underset{j}{0} \cdots] \text{ for } \delta(\sigma) = z_j$$

9.3.2 Modular Controllers

For large systems, it is desirable to design modular controllers instead of a centralized controller as they are easier to design, verify and modify. It is also easier to implement several small modular controllers, as discussed in Section 7.2, rather than one massive one.

For clarity, we define the modular implementation for the case of two modular DES supervisors, $\mathbf{S}_1$ and $\mathbf{S}_2$. The definition can easily be extended to n supervisors.

First, the two supervisors are converted to centralized CMSSM as above. But, instead of actually controlling the system's outputs, they generate 'local' outputs. Thus we have the following controllers:

$$H_1 = (I, Z_1, Q_1, \Omega_1, \Phi_1, q_{Res_1}, T) \quad \text{and} \quad H_2 = (I, Z_2, Q_2, \Omega_2, \Phi_2, q_{Res_2}, T)$$

The Composite Controller, H_{mod}, is then implemented by taking the conjunction of the two modular controllers' outputs. Thus the system output is:

$$Z = Z_1 \wedge Z_2 = [z_{1_0} \wedge z_{2_0}, \ \ldots, z_{1_{r-1}} \wedge z_{2_{r-1}}]$$

We then define the following maps:

i) $\Omega = \Omega_1 \text{ x } \Omega_2 \; (Q_1 \text{ x } Q_2) \;=\; \Omega_1(Q_1) \text{ x } \Omega_2(Q_2)$

ii) $\Phi = \Phi_1 \wedge \Phi_2 \; (Q_1 \text{ x } Q_2) \;=\; \Phi_1(Q_1) \wedge \Phi_2(Q_2)$

Then, let $Q = Q_1 \,\text{x}\, Q_2$, and $q_{Res} = q_{Res_1} \,\text{x}\, q_{Res_2}$. Thus the composite controller is defined:

$$H_{mod} = (I, Z_1 \wedge Z_2, Q_1 \,\text{x}\, Q_2, \Omega, \Phi, q_{Res_1} \,\text{x}\, q_{Res_2}, T) = (I, Z, Q, \Omega, \Phi, q_{Res}, T)$$

9.3.3 Implementation Details

In this section we discuss the implementation of the synchronous state machine's mapping functions, and the execution algorithm used to implement the CMSSM.

For the state machine, both the next state function (Ω) and the output map (Φ) are implemented in a modular manner. The next state function is implemented as l boolean logic functions. Thus,

$$\Omega(Q(k), I(k+1)) = [\omega_0(Q(k), I(k+1)), \; \ldots \; , \omega_{l-1}(Q(k), I(k+1))]$$

Similarly, the output map is implemented as r boolean logic functions. Thus,

$$\Phi(Q(k+1)) = [\phi_0(Q(k+1)), \; \ldots \; , \phi_{r-1}(Q(k+1))]$$

The execution algorithm used to implement the clocked Moore synchronous state machines for a centralized implementation is as follows:

1. Initialize state machine by setting $Q(0) = q_{Res}$, $Z(0) = \Phi(q_{Res})$and $k = 0$.

2. Wait for next clock pulse $(k + 1)$. Sample inputs and set $I(k + 1)$ to these values.

3. Compute $Q(k+1) = \Omega(Q(k), I(k+1))$ and $Z(k+1) = \Phi(Q(k+1)$.

4. Set $k = k + 1$ and then go to step 2.

For the modular implementation, the algorithm would be essentially the same. Step 3 of the algorithm would be applied to every modular controller, and then the global output would be evaluated using the local outputs generated by each controller. Step 4 would then occur and the process would repeat.

9.3.4 Interpretation of Events

The interpretation of the input variables (i_0, i_1 etc.) is that the CMSSM senses the occurrence of an event σ when its corresponding input $\gamma(\sigma)$ is TRUE (=1). The interpretation of the output variables (z_0, z_1 etc.) is that a controllable event, $\sigma \in \Sigma_c$, is enabled when its corresponding output $\delta(\sigma)$ is TRUE. Otherwise, the event is disabled.

PLC Interpretation

With respect to the PLC implementation we are discussing, Figure 9.2 shows what the above interpretation means in terms of actual input and output signals. The state machines sample inputs and generate outputs with regard to the clock signal, *PLC_clock.* Just before the falling edge of *PLC_clock,* input signals (such as $\overline{\mathrm{In}_\beta}$) are sampled, and, just after the same edge, output signals (such as the enablement signal $\overline{\mathrm{En}_\alpha}$) are generated.

NOTE: The bar over the signals in Figure 9.2 indicates that the signal is TRUE when the signal is low.

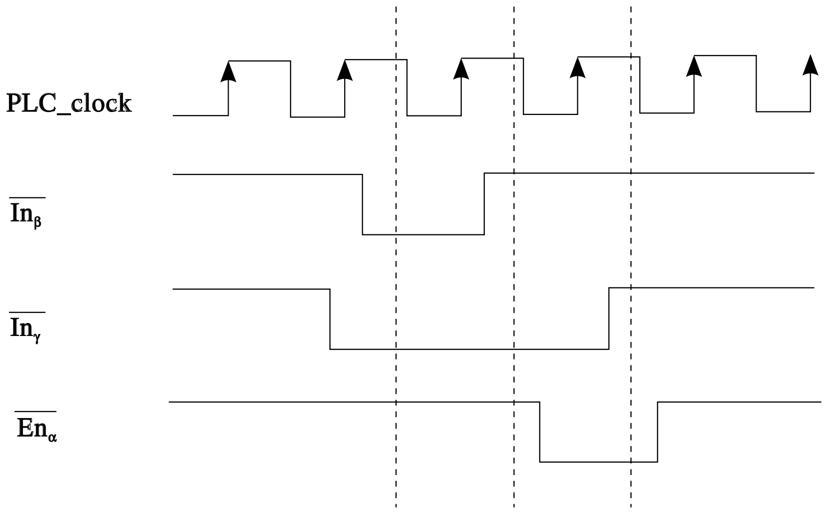

PLC_clock
$\overline{\mathrm{In}_\beta}$
$\overline{\mathrm{In}_\gamma}$
$\overline{\mathrm{En}_\alpha}$

Figure 9.2: Input and Output Signals

In the case of controllable events generated by the supervisor (see Section 6.2), the enablement output is monitored as the input event. For the controllable event α, generated by the supervisor, the signal $\overline{\mathrm{En}_\alpha}$ is both the output signal (enablement) and the input signal (occurrence indicator) of the event. In the diagram, the vertical dashed line indicates approximately where the state machine evaluates its inputs. One sees that $\overline{\mathrm{En}_\alpha}$ occurs after the line and thus doesn't get seen until the next falling edge. At this point α is considered to have occurred.

This interpretation works well with the modular implementation method. Each modular state machine generates its local value for each output. The conjunction of these local values is taken and the result is used to generate the output signal. Thus, the local controllers don't have to know the decisions of the other controllers; they simply monitor the output signal, and when it goes low, they know they were finally in agreement. This permits them to work together, yet not fall out of sync, falsely believing that a controllable event occurred just because one (or more, but not all) individual controller permitted it.

Event Demarcation

An important issue in the implementation of supervisors, is how to correctly determine the occurrence of an event: in particular, how to be able to determine from an input signal that an event has occurred and how many times that it has occurred. For instance, in Figure 9.2, the input signal $\overline{\mathrm{In}_\gamma}$ stretches over two falling edges of the clock pulse; thus it is seen twice by the state machines. The question is, "Does this mean the event occurred once or twice?" The answer is that it depends on the event and how it is defined.

The signal $\overline{\mathrm{In}_\gamma}$ is typical of a sensor input that a train is present. If you define the input to mean the event that a train is present, then the event occurred twice. If you define the input to mean that a train arrived at the sensor, then it has occurred once (when the signal is high (FALSE), this would be the event that no train is present). This requires that the input signal be conditioned to generate only one pulse of the correct length or that the state machine, after seeing the event once, moves to a state in which it will ignore the event (the event is self-looped) until such a time that the signal is FALSE again.

For controllable events generated by the supervisor, the event is considered to have occurred once when the enablement signal is low (TRUE) for one clock cycle. If the event is TRUE for two clock cycles, it is considered to have occurred twice.

NOTE: It is important that both output and input signals have a pulse width slightly longer (to allow for any setup and hold time required) than the period of *PLC_clock* to prevent an event from being missed by the state machines.

9.4 Control Equivalence

For the purpose of this work, we are only interested in whether a CMSSM will enforce the same control action as the original DES supervisor. The effects of transmission delay and concurrent events are not considered at this time and are left for future research. For a discussion on these subjects, refer to [6], [18] and [27]. To aid in the following discussion, some definitions are useful.

Definition: A DES supervisor and a CMSSM are control equivalent if they both take the same control action based on any sequence of events generated by the plant.

Definition: Let $\zeta : X \to Z$ map the control action specified in the supervisor at a given state to the same representation used by the CMSSM.

9.4.1 Centralized Controller

For centralized equivalence we have the following:

Theorem 5 *If* $\mathbf{S}$ *is controllable for plant* $\mathbf{G}$ *then the centralized implementation,* H*, is control equivalent to* $\mathbf{S}$.

Proof:

To prove Theorem 5, it is sufficient to show that Figure 9.3 commutes. It is clear from the definitions of λ, Γ, Ω, Φ, and ζ that this is in fact the case.

□

$$\begin{tikzcd}
\mathbf{X} \arrow[dd, "\lambda"'] & \times & \mathbf{\Sigma} \arrow[dd, "\Gamma"'] \arrow[r, "\xi"] & \mathbf{X} \arrow[dr, "\zeta"] & \\
 & & & & \mathbf{Z} \\
\mathbf{Q} & \times & \mathbf{I} \arrow[r, "\Omega"] & \mathbf{Q} \arrow[ur, "\Phi"'] &
\end{tikzcd}$$

Figure 9.3: Centralized Control Equivalence Diagram

9.4.2 Modular Controllers

For modular equivalence, we present the following theorem:

Theorem 6 *If* $\mathbf{S}_1$ *and* $\mathbf{S}_2$ *are controllable for plant* $\mathbf{G}$ *then the modular implementation,* H_{mod}*, is control equivalent to* $\mathbf{S}_1 \wedge \mathbf{S}_2$*.*

For the proof of Theorem 6, we require the following definition:

Definition: Let $\zeta : X_1 \times X_2 \rightarrow Z$ map the control action specified in $\mathbf{S}_1 \wedge \mathbf{S}_2$ at a given state to the same representation used by the CMSSM.

Proof:

To prove Theorem 6, it is sufficient to show that Figure 9.4 commutes, namely whenever the upper path through the diagram is defined, then so is the lower, and yields the same results. This follows directly from the definitions of the relevant maps. This result can easily be extended to n supervisors.

□

$$\begin{tikzcd}
X_1 \arrow[dd, "\lambda_1"'] & \times & X_2 \arrow[dd, "\lambda_2"'] & \times & \Sigma \arrow[dd, "\Gamma"'] \arrow[r, "\xi"] & X_1 \times X_2 \arrow[dr, "\zeta"] & \\
 & & & & & & Z \\
Q_1 & \times & Q_2 & \times & I \arrow[r, "\Omega"] & Q_1 \times Q_2 \arrow[ur, "\Phi"'] &
\end{tikzcd}$$

Figure 9.4: Modular Control Equivalence Diagram

Chapter 10

PLC Implementation

In this chapter, the PLC implementation of the clocked Moore synchronous state machine (CMSSM) is discussed. First, programmable logic controllers (PLC:[15],[19], and [29]) are examined, followed by a discussion of how the CMSSM are implemented on the PLC. Then, an overview of the PLC processor file that controls the testbed is given. Finally, the PLC controller implementation of the testbed supervisors is discussed.

10.1 Introduction to Programmable Logic Controllers

A programmable logic controller is similar to a microcomputer, but specialized for control purposes. The PLC is designed to be highly reliable, and robust enough to operate in hazardous industrial environments. An example of a PLC is shown in Figure 10.1. The PLC was chosen for the physical implementation of the synchronous state machine because of its robustness, and its wide use in industry.

The PLC traces its origins back to the 1960's. At the time, automated manufacturing lines were controlled by a wiring panel of relays, often filling an entire wall. The control logic would be designed by an engineer, and then an electrician would hardwire the various components together, using a blueprint (similar to the ladder diagram in Figure 5.4) given to him by the engineer. Wiring panels were obviously undesirable since they meant that changing the control program required re-wiring the entire panel. During this time, the entire manufacturing line would have to be shut down.

In 1968, General Motors Hydromatic Division drafted the design specifications for the first programmable logic controller. Their requirements were:

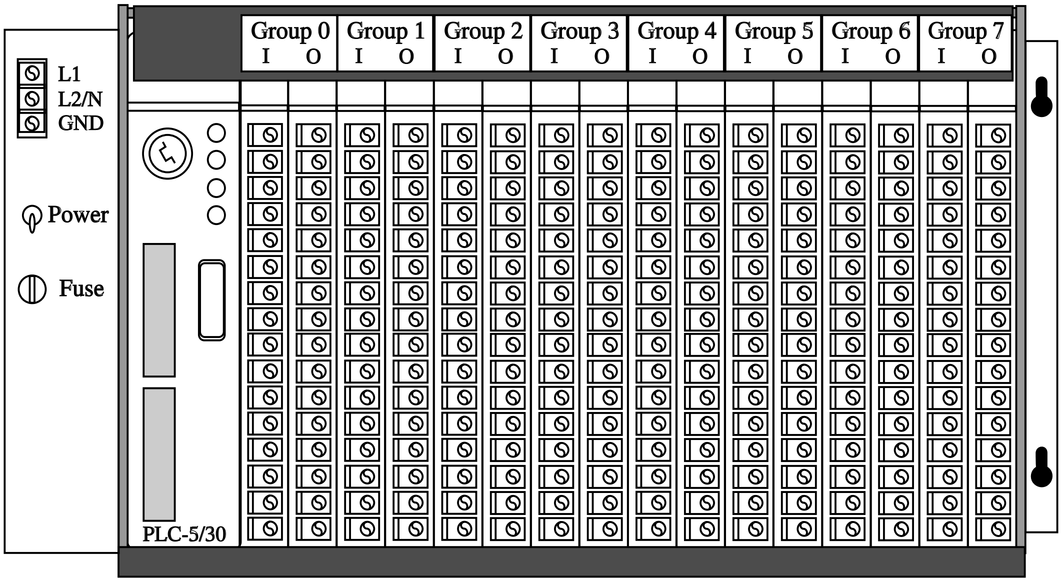


Figure 10.1: Allen-Bradley PLC-5/L30B

1. The PLC must be solid state (no mechanical devices).

2. It had to be programmable like a computer.

3. The PLC had to be robust enough to operate in an industrial environment.

4. Electricians and technicians must be able to easily program and use the PLC.

The last item is the main reason that programmable logic controllers are so popular today. The first PLC (developed by Gould Modicon) was programmed using relay ladder logic (discussed in Section 10.3.1) which looked a lot like the blueprint used by the electricians for wiring the control panel. This meant that an electrician would be able to easily understand and write relay ladder logic code.

Today, the PLC is a part of most manufacturing processes. It is used to control automobile manufacturing lines, robots, plastic injection molding machines, large flexographic printing presses, food packaging lines, and even to control railroad cars that perform maintenance (re-grind and true) on railroad tracks.

10.2 PLC Interpretation

The execution algorithm given in Section 9.3.3 will be translated into a RLL program. The code will have five main parts: initialize variables, evaluate next state logic, assign temporary outputs, set new states to be current states, and combine temporary outputs to generate system outputs. This code will compose the main control loop of the PLC program discussed in Section 10.3.5.

10.3 PLC Processor File

This section discusses the structure of the files that operate the testbed. The processor file is the file that gets loaded into the memory of the PLC. This file contains configuration information, data files, and relay ladder logic and sequential function chart programs. The relevant components of the processor file for the testbed are discussed below.

10.3.1 Relay Ladder Logic Files

The CMSSM will be implemented on the PLC in the form of a relay ladder logic (RLL:[3], [4]) program. A RLL program is made up of several steps called rungs. A rung is composed of a logical evaluation statement and an output assignment statement. If the logic evaluates to TRUE, then the output is energized. The basic components of a rung are shown in Fig. 10.2. In the example rung in the diagram, the output would evaluate as $Z = (X_0 \wedge \neg X_1) \vee \neg X_2$. A RLL program is evaluated by starting at the first rung and then processing each rung in order.

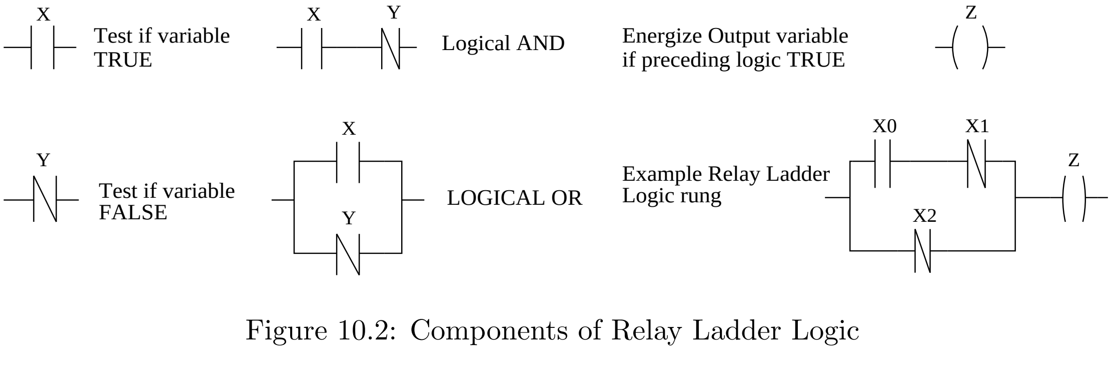

X
Test if variable TRUE
X
Y
Logical AND
Energize Output variable if preceding logic TRUE
Z
Y
Test if variable FALSE
X
Y
LOGICAL OR
Example Relay Ladder Logic rung
X0
X1
X2
Z

Figure 10.2: Components of Relay Ladder Logic

For the testbed, several RLL programs are used to implement the CMSSM. There are two initialization routines, *init1* and *init2* that set track switches and initialize variables. The main RLL program is *supvis.* This RLL program evaluates the state machines and determines the system outputs. The source code listing for the RLL programs is given in Appendix D.

10.3.2 Sequential Function Charts

A sequential function chart (SFC:[3], [4]) provides a means to divide a long program into several sections. Figure 10.3 shows the basic structure of a SFC. The SFC is composed of steps and transitions. Each step is related to an RLL program file. Each transition is related to a RLL transition file, similar to the example shown in the diagram. The PLC starts at the first step of the SFC, executes the RLL file associated with it, and then evaluates the transition. The current step is executed repeatedly until the transition evaluates to TRUE. The testbed is controlled by the SFC *main_des.* The structure of *main_des* is shown in

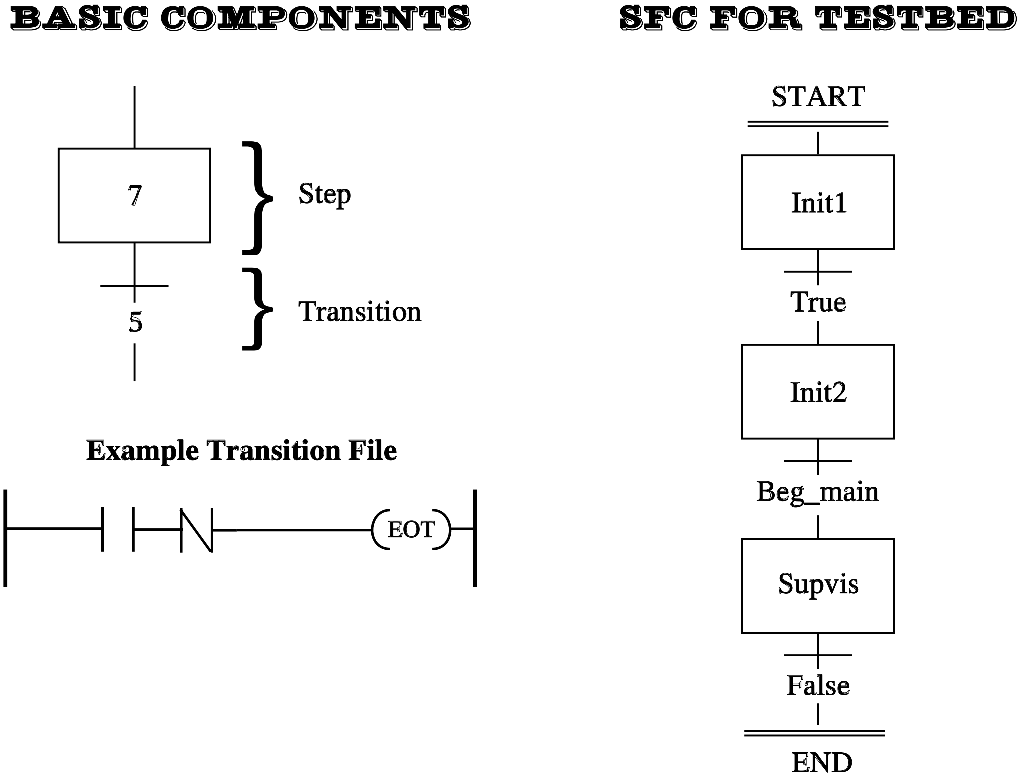


Figure 10.3: Structure of SFC

Figure 10.3. The SFC calls the RLL programs discussed in the section above. The source

code listing for *main_des* is given in Appendix D.

10.3.3 Switch Management Code

The RLL program *supvis* contains code that operates the 3-way switches (switches 3 and 8) and implements the switch request handlers discussed in Section 6.4.4. This code is not part of the supervisor, but part of the plant. The 3-way switch operators are required since the switches change position in two steps. First, the switch is moved to the straight track position, and then it is moved to the new curved position. The switch operators hide these details from the supervisors.

Both the 3-way switch operators, and the switch request handlers are implemented as state machines. The state machine for the operator of switch 3 is called *def_sw3*, and the state machine for switch 8 is called *def_sw8.* The PLC code for the switch operators starts on page 236 while the code for the request handlers starts on page 234.

10.3.4 PLC Data Files

The RLL programs for the testbed utilize data files to store its data. The integer file *N7* is used to store state variables and temporary outputs for the state machines. Initialization data as well as information for the operation of the timers (see following section) is also stored in file *N7.* The file *N7* must have at least 40 16-bit elements in it.

The binary file *B3* stores data used to "process" sensor inputs. This file must always have at least 18 16-bit elements in it.

Note: Most of the sensor inputs are "processed" by attaching them to a one-shot command. The output of the one shot goes TRUE for exactly one clock cycle when the input of the one shot makes a low to high transition. This is used to generate a single pulse of known duration when a sensor is activated. This is to prevent a single activation from being "seen" by a supervisor more than once. The output of the one shot is monitored by the supervisor instead of the actual sensor input.

10.3.5 Structure of Testbed PLC Program

The operation of the testbed's PLC program that implements the DES supervisors is shown in Figure 10.4. The relay ladder logic program that is currently being executed is labeled at the top of the box that marks entry into the program.

10.4 PLC Testbed Controllers

The 29 DES supervisors were implemented using the modular method outlined in Section 9.3.2. The resulting CMSSM were then converted into PLC code as described in Section 10.2. Below, example implementations of the main types of the testbed supervisors are given. The relay ladder code of the remaining supervisors can be found in Section D.2.3. From this code, it is easy to re-create the CMSSM that corresponds to the given supervisor.

It is important to note the following points:

- The state machines shown below only show inputs that are relevant to deciding the next state, and only those outputs that the controller actually disables.
- Each state in the CMSSMs shown below has a self-looped transition labeled *OTHER*. This means that, if the logical conditions on the other arrows leaving the state are not met, then this "catch-all" condition takes effect, and the CMSSM stays in its current state.

10.4.1 Discussion of Concurrency

As discussed in Section 9.4, this work does not formally deal with concurrent events; that is left for future work. However, the collision protection supervisors each contain instances in which the issue of concurrent events couldn't be avoided. These issues were handled in an ad hoc manner. If time permitted, they could be formalized by adding new events to the plant model that stood for the simultaneous occurrence of the events in question. Thus, the supervisors could be specifically instructed on how to handle these situations. In the section below concerning the collision protection controllers, specific instances of concurrent events will be discussed.

10.4.2 Crane Controllers

As an example of the crane supervisors, the implementation of supervisor **sup_crane1** (shown on page 84) is shown below. The CMSSM representation is shown in Figure 10.5.

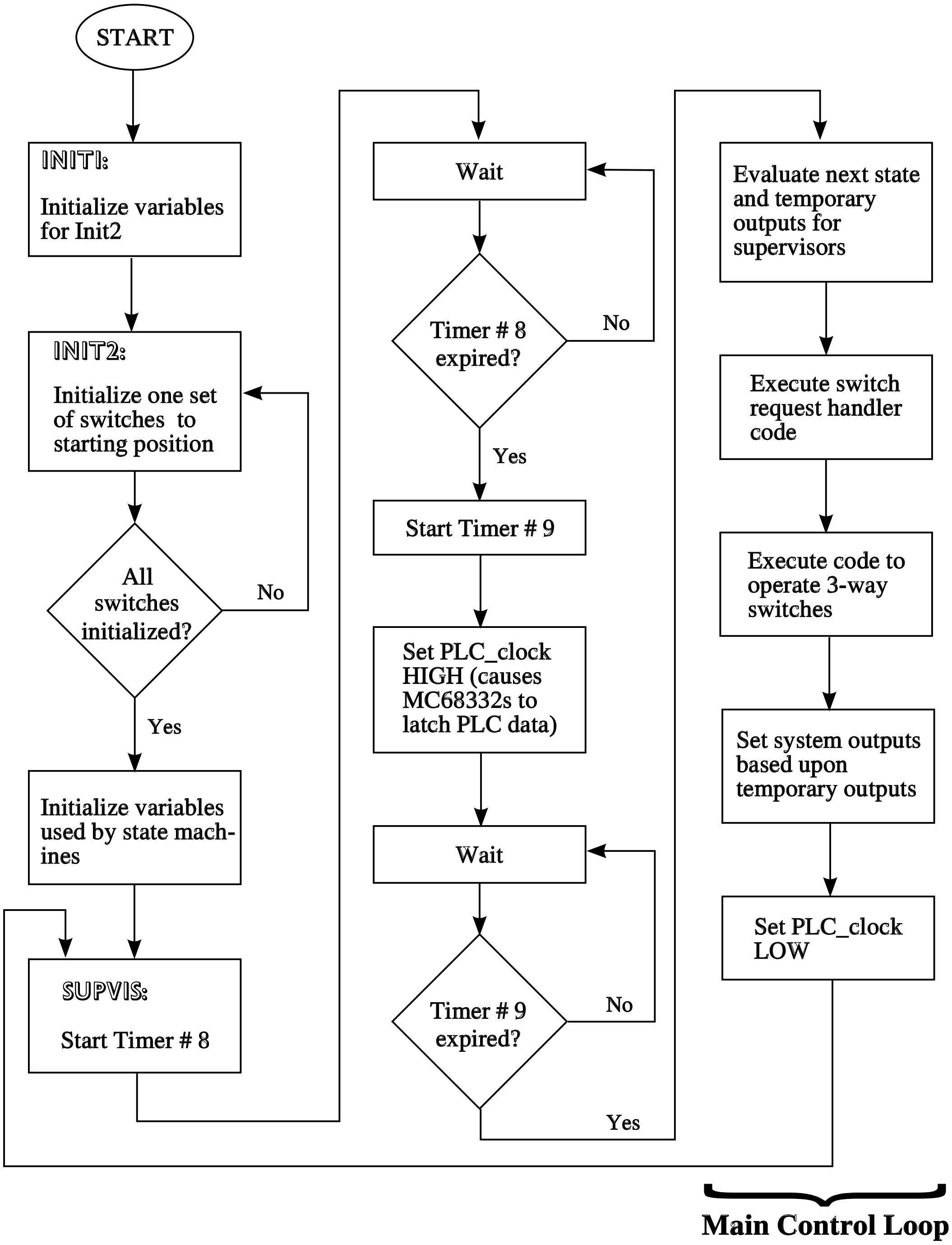

START
INIT1:
Initialize variables for Init2
INIT2:
Initialize one set of switches to starting position
All switches initialized?
No
Yes
Initialize variables used by state mach-ines
SUPVIS:
Start Timer # 8
Wait
Timer # 8 expired?
No
Yes
Start Timer # 9
Set PLC_clock HIGH (causes MC68332s to latch PLC data)
Wait
Timer # 9 expired?
No
Yes
Evaluate next state and temporary outputs for supervisors
Execute switch request handler code
Execute code to operate 3-way switches
Set system outputs based upon temporary outputs
Set PLC_clock LOW
Main Control Loop

Figure 10.4: Flow Diagram for PLC Program

The reset state, next state equations, and the local output equations[1] are defined below. Figure 10.6 shows the equivalent Relay Ladder Logic.

NOTE: In Figure 10.5, the output *"s_ld1"* is the actual system output. This is because supervisor **sup_crane1** is the only supervisor that disables the event.

$A_{Res} = [A_{1Res}, A_{0Res}] = [0, 0]$

$$A_{1new} = (\neg A_1 \ \wedge \ \neg A_0 \ \wedge \ \ \text{t2\_at13}) \vee \ (A_1 \wedge \ \neg \ \text{f\_ld1})$$

$$A_{0new} = (\neg A_1 \ \wedge \ \neg A_0 \ \wedge \ \ \text{t1\_at13}) \vee \ (A_0 \wedge \ \neg \ \text{f\_ld1})$$

$\text{en\_t2a} = \neg A_{1new}$

$en\_t1a = \neg A_{0new}$

$s\_ld1 \ \ = A_{1new} \ \vee \ A_{0new}$

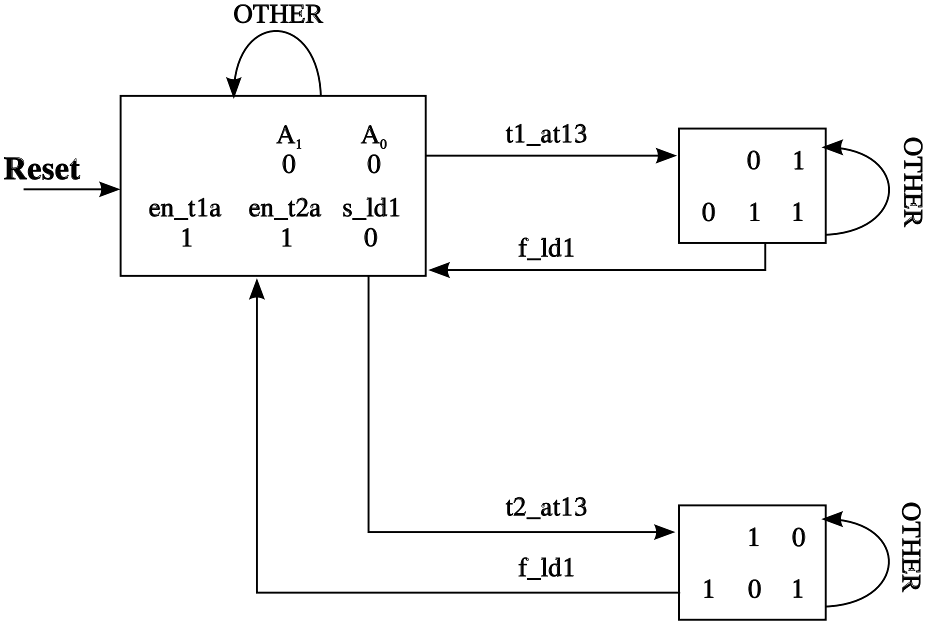


Figure 10.5: CMSSM for **sup_crane1**

[1] Only local outputs are shown here. For the evaluation of the system outputs, refer to the PLC source code starting on page 238.

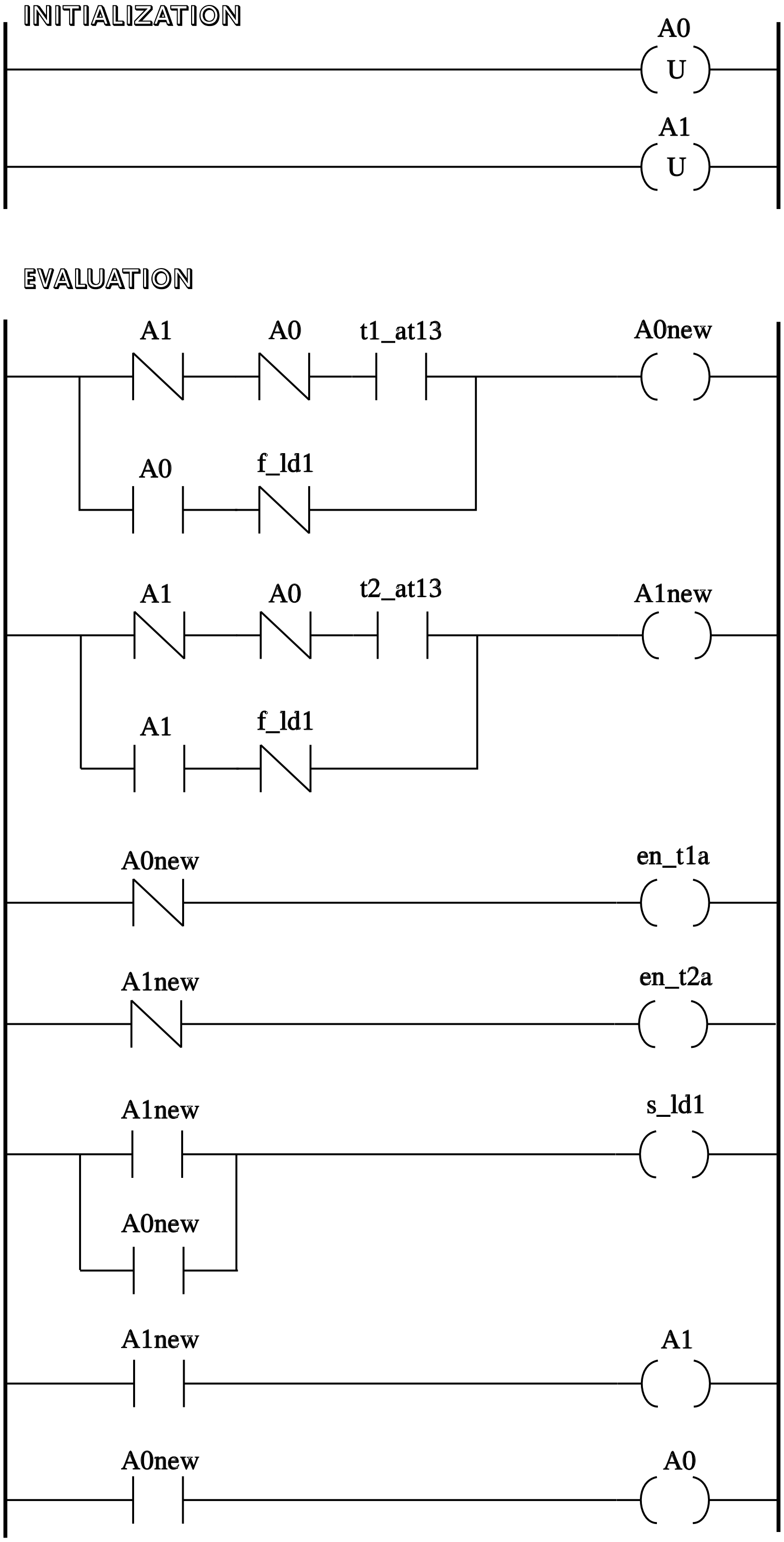


Figure 10.6: RLL for **sup_crane1**

10.4.3 Routing Controllers

As an example of the routing supervisors, the state machine implementation of supervisor **sup_rt2_sw8** (shown on page 91) is shown in Figure 10.7. Refer to the PLC source code on page 228 for the logic equations that define the CMSSM.

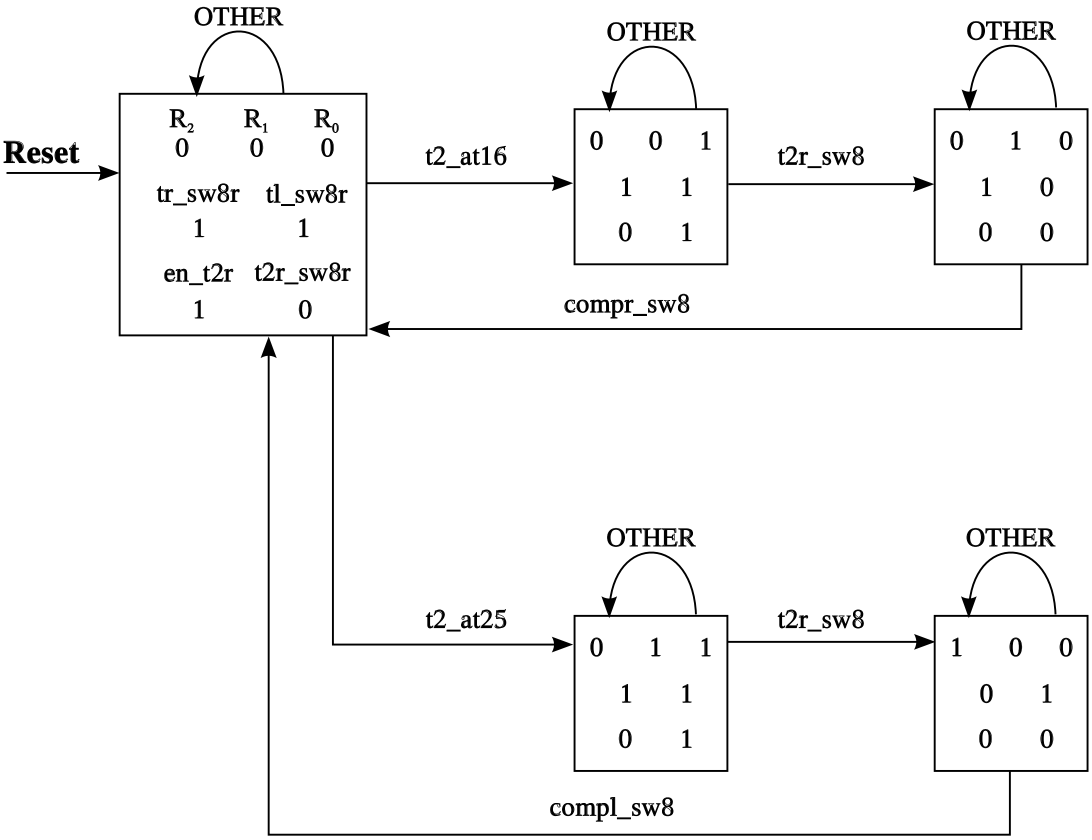


Figure 10.7: CMSSM for **sup_rt2_sw8**

10.4.4 Collision Protection Controllers

The collision protection supervisors are broken down into three classes of supervisors of similar structure. This is done purely as a descriptive aide. An example implementation for each class is given below.

Class I Supervisors

As an example of this class of supervisors, the state machine implementation of supervisor **sup_cprot_23_24** (shown on page 93) is shown in Figure 10.8. Refer to the PLC source code on page 216 for the logic equations that define the CMSSM.

Examination of **sup_cprot_23_24** shows that both states 1 and 3 contain the possibility of concurrent events[2]. At state 1, events *"t2_at22"* and *"t1_at26"* could both conceivably occur concurrently. The order of occurrence would dictate which states the supervisor would enter before reaching the same final state. If *"t2_at22"* occurred first, then the supervisor would enter state 2 before entering state 0. Following this path would mean that the event *"en_train1"* would be disabled briefly. The event would not have been disabled at all if *"t1_at26"* had occurred first. Since a variation of an output could occur, the order the events actually occur may be important. For the testbed, all that is important at this point is that train 1 had reached sensor 26 thus leaving that section of track. It is no longer important that train 2 is about to enter the track section, so there is no need to disable train 2. The CMMSM in Figure 10.8 returns to state 0 if events *"t2_at22"* and *"t1_at26"* occur concurrently. A similar situation is present at state 3 of the supervisor.

[2] For the purposes of this work, concurrent events are taken to mean two or more events that occur so close together that it is impossible to distinguish which one occurred first.

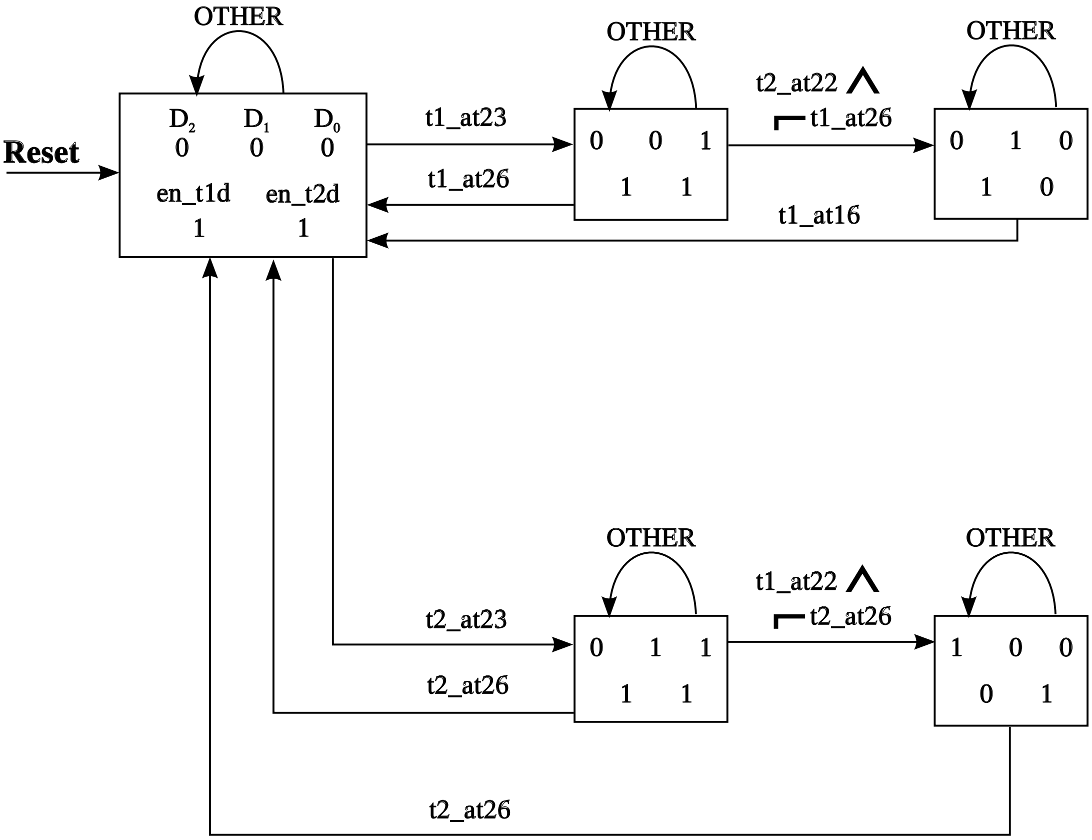


Figure 10.8: CMSSM for **sup_cprot_23_24**

Class II Supervisors

As an example of this class of supervisor, the state machine implementation of supervisor **sup_cprot_11_13** (shown on page 98) is shown in Figure 10.9. Refer to the PLC source code on page 224 for the logic equations that define the CMSSM.

For supervisor **sup_cprot_11_13**, states 1 and 3 contain possibilities for concurrent events similar to the instances in supervisor **sup_cprot_23_24**. Comparing the original supervisor and the CMSSM implementation, one can easily determine the relevant issues and the solution employed to resolve them.

Class III Supervisors

As an example of this class of supervisor, the state machine implementation of supervisor **sup_cprot_19_20** (shown on page 101) is shown in Figure 10.10. Refer to the PLC source code on page 218 for the logic equations that define the CMSSM.

For supervisor **sup_cprot_19_20**, states 1, 3, 5, 7, 9, and 10 contain possibilities for concurrent events. Comparing the original supervisor and the CMSSM implementation, one can determine the relevant issues and the solution employed to resolve them.

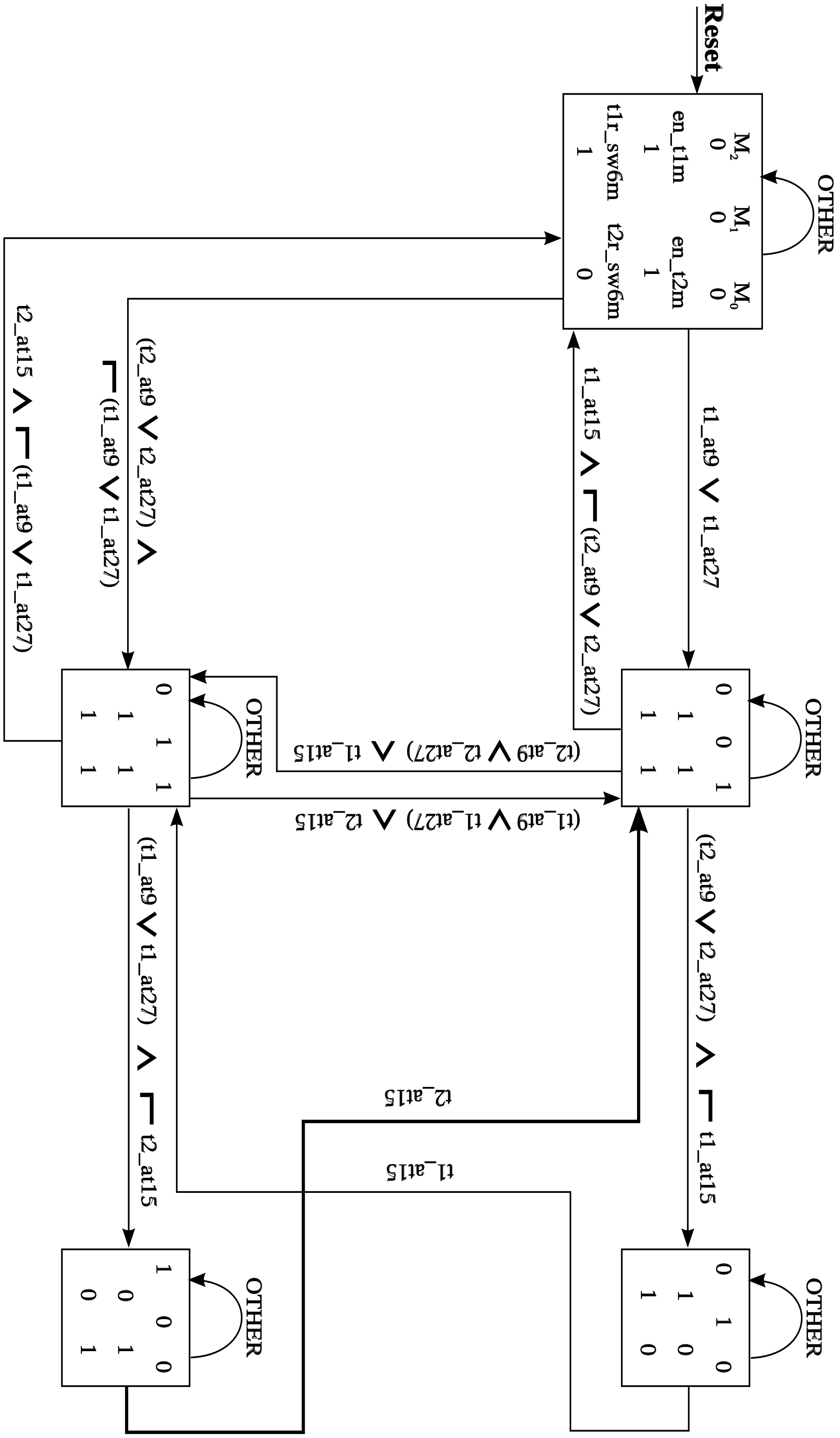

Reset
OTHER
M2 M1 M0
0 0 0
en_t1m en_t2m
1 1
t1r_sw6m t2r_sw6m
1 0
t1_at9 ∨ t1_at27
t1_at15 ∧ ¬(t2_at9 ∨ t2_at27)
(t2_at9 ∨ t2_at27) ∧ ¬(t1_at9 ∨ t1_at27)
t2_at15 ∧ ¬(t1_at9 ∨ t1_at27)
0 0 1
1 1 1
1 1 1
0 1 1
1 1 1
1 1 1
(t2_at9 ∨ t2_at27) ∧ t1_at15
(t1_at9 ∨ t1_at27) ∧ t2_at15
(t2_at9 ∨ t2_at27) ∧ ¬ t1_at15
(t1_at9 ∨ t1_at27) ∧ ¬ t2_at15
t2_at15
t1_at15
0 1 0
1 0 1
1 0 0
0 1 1
0 0 1

Figure 10.9: CMSSM for **sup_cprot_11_13**

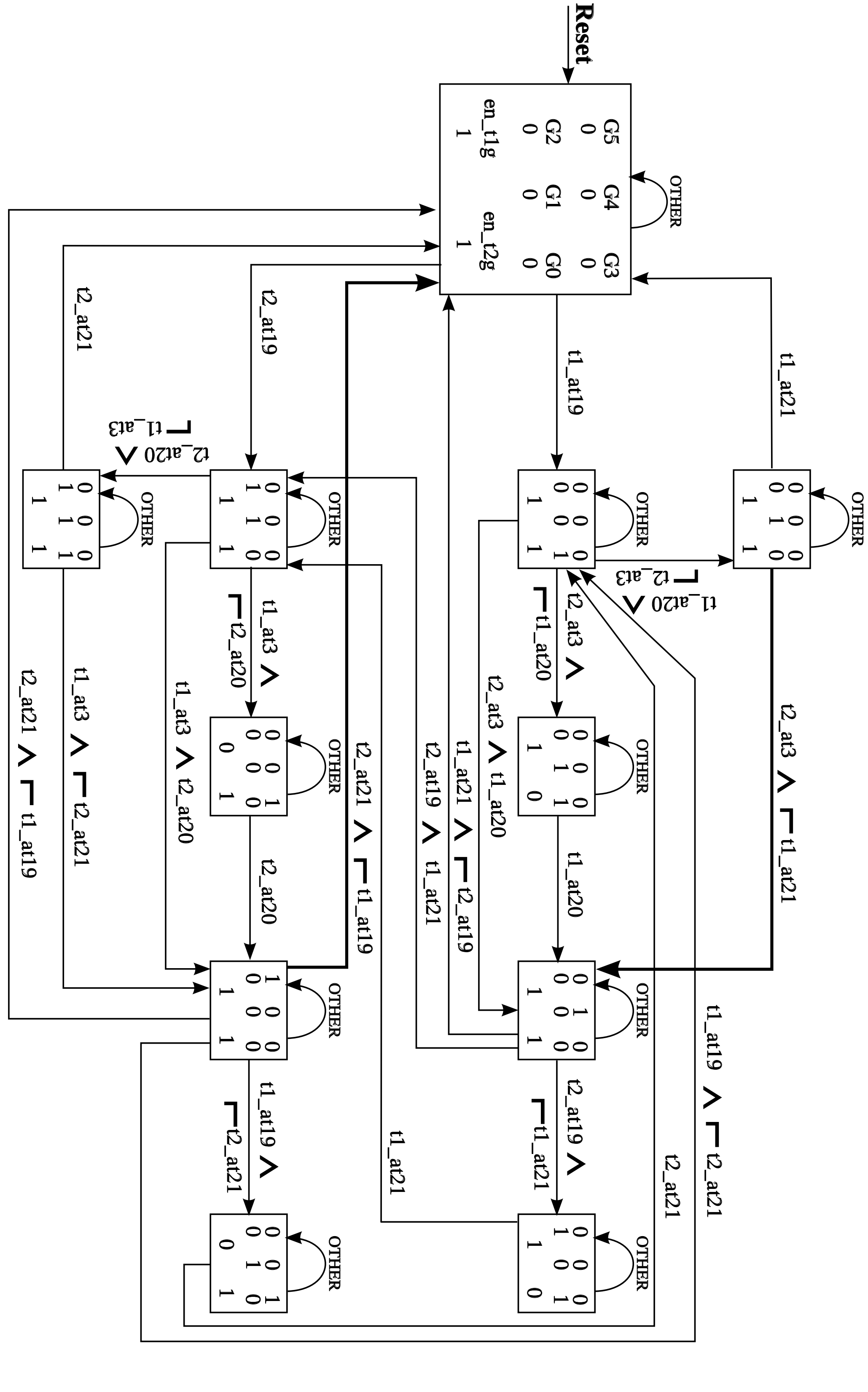


Figure 10.10: CMSSM for **sup_cprot_19_20**

Chapter 11

Discussion of Results

The manufacturing testbed has been designed, assembled and tested. The testbed hardware is operational, although there still exist some minor hardware problems. A detailed plant model composed of 59 sub-models has been developed, giving a composite model with state size on the order of 10^{16}. Then, 29 supervisors were designed to implement the required control specifications. Since the size of the plant model was beyond the capacity of current methods to handle, model reduction theorems were created that enabled the verification of the controllability of the supervisors, using sub-plants of the original model as small as (16 states, 72 transitions) and as large as (4761 states, 136 896 transitions). As yet non-blocking has not been verified due to the high interaction of the sub-models, but non-blocking theorems have been created that, although not sufficient by themselves to verify non-blocking, greatly reduce the work involved.

The testbed model was developed by interpreting the model as an interface to the underlying software that physically controls the testbed. This creates a specification dictating how the DES controllers interact with the low level software. This effectively creates an abstraction layer that isolates the high level DES supervisors from the low level hardware. The abstraction layer permits changes in the hardware or software without requiring changes in the DES supervisor.

The physical interpretation of “events occurring” as well as the “enablement” of controllable events has been defined. A closed-looped feedback interpretation has been presented that permits events to be generated by the supervisor. This makes it easy to design supervisors that need to send commands to the plant. Also, guidelines to aid in modeling and designing supervisors for real plants have been given.

The supervisor implementation has been formally defined as a clocked Moore synchronous state machine (CMSSM). A method to translate a DES supervisor into an equivalent CMSSM is outlined. The implementation of the CMSSM as a relay ladder logic program on a PLC was then specified. This general methodology does not yet deal with transmission delay and concurrent events.

Software programs for the MC68332 processors have been written to generate the discrete interface required by the plant model specification. The supervisors have been translated into CMSSM. PLC source code has been written to implement the CMSSM and provide an interface to the rest of the testbed. All software has been tested, and the testbed is now operational, demonstrating that the supervisor implementation method works.

Chapter 12

Conclusions and Future Work

This chapter contains a discussion of conclusions and future work. They are discussed in the following sections.

12.1 Conclusions

After completion of the project described in this thesis, several conclusions are evident.

- The success of the testbed demonstrates that supervisory control can effectively be used as a design tool for reasonably large, real systems.

- It is evident that modular design of both the plant model and the supervisor is essential to deal with large, real systems. The size of such systems currently makes any other method infeasible.

- The model reduction theorems greatly extend the possible size of plants that RW theory can be applied to. The theorems immensely reduce the amount of computations necessary.

- By considering the plant model as an interface specification, RW theory can easily be applied to programmable plants. The interface specification provides a necessary abstraction level between the high level DES supervisors and the low level software and hardware.

- The generic implementation methodology presented provides an excellent means to bridge the gap between DES theory and applications. By giving a physical interpretation to the theory, the implementation method makes it easier for designers to model real plants, and to design effective supervisors. Because the method is generic, it can be applied to a wide range of applications. Also, since the method uses CMSSM as an intermediate step, it can easily be modified for implementation on any digital logic device.

- To allow for easy implementation, it is important that the implementation methodology be extended to handle transmission delay and concurrent events. The formalism presented in this work provides an excellent starting point for this extension.

- The manufacturing testbed was an excellent instrument for investigating implementation issues. It posed a myriad of relevant, unexpected problems which yielded a wealth of insight. The testbed is useful for testing theories as well as providing inspiration for developing new ones.

- The current failure to verify non-blocking for the testbed shows that the non-blocking issue is one of the main obstacles to be overcome to enable RW theory to handle large systems.

12.2 Future Work

There are several potential future projects that are related or inspired by this work. They are listed below.

- The implementation algorithm should be extended to handle transmission delay and concurrent events. A computer program should be written to implement the new algorithm to make implementing supervisors easier.

- It is imperative that a method be developed for the verification of non-blocking for large plants. A possible approach for verifying that a given supervisor is non-blocking for the plant, would be to remove "extraneous"[1] events and generate a new, smaller plant model that still retains the information necessary to verify non-blocking. Ideally, this method would be applied to the sub-models before they are combined to form the composite model. Various other methods of abstraction should also be investigated.

[1] This refers to events that are "extraneous" for a given supervisor.

- Develop an approach to handle "unreliable"[2] events. Because of pulse widths being too short, some of the testbed's sensor outputs sometimes get missed by the PLC controllers. The ineffective signal outputs cause the sensor events to be "unreliable". In most cases, there is another sensor shortly after the ineffective one that works properly and can be used as a backup if the first sensor fails. To obtain an optimal yet dependable supervisor, the supervisor should check the first sensor event, and then check the second sensor event as a backup. It would be useful to be able to generate a supervisor that could do this automatically. Even if all events were "reliable", this approach would be useful for important systems that require redundancy.

[2] An event is "unreliable" if it sometimes fails to occur when it is supposed to.

Bibliography

[1] Allen-Bradley, Inc. *1785 PLC-5 Family Programmable Controllers Hardware Installation Manual*, 4.4th edition, 1993.

[2] Allen-Bradley, Inc. *1785 PLC-5 Programmable Controllers Design Manual*, 4.4th edition, 1993.

[3] Allen-Bradley, Inc. *PLC-5 Programming Software, Instruction Set Reference*, 4.4th edition, 1993.

[4] Allen-Bradley, Inc. *PLC-5 Programming Software, Programming*, 4.4th edition, 1993.

[5] Allen-Bradley, Inc. *PLC-5 Programming Software, Software Configuration and Maintenance*, 4.4th edition, 1993.

[6] S. Balemi. Input/output discrete event processes and communication delays. *Discrete Event Dynamic Systems: Theory and Applications*, 4(1):41–85, 1994.

[7] M. Bolton. *Digital Systems Design with Programmable Logic*. Addison-Wesley, 1990.

[8] B. A. Brandin. *Real-Time Supervisory Control of Automated Manufacturing Systems*. PhD thesis, Department of Electrical Engineering, University of Toronto, 1993. Also appears as Systems Control Group technical report # 9302, Department of Electrical Engineering, University of Toronto, February 1993.

[9] B. A. Brandin and W. M. Wonham. Final Report: Ontario/Rhône-Alpes Project on the Supervisory Control of Automated Manufacturing Systems. Technical report, Department of Electrical and Computer Engineering, University of Toronto, 1995.

[10] B.A. Brandin, W.M. Wonham, and B. Benhabib. Discrete-event systems supervisory control applied to the management of manufacturing workcells. In *7th International*

Conference on Computer Aided Manufacturing Engineering, pages 527–536, Cookeville, Elsevier, 1991.

[11] T. Catherall. *A User's Guide to the Märklin Digital System.* Märklin, Inc., 4th edition, 1991.

[12] Data I/O Corporation. *ABEL-HDL Design Software, Language Reference*, October 1991.

[13] W. Ford and W. Topp. *MC68000, Assembly Language and Systems Programming.* D.C. Heath and Company, 1987.

[14] Gebr. Märklin & Cie GmbH. *Märklin Digital Control Unit 6021*, 1993.

[15] D. Johnson. *Programmable Controllers for Factory Automation.* Marcel Dekker, 1987.

[16] R.J. Leduc and W.M. Wonham. Discrete event systems modeling and control of a manufacturing testbed. In *Canadian Conference on Electrical and Computer Engineering*, pages 793–796, Sept 1995.

[17] R.J. Leduc and W.M. Wonham. PLC implementation of a DES supervisor for a manufacturing testbed. In *Thirty-third Annual Allerton Conference on Communication, Control, and Computing*, Oct 1995.

[18] Yong Li. Supervisory control of real-time discrete-event systems. Master's thesis, Department of Electrical Engineering, University of Toronto, Toronto, ONT, 1986.

[19] G. Michel. *Programmable Logic Controllers- Architecture and Applications.* Wiley, 1990.

[20] M.P.O Morford and R.J. Lenardon. *Classical Mythology.* David McKay Company, 1976.

[21] Motorola, Inc. *Motorola Freeware 32-bit Cross Assembler User's Manual.*

[22] Motorola, Inc. *MC68681 Dual Asynchronous Receiver/Transceiver*, 1985.

[23] Motorola, Inc. *MC68332 User's Manual*, 1990.

[24] Motorola, Inc. *CPU32Bug Debug Monitor User's Manual*, 1991.

[25] Motorola, Inc. *MC68332EVK Evaluation Kit User's Manual*, 1993.

[26] Motorola, Inc. *Modular Microcontroller Family Time Processor Unit*, 1993.

[27] T.- M. Pai. Real-time implementation of discrete-event controllers. Master's thesis, Department of Electrical Engineering, University of Toronto, Toronto, ONT, 1990.

[28] P. Ramadge and W. Wonham. Supervisory control of a class of discrete-event processes. *SIAM J. Control Optim*, 25(1):206–230, 1987.

[29] J. Stenerson. *Fundamentals of Programmable Logic Controllers, Sensors, and Communications*. Prentice Hall, Inc., 1993.

[30] R. Strobel and A. Johnson. Pocket pagers in lots of one. *IEEE Spectrum*, pages 29–32, Sept 1993.

[31] J. Wakerley. *Digital Design Principles*. Prentice-Hall, Inc., 1990.

[32] W. Wonham and P. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J. Control Optim*, 25(3):637–659, 1987.

[33] W.M. Wonham. *Notes on Control of Discrete-Event Systems*. Department of Electrical and Computer Engineering, University of Toronto, 1995.

Appendix A

Circuit Schematics

This appendix contains the circuit schematics for the circuit boards described in Chapter 4. The schematics for the sensor boards, the remote switching board, and the expansion boards are given below.

NOTE: In some of the following schematic pages, there are extra graphics that extend past the diagram. This is due to an error in the program that generates the schematics.

A.1 Sensor Board Circuit Schematics

This section contains the schematics for sensor boards 0 to 6, as well as the RS-232 board that goes with sensor board 6. Figure A.1 shows the board layout for each board.

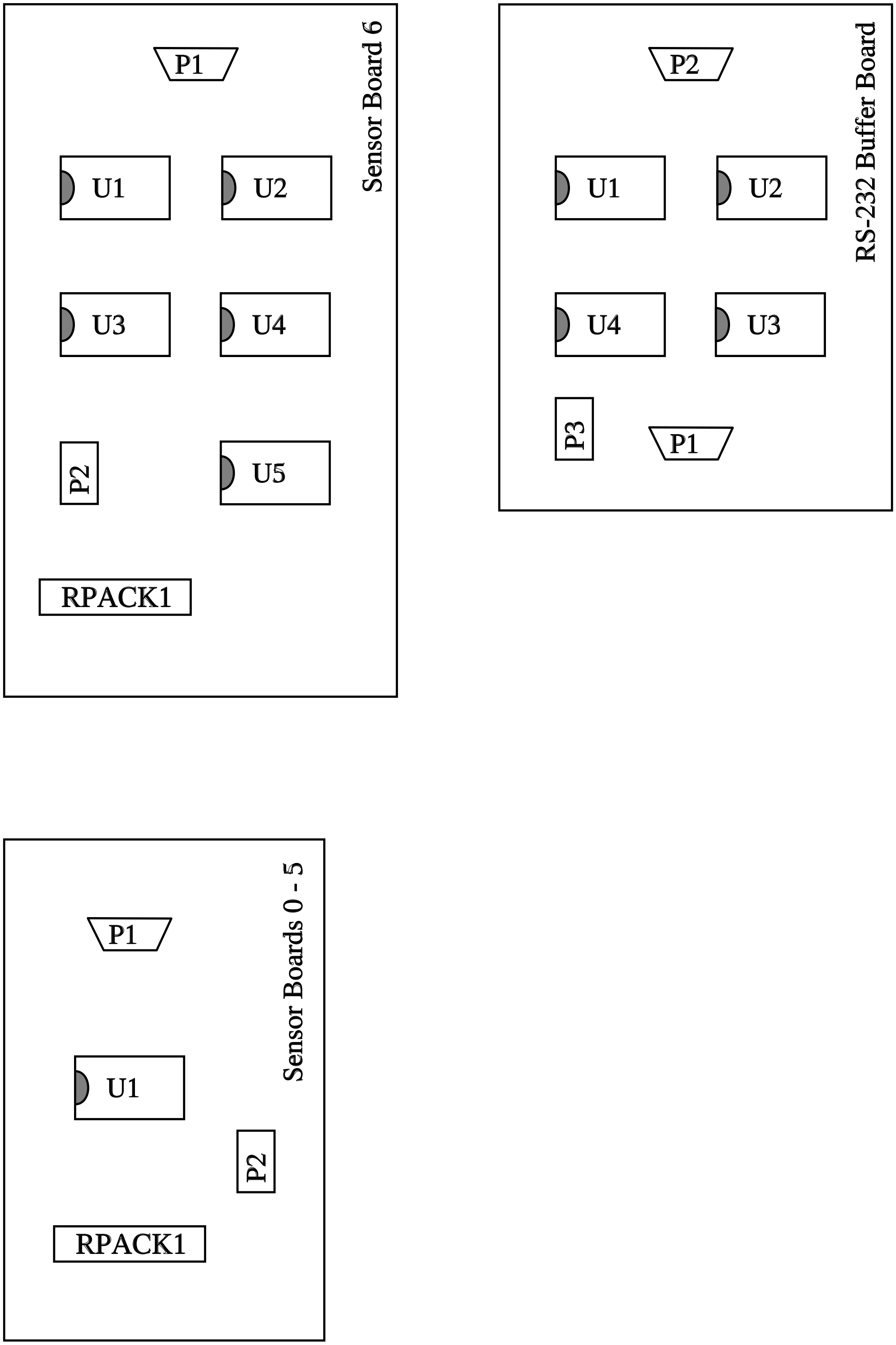
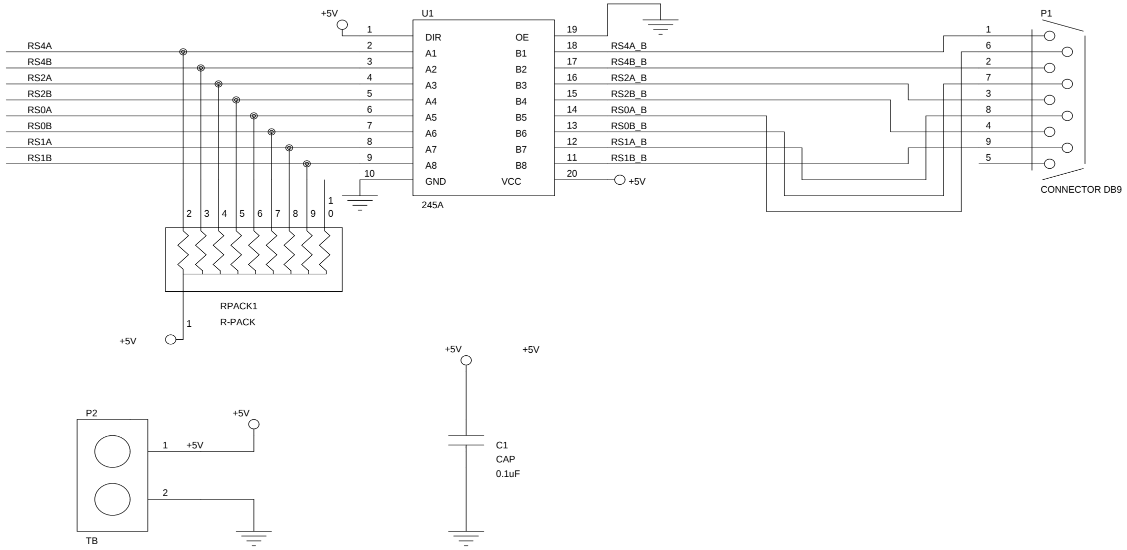


Figure A.1: Layout of Sensor Boards

SENSOR BOARD 0

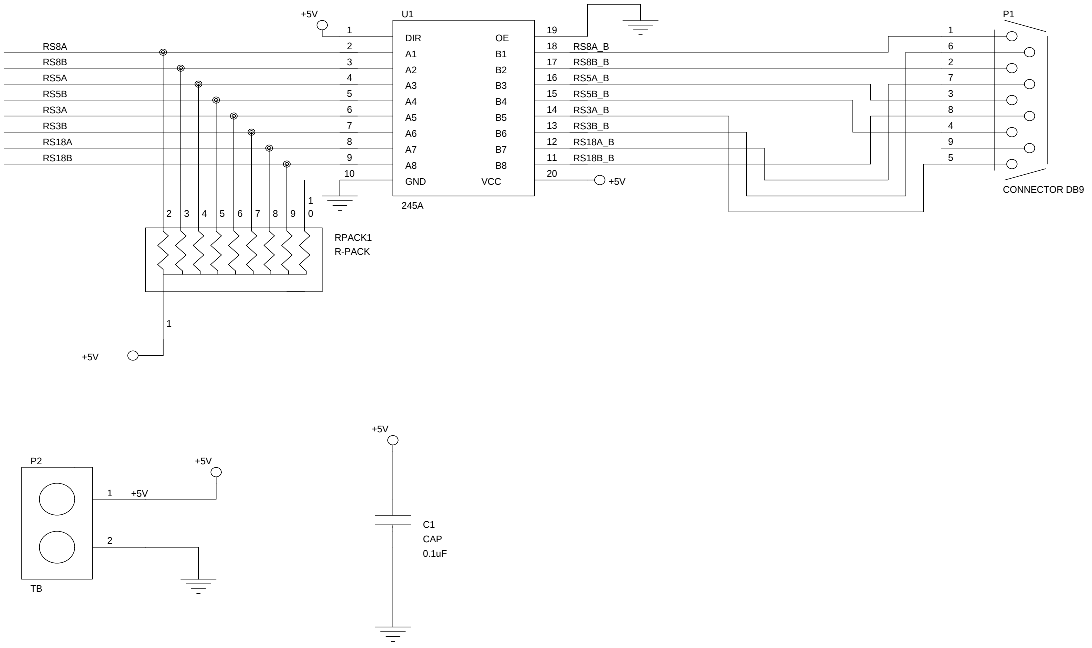
Signals Hardwired to Test bed. There are no connectors.



UNIVERSITY OF TORONTO		
Title Sensor Board 0		
Size A	Document Number 1	REV 1.0
Date: December 5, 1995	Sheet 1 of 1	

SENSOR BOARD 1

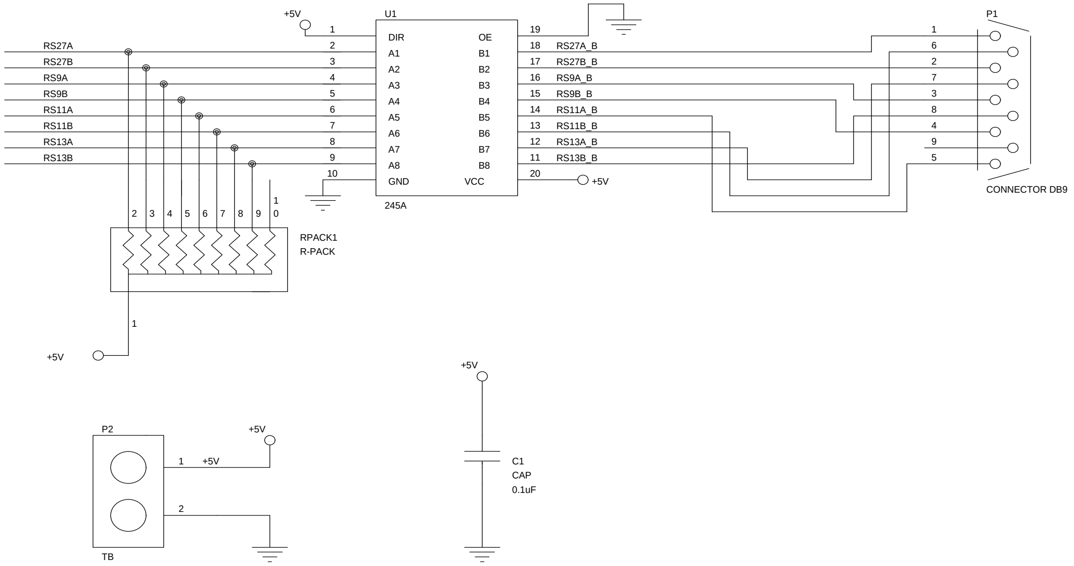
Signals Hardwired to Test bed. There are no connectors.



University of Toronto		
Title Sensor Board 1		
Size A	Document Number 1	REV 1.0
Date: December 5, 1995	Sheet 1 of 1	

SENSOR BOARD 2

Signals Hardwired to Test bed. There are no connectors.



UNIVERSITY OF TORONTO

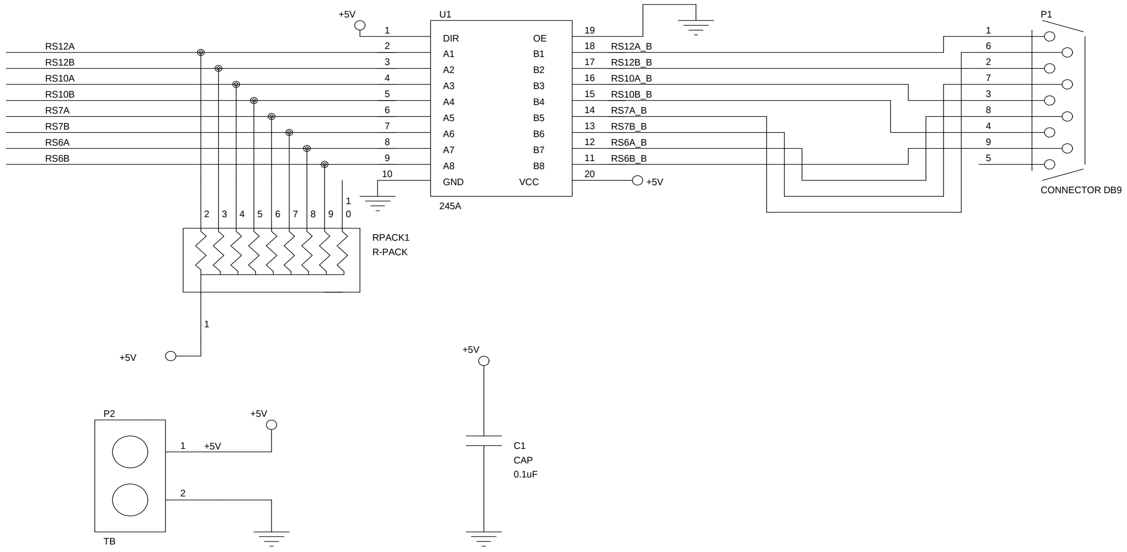
Title
Sensor Board 2

Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 Sheet 1 of 1

SENSOR BOARD 3

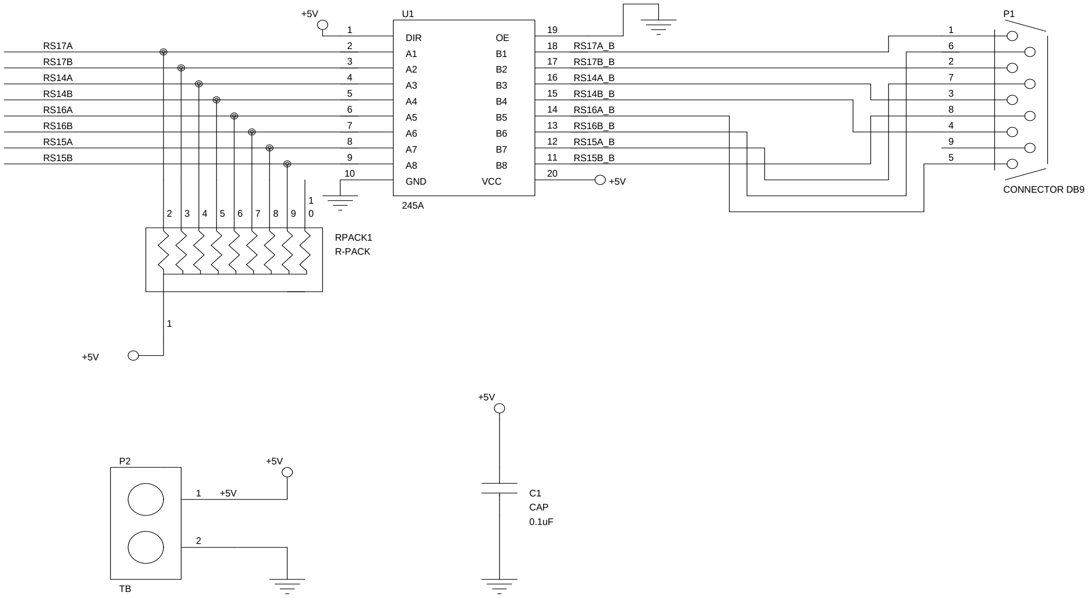
Signals Hardwired to Test bed. There are no connectors.



UNIVERSITY OF TORONTO			
Title Sensor Board 3			
Size A	Document Number 1		REV 1.0
Date: December 5, 1995	Sheet 1	of 1	

SENSOR BOARD 4

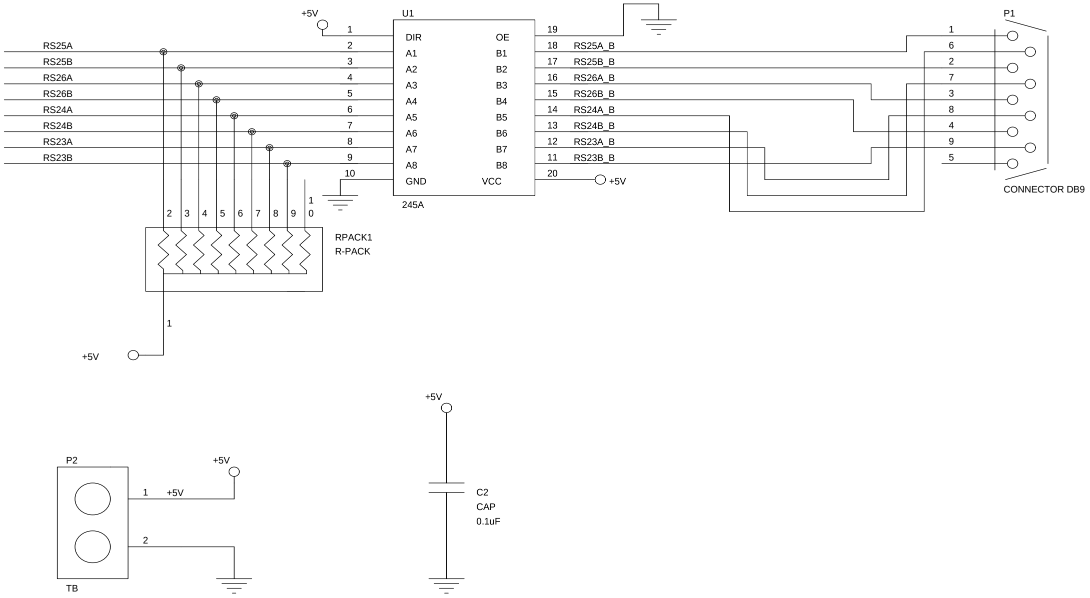
Signals Hardwired to Test bed. There are no connectors.



UNIVERSITY OF TORONTO		
Title Sensor Board 4		
Size A	Document Number 1	REV 1.0
Date: December 5, 1995	Sheet 1 of 1	

SENSOR BOARD 5

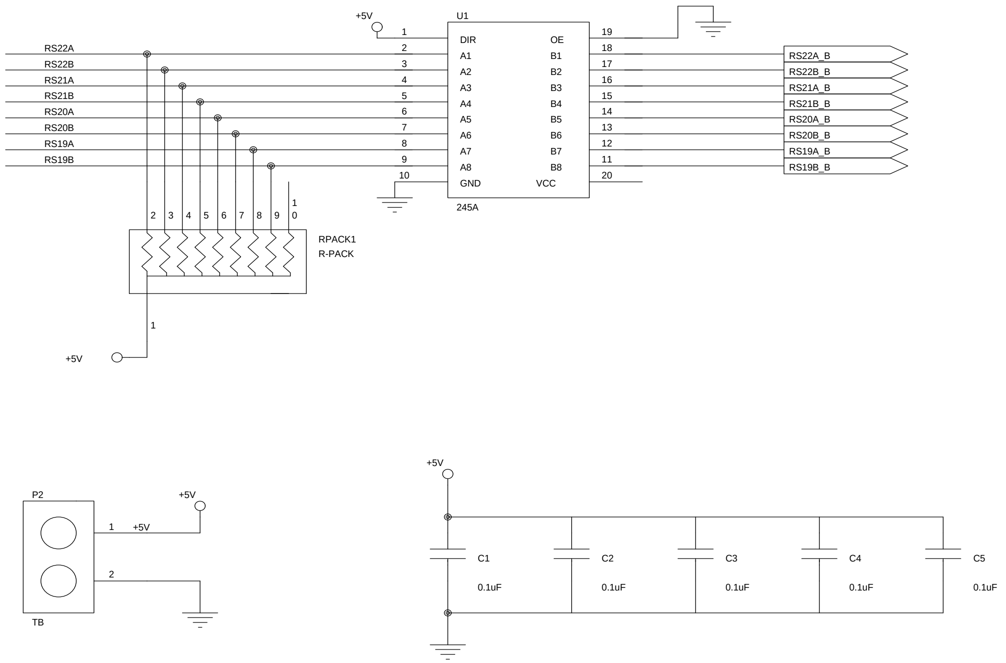
Signals Hardwired to Test bed. There are no connectors.



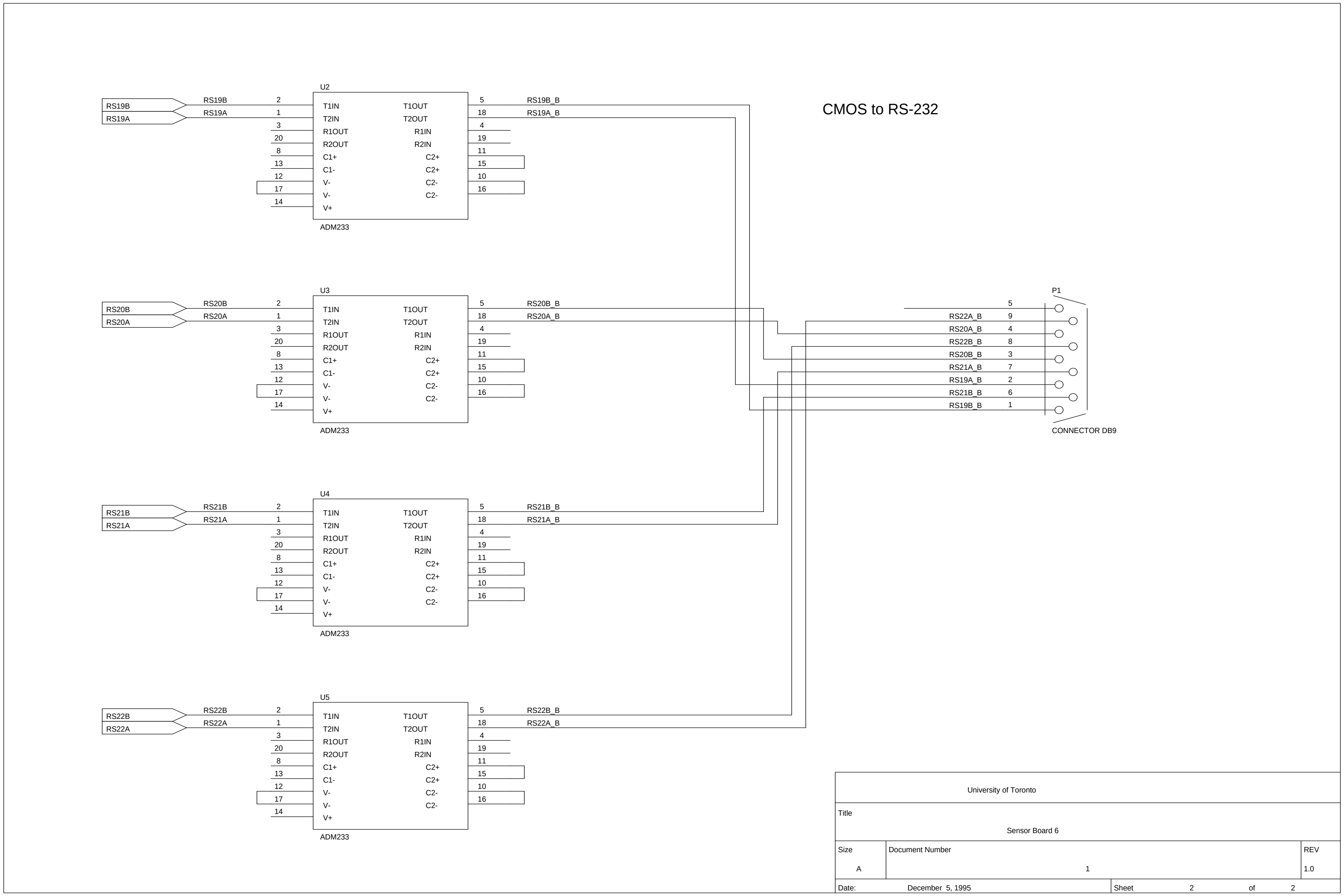
UNIVERSITY OF TORONTO				
Title	Sensor Board 5			
Size	Document Number			REV
A	1			1.0
Date:	December 5, 1995	Sheet	1 of 1	

SENSOR BOARD 6

Signals Hardwired to Test bed. There are no connectors.

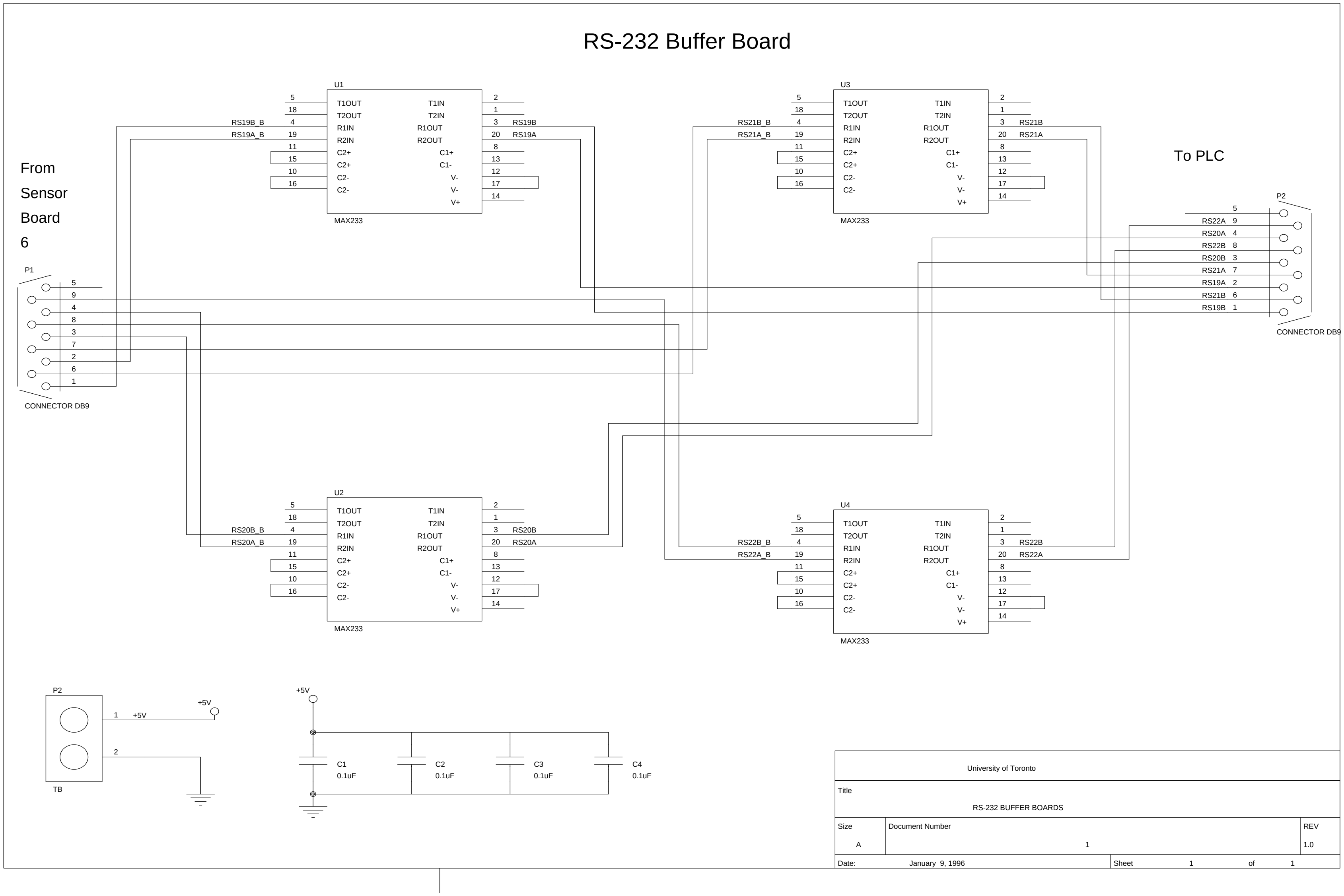


UNIVERSITY OF TORONTO		
Title Sensor Board 6		
Size A	Document Number 1	REV 1.0
Date: December 5, 1995	Sheet 1 of 2	



University of Toronto			
Title Sensor Board 6			
Size A	Document Number 1		REV 1.0
Date: December 5, 1995	Sheet 2	of 2	

RS-232 Buffer Board



University of Toronto

Title

RS-232 BUFFER BOARDS

Size	Document Number	REV
A	1	1.0

Date: January 9, 1996 Sheet 1 of 1

A.2 Remote Switching Board Circuit Schematics

This section contains the schematics for the remote switching circuit. Figure A.2 shows the layout of the board.

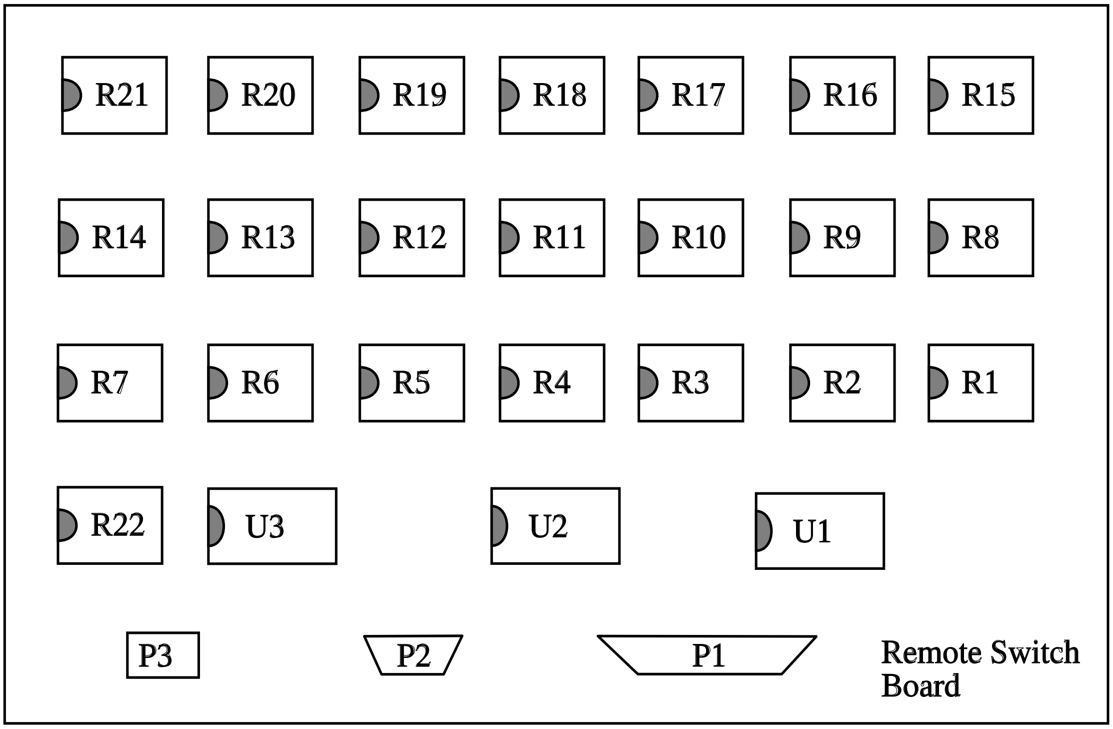
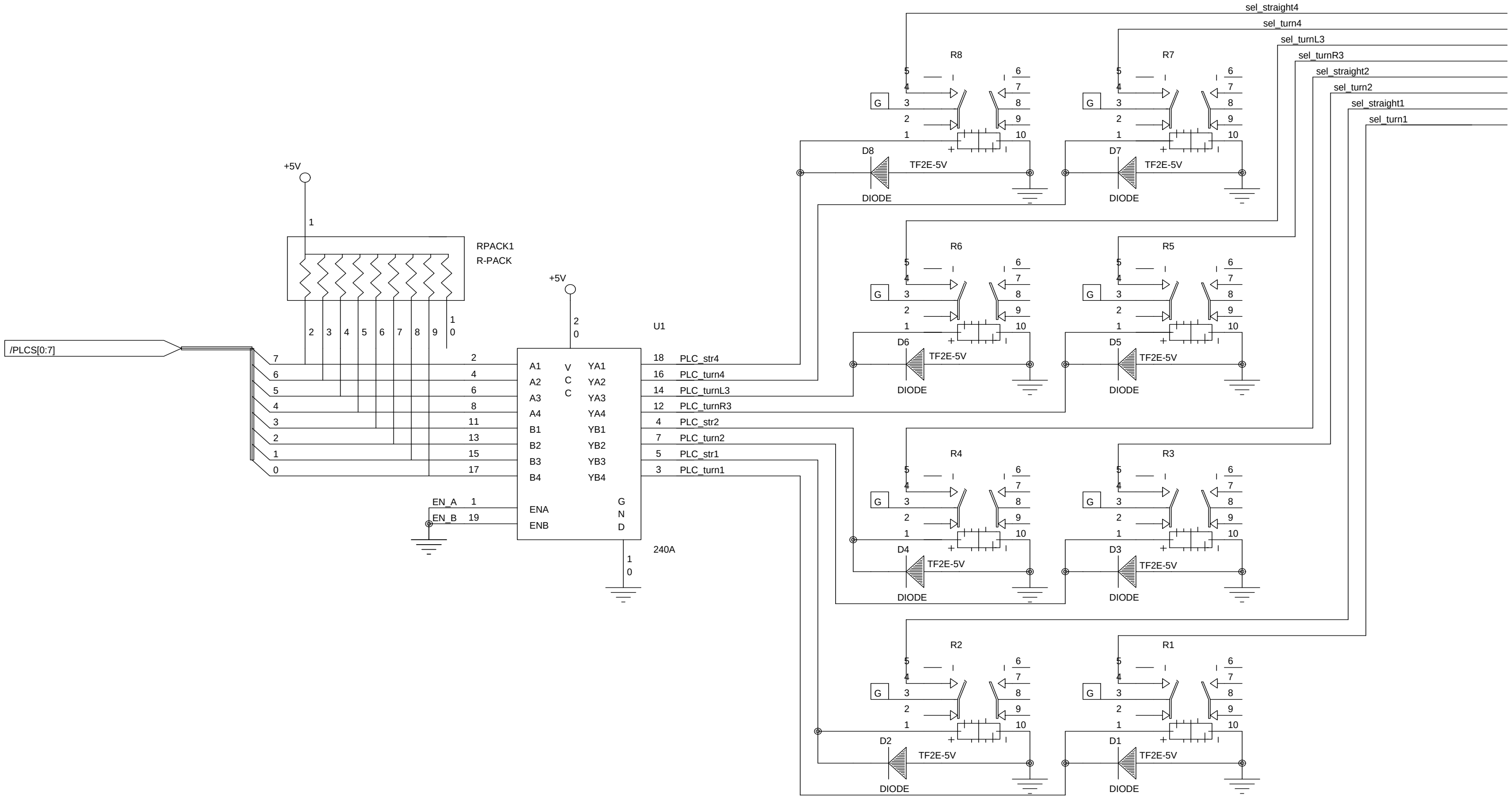


Figure A.2: Layout of Remote Switching Board

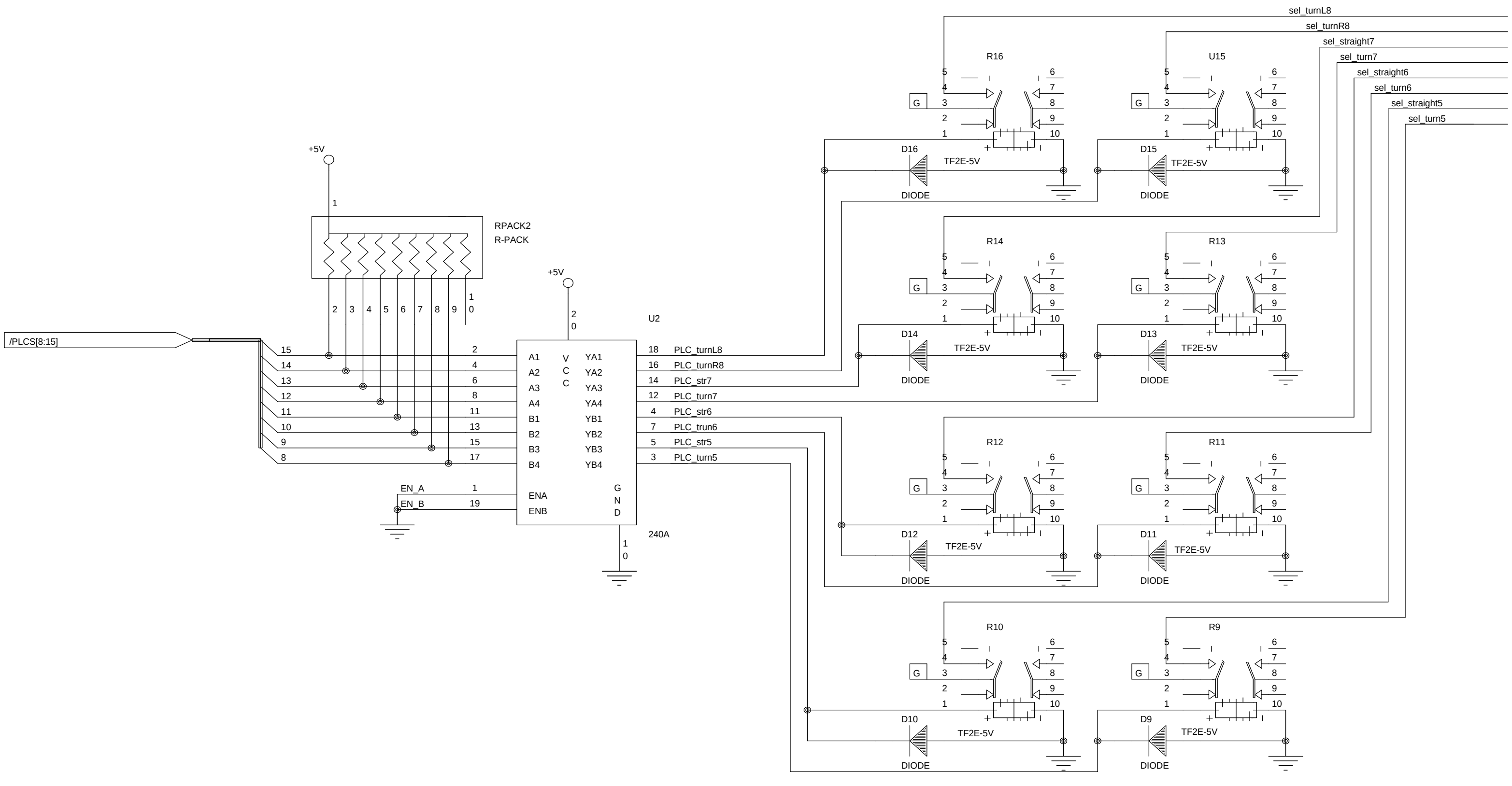
NOTE: Signals leaving relays
are hardwired to testbed
No connectors are used.

Remote Switching Circuit



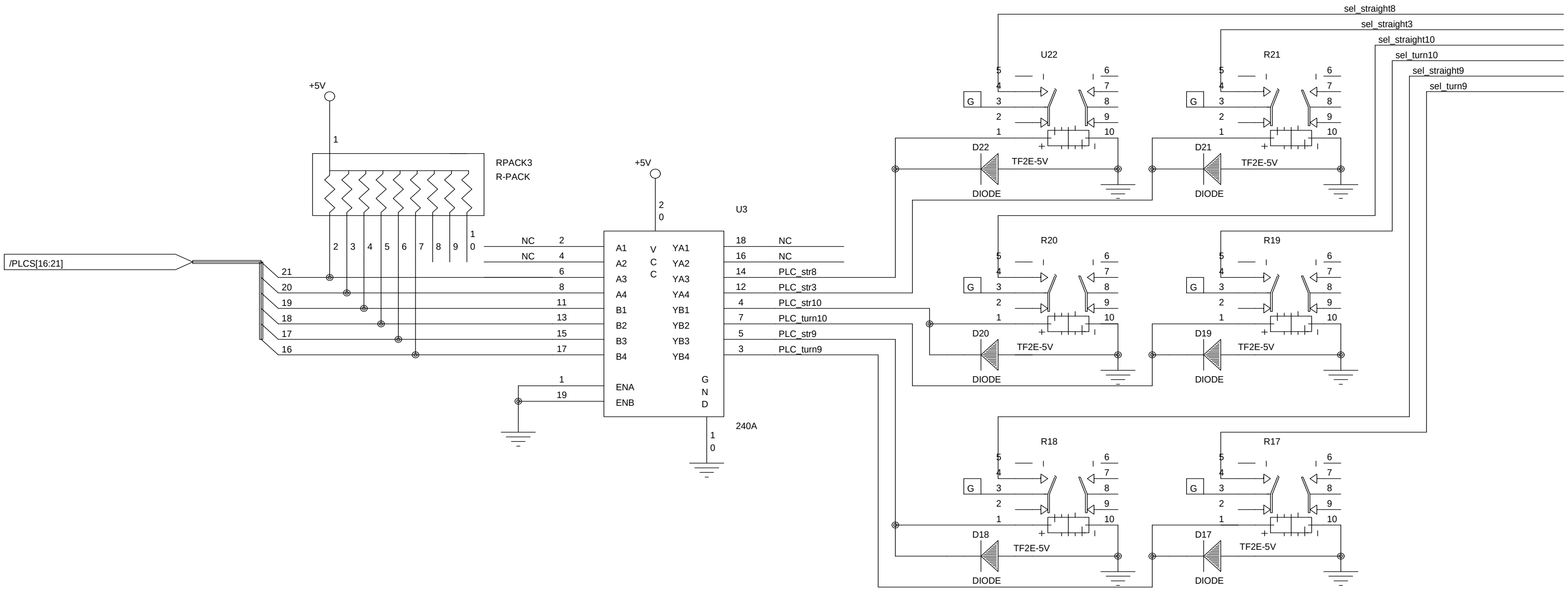
UNIVERSITY OF TORONTO				
Title Remote Switching Circuit				
Size A	Document Number 1			REV 1.0
Date: December 5, 1995		Sheet 1	of 4	

Remote Switching Circuit



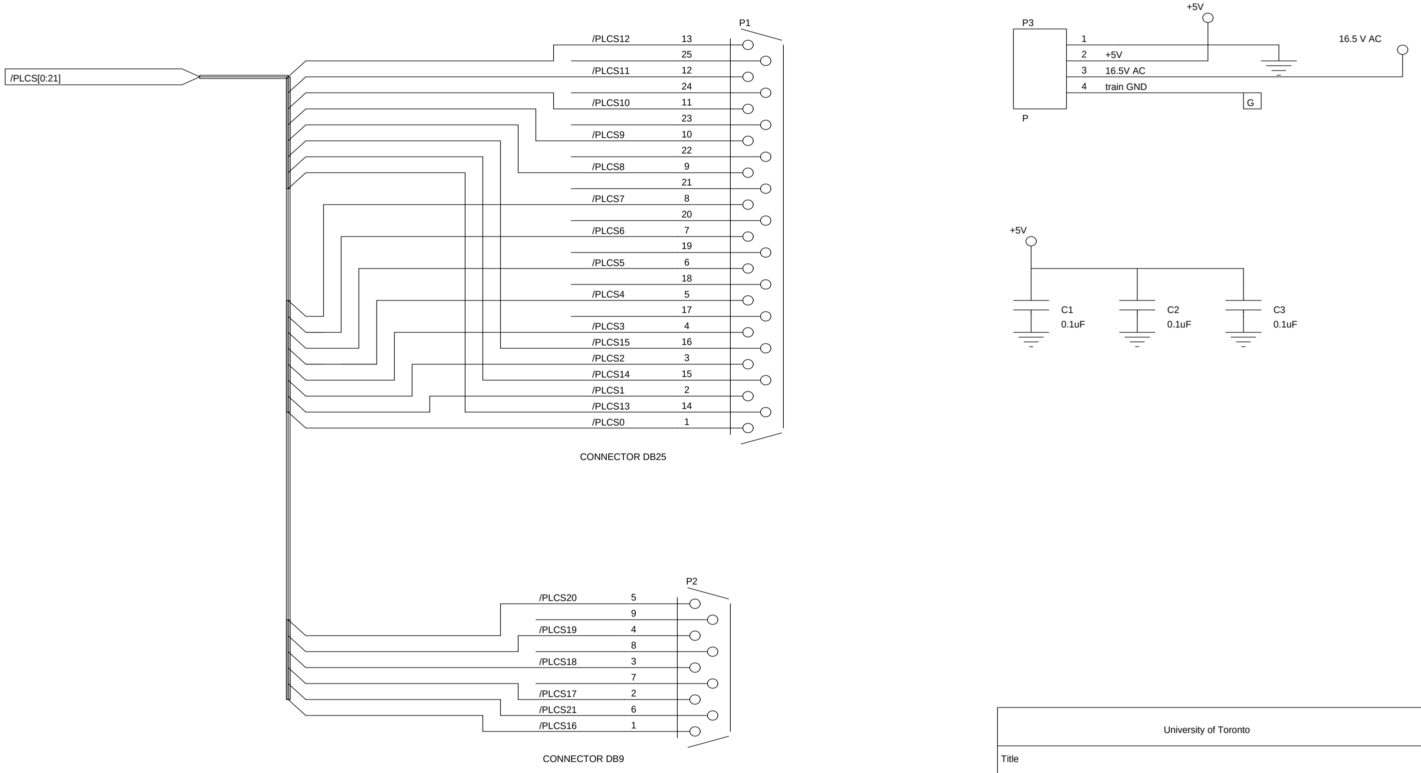
UNIVERSITY OF TORONTO		
Title Remote Switching Circuit		
Size A	Document Number 1	REV 1.0
Date: December 5, 1995	Sheet 2 of 4	

Remote Switching Circuit



UNIVERSITY OF TORONTO				
Title Remote Switching Circuit				
Size A	Document Number 1			REV 1.0
Date:	December 5, 1995	Sheet	3	of 4

Remote Switching Circuit



University of Toronto			
Title Remote Switching Circuit			
Size A	Document Number 1		REV 1.0
Date: December 5, 1995		Sheet 4 of 4	

A.3 Expansion Board Circuit Schematics

This section contains the schematics for the expansion boards for CPU A and CPU B. One set of schematics is given, and any differences between the two boards are pointed out in the schematics, where appropriate. Figure A.3 shows the layout of the expansion board for CPU B. The board for CPU A is almost identical.

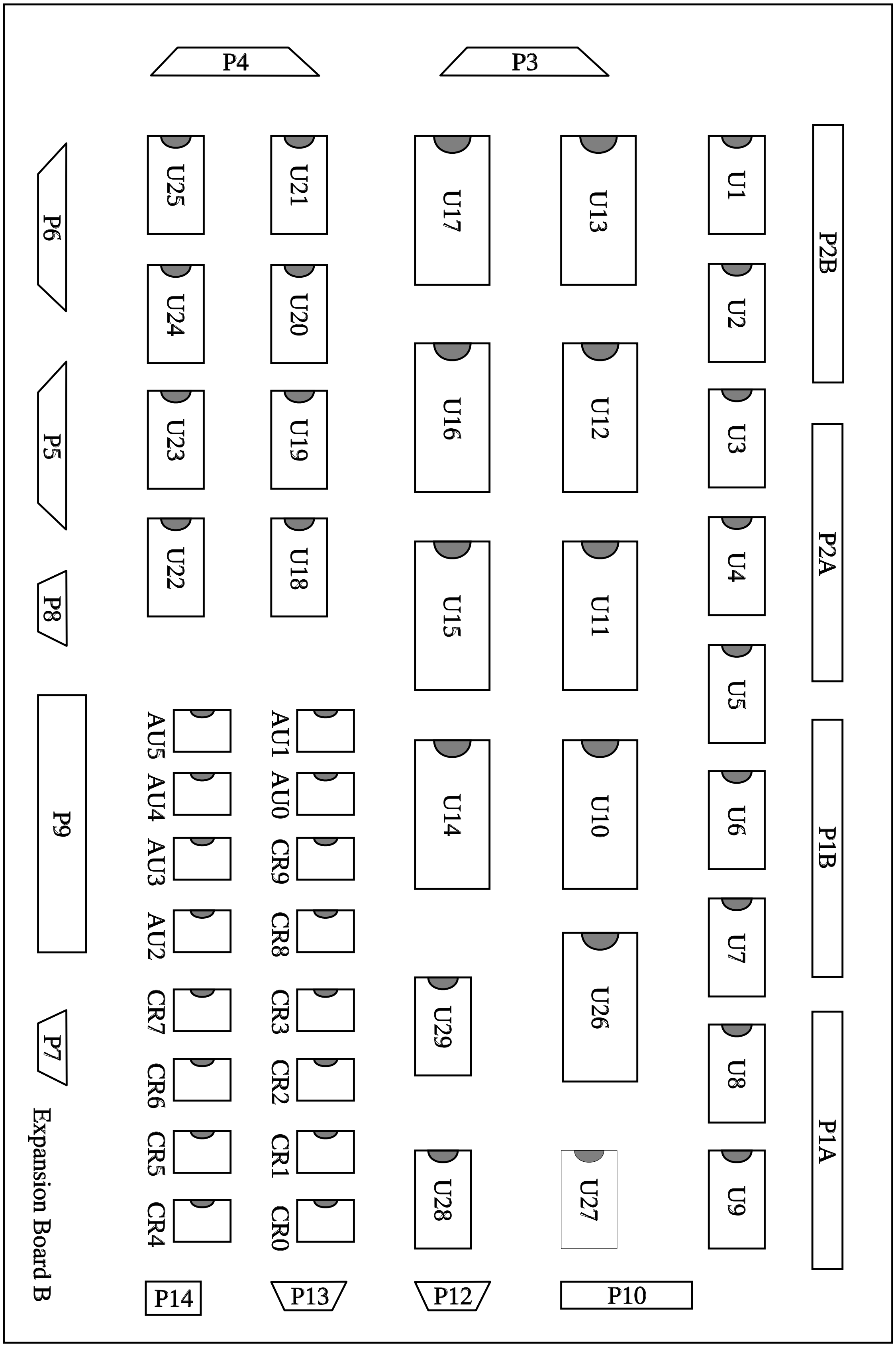
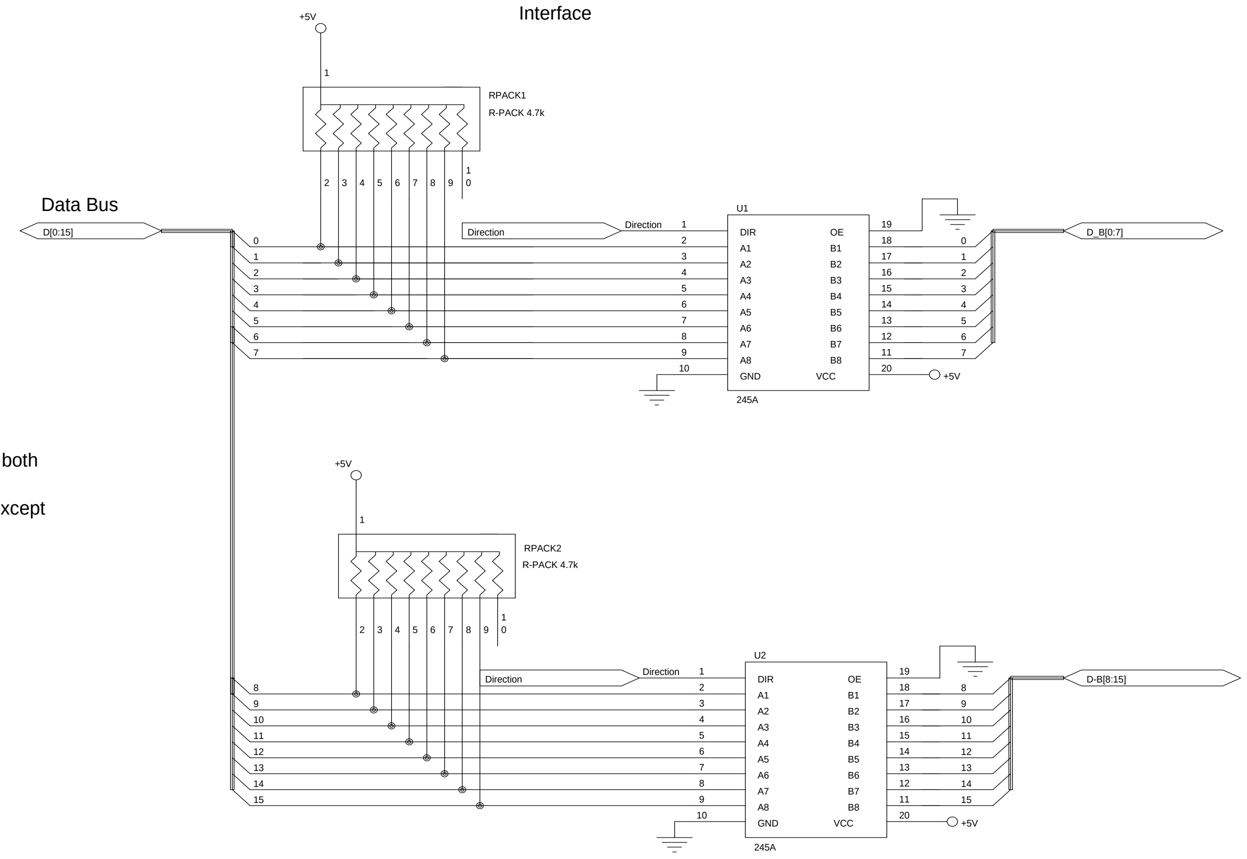


Figure A.3: Layout of Expansion Board B

Expansion Board for CPUA and CPUB



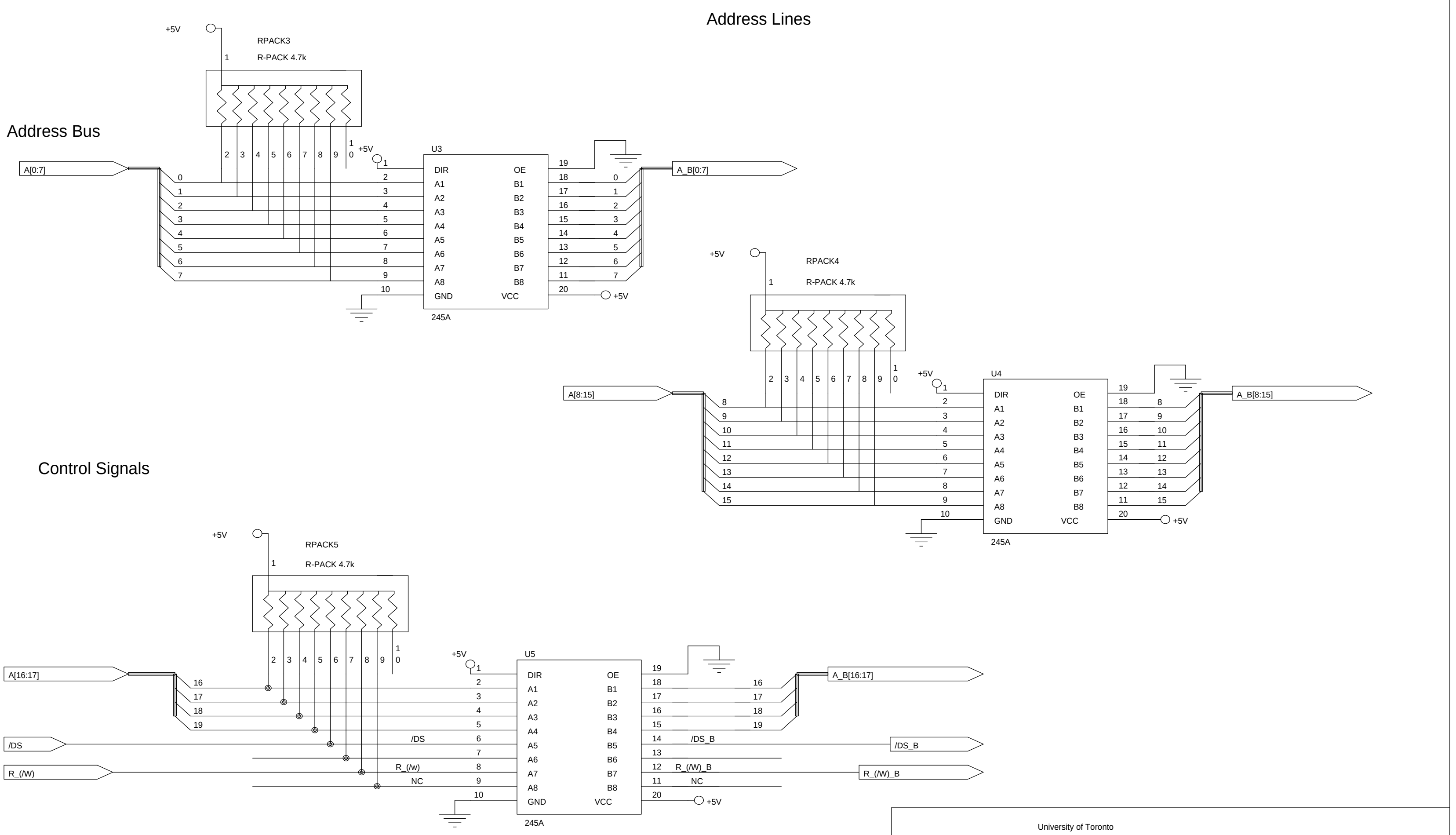
Schematics apply to both

CPU A and CPU B except

where indicated.

University of Toronto

Title Expansion Boards for CPUA and CPUB, DES TB.		
Size A	Document Number 1	REV 1.0
Date: December 5, 1995	Sheet 1 of 20	



University of Toronto

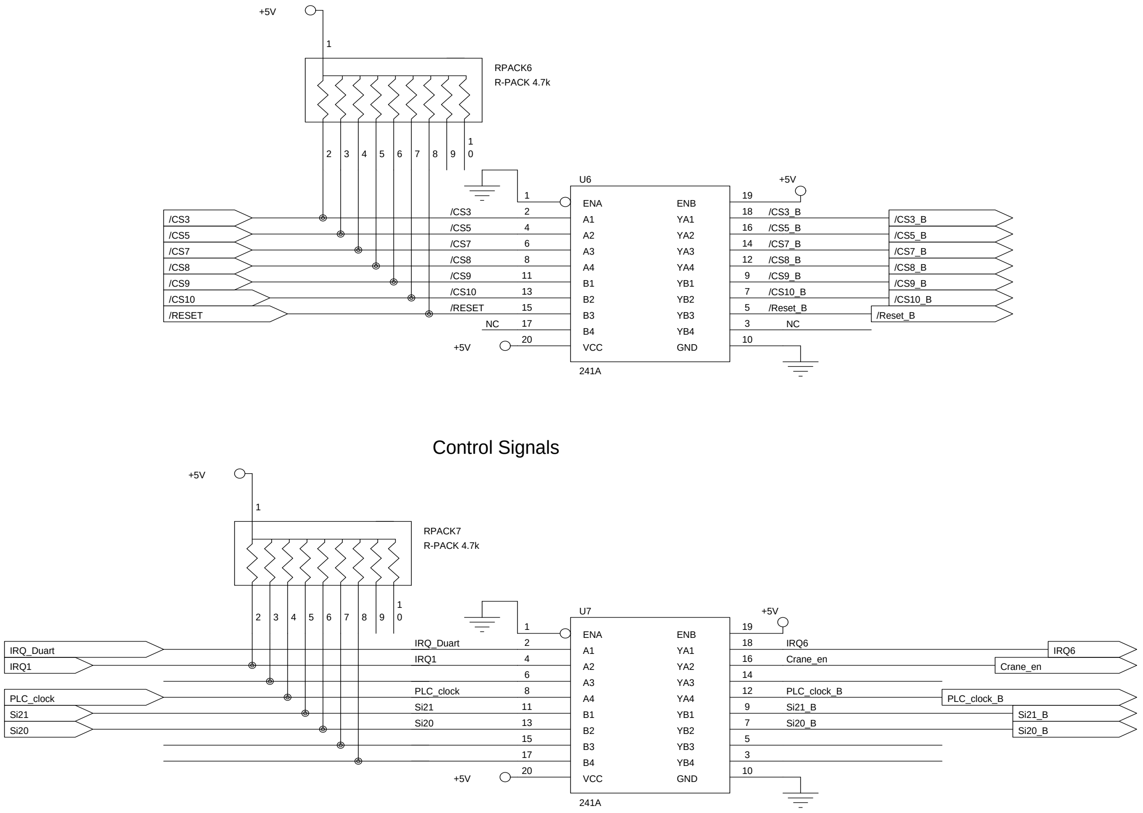
Title

Expansion Boards for CPUA and CPUB, DES TB.

Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 Sheet 2 of 20

Control Signals



Control Signals

University of Toronto

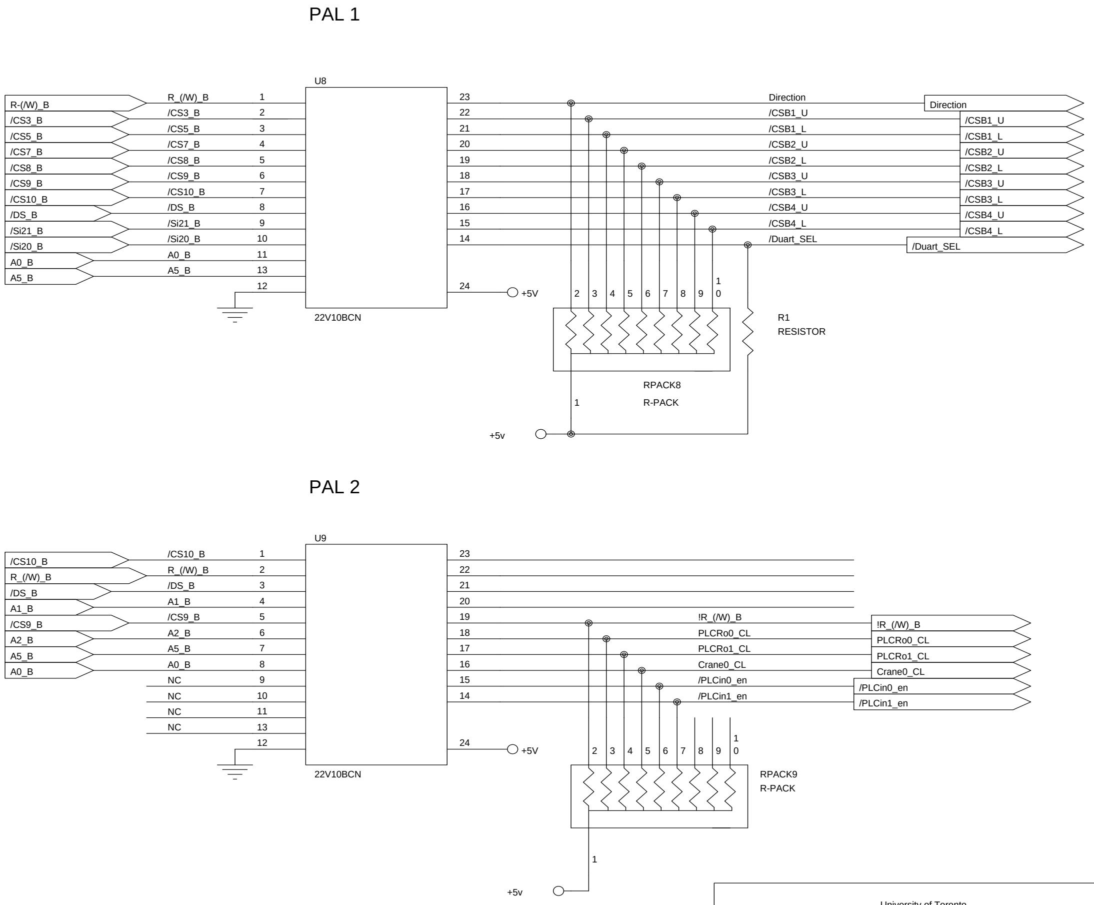
Title

Expansion Boards for CPUA and CPUB, DES TB.

Size	Document Number	REV
A	1	1.0

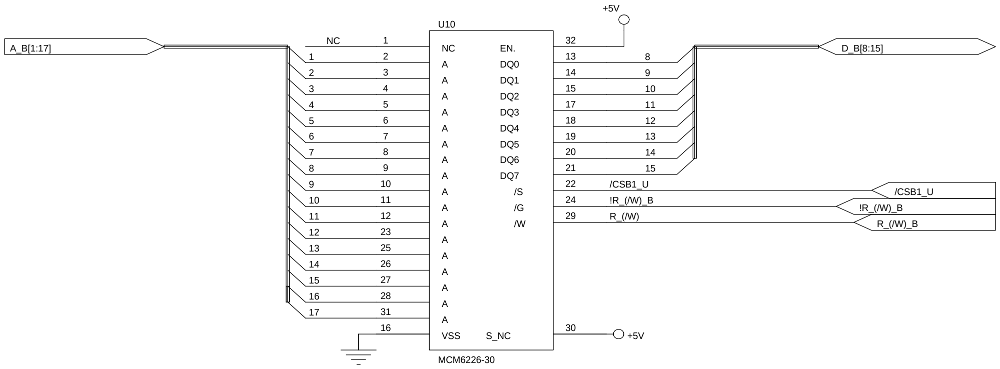
Date: December 5, 1995 Sheet 3 of 20

PAL 1

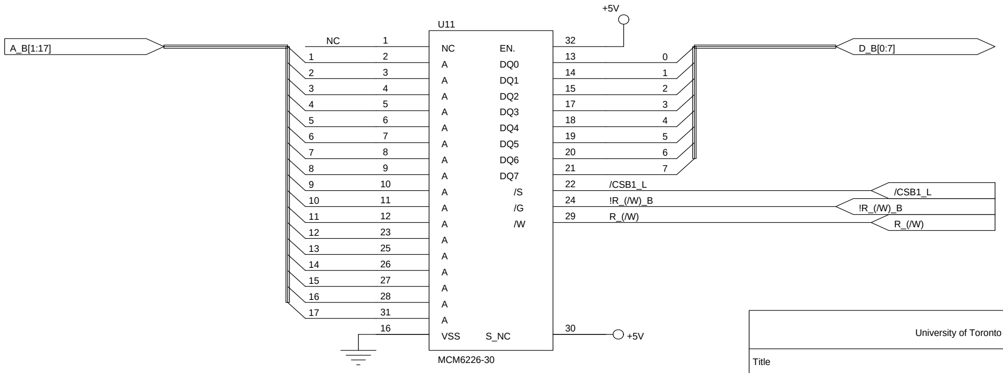


University of Toronto		
Title Expansion Boards for CPUA and CPUB, DES TB.		
Size A	Document Number 1	REV 1.0
Date: December 5, 1995	Sheet 4 of 20	

RAM Bank 1 Upper Byte



RAM Bank 1 Lower Byte



University of Toronto

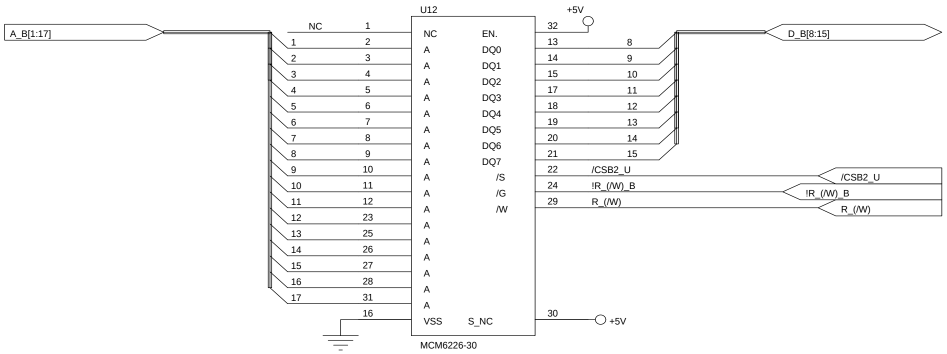
Title

Expansion Boards for CPUA and CPUB, DES TB.

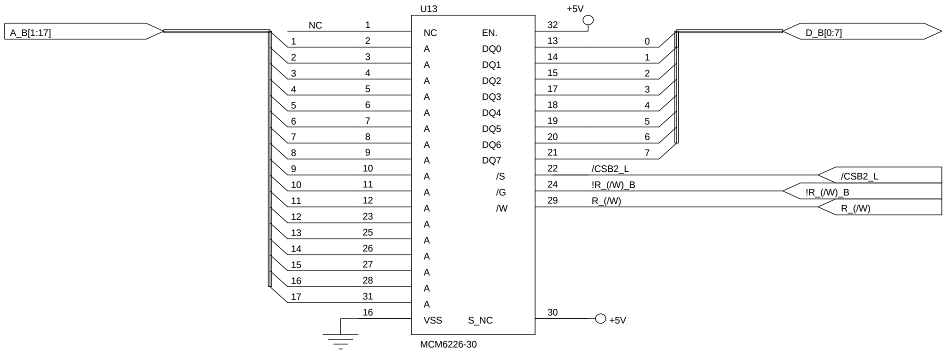
Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 Sheet 5 of 20

RAM Bank 2 Upper Byte



RAM Bank 2 Lower Byte



UNIVERSITY OF TORONTO

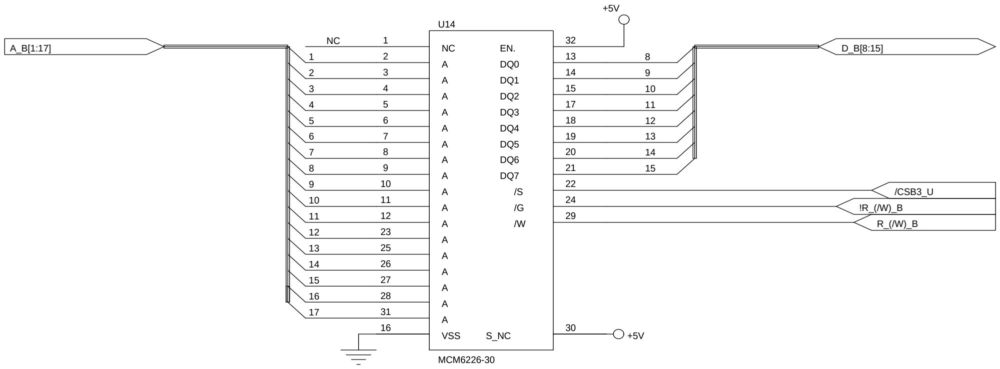
Title

Expansion Borads for CPUA and CPUB, DES TB.

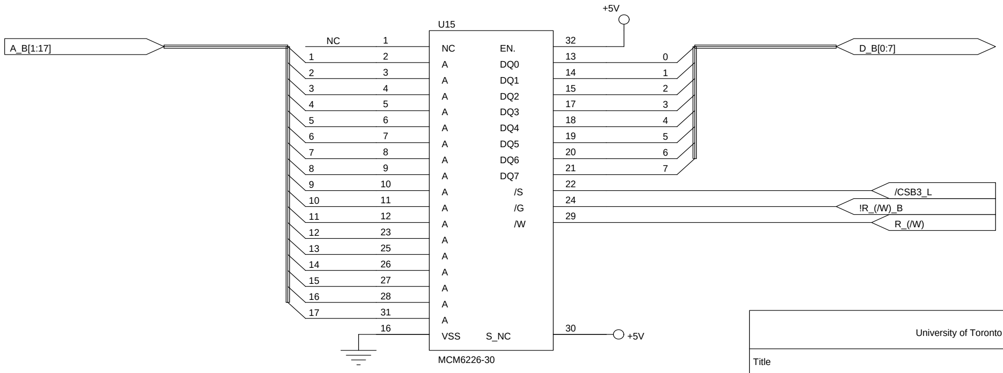
Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 Sheet 6 of 20

RAM Bank 3 Upper Byte



RAM Bank 3 Lower Byte



University of Toronto

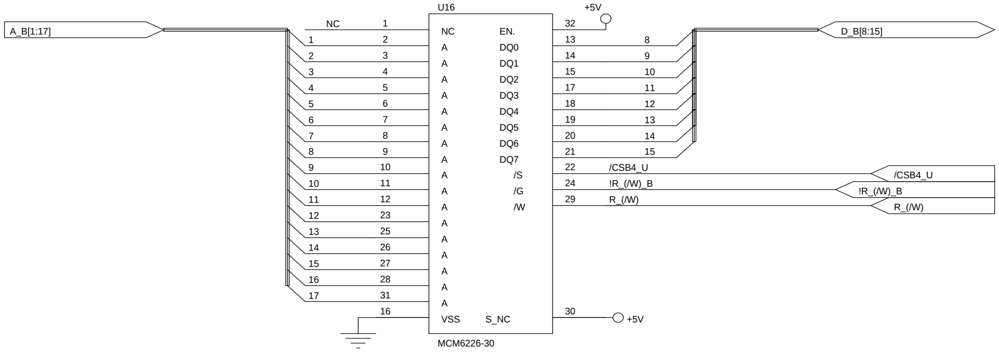
Title

Expansion Boards for CPUA and CPUB, DES TB.

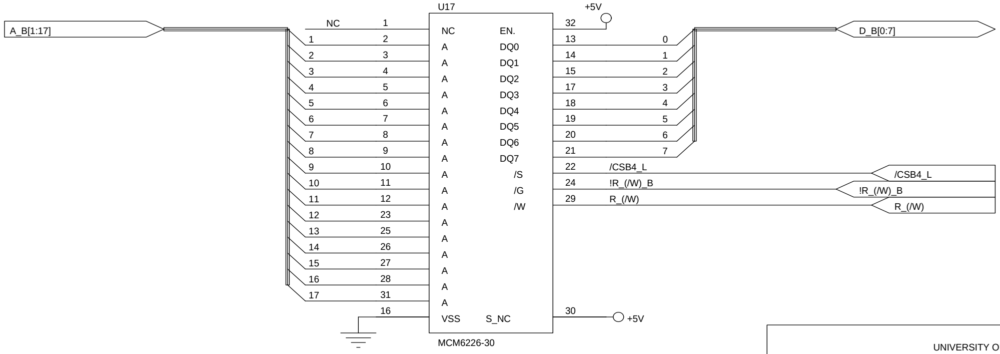
Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 | Sheet 7 of 20

RAM Bank 4 Upper Byte

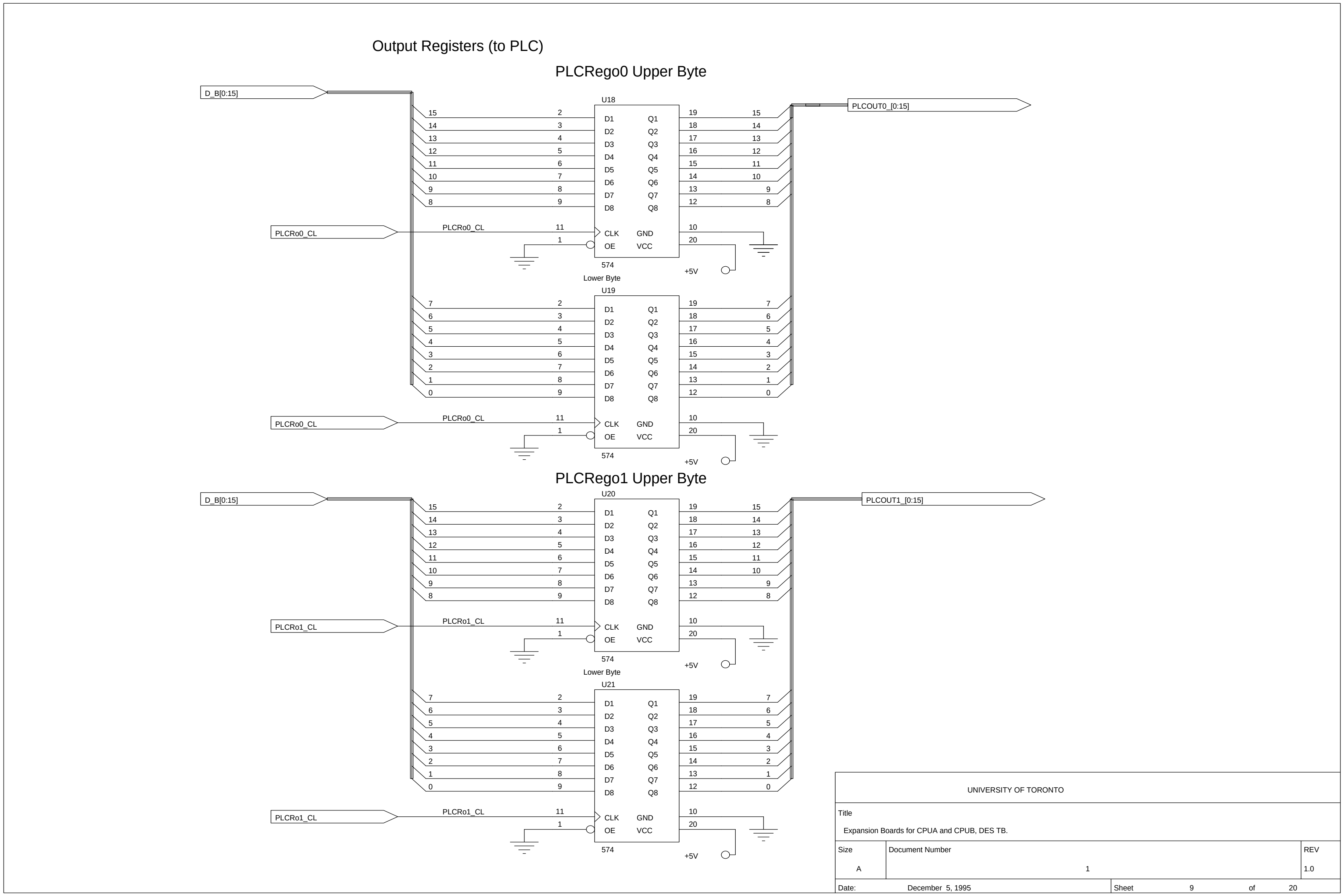


RAM Bank 4 Lower Byte



UNIVERSITY OF TORONTO

Title Expansion Borads for CPUA and CPUB, DES TB.					
Size A	Document Number 1				REV 1.0
Date:	December 5, 1995	Sheet	8	of	20



UNIVERSITY OF TORONTO

Title

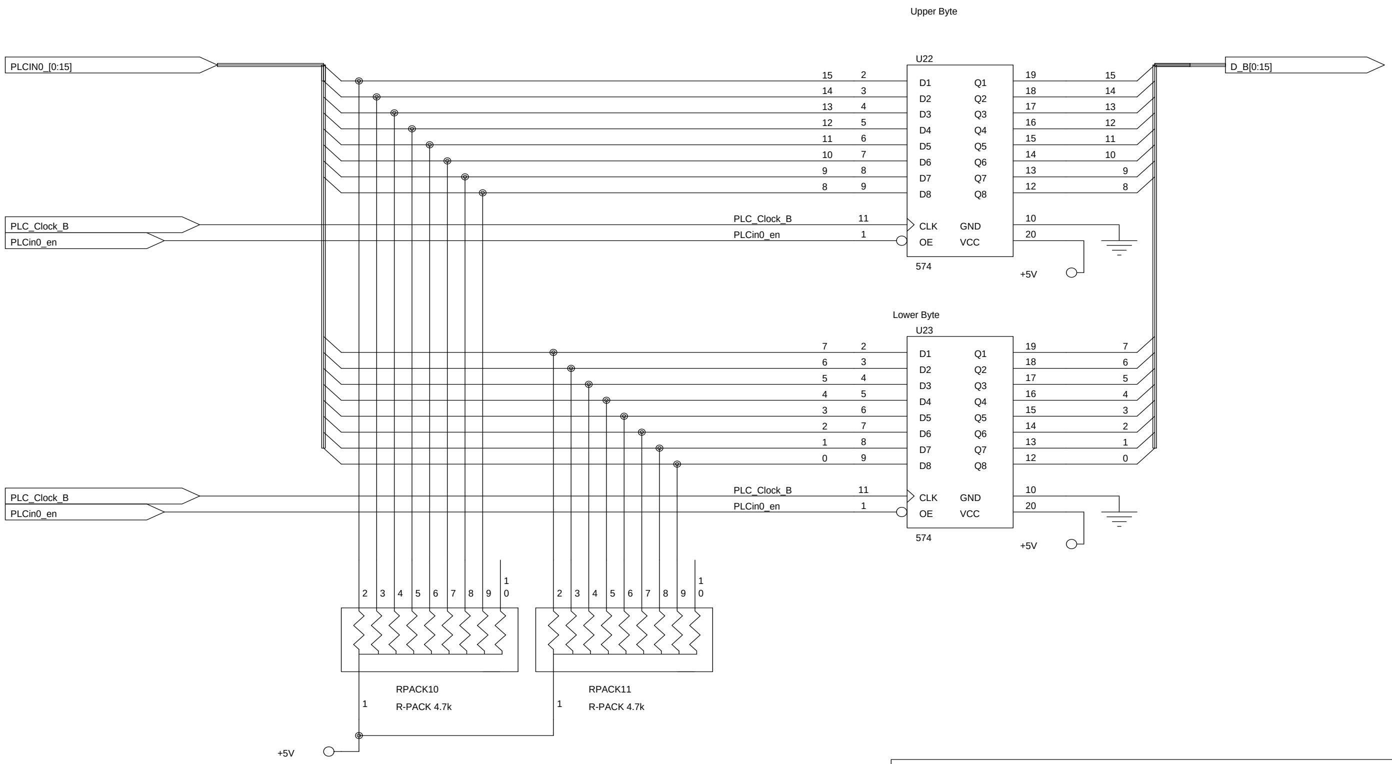
Expansion Boards for CPUA and CPUB, DES TB.

Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 Sheet 9 of 20

Input Registers From PLC

PLCRegi0



UNIVERSITY OF TORONTO

Title

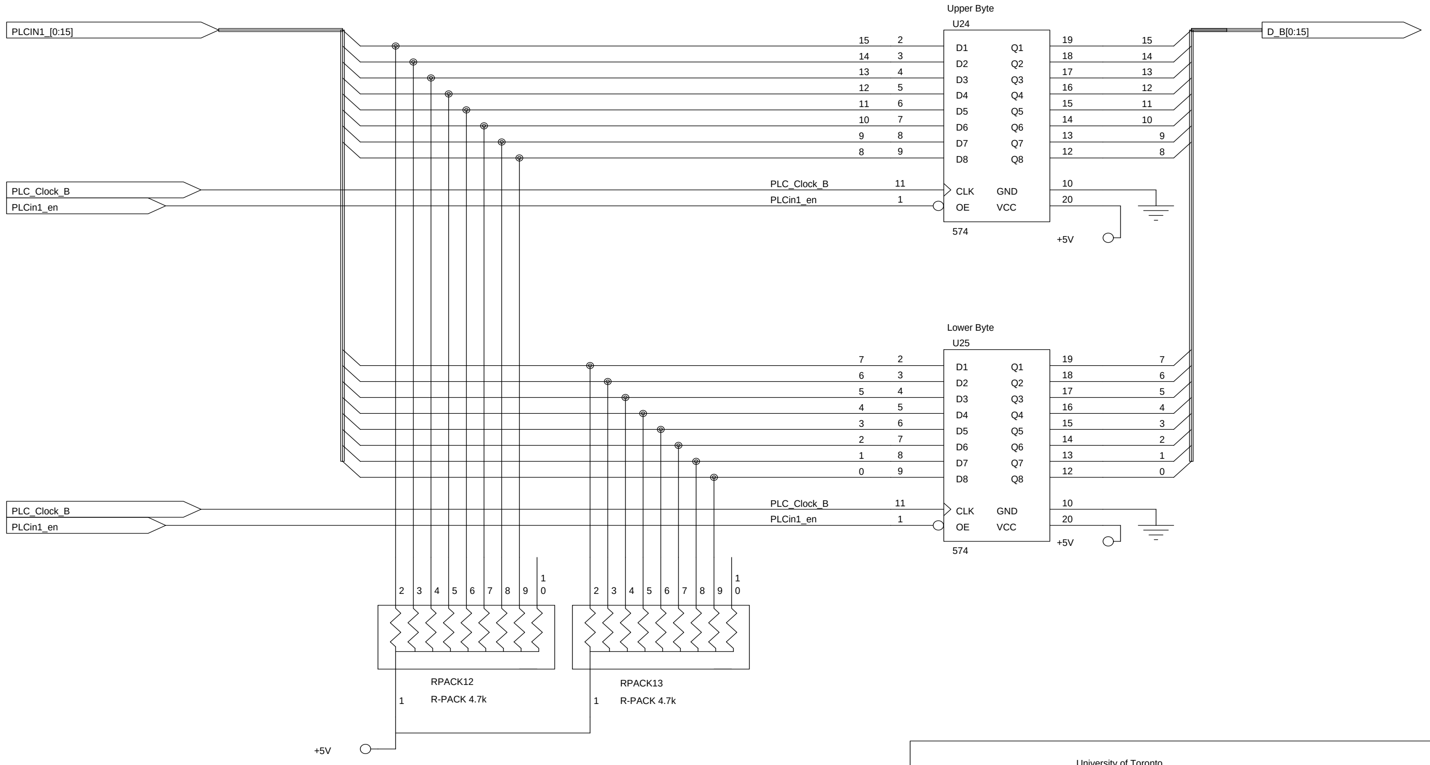
Expansion Borads for CPUA and CPUB, DES TB.

Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 Sheet 10 of 20

Input Register from PLC

PLCRegi1



University of Toronto

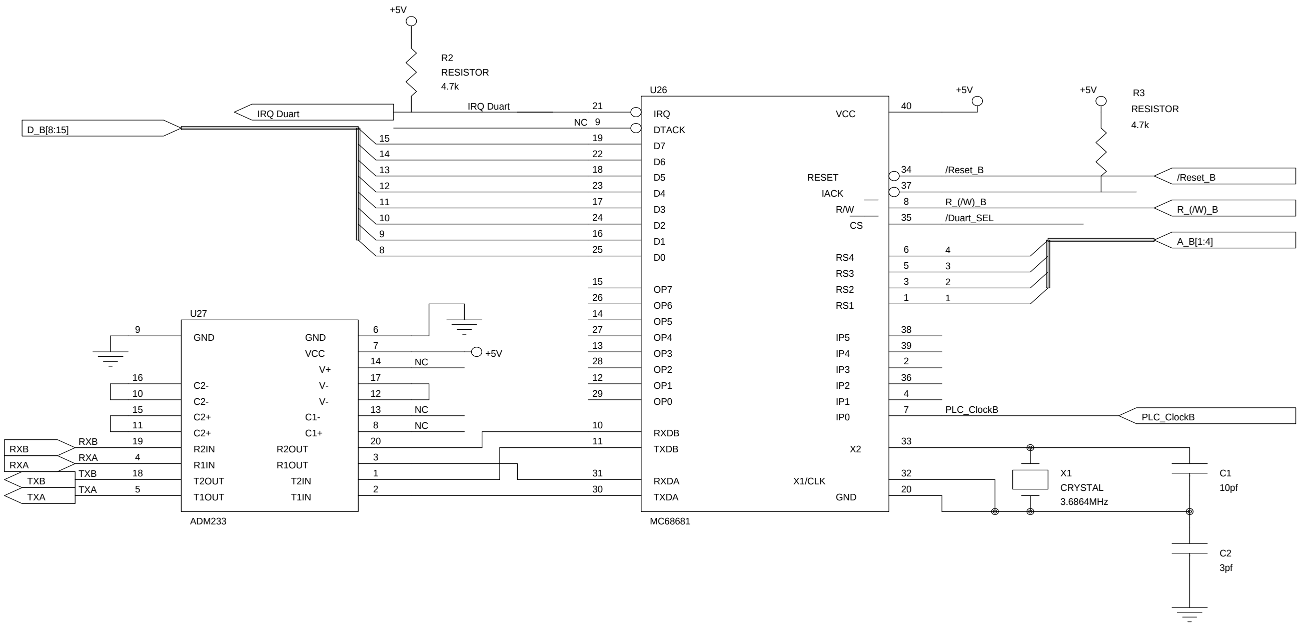
Title

Expansion Boards for CPUA and CPUB, DES TB.

Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 Sheet 11 of 20

DUART



UNIVERSITY OF TORONTO

Title

Expansion Boards for CPUA and CPUB, DES TB.

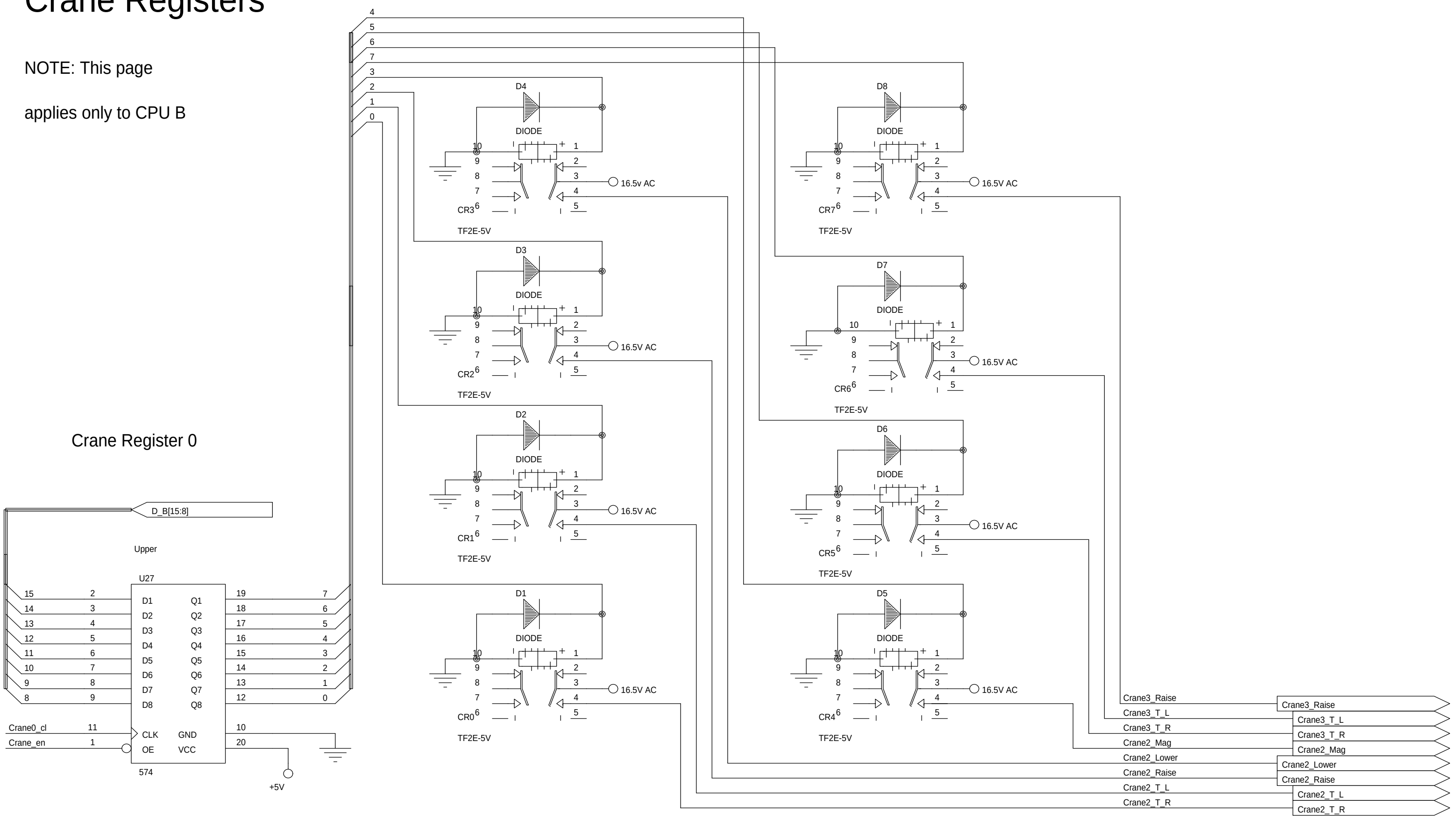
Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 Sheet 12 of 20

Crane Registers

NOTE: This page

applies only to CPU B



UNIVERSITY OF TORONTO

Title

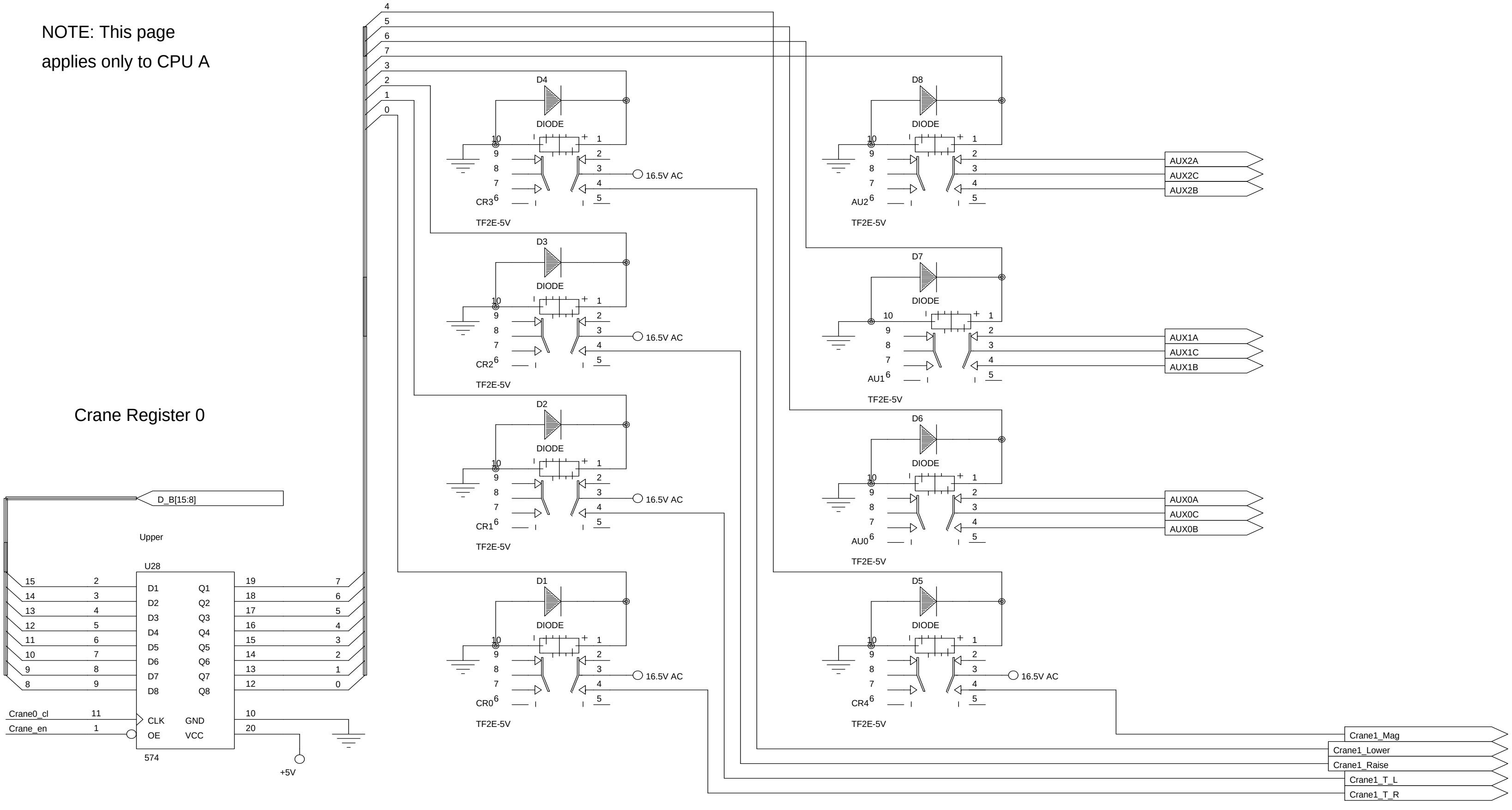
Expansion Boards for CPUA and CPUB, DES TB.

Size	Document Number	REV
A	1	1.0

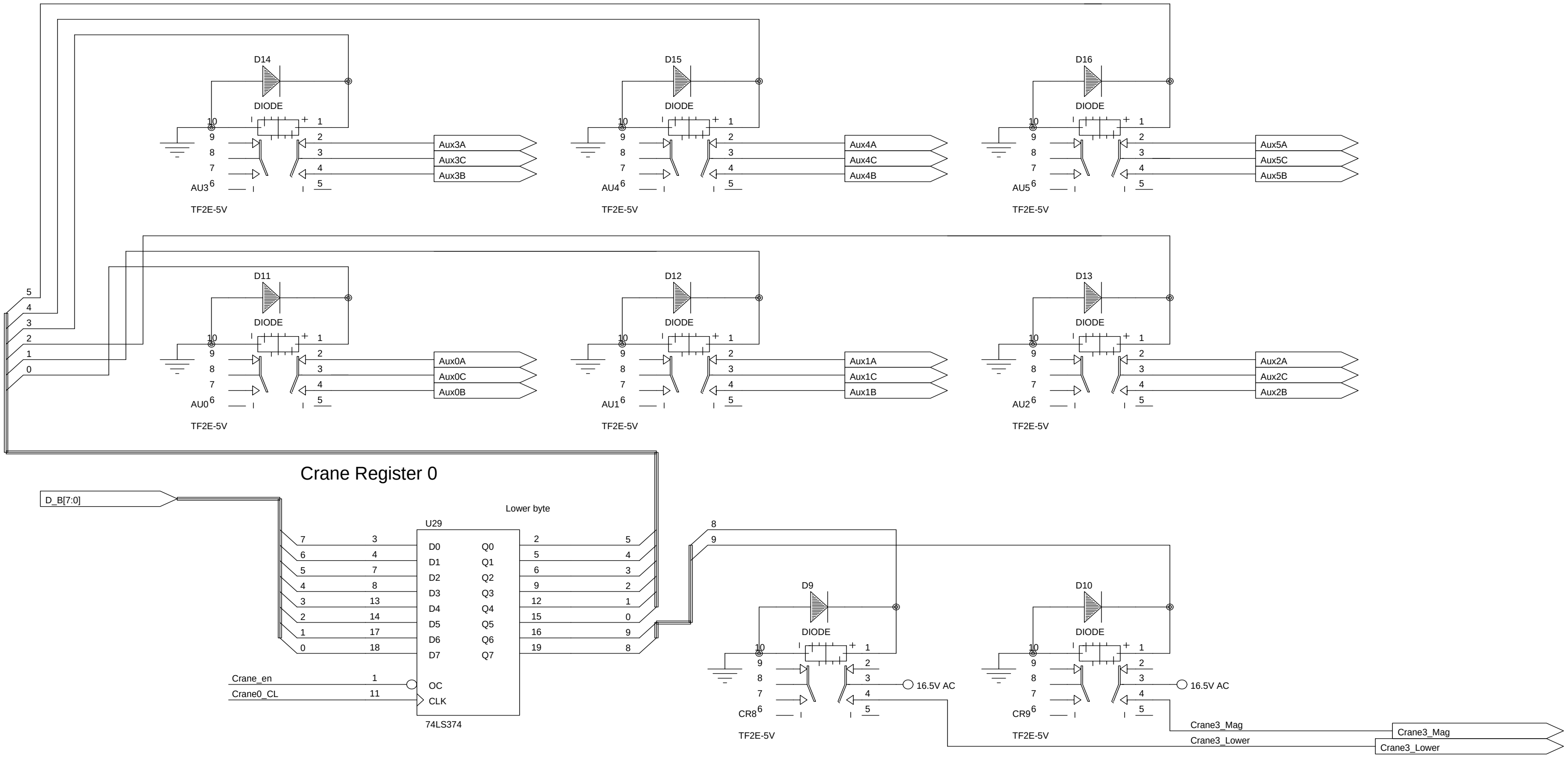
Date: December 8, 1995 Sheet 13 of 20

Crane Registers

NOTE: This page applies only to CPU A



UNIVERSITY OF TORONTO		
Title Expansion Boards for CPUA and CPUB, DES TB.		
Size A	Document Number 1	REV 1.0
Date: December 8, 1995	Sheet 14 of 20	



Crane Registers

NOTE: This applies

only to CPU B

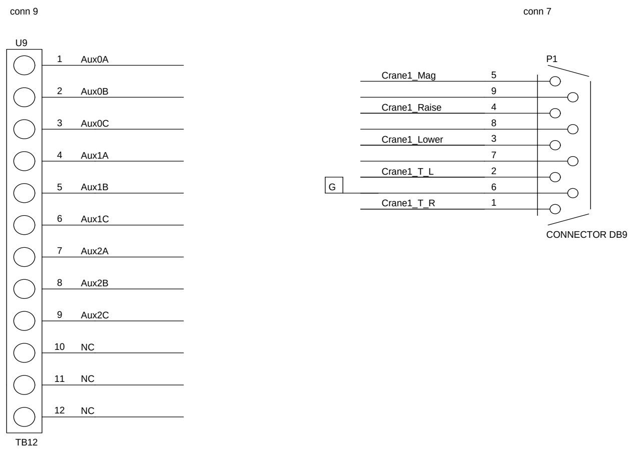
UNIVERSITY OF TORONTO			
Title Expansion Boards for CPUA and CPUB, DES TB.			
Size A	Document Number 1		REV 1.0
Date: December 5, 1995	Sheet 15	of 20	

For CPU A only

NOTE:

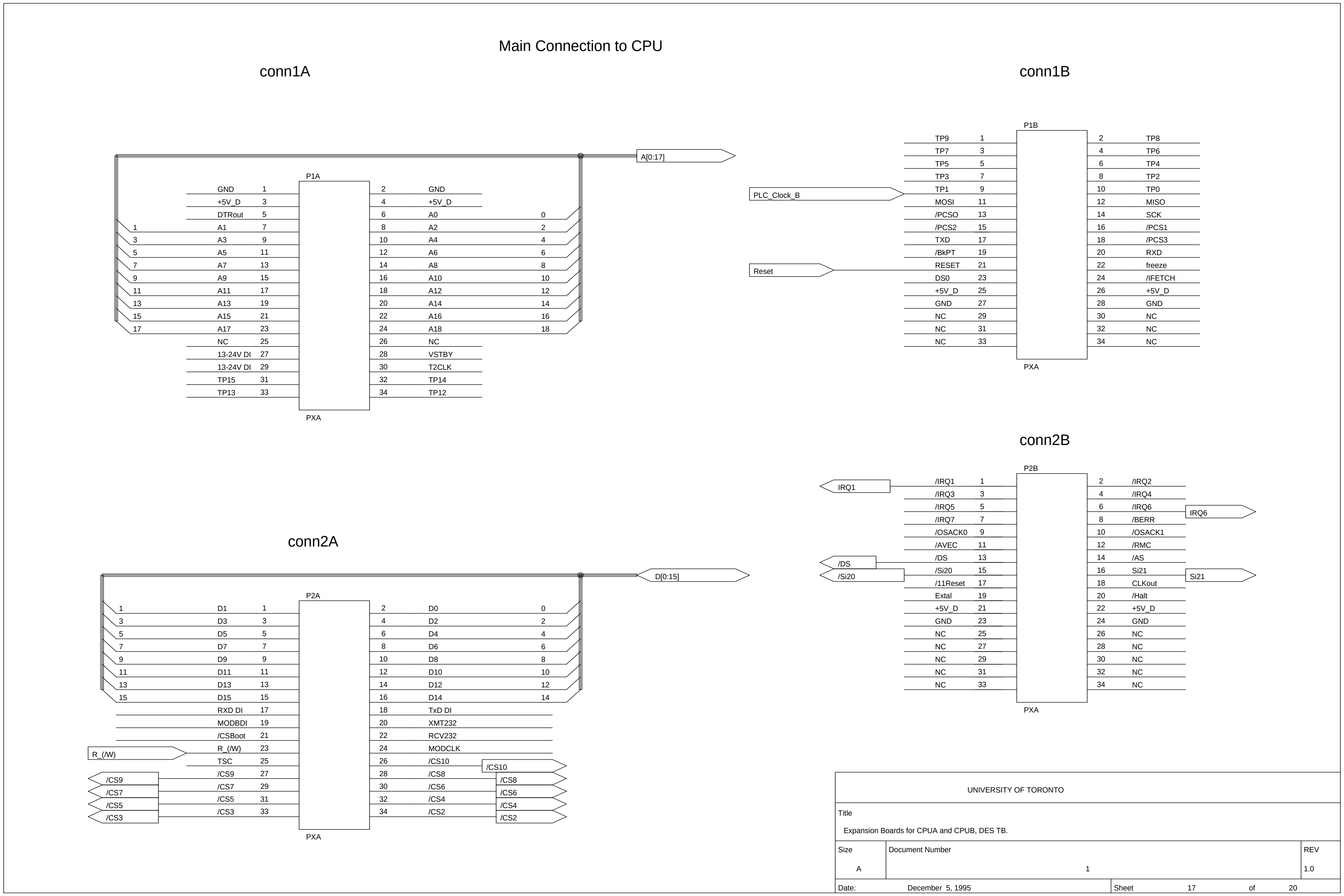
conn 8 for cpu A does not exist.

For CPU A, Crane Register 0 lower does not exist.



University of Toronto			
Title Expansion Borads for CPUA and CPUB, DES TB.			
Size A	Document Number 1		REV 1.0
Date: December 8, 1995		Sheet 16 of 20	

Main Connection to CPU



UNIVERSITY OF TORONTO

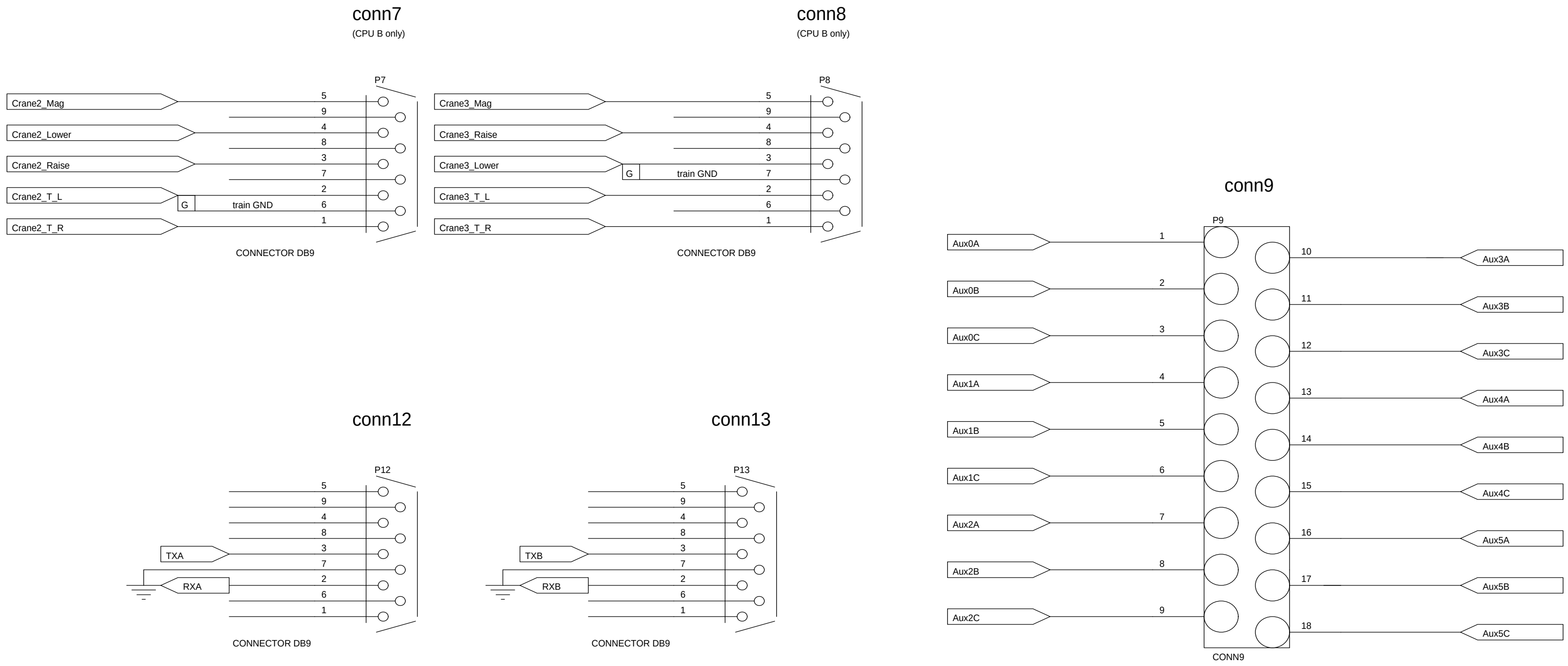
Title

Expansion Boards for CPUA and CPUB, DES TB.

Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 | Sheet 17 of 20

CONNECTORS



UNIVERSITY OF TORONTO

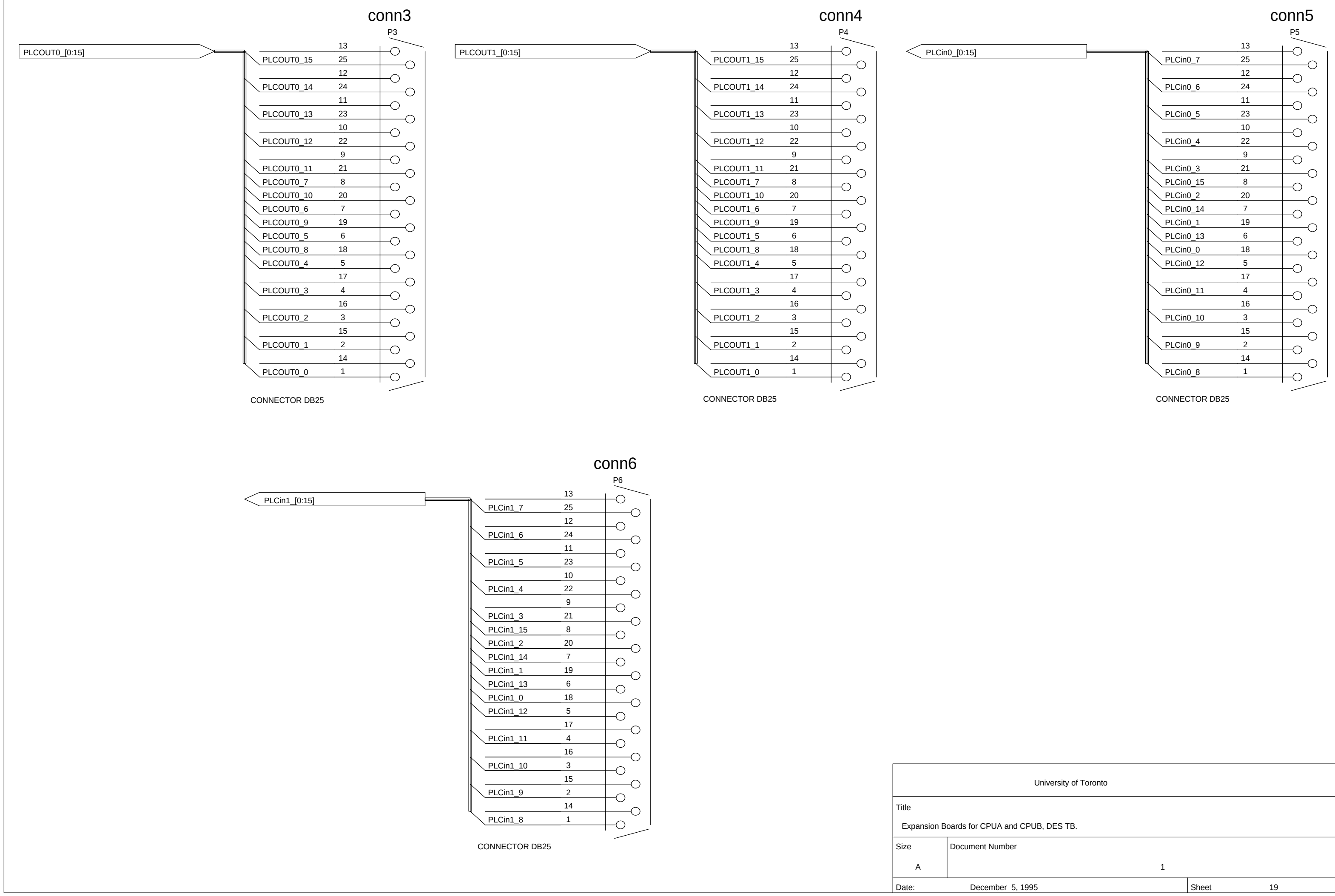
Title

Expansion Boards for CPUA and CPUB, DES TB.

Size	Document Number	REV
A	1	1.0

Date: December 8, 1995 | Sheet 18 of 20

Connectors



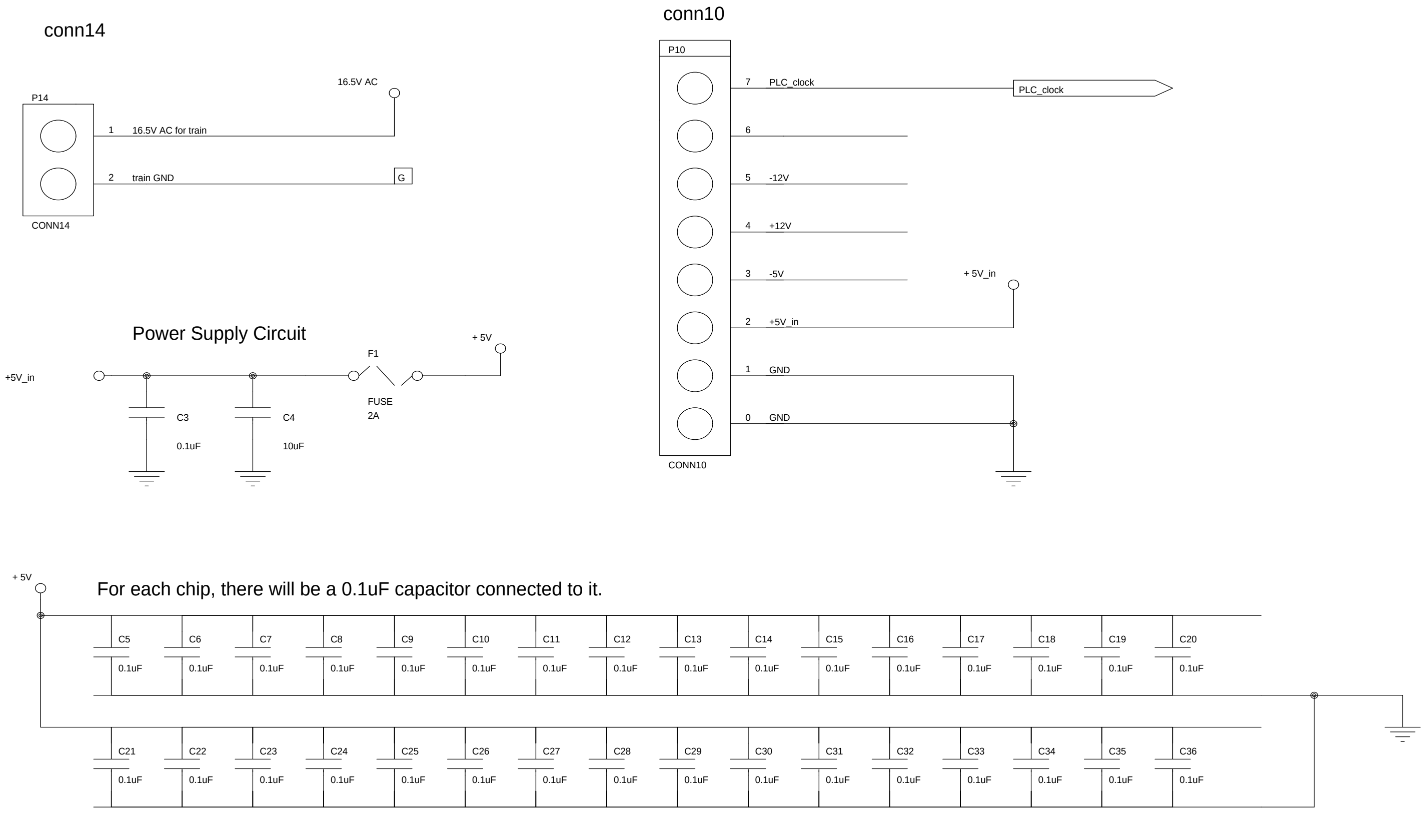
University of Toronto

Title

Expansion Boards for CPUA and CPUB, DES TB.

Size	Document Number	REV
A	1	1.0

Date: December 5, 1995 | Sheet 19 of 20



University of Toronto

Title Expansion Borads for CPUA and CPUB, DES TB.				
Size A	Document Number 1			REV 1.0
Date:	December 5, 1995	Sheet 20	of 20	

Appendix B

Component Listing

This chapter lists the essential materials needed to build the testbed. For the expansion boards, quantities are given only for board B. The same components would be required for board A. The electronic components are listed in Table B.1, the model train components in Table B.2, and the PLC components in Table B.3.

Quantity	*Description*	*Quantity*	*Description*
12	74HC245A	9	74HC574
21	9-pin 4.7kΩ resistor pack	38	1N4001 diodes
5	DB25 connector, female	1	MC68681
1	18 pin terminal block	1	3.6864MHz crystal
14	DB9 connector, female	9	ADM233
1	8 pin terminal block	1	10pF capacitor
10	2 pin terminal block	1	3pF capacitor
1	2A fuse	3	4.7kΩ resistor
45	0.1μF capacitor	4	34 pin male conn. (3431-5002 by 3M)
1	10μF capacitor	8	MCM6226-30 RAM
2	74HC241A	3	74HC240A
38	TF2E-5V Aromat relays	2	22V10BCN (TTL) PAL
1	74HC374	4	34 pin female conn. (3431-5202 by 3M)
14	DB9 female cable connector	5	DB25 female cable connector
1	486 based PC	1	3 way switch box
1	5V-6A power supply	2	MC68332 evaluation kit (M68332EVK)

Table B.1: Electronic Components

Quantity	*Description*	*Quantity*	*Description*
1	Str track 7-3/32” power feed (2290)	4	L and R remote switches (2261)
12	Str track 7-3/32” (2200)	2	3-way remote switch (2270)
12	Str track 3-9/16” (2201)	2	Digital locomotive (3665)
12	Str track 1-3/16” (2203)	2	Gondola car (4465)
10	Str track 8-9/16” (2209)	3	Electric crane (7051)
3	Str track 1-5/8” (2293)	1	Digital transformer (6001)
2	Str track 6-5/8” (2206)	2	Analog transformer (6667A)
3	Str track 7/8” (2204)	1	Central unit (6021)
12	Curved track 30 degree (2221)	1	Computer interface (6050)
12	Curved track 15 degree (2223)	2	3/4 inch - 3.75 feet X 3.75 feet plywood
12	Curved track 22 degree 30’ (2232)	2	Set of metal banquet legs

Table B.2: Train Components

Quantity	*Description*	*Quantity*	*Description*
1	Processor 1785 PLC-5/30 (1785L30B)	1	Communication Cable (1784-CP10)
1	16 slot chassis	8	16 pt DC input card (1771-IGD)
1	Power supply (1771-P7)	6	16 pt DC output card (1771-OGD)
1	Cable for power supply (1771-CP1)	1	Software for a PC (6203-PLC5)

Table B.3: PLC Components

Appendix C

Cables and Connectors

To interconnect the various components of the testbed, 26 cables were used. The cables and their attached connectors are detailed in the sections below. In the case of cables that go to the PLC (excluding the cable connecting the PLC to its programming station), they contain a connector only on the end opposite of the PLC. Figure C.1 shows all 26 discussed cables, showing their various connections.

NOTE: The signal "NC" stands for no connection.

C.1 Remote Sensor Cables

Sensor boards 0 to 5 each are attached by a cable (cables 1-6) to the PLC. Sensor board 6 is connected by cable 7 to the RS-232, who is connected to the PLC by cable 8. The pinouts of the sensor board (including the RS-232 board) connectors for each cable is given in Table C.1.

C.2 Track Switch Cables

The remote switching board is connected to the PLC by cables 9 and 10. The pinouts of the cables are given in Table C.2. Any pins not listed are unconnected.

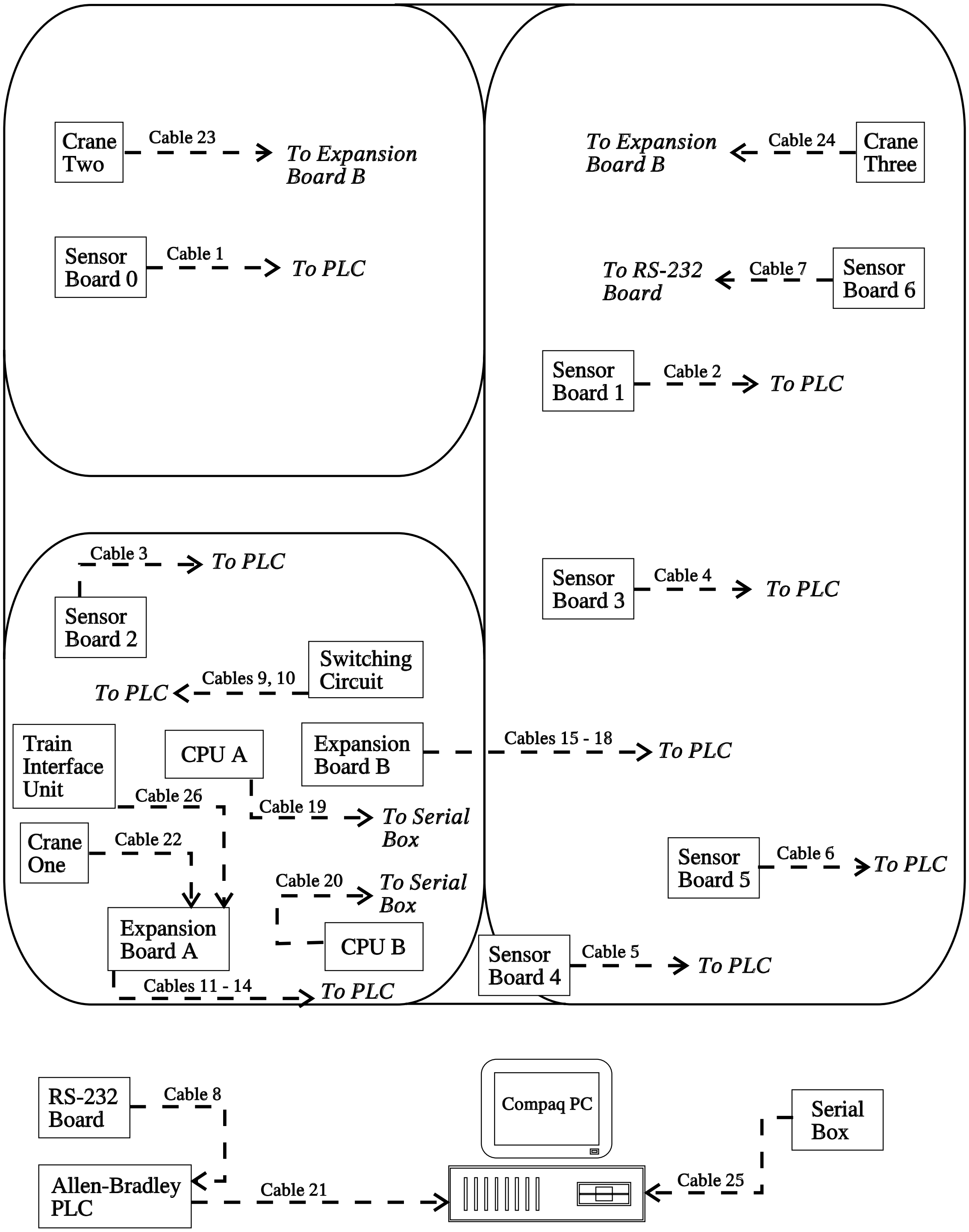


Figure C.1: Cable Connection for Testbed

Cable 1		*Cable 2*		*Cable 3*		*Cable 4*		*Cable 5*	
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
1	RS4A_B	1	RS8A_B	1	RS27A_B	1	RS12A_B	1	RS17A_B
2	RS4B_B	2	RS8B_B	2	RS27B_B	2	RS12B_B	2	RS17B_B
3	RS2A_B	3	RS5A_B	3	RS9A_B	3	RS10A_B	3	RS14A_B
4	RS2B_B	4	RS5B_B	4	RS9B_B	4	RS10B_B	4	RS14B_B
5	NC	5	RS3A_B	5	NC	5	NC	5	RS16A_B
6	RS0A_B	6	RS3B_B	6	RS11A_B	6	RS7A_B	6	RS16B_B
7	RS0B_B	7	RS18A_B	7	RS11B_B	7	RS7B_B	7	RS15A_B
8	RS1A_B	8	RS18B_B	8	RS13A_B	8	RS6A_B	8	RS15B_B
9	RS1B_B	9	NC	9	RS13B_B	9	RS6B_B	9	NC
Cable 6		*Cable 7 (SB6)*		*Cable 7 (RS-232B)*		*Cable 8*			
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*		
1	RS25A_B	1	RS19B_B	1	RS19B_B	1	RS19B		
2	RS25B_B	2	RS19A_B	2	RS19A_B	2	RS19A		
3	RS26A_B	3	RS20B_B	3	RS20B_B	3	RS20B		
4	RS26B_B	4	RS20A_B	4	RS20A_B	4	RS20A		
5	NC	5	NC	5	NC	5	NC		
6	RS24A_B	6	RS21B_B	6	RS21B_B	6	RS21B		
7	RS24B_B	7	RS21A_B	7	RS21A_B	7	RS21A		
8	RS23A_B	8	RS22B_B	8	RS22B_B	8	RS22B		
9	RS23B_B	9	RS22A_B	9	RS22A_B	9	RS22A		

Table C.1: Sensor Board Connectors' Pinout

Cable 9				*Cable 10*	
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
1	PLCS0	9	PLCS8	1	PLCS16
2	PLCS1	10	PLCS9	2	PLCS17
3	PLCS2	11	PLCS10	3	PLCS18
4	PLCS3	12	PLCS11	4	PLCS19
5	PLCS4	13	PLCS12	5	PLCS20
6	PLCS5	14	PLCS13	6	PLCS21
7	PLCS6	15	PLCS14		
8	PLCS7	16	PLCS15		

Table C.2: Remote switching Board Connectors' Pinout

C.3 PLC Discrete Interface Cables

Each MC68332 processor has (via its respective expansion board) a 32-bit discrete output bus, and a 32-bit discrete input bus to/from the PLC. For CPU A, cables 11 and 12 provide the output bus, and cables 13 and 14 provide the input bus. The pinout of the cables' connectors are given in Table C.3. For CPU B, cables 15 and 16 provide the output bus, and cables 17 and 18 provide the input bus. The pinout of the cables' connectors are given in Table C.4. Any pins not listed are unconnected.

Cable 11				*Cable 12*			
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
1	PLCOUT0_0	18	PLCOUT0_8	1	PLCOUT1_0	18	PLCOUT1_8
2	PLCOUT0_1	19	PLCOUT0_9	2	PLCOUT1_1	19	PLCOUT1_9
3	PLCOUT0_2	20	PLCOUT0_10	3	PLCOUT1_2	20	PLCOUT1_10
4	PLCOUT0_3	21	PLCOUT0_11	4	PLCOUT1_3	21	PLCOUT1_11
5	PLCOUT0_4	22	PLCOUT0_12	5	PLCOUT1_4	22	PLCOUT1_12
6	PLCOUT0_5	23	PLCOUT0_13	6	PLCOUT1_5	23	PLCOUT1_13
7	PLCOUT0_6	24	PLCOUT0_14	7	PLCOUT1_6	24	PLCOUT1_14
8	PLCOUT0_7	25	PLCOUT0_15	8	PLCOUT1_7	25	PLCOUT1_15
Cable 13				*Cable 14*			
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
1	PLCin0_8	18	PLCin0_0	1	PLCin1_8	18	PLCin1_0
2	PLCin0_9	19	PLCin0_1	2	PLCin1_9	19	PLCin1_1
3	PLCin0_10	20	PLCin0_2	3	PLCin1_10	20	PLCin1_2
4	PLCin0_11	21	PLCin0_3	4	PLCin1_11	21	PLCin1_3
5	PLCin0_12	22	PLCin0_4	5	PLCin1_12	22	PLCin1_4
6	PLCin0_13	23	PLCin0_5	6	PLCin1_13	23	PLCin1_5
7	PLCin0_14	24	PLCin0_6	7	PLCin1_14	24	PLCin1_6
8	PLCin0_15	25	PLCin0_7	8	PLCin1_15	25	PLCin1_7

Table C.3: PLC Discrete Interface Cable Connectors' Pinout, CPU A

C.4 Crane and Train Cables

Each crane is connected to one of the expansion boards by a cable (cable 22 for crane 1, cable 23 for crane 2, and cable 24 for crane 3). The train interface is connected to expansion board A by cable 26. The pinout of the cable's connectors are given in Table C.5. Any pins not listed are unconnected.

NOTE: The connector for the Märklin train interface unit is a custom 6 pin connector that was supplied with the interface unit.

Cable 15				*Cable 16*			
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
1	PLCOUT0_0	18	PLCOUT0_8	1	PLCOUT1_0	18	PLCOUT1_8
2	PLCOUT0_1	19	PLCOUT0_9	2	PLCOUT1_1	19	PLCOUT1_9
3	PLCOUT0_2	20	PLCOUT0_10	3	PLCOUT1_2	20	PLCOUT1_10
4	PLCOUT0_3	21	PLCOUT0_11	4	PLCOUT1_3	21	PLCOUT1_11
5	PLCOUT0_4	22	PLCOUT0_12	5	PLCOUT1_4	22	PLCOUT1_12
6	PLCOUT0_5	23	PLCOUT0_13	6	PLCOUT1_5	23	PLCOUT1_13
7	PLCOUT0_6	24	PLCOUT0_14	7	PLCOUT1_6	24	PLCOUT1_14
8	PLCOUT0_7	25	PLCOUT0_15	8	PLCOUT1_7	25	PLCOUT1_15
Cable 17				*Cable 18*			
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
1	PLCin0_8	18	PLCin0_0	1	PLCin1_8	18	PLCin1_0
2	PLCin0_9	19	PLCin0_1	2	PLCin1_9	19	PLCin1_1
3	PLCin0_10	20	PLCin0_2	3	PLCin1_10	20	PLCin1_2
4	PLCin0_11	21	PLCin0_3	4	PLCin1_11	21	PLCin1_3
5	PLCin0_12	22	PLCin0_4	5	PLCin1_12	22	PLCin1_4
6	PLCin0_13	23	PLCin0_5	6	PLCin1_13	23	PLCin1_5
7	PLCin0_14	24	PLCin0_6	7	PLCin1_14	24	PLCin1_6
8	PLCin0_15	25	PLCin0_7	8	PLCin1_15	25	PLCin1_7

Table C.4: PLC Discrete Interface Cable Connectors' Pinout, CPU B

Cable 22		*Cable 23*		*Cable 24*	
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
1	Crane1_T_R	1	Crane2_T_R	1	Crane3_T_R
2	Crane1_T_L	2	Crane2_T_L	2	Crane3_T_L
3	Crane1_Lower	3	Crane2_Lower	3	Crane3_Lower
4	Crane1_Raise	4	Crane2_Raise	4	Crane3_Raise
5	Crane1_Mag	5	Crane2_Mag	5	Crane3_Mag
6	train GND	6	train GND	7	train GND
Cable 26 (Train)		*Cable 26 (CPU)*			
Pin	*Signal*	*Pin*	*Signal*		
1	RD	2	RXA (TD)		
3	GND	3	TXA (RD)		
4	TD	7	GND		
5	CTS				

Table C.5: Crane and Train Cable Connectors' Pinout

C.5 Programming Terminal Cables

The two MC68332 processors are both connected to the serial box by cables (CPU A by cable 19, and CPU B by cable 20). The serial box is then connected to the Compaq PC by cable 25. The PLC is connected to the Compaq PC by cable 21. The pinout of the cable's connectors are given in Table C.6. Any pins not listed are unconnected.

NOTE: Cable 21 is a custom cable (model 1784-CP10) supplied by Allen-Bradley. For cable 21, several sets of pins on the connectors are tied together. On the PC side, they are {4, 6} and {7, 88}. On the PLC side, they are {4, 5} and {6, 8, 20}.

Cable 19 (CPU A)		*Cable 20 (Serial box)*		*Cable 20 (CPU B)*		*Cable 20 (Serial box)*	
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
2	RXD	2	TXD	2	RXD	2	TXD
3	TXD	3	RXD	3	TXD	3	RXD
5	GND	7	GND	5	GND	7	GND
Cable 25 (PC)		*Cable 25 (Serial box)*		*Cable 21 (PLC)*		*Cable 21 (PC)*	
Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
2	TXD	2	TXD	2	RXD	2	RXD
3	RXD	3	RXD	7	GND	5	GND
7	GND	7	GND	3	TXD	3	TXD
				4	RTS	4	DTR
				5	CTS	6	DSR
				6	DSR	7	RTS
				8	DCD	8	CTS
				20	DTR		

Table C.6: Programming Terminal Cable Connectors' Pinout

Appendix D

PLC Source Code

This Appendix contains the PLC source code that controls the testbed, and implements the supervisors described in Chapter 7. First, the source code for the sequential function chart (SFC) program that calls the other programs, is given. After the listing of the SFC program, the source code for the relay ladder logic (RLL) programs that the SFC calls is given. This includes the two intialization programs, *init1* and *init2*, as well as the main program, *supvis,* that implements the DES supervisors.

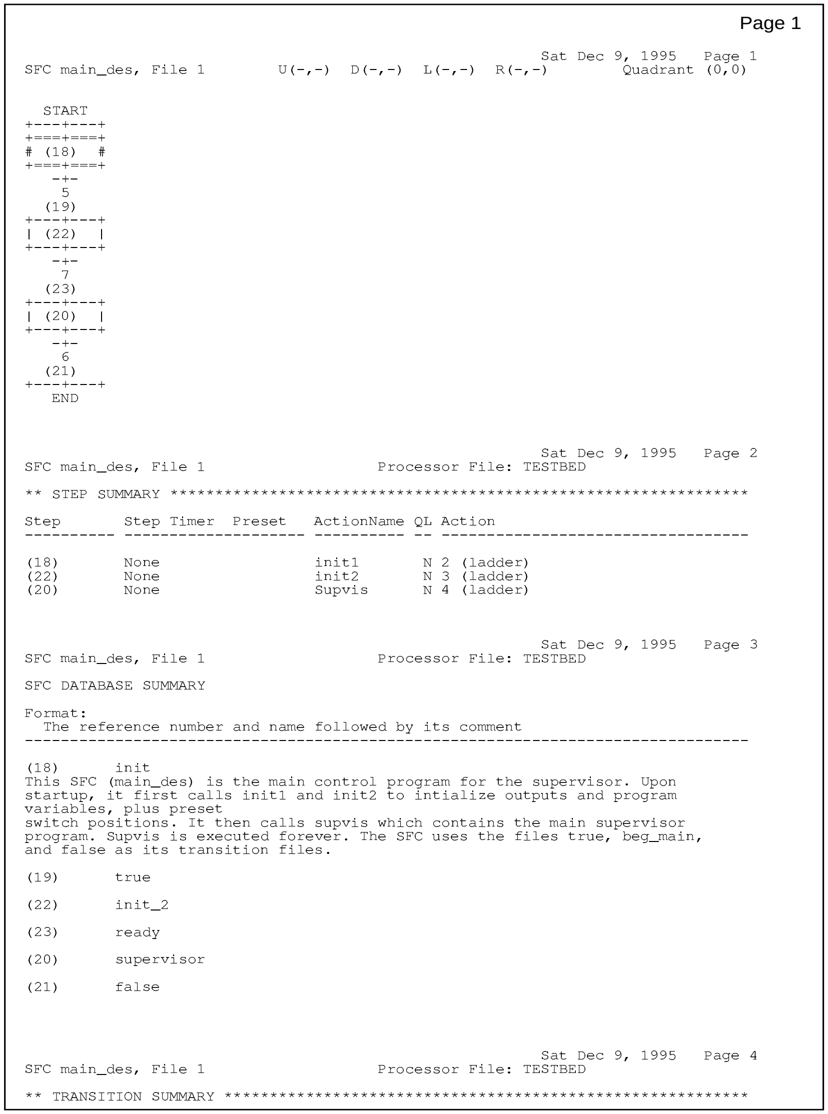
NOTE: Please excuse the fact that each set of pages is numbered "1" and "2". This is an uncorrectable quirk of the method used to generate the program listing in the two page format.

D.1 SFC Program

This section contains the program listing for the sequential function chart file, *main_des*, as well as the listing for the three RLL files *(true, false,* and *beg_main)* that the SFC program uses as its transition files.

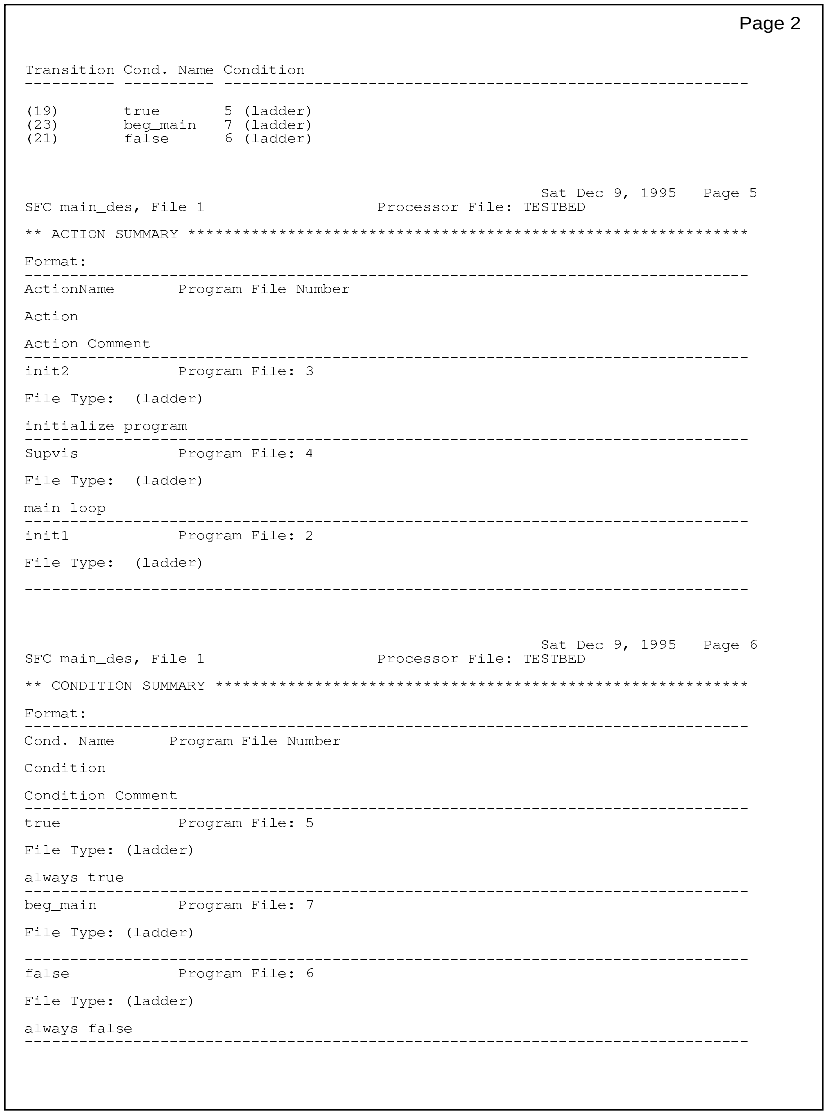
D.1.1 Main_des

This is the program listing for the SFC program *main_des.* This file is the main control program for the PLC, thus it begins execution as soon as the PLC enters run mode. This program, in turn, calls the RLL programs.



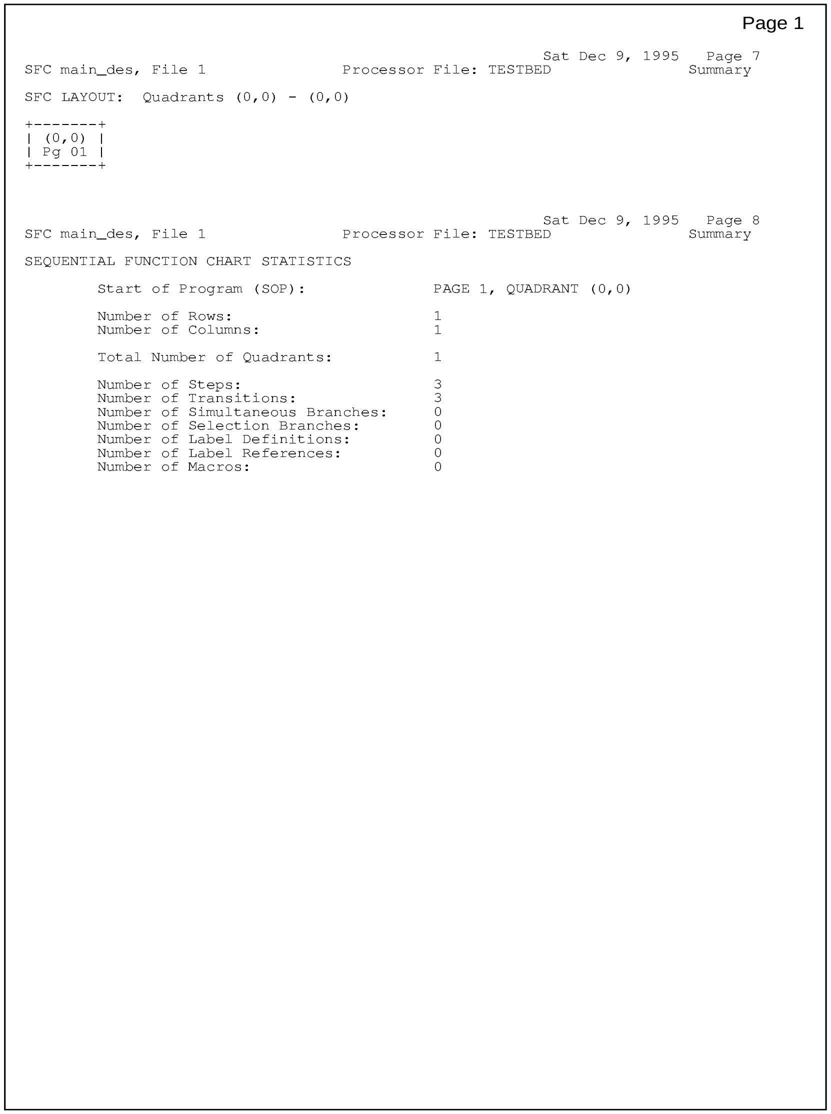
Page 1

```
                                                         Sat Dec 9, 1995   Page 1
SFC main_des, File 1        U(-,-)  D(-,-)  L(-,-)  R(-,-)        Quadrant (0,0)


  START
+---+---+
+===+===+
# (18)  #
+===+===+
   -+-
    5
  (19)
+---+---+
| (22)  |
+---+---+
   -+-
    7
  (23)
+---+---+
| (20)  |
+---+---+
   -+-
    6
  (21)
+---+---+
   END



                                                         Sat Dec 9, 1995   Page 2
SFC main_des, File 1                   Processor File: TESTBED

** STEP SUMMARY ****************************************************************

Step       Step Timer  Preset   ActionName QL Action
---------- -------------------- ---------- -- ----------------------------------

(18)       None                 init1       N 2 (ladder)
(22)       None                 init2       N 3 (ladder)
(20)       None                 Supvis      N 4 (ladder)



                                                         Sat Dec 9, 1995   Page 3
SFC main_des, File 1                   Processor File: TESTBED

SFC DATABASE SUMMARY

Format:
  The reference number and name followed by its comment
--------------------------------------------------------------------------------

(18)      init
This SFC (main_des) is the main control program for the supervisor. Upon
startup, it first calls init1 and init2 to intialize outputs and program
variables, plus preset
switch positions. It then calls supvis which contains the main supervisor
program. Supvis is executed forever. The SFC uses the files true, beg_main,
and false as its transition files.

(19)      true

(22)      init_2

(23)      ready

(20)      supervisor

(21)      false



                                                         Sat Dec 9, 1995   Page 4
SFC main_des, File 1                   Processor File: TESTBED

** TRANSITION SUMMARY **********************************************************
```



Page 2

```
Transition Cond. Name Condition
---------- ---------- ----------------------------------------------------------

(19)       true       5 (ladder)
(23)       beg_main   7 (ladder)
(21)       false      6 (ladder)



                                                         Sat Dec 9, 1995   Page 5
SFC main_des, File 1                   Processor File: TESTBED

** ACTION SUMMARY **************************************************************

Format:
--------------------------------------------------------------------------------
ActionName       Program File Number

Action

Action Comment
--------------------------------------------------------------------------------
init2            Program File: 3

File Type:  (ladder)

initialize program
--------------------------------------------------------------------------------
Supvis           Program File: 4

File Type:  (ladder)

main loop
--------------------------------------------------------------------------------
init1            Program File: 2

File Type:  (ladder)

--------------------------------------------------------------------------------



                                                         Sat Dec 9, 1995   Page 6
SFC main_des, File 1                   Processor File: TESTBED

** CONDITION SUMMARY ***********************************************************

Format:
--------------------------------------------------------------------------------
Cond. Name      Program File Number

Condition

Condition Comment
--------------------------------------------------------------------------------
true             Program File: 5

File Type: (ladder)

always true
--------------------------------------------------------------------------------
beg_main         Program File: 7

File Type: (ladder)

--------------------------------------------------------------------------------
false            Program File: 6

File Type: (ladder)

always false
--------------------------------------------------------------------------------
```

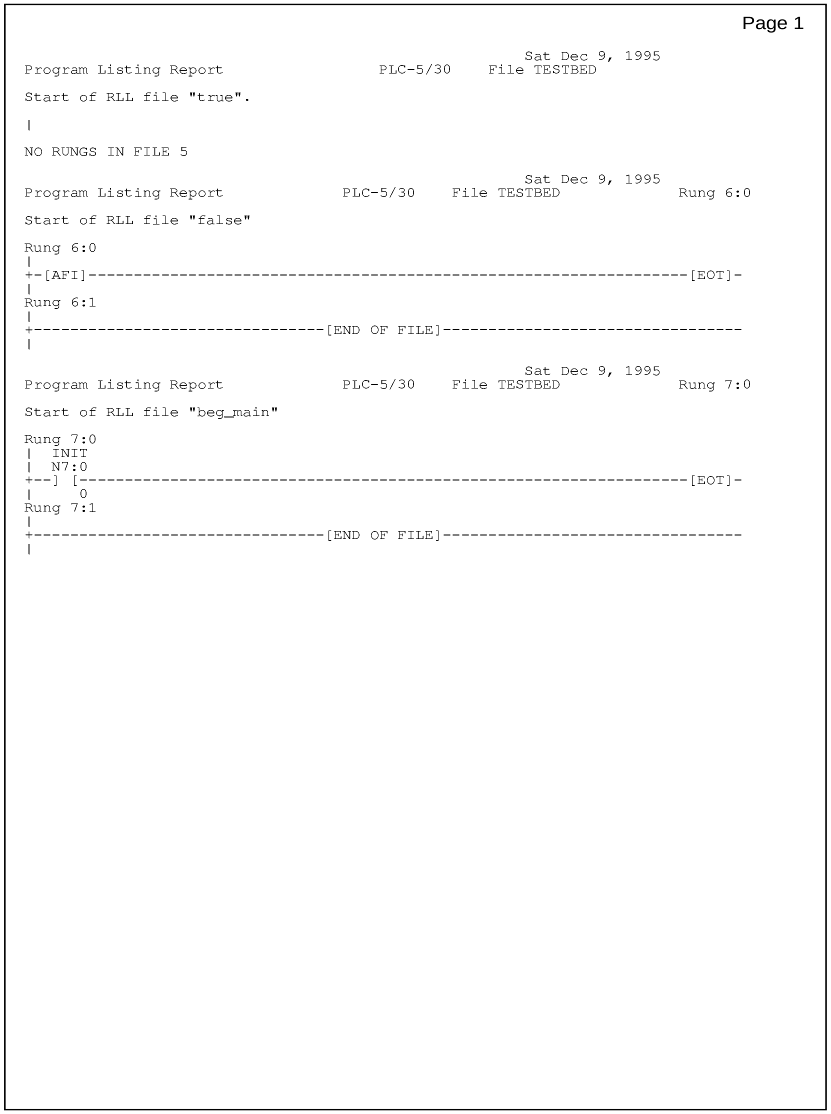


Page 1

```
                                                         Sat Dec 9, 1995   Page 7
SFC main_des, File 1              Processor File: TESTBED               Summary

SFC LAYOUT:  Quadrants (0,0) - (0,0)

+--------+
| (0,0) |
| Pg 01 |
+--------+



                                                         Sat Dec 9, 1995   Page 8
SFC main_des, File 1              Processor File: TESTBED               Summary

SEQUENTIAL FUNCTION CHART STATISTICS

        Start of Program (SOP):              PAGE 1, QUADRANT (0,0)

        Number of Rows:                      1
        Number of Columns:                   1

        Total Number of Quadrants:           1

        Number of Steps:                     3
        Number of Transitions:               3
        Number of Simultaneous Branches:     0
        Number of Selection Branches:        0
        Number of Label Definitions:         0
        Number of Label References:          0
        Number of Macros:                    0
```

D.1.2 Transition Files

The soure code for the transition files *true, false,* and *begin_main* are given below. These are the transition files for the SFC program *main_des.*



Page 1

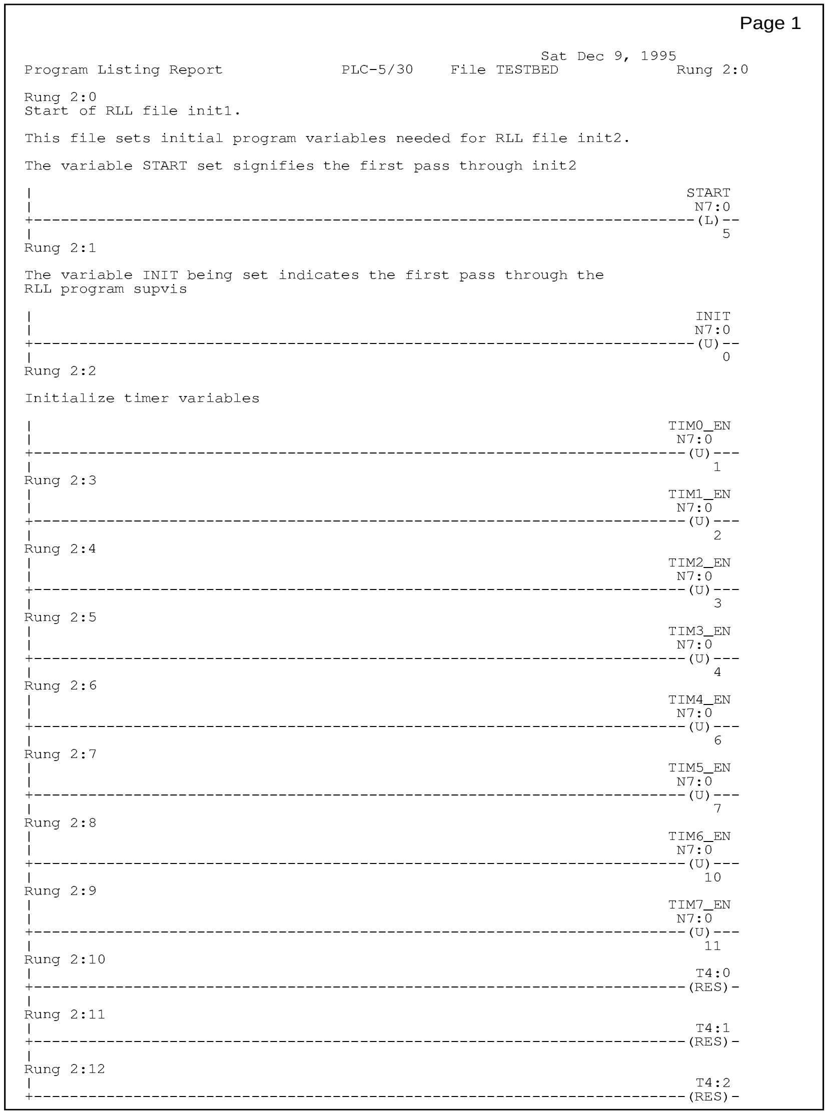
```
                                                       Sat Dec 9, 1995
Program Listing Report                 PLC-5/30    File TESTBED

Start of RLL file "true".

|

NO RUNGS IN FILE 5

                                                       Sat Dec 9, 1995
Program Listing Report             PLC-5/30    File TESTBED             Rung 6:0

Start of RLL file "false"

Rung 6:0
|
+-[AFI]------------------------------------------------------------------[EOT]-
|
Rung 6:1
|
+---------------------------------[END OF FILE]---------------------------------
|

                                                       Sat Dec 9, 1995
Program Listing Report             PLC-5/30    File TESTBED             Rung 7:0

Start of RLL file "beg_main"

Rung 7:0
|  INIT
|  N7:0
+--] [-------------------------------------------------------------------[EOT]-
|     0
Rung 7:1
|
+---------------------------------[END OF FILE]---------------------------------
|
```

D.2 RLL Programs

This section contains the program listings for the relay ladder logic programs *init1, init2,* and *supvis.* These programs are called by the SFC program *main_des* in the order they are listed above.

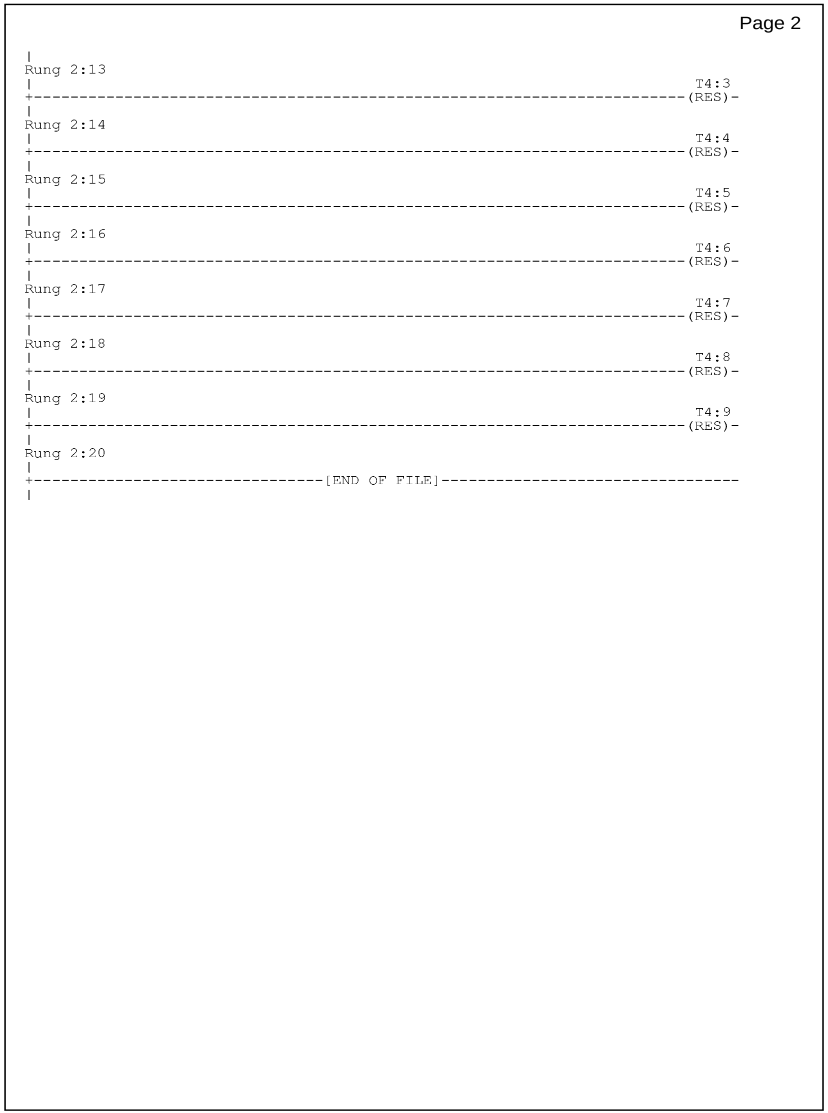
D.2.1 Init1

The listing for the RLL program *init1* is given below. This is the first initialization file called. It performs initialization for *init2.*



Page 1

```
                                                         Sat Dec 9, 1995
Program Listing Report             PLC-5/30    File TESTBED             Rung 2:0

Rung 2:0
Start of RLL file init1.

This file sets initial program variables needed for RLL file init2.

The variable START set signifies the first pass through init2

|                                                                         START
|                                                                          N7:0
+------------------------------------------------------------------------(L)--
|                                                                             5
Rung 2:1

The variable INIT being set indicates the first pass through the
RLL program supvis

|                                                                          INIT
|                                                                          N7:0
+------------------------------------------------------------------------(U)--
|                                                                             0
Rung 2:2

Initialize timer variables

|                                                                       TIM0_EN
|                                                                        N7:0
+-----------------------------------------------------------------------(U)---
|                                                                            1
Rung 2:3
|                                                                       TIM1_EN
|                                                                        N7:0
+-----------------------------------------------------------------------(U)---
|                                                                            2
Rung 2:4
|                                                                       TIM2_EN
|                                                                        N7:0
+-----------------------------------------------------------------------(U)---
|                                                                            3
Rung 2:5
|                                                                       TIM3_EN
|                                                                        N7:0
+-----------------------------------------------------------------------(U)---
|                                                                            4
Rung 2:6
|                                                                       TIM4_EN
|                                                                        N7:0
+-----------------------------------------------------------------------(U)---
|                                                                            6
Rung 2:7
|                                                                       TIM5_EN
|                                                                        N7:0
+-----------------------------------------------------------------------(U)---
|                                                                            7
Rung 2:8
|                                                                       TIM6_EN
|                                                                        N7:0
+-----------------------------------------------------------------------(U)---
|                                                                           10
Rung 2:9
|                                                                       TIM7_EN
|                                                                        N7:0
+-----------------------------------------------------------------------(U)---
|                                                                           11
Rung 2:10
|                                                                          T4:0
+-----------------------------------------------------------------------(RES)-
|
Rung 2:11
|                                                                          T4:1
+-----------------------------------------------------------------------(RES)-
|
Rung 2:12
|                                                                          T4:2
+-----------------------------------------------------------------------(RES)-
```

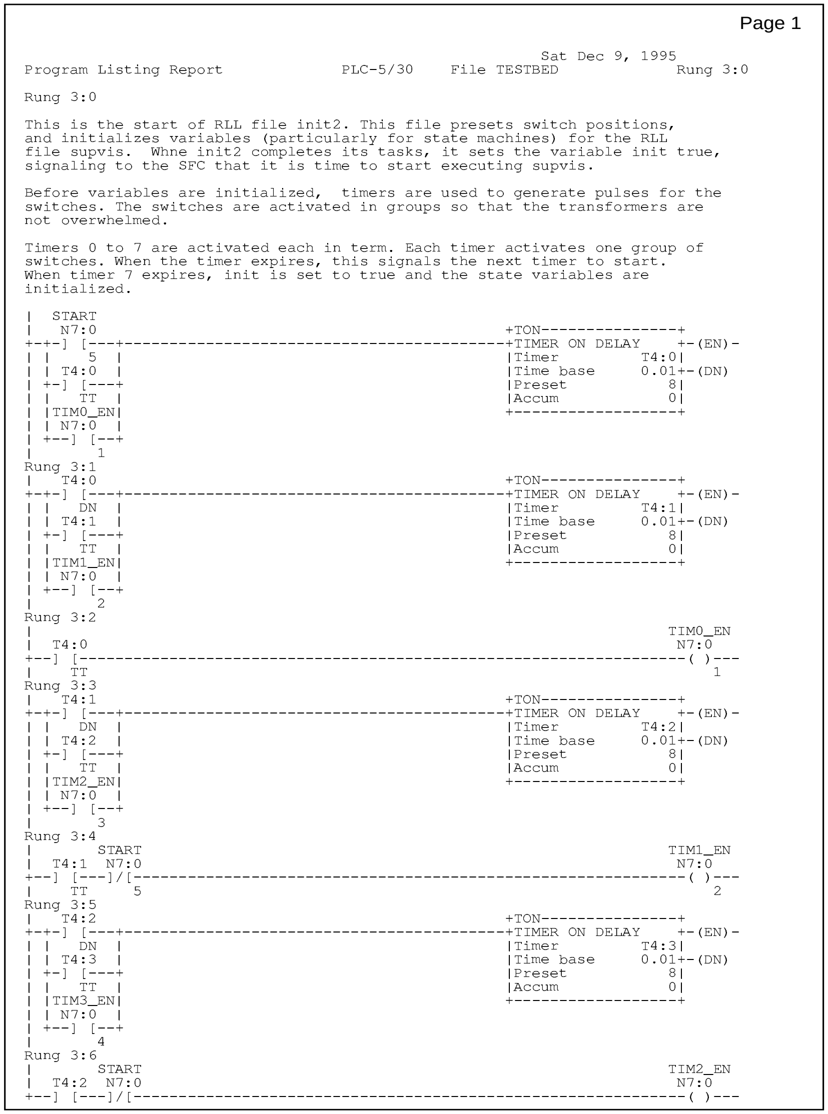


Page 2

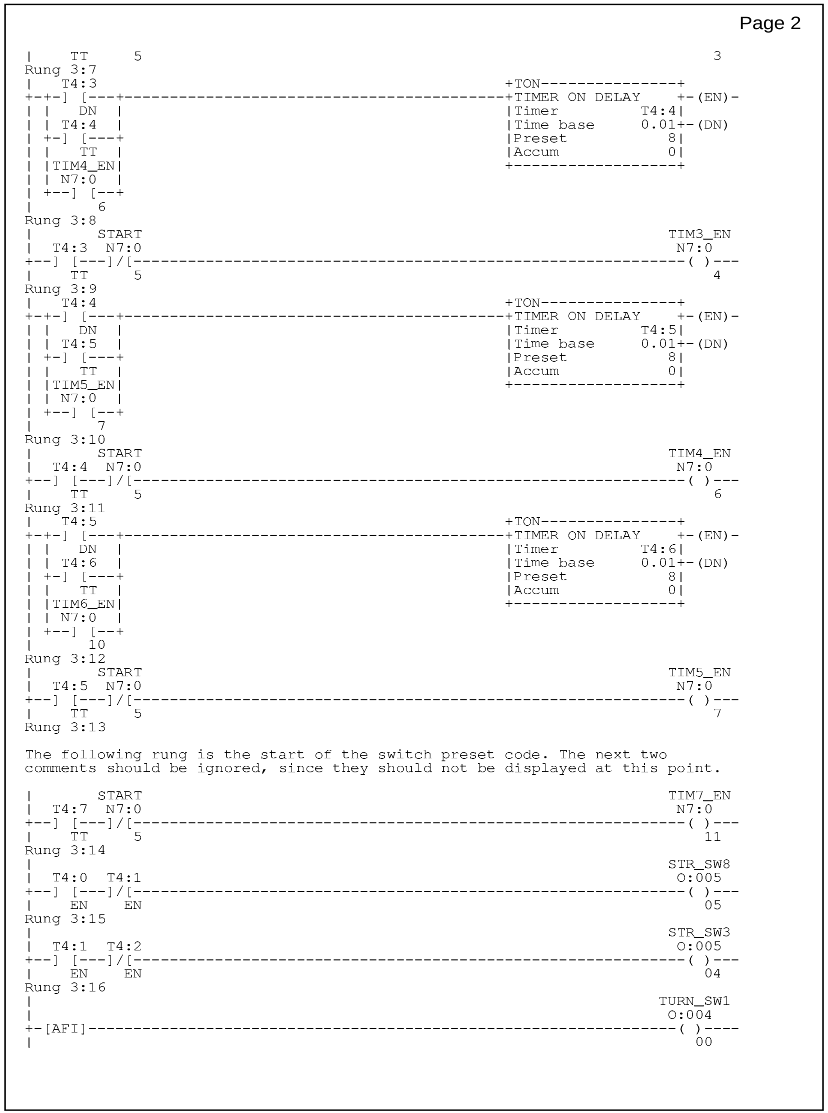
```
|
Rung 2:13
|                                                                          T4:3
+-----------------------------------------------------------------------(RES)-
|
Rung 2:14
|                                                                          T4:4
+-----------------------------------------------------------------------(RES)-
|
Rung 2:15
|                                                                          T4:5
+-----------------------------------------------------------------------(RES)-
|
Rung 2:16
|                                                                          T4:6
+-----------------------------------------------------------------------(RES)-
|
Rung 2:17
|                                                                          T4:7
+-----------------------------------------------------------------------(RES)-
|
Rung 2:18
|                                                                          T4:8
+-----------------------------------------------------------------------(RES)-
|
Rung 2:19
|                                                                          T4:9
+-----------------------------------------------------------------------(RES)-
|
Rung 2:20
|
+--------------------------------[END OF FILE]---------------------------------
|
```

D.2.2 Init2

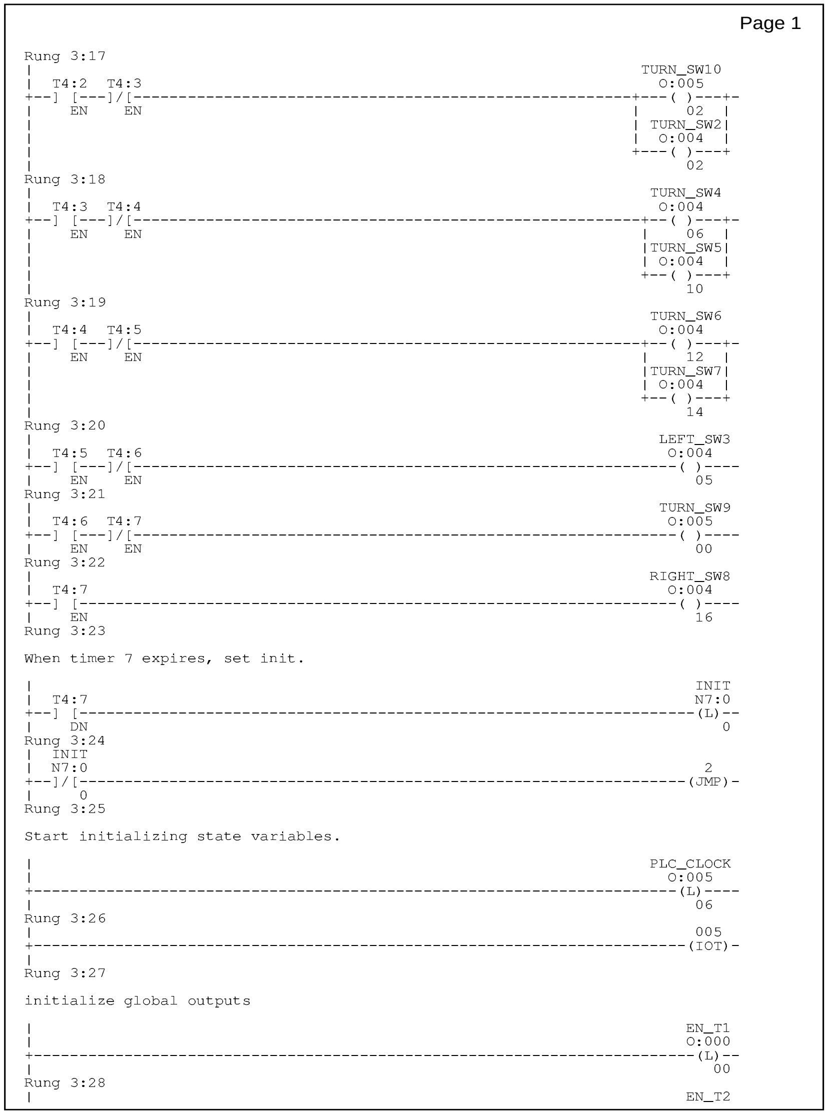
The listing for the RLL program *init2* is given below. This is the initialization file called after *init1*. It initializes the switches to their starting position, and then initializes variables required by the main program, *supvis.*



```
                                                                 Page 1

                                                         Sat Dec 9, 1995
Program Listing Report             PLC-5/30    File TESTBED             Rung 3:0

Rung 3:0

This is the start of RLL file init2. This file presets switch positions,
and initializes variables (particularly for state machines) for the RLL
file supvis.  Whne init2 completes its tasks, it sets the variable init true,
signaling to the SFC that it is time to start executing supvis.

Before variables are initialized,  timers are used to generate pulses for the
switches. The switches are activated in groups so that the transformers are
not overwhelmed.

Timers 0 to 7 are activated each in term. Each timer activates one group of
switches. When the timer expires, this signals the next timer to start.
When timer 7 expires, init is set to true and the state variables are
initialized.

|  START
|   N7:0                                             +TON---------------+
+-+-] [---+------------------------------------------+TIMER ON DELAY    +-(EN)-
| |    5  |                                          |Timer         T4:0|
| | T4:0  |                                          |Time base     0.01+-(DN)
| +-] [---+                                          |Preset           8|
| |   TT  |                                          |Accum            0|
| |TIM0_EN|                                          +------------------+
| | N7:0  |
| +--] [--+
|       1
Rung 3:1
|   T4:0                                             +TON---------------+
+-+-] [---+------------------------------------------+TIMER ON DELAY    +-(EN)-
| |   DN  |                                          |Timer         T4:1|
| | T4:1  |                                          |Time base     0.01+-(DN)
| +-] [---+                                          |Preset           8|
| |   TT  |                                          |Accum            0|
| |TIM1_EN|                                          +------------------+
| | N7:0  |
| +--] [--+
|       2
Rung 3:2
|                                                                      TIM0_EN
|  T4:0                                                                 N7:0
+--] [-------------------------------------------------------------------( )---
|    TT                                                                    1
Rung 3:3
|   T4:1                                             +TON---------------+
+-+-] [---+------------------------------------------+TIMER ON DELAY    +-(EN)-
| |   DN  |                                          |Timer         T4:2|
| | T4:2  |                                          |Time base     0.01+-(DN)
| +-] [---+                                          |Preset           8|
| |   TT  |                                          |Accum            0|
| |TIM2_EN|                                          +------------------+
| | N7:0  |
| +--] [--+
|       3
Rung 3:4
|       START                                                          TIM1_EN
|  T4:1  N7:0                                                           N7:0
+--] [---]/[-------------------------------------------------------------( )---
|    TT     5                                                              2
Rung 3:5
|   T4:2                                             +TON---------------+
+-+-] [---+------------------------------------------+TIMER ON DELAY    +-(EN)-
| |   DN  |                                          |Timer         T4:3|
| | T4:3  |                                          |Time base     0.01+-(DN)
| +-] [---+                                          |Preset           8|
| |   TT  |                                          |Accum            0|
| |TIM3_EN|                                          +------------------+
| | N7:0  |
| +--] [--+
|       4
Rung 3:6
|       START                                                          TIM2_EN
|  T4:2  N7:0                                                           N7:0
+--] [---]/[-------------------------------------------------------------( )---
```

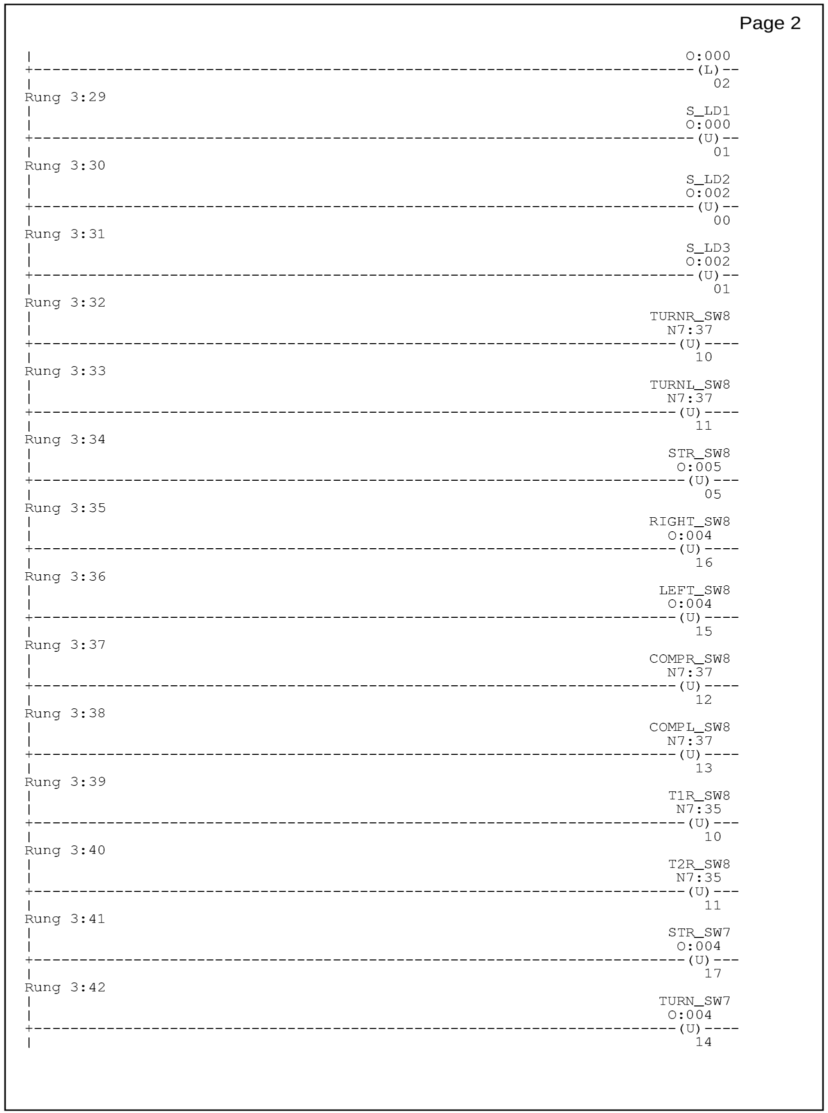


```
                                                                 Page 2

|    TT     5                                                              3
Rung 3:7
|   T4:3                                             +TON---------------+
+-+-] [---+------------------------------------------+TIMER ON DELAY    +-(EN)-
| |   DN  |                                          |Timer         T4:4|
| | T4:4  |                                          |Time base     0.01+-(DN)
| +-] [---+                                          |Preset           8|
| |   TT  |                                          |Accum            0|
| |TIM4_EN|                                          +------------------+
| | N7:0  |
| +--] [--+
|       6
Rung 3:8
|       START                                                          TIM3_EN
|  T4:3  N7:0                                                           N7:0
+--] [---]/[-------------------------------------------------------------( )---
|    TT     5                                                              4
Rung 3:9
|   T4:4                                             +TON---------------+
+-+-] [---+------------------------------------------+TIMER ON DELAY    +-(EN)-
| |   DN  |                                          |Timer         T4:5|
| | T4:5  |                                          |Time base     0.01+-(DN)
| +-] [---+                                          |Preset           8|
| |   TT  |                                          |Accum            0|
| |TIM5_EN|                                          +------------------+
| | N7:0  |
| +--] [--+
|       7
Rung 3:10
|       START                                                          TIM4_EN
|  T4:4  N7:0                                                           N7:0
+--] [---]/[-------------------------------------------------------------( )---
|    TT     5                                                              6
Rung 3:11
|   T4:5                                             +TON---------------+
+-+-] [---+------------------------------------------+TIMER ON DELAY    +-(EN)-
| |   DN  |                                          |Timer         T4:6|
| | T4:6  |                                          |Time base     0.01+-(DN)
| +-] [---+                                          |Preset           8|
| |   TT  |                                          |Accum            0|
| |TIM6_EN|                                          +------------------+
| | N7:0  |
| +--] [--+
|      10
Rung 3:12
|       START                                                          TIM5_EN
|  T4:5  N7:0                                                           N7:0
+--] [---]/[-------------------------------------------------------------( )---
|    TT     5                                                              7
Rung 3:13

The following rung is the start of the switch preset code. The next two
comments should be ignored, since they should not be displayed at this point.

|       START                                                          TIM7_EN
|  T4:7  N7:0                                                           N7:0
+--] [---]/[-------------------------------------------------------------( )---
|    TT     5                                                             11
Rung 3:14
|                                                                      STR_SW8
|  T4:0  T4:1                                                           O:005
+--] [---]/[-------------------------------------------------------------( )---
|    EN    EN                                                             05
Rung 3:15
|                                                                      STR_SW3
|  T4:1  T4:2                                                           O:005
+--] [---]/[-------------------------------------------------------------( )---
|    EN    EN                                                             04
Rung 3:16
|                                                                     TURN_SW1
|                                                                      O:004
+-[AFI]-----------------------------------------------------------------( )----
|                                                                        00
```



Page 1

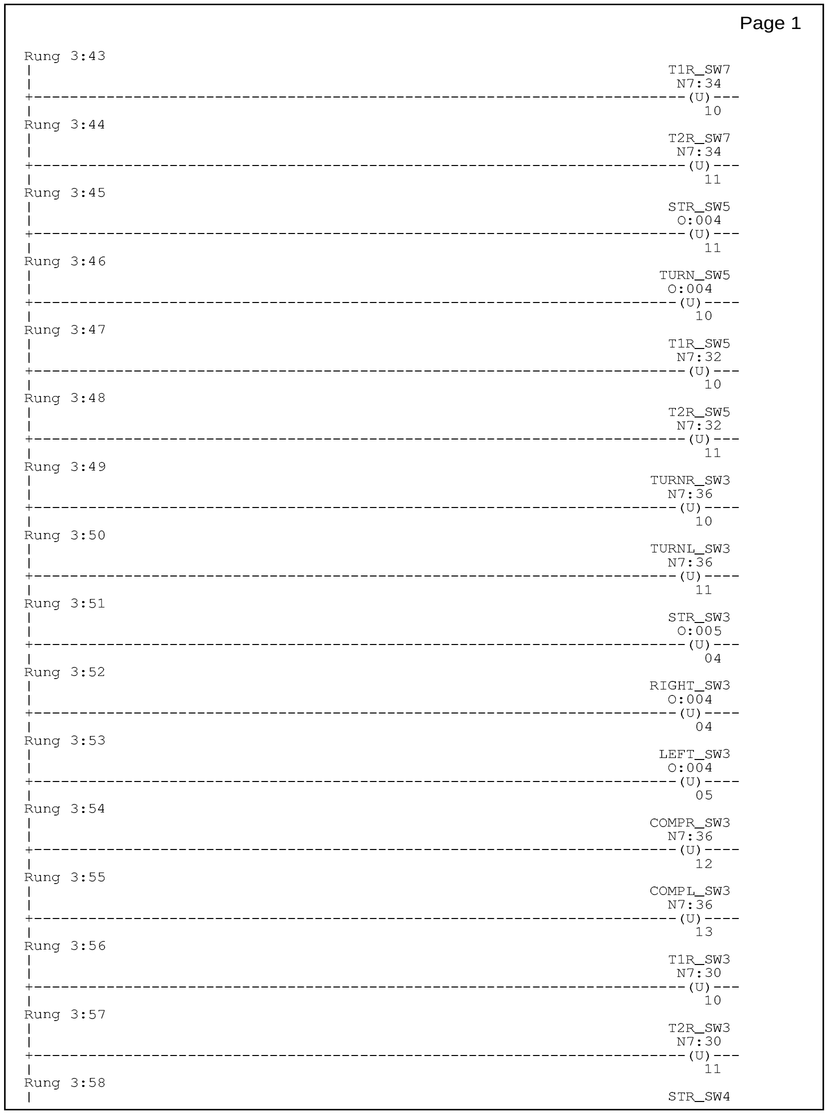
```
Rung 3:17
|                                                                    TURN_SW10
|  T4:2  T4:3                                                          O:005
+--] [---]/[--------------------------------------------------------+---( )---+-
|    EN    EN                                                       |    02   |
|                                                                   | TURN_SW2|
|                                                                   |  O:004  |
|                                                                   +---( )---+
|                                                                        02
Rung 3:18
|                                                                     TURN_SW4
|  T4:3  T4:4                                                          O:004
+--] [---]/[---------------------------------------------------------+--( )---+-
|    EN    EN                                                        |   06   |
|                                                                    |TURN_SW5|
|                                                                    | O:004  |
|                                                                    +--( )---+
|                                                                        10
Rung 3:19
|                                                                     TURN_SW6
|  T4:4  T4:5                                                          O:004
+--] [---]/[---------------------------------------------------------+--( )---+-
|    EN    EN                                                        |   12   |
|                                                                    |TURN_SW7|
|                                                                    | O:004  |
|                                                                    +--( )---+
|                                                                        14
Rung 3:20
|                                                                      LEFT_SW3
|  T4:5  T4:6                                                           O:004
+--] [---]/[-------------------------------------------------------------( )----
|    EN    EN                                                              05
Rung 3:21
|                                                                      TURN_SW9
|  T4:6  T4:7                                                           O:005
+--] [---]/[-------------------------------------------------------------( )----
|    EN    EN                                                              00
Rung 3:22
|                                                                     RIGHT_SW8
|  T4:7                                                                 O:004
+--] [-------------------------------------------------------------------( )----
|    EN                                                                    16
Rung 3:23

When timer 7 expires, set init.

|                                                                          INIT
|  T4:7                                                                    N7:0
+--] [---------------------------------------------------------------------(L)--
|    DN                                                                       0
Rung 3:24
|  INIT
|  N7:0                                                                     2
+--]/[--------------------------------------------------------------------(JMP)-
|     0
Rung 3:25

Start initializing state variables.

|                                                                     PLC_CLOCK
|                                                                       O:005
+------------------------------------------------------------------------(L)----
|                                                                        06
Rung 3:26
|                                                                          005
+-------------------------------------------------------------------------(IOT)-
|
Rung 3:27

initialize global outputs

|                                                                         EN_T1
|                                                                         O:000
+--------------------------------------------------------------------------(L)--
|                                                                            00
Rung 3:28
|                                                                         EN_T2
```



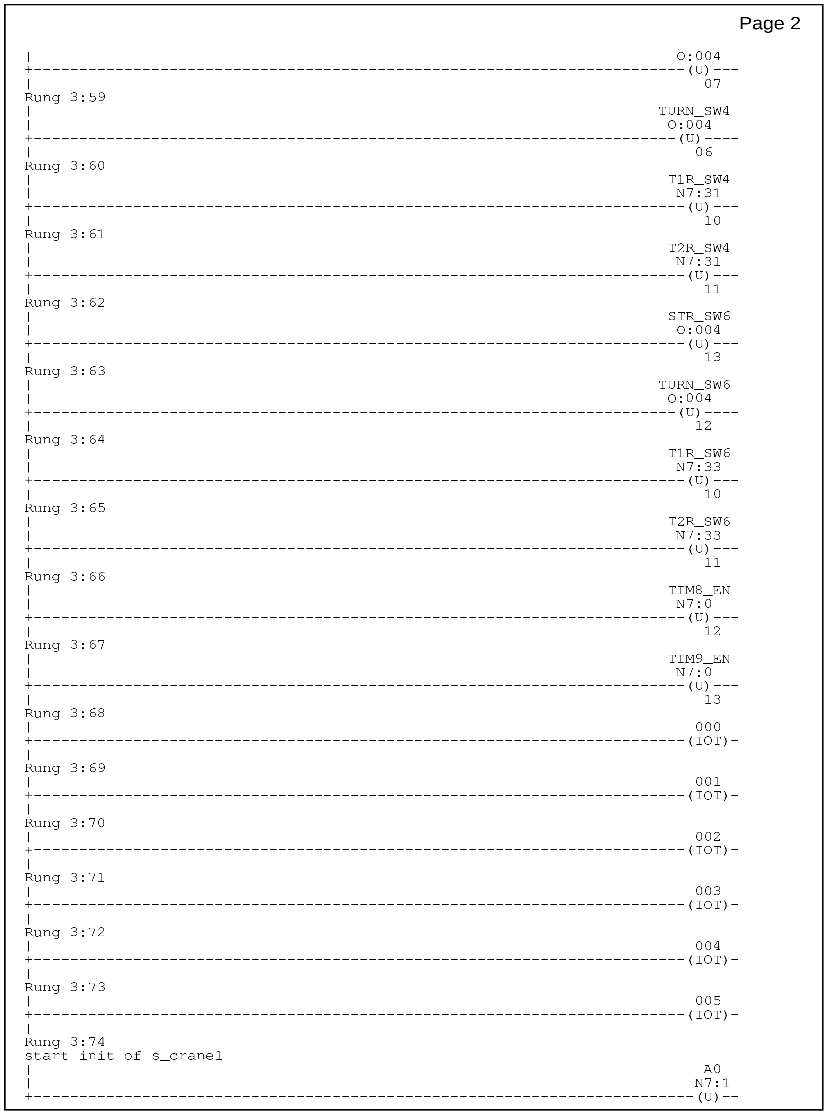
Page 2

```
|                                                                         O:000
+--------------------------------------------------------------------------(L)--
|                                                                            02
Rung 3:29
|                                                                         S_LD1
|                                                                         O:000
+--------------------------------------------------------------------------(U)--
|                                                                            01
Rung 3:30
|                                                                         S_LD2
|                                                                         O:002
+--------------------------------------------------------------------------(U)--
|                                                                            00
Rung 3:31
|                                                                         S_LD3
|                                                                         O:002
+--------------------------------------------------------------------------(U)--
|                                                                            01
Rung 3:32
|                                                                     TURNR_SW8
|                                                                       N7:37
+------------------------------------------------------------------------(U)----
|                                                                          10
Rung 3:33
|                                                                     TURNL_SW8
|                                                                       N7:37
+------------------------------------------------------------------------(U)----
|                                                                          11
Rung 3:34
|                                                                       STR_SW8
|                                                                        O:005
+-------------------------------------------------------------------------(U)---
|                                                                           05
Rung 3:35
|                                                                     RIGHT_SW8
|                                                                       O:004
+------------------------------------------------------------------------(U)----
|                                                                          16
Rung 3:36
|                                                                      LEFT_SW8
|                                                                       O:004
+------------------------------------------------------------------------(U)----
|                                                                          15
Rung 3:37
|                                                                     COMPR_SW8
|                                                                       N7:37
+------------------------------------------------------------------------(U)----
|                                                                          12
Rung 3:38
|                                                                     COMPL_SW8
|                                                                       N7:37
+------------------------------------------------------------------------(U)----
|                                                                          13
Rung 3:39
|                                                                       T1R_SW8
|                                                                        N7:35
+-------------------------------------------------------------------------(U)---
|                                                                           10
Rung 3:40
|                                                                       T2R_SW8
|                                                                        N7:35
+-------------------------------------------------------------------------(U)---
|                                                                           11
Rung 3:41
|                                                                       STR_SW7
|                                                                        O:004
+-------------------------------------------------------------------------(U)---
|                                                                           17
Rung 3:42
|                                                                      TURN_SW7
|                                                                       O:004
+------------------------------------------------------------------------(U)----
|                                                                          14
```

1

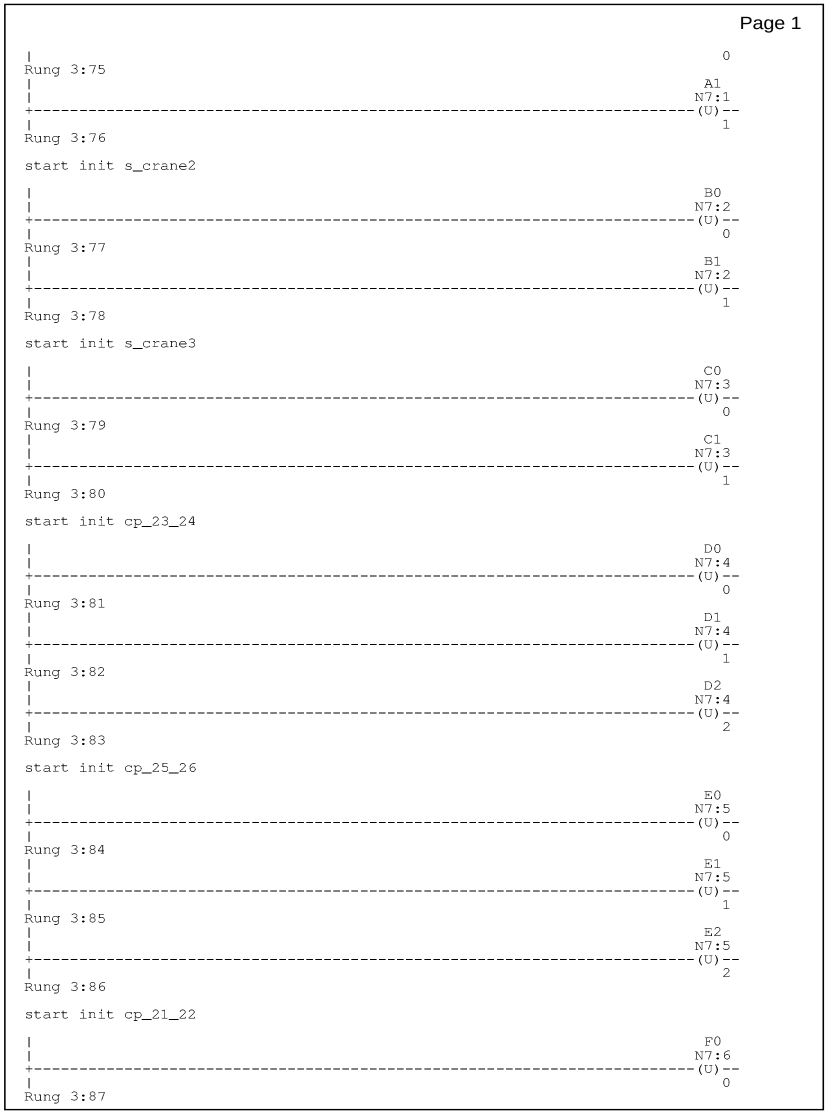


```
                                                                          Page 1
Rung 3:43
|                                                                      T1R_SW7
|                                                                       N7:34
+------------------------------------------------------------------------(U)---
|                                                                          10
Rung 3:44
|                                                                      T2R_SW7
|                                                                       N7:34
+------------------------------------------------------------------------(U)---
|                                                                          11
Rung 3:45
|                                                                      STR_SW5
|                                                                       O:004
+------------------------------------------------------------------------(U)---
|                                                                          11
Rung 3:46
|                                                                     TURN_SW5
|                                                                      O:004
+-----------------------------------------------------------------------(U)----
|                                                                         10
Rung 3:47
|                                                                      T1R_SW5
|                                                                       N7:32
+------------------------------------------------------------------------(U)---
|                                                                          10
Rung 3:48
|                                                                      T2R_SW5
|                                                                       N7:32
+------------------------------------------------------------------------(U)---
|                                                                          11
Rung 3:49
|                                                                    TURNR_SW3
|                                                                      N7:36
+-----------------------------------------------------------------------(U)----
|                                                                         10
Rung 3:50
|                                                                    TURNL_SW3
|                                                                      N7:36
+-----------------------------------------------------------------------(U)----
|                                                                         11
Rung 3:51
|                                                                      STR_SW3
|                                                                       O:005
+------------------------------------------------------------------------(U)---
|                                                                          04
Rung 3:52
|                                                                    RIGHT_SW3
|                                                                      O:004
+-----------------------------------------------------------------------(U)----
|                                                                         04
Rung 3:53
|                                                                     LEFT_SW3
|                                                                      O:004
+-----------------------------------------------------------------------(U)----
|                                                                         05
Rung 3:54
|                                                                    COMPR_SW3
|                                                                      N7:36
+-----------------------------------------------------------------------(U)----
|                                                                         12
Rung 3:55
|                                                                    COMPL_SW3
|                                                                      N7:36
+-----------------------------------------------------------------------(U)----
|                                                                         13
Rung 3:56
|                                                                      T1R_SW3
|                                                                       N7:30
+------------------------------------------------------------------------(U)---
|                                                                          10
Rung 3:57
|                                                                      T2R_SW3
|                                                                       N7:30
+------------------------------------------------------------------------(U)---
|                                                                          11
Rung 3:58
|                                                                      STR_SW4
```



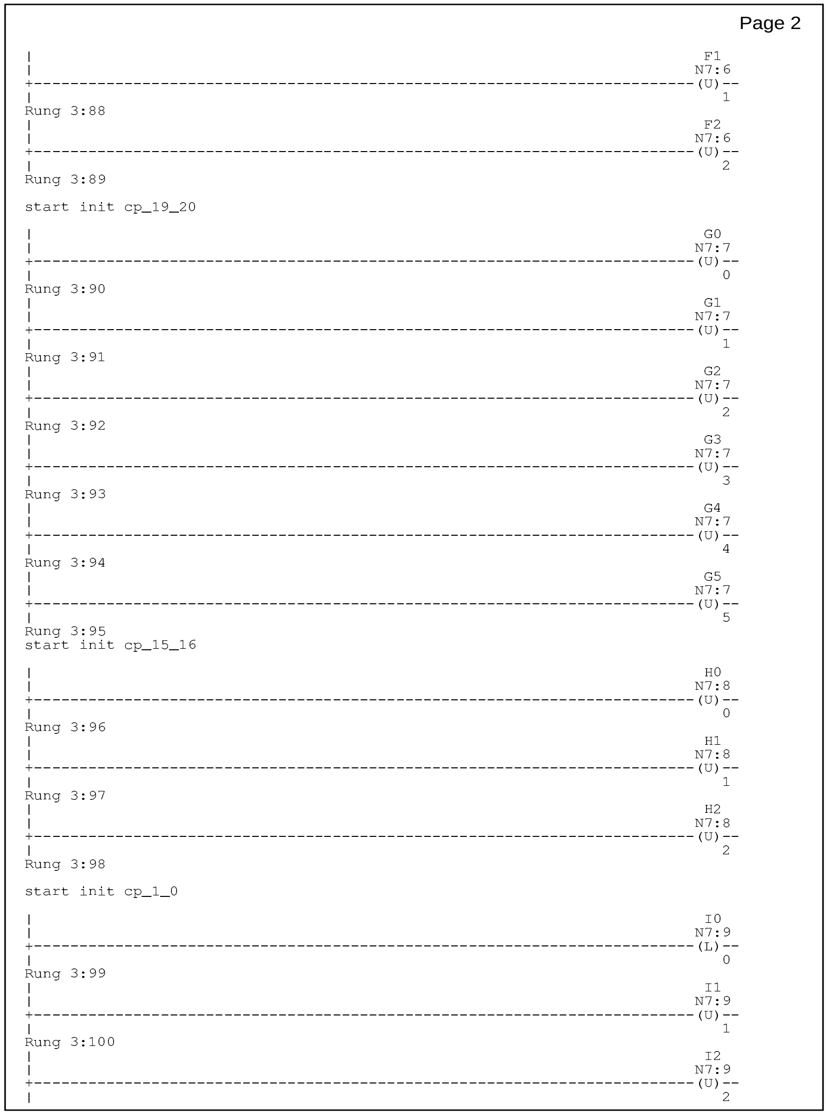
```
                                                                          Page 2
|                                                                       O:004
+------------------------------------------------------------------------(U)---
|                                                                          07
Rung 3:59
|                                                                     TURN_SW4
|                                                                      O:004
+-----------------------------------------------------------------------(U)----
|                                                                         06
Rung 3:60
|                                                                      T1R_SW4
|                                                                       N7:31
+------------------------------------------------------------------------(U)---
|                                                                          10
Rung 3:61
|                                                                      T2R_SW4
|                                                                       N7:31
+------------------------------------------------------------------------(U)---
|                                                                          11
Rung 3:62
|                                                                      STR_SW6
|                                                                       O:004
+------------------------------------------------------------------------(U)---
|                                                                          13
Rung 3:63
|                                                                     TURN_SW6
|                                                                      O:004
+-----------------------------------------------------------------------(U)----
|                                                                         12
Rung 3:64
|                                                                      T1R_SW6
|                                                                       N7:33
+------------------------------------------------------------------------(U)---
|                                                                          10
Rung 3:65
|                                                                      T2R_SW6
|                                                                       N7:33
+------------------------------------------------------------------------(U)---
|                                                                          11
Rung 3:66
|                                                                      TIM8_EN
|                                                                       N7:0
+------------------------------------------------------------------------(U)---
|                                                                          12
Rung 3:67
|                                                                      TIM9_EN
|                                                                       N7:0
+------------------------------------------------------------------------(U)---
|                                                                          13
Rung 3:68
|                                                                         000
+------------------------------------------------------------------------(IOT)-
|
Rung 3:69
|                                                                         001
+------------------------------------------------------------------------(IOT)-
|
Rung 3:70
|                                                                         002
+------------------------------------------------------------------------(IOT)-
|
Rung 3:71
|                                                                         003
+------------------------------------------------------------------------(IOT)-
|
Rung 3:72
|                                                                         004
+------------------------------------------------------------------------(IOT)-
|
Rung 3:73
|                                                                         005
+------------------------------------------------------------------------(IOT)-
|
Rung 3:74
start init of s_crane1
|                                                                          A0
|                                                                         N7:1
+-------------------------------------------------------------------------(U)--
```

1



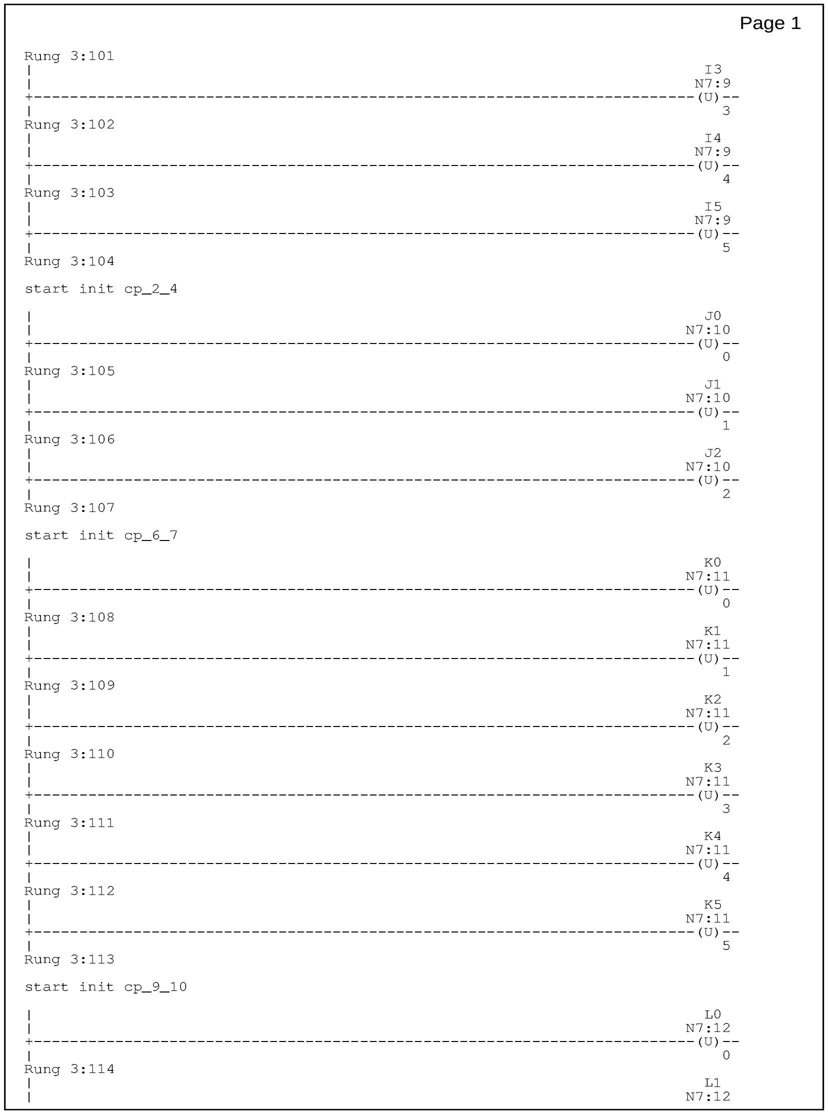
Page 1

```
|                                                                          0
Rung 3:75
|                                                                         A1
|                                                                        N7:1
+-----------------------------------------------------------------------(U)--
|                                                                          1
Rung 3:76

start init s_crane2

|                                                                         B0
|                                                                        N7:2
+-----------------------------------------------------------------------(U)--
|                                                                          0
Rung 3:77
|                                                                         B1
|                                                                        N7:2
+-----------------------------------------------------------------------(U)--
|                                                                          1
Rung 3:78

start init s_crane3

|                                                                         C0
|                                                                        N7:3
+-----------------------------------------------------------------------(U)--
|                                                                          0
Rung 3:79
|                                                                         C1
|                                                                        N7:3
+-----------------------------------------------------------------------(U)--
|                                                                          1
Rung 3:80

start init cp_23_24

|                                                                         D0
|                                                                        N7:4
+-----------------------------------------------------------------------(U)--
|                                                                          0
Rung 3:81
|                                                                         D1
|                                                                        N7:4
+-----------------------------------------------------------------------(U)--
|                                                                          1
Rung 3:82
|                                                                         D2
|                                                                        N7:4
+-----------------------------------------------------------------------(U)--
|                                                                          2
Rung 3:83

start init cp_25_26

|                                                                         E0
|                                                                        N7:5
+-----------------------------------------------------------------------(U)--
|                                                                          0
Rung 3:84
|                                                                         E1
|                                                                        N7:5
+-----------------------------------------------------------------------(U)--
|                                                                          1
Rung 3:85
|                                                                         E2
|                                                                        N7:5
+-----------------------------------------------------------------------(U)--
|                                                                          2
Rung 3:86

start init cp_21_22

|                                                                         F0
|                                                                        N7:6
+-----------------------------------------------------------------------(U)--
|                                                                          0
Rung 3:87
```



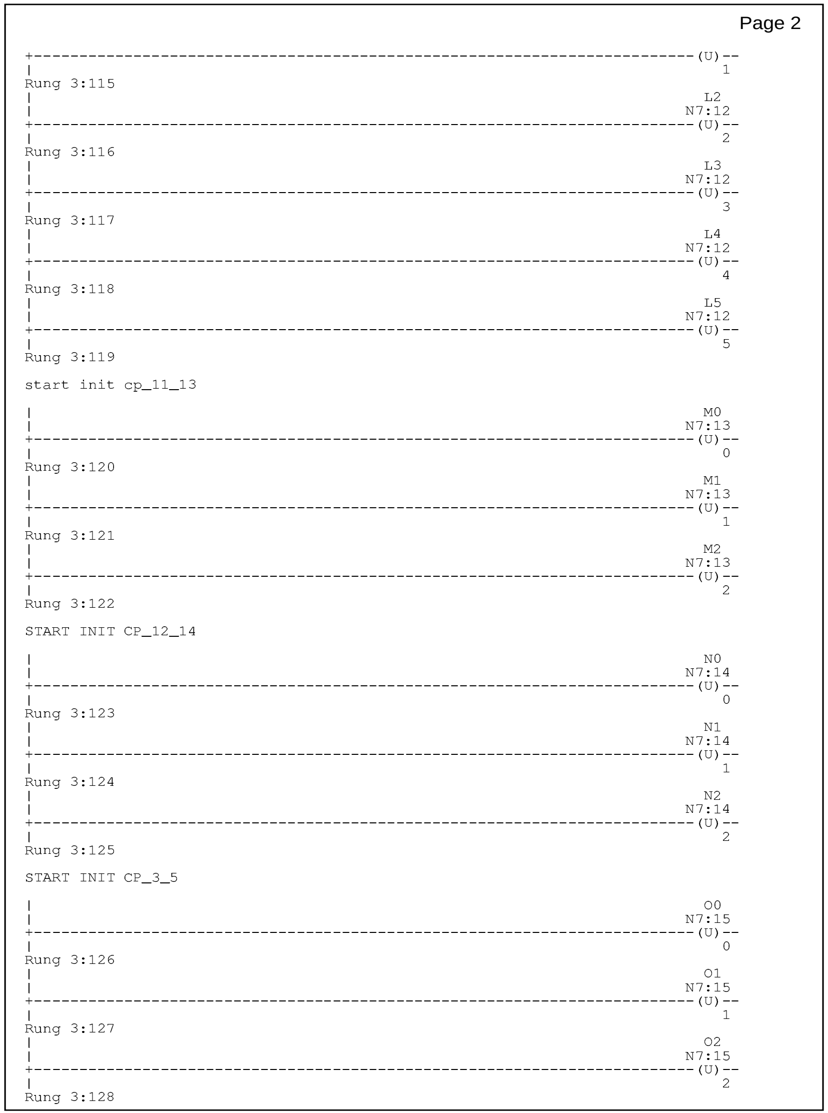
Page 2

```
|                                                                         F1
|                                                                        N7:6
+-----------------------------------------------------------------------(U)--
|                                                                          1
Rung 3:88
|                                                                         F2
|                                                                        N7:6
+-----------------------------------------------------------------------(U)--
|                                                                          2
Rung 3:89

start init cp_19_20

|                                                                         G0
|                                                                        N7:7
+-----------------------------------------------------------------------(U)--
|                                                                          0
Rung 3:90
|                                                                         G1
|                                                                        N7:7
+-----------------------------------------------------------------------(U)--
|                                                                          1
Rung 3:91
|                                                                         G2
|                                                                        N7:7
+-----------------------------------------------------------------------(U)--
|                                                                          2
Rung 3:92
|                                                                         G3
|                                                                        N7:7
+-----------------------------------------------------------------------(U)--
|                                                                          3
Rung 3:93
|                                                                         G4
|                                                                        N7:7
+-----------------------------------------------------------------------(U)--
|                                                                          4
Rung 3:94
|                                                                         G5
|                                                                        N7:7
+-----------------------------------------------------------------------(U)--
|                                                                          5
Rung 3:95
start init cp_15_16

|                                                                         H0
|                                                                        N7:8
+-----------------------------------------------------------------------(U)--
|                                                                          0
Rung 3:96
|                                                                         H1
|                                                                        N7:8
+-----------------------------------------------------------------------(U)--
|                                                                          1
Rung 3:97
|                                                                         H2
|                                                                        N7:8
+-----------------------------------------------------------------------(U)--
|                                                                          2
Rung 3:98

start init cp_1_0

|                                                                         I0
|                                                                        N7:9
+-----------------------------------------------------------------------(L)--
|                                                                          0
Rung 3:99
|                                                                         I1
|                                                                        N7:9
+-----------------------------------------------------------------------(U)--
|                                                                          1
Rung 3:100
|                                                                         I2
|                                                                        N7:9
+-----------------------------------------------------------------------(U)--
|                                                                          2
```



Page 1

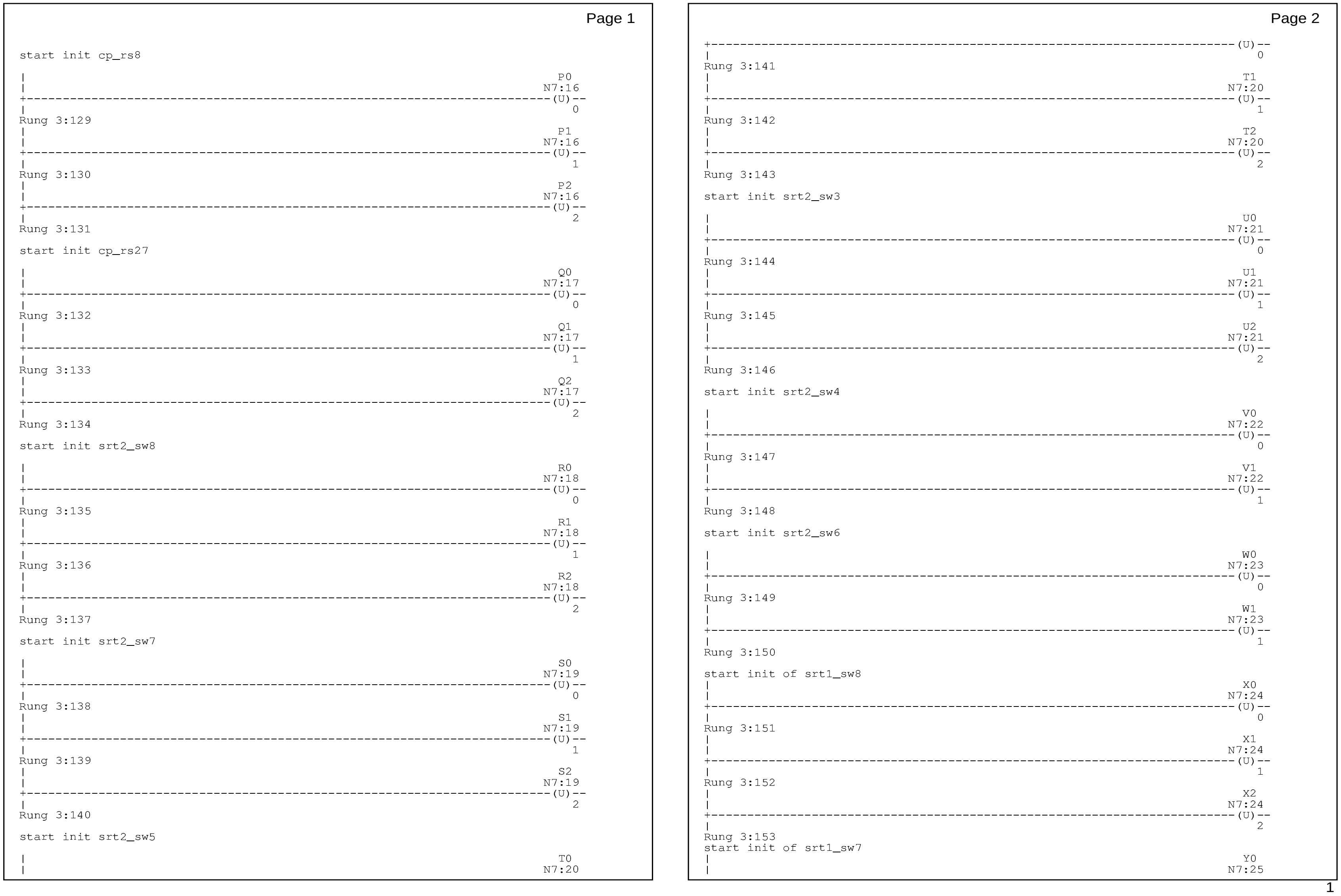
```
Rung 3:101
|                                                                         I3
|                                                                        N7:9
+-------------------------------------------------------------------------(U)--
|                                                                            3
Rung 3:102
|                                                                         I4
|                                                                        N7:9
+-------------------------------------------------------------------------(U)--
|                                                                            4
Rung 3:103
|                                                                         I5
|                                                                        N7:9
+-------------------------------------------------------------------------(U)--
|                                                                            5
Rung 3:104

start init cp_2_4

|                                                                         J0
|                                                                       N7:10
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:105
|                                                                         J1
|                                                                       N7:10
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:106
|                                                                         J2
|                                                                       N7:10
+-------------------------------------------------------------------------(U)--
|                                                                            2
Rung 3:107

start init cp_6_7

|                                                                         K0
|                                                                       N7:11
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:108
|                                                                         K1
|                                                                       N7:11
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:109
|                                                                         K2
|                                                                       N7:11
+-------------------------------------------------------------------------(U)--
|                                                                            2
Rung 3:110
|                                                                         K3
|                                                                       N7:11
+-------------------------------------------------------------------------(U)--
|                                                                            3
Rung 3:111
|                                                                         K4
|                                                                       N7:11
+-------------------------------------------------------------------------(U)--
|                                                                            4
Rung 3:112
|                                                                         K5
|                                                                       N7:11
+-------------------------------------------------------------------------(U)--
|                                                                            5
Rung 3:113

start init cp_9_10

|                                                                         L0
|                                                                       N7:12
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:114
|                                                                         L1
|                                                                       N7:12
```



Page 2

```
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:115
|                                                                         L2
|                                                                       N7:12
+-------------------------------------------------------------------------(U)--
|                                                                            2
Rung 3:116
|                                                                         L3
|                                                                       N7:12
+-------------------------------------------------------------------------(U)--
|                                                                            3
Rung 3:117
|                                                                         L4
|                                                                       N7:12
+-------------------------------------------------------------------------(U)--
|                                                                            4
Rung 3:118
|                                                                         L5
|                                                                       N7:12
+-------------------------------------------------------------------------(U)--
|                                                                            5
Rung 3:119

start init cp_11_13

|                                                                         M0
|                                                                       N7:13
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:120
|                                                                         M1
|                                                                       N7:13
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:121
|                                                                         M2
|                                                                       N7:13
+-------------------------------------------------------------------------(U)--
|                                                                            2
Rung 3:122

START INIT CP_12_14

|                                                                         N0
|                                                                       N7:14
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:123
|                                                                         N1
|                                                                       N7:14
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:124
|                                                                         N2
|                                                                       N7:14
+-------------------------------------------------------------------------(U)--
|                                                                            2
Rung 3:125

START INIT CP_3_5

|                                                                         O0
|                                                                       N7:15
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:126
|                                                                         O1
|                                                                       N7:15
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:127
|                                                                         O2
|                                                                       N7:15
+-------------------------------------------------------------------------(U)--
|                                                                            2
Rung 3:128
```

1



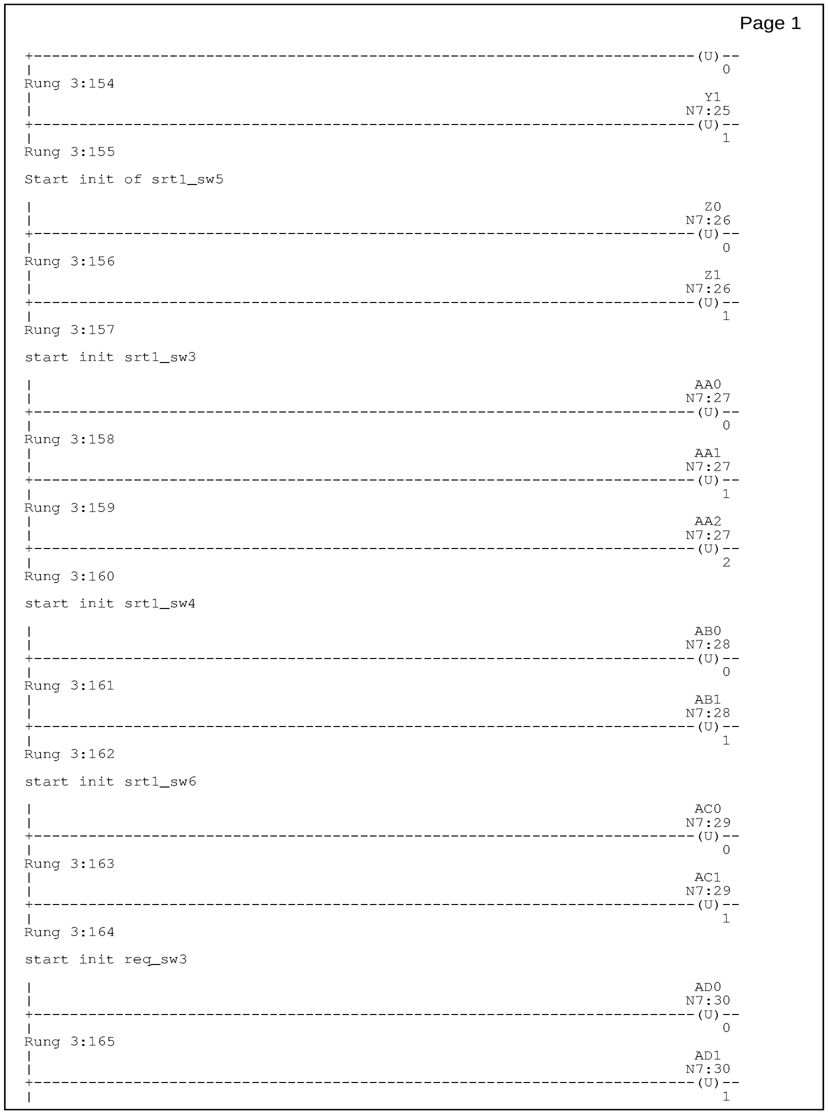
Page 1

```
start init cp_rs8

|                                                                     P0
|                                                                   N7:16
+-------------------------------------------------------------------(U)--
|                                                                      0
Rung 3:129
|                                                                     P1
|                                                                   N7:16
+-------------------------------------------------------------------(U)--
|                                                                      1
Rung 3:130
|                                                                     P2
|                                                                   N7:16
+-------------------------------------------------------------------(U)--
|                                                                      2
Rung 3:131

start init cp_rs27

|                                                                     Q0
|                                                                   N7:17
+-------------------------------------------------------------------(U)--
|                                                                      0
Rung 3:132
|                                                                     Q1
|                                                                   N7:17
+-------------------------------------------------------------------(U)--
|                                                                      1
Rung 3:133
|                                                                     Q2
|                                                                   N7:17
+-------------------------------------------------------------------(U)--
|                                                                      2
Rung 3:134

start init srt2_sw8

|                                                                     R0
|                                                                   N7:18
+-------------------------------------------------------------------(U)--
|                                                                      0
Rung 3:135
|                                                                     R1
|                                                                   N7:18
+-------------------------------------------------------------------(U)--
|                                                                      1
Rung 3:136
|                                                                     R2
|                                                                   N7:18
+-------------------------------------------------------------------(U)--
|                                                                      2
Rung 3:137

start init srt2_sw7

|                                                                     S0
|                                                                   N7:19
+-------------------------------------------------------------------(U)--
|                                                                      0
Rung 3:138
|                                                                     S1
|                                                                   N7:19
+-------------------------------------------------------------------(U)--
|                                                                      1
Rung 3:139
|                                                                     S2
|                                                                   N7:19
+-------------------------------------------------------------------(U)--
|                                                                      2
Rung 3:140

start init srt2_sw5

|                                                                     T0
|                                                                   N7:20
```

Page 2

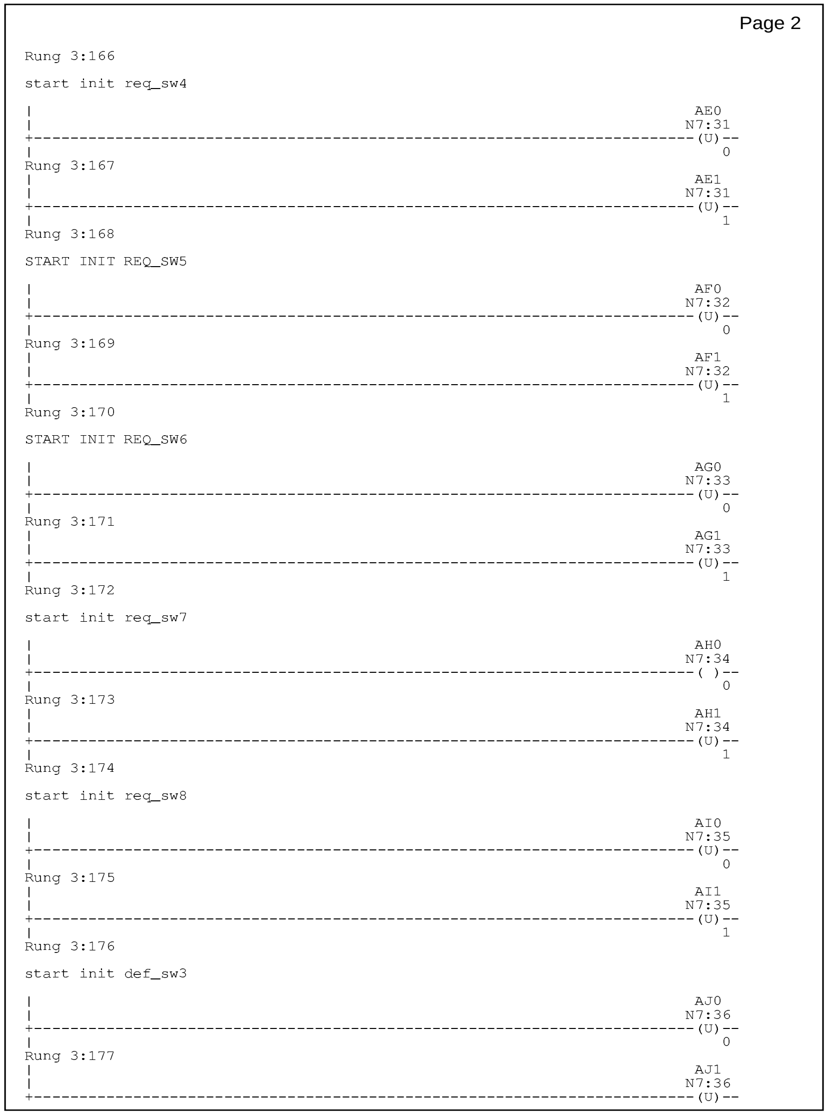
```
+-------------------------------------------------------------------(U)--
|                                                                      0
Rung 3:141
|                                                                     T1
|                                                                   N7:20
+-------------------------------------------------------------------(U)--
|                                                                      1
Rung 3:142
|                                                                     T2
|                                                                   N7:20
+-------------------------------------------------------------------(U)--
|                                                                      2
Rung 3:143

start init srt2_sw3

|                                                                     U0
|                                                                   N7:21
+-------------------------------------------------------------------(U)--
|                                                                      0
Rung 3:144
|                                                                     U1
|                                                                   N7:21
+-------------------------------------------------------------------(U)--
|                                                                      1
Rung 3:145
|                                                                     U2
|                                                                   N7:21
+-------------------------------------------------------------------(U)--
|                                                                      2
Rung 3:146

start init srt2_sw4

|                                                                     V0
|                                                                   N7:22
+-------------------------------------------------------------------(U)--
|                                                                      0
Rung 3:147
|                                                                     V1
|                                                                   N7:22
+-------------------------------------------------------------------(U)--
|                                                                      1
Rung 3:148

start init srt2_sw6

|                                                                     W0
|                                                                   N7:23
+-------------------------------------------------------------------(U)--
|                                                                      0
Rung 3:149
|                                                                     W1
|                                                                   N7:23
+-------------------------------------------------------------------(U)--
|                                                                      1
Rung 3:150

start init of srt1_sw8
|                                                                     X0
|                                                                   N7:24
+-------------------------------------------------------------------(U)--
|                                                                      0
Rung 3:151
|                                                                     X1
|                                                                   N7:24
+-------------------------------------------------------------------(U)--
|                                                                      1
Rung 3:152
|                                                                     X2
|                                                                   N7:24
+-------------------------------------------------------------------(U)--
|                                                                      2
Rung 3:153
start init of srt1_sw7
|                                                                     Y0
|                                                                   N7:25
```

1



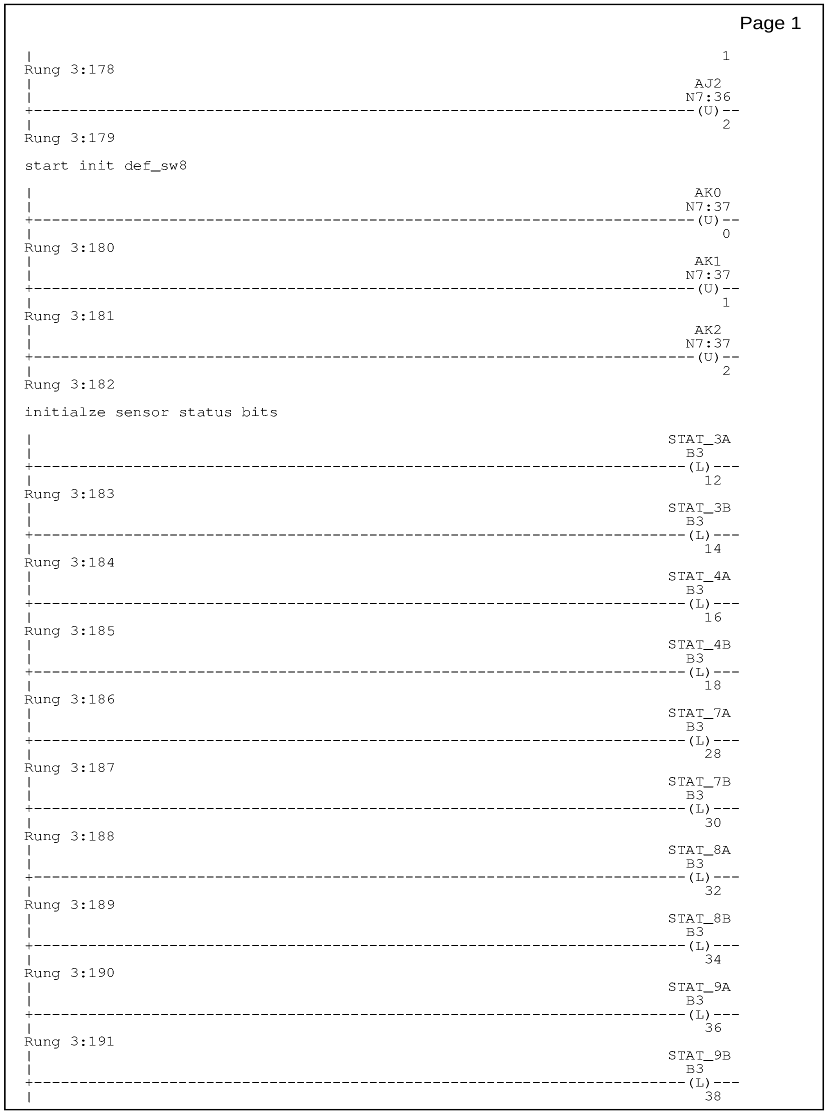
Page 1

```
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:154
|                                                                          Y1
|                                                                        N7:25
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:155

Start init of srt1_sw5

|                                                                          Z0
|                                                                        N7:26
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:156
|                                                                          Z1
|                                                                        N7:26
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:157

start init srt1_sw3

|                                                                         AA0
|                                                                        N7:27
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:158
|                                                                         AA1
|                                                                        N7:27
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:159
|                                                                         AA2
|                                                                        N7:27
+-------------------------------------------------------------------------(U)--
|                                                                            2
Rung 3:160

start init srt1_sw4

|                                                                         AB0
|                                                                        N7:28
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:161
|                                                                         AB1
|                                                                        N7:28
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:162

start init srt1_sw6

|                                                                         AC0
|                                                                        N7:29
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:163
|                                                                         AC1
|                                                                        N7:29
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:164

start init req_sw3

|                                                                         AD0
|                                                                        N7:30
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:165
|                                                                         AD1
|                                                                        N7:30
+-------------------------------------------------------------------------(U)--
|                                                                            1
```



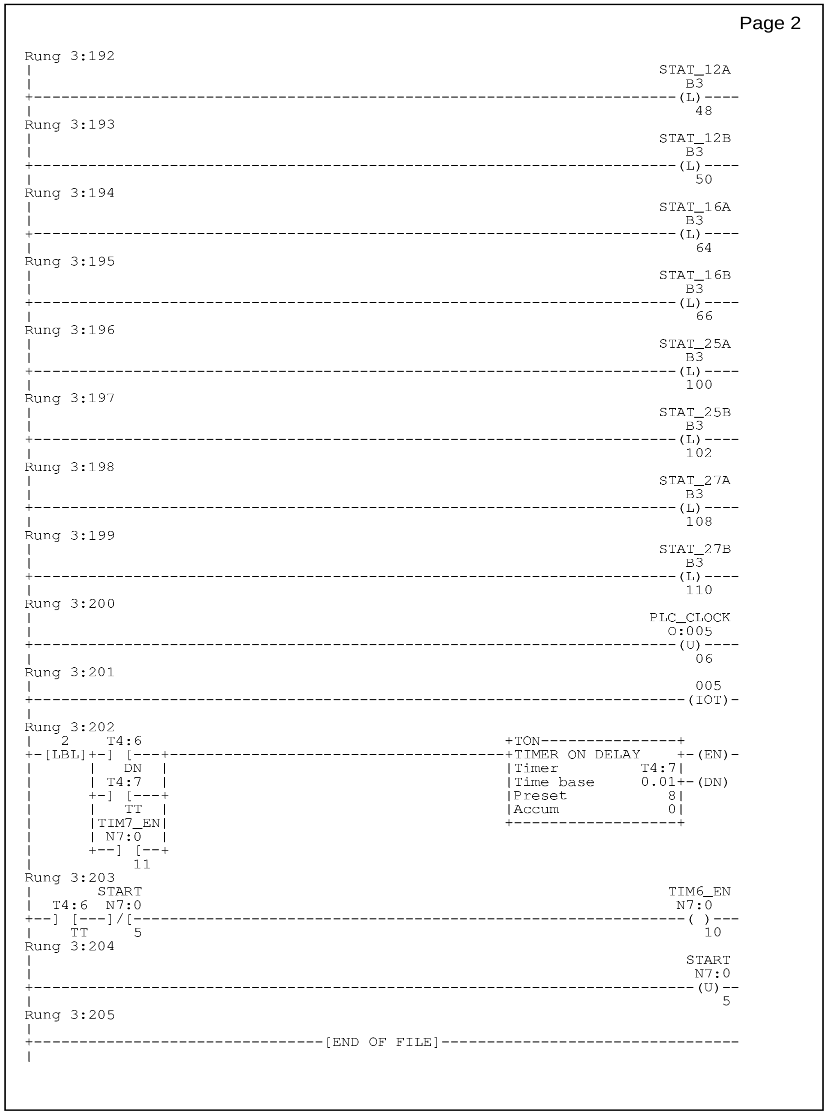
Page 2

```
Rung 3:166

start init req_sw4

|                                                                         AE0
|                                                                        N7:31
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:167
|                                                                         AE1
|                                                                        N7:31
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:168

START INIT REQ_SW5

|                                                                         AF0
|                                                                        N7:32
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:169
|                                                                         AF1
|                                                                        N7:32
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:170

START INIT REQ_SW6

|                                                                         AG0
|                                                                        N7:33
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:171
|                                                                         AG1
|                                                                        N7:33
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:172

start init req_sw7

|                                                                         AH0
|                                                                        N7:34
+-------------------------------------------------------------------------( )--
|                                                                            0
Rung 3:173
|                                                                         AH1
|                                                                        N7:34
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:174

start init req_sw8

|                                                                         AI0
|                                                                        N7:35
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:175
|                                                                         AI1
|                                                                        N7:35
+-------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:176

start init def_sw3

|                                                                         AJ0
|                                                                        N7:36
+-------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:177
|                                                                         AJ1
|                                                                        N7:36
+-------------------------------------------------------------------------(U)--
```



Page 1

```
|                                                                            1
Rung 3:178
|                                                                         AJ2
|                                                                        N7:36
+--------------------------------------------------------------------------(U)--
|                                                                            2
Rung 3:179

start init def_sw8

|                                                                         AK0
|                                                                        N7:37
+--------------------------------------------------------------------------(U)--
|                                                                            0
Rung 3:180
|                                                                         AK1
|                                                                        N7:37
+--------------------------------------------------------------------------(U)--
|                                                                            1
Rung 3:181
|                                                                         AK2
|                                                                        N7:37
+--------------------------------------------------------------------------(U)--
|                                                                            2
Rung 3:182

initialze sensor status bits

|                                                                      STAT_3A
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          12
Rung 3:183
|                                                                      STAT_3B
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          14
Rung 3:184
|                                                                      STAT_4A
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          16
Rung 3:185
|                                                                      STAT_4B
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          18
Rung 3:186
|                                                                      STAT_7A
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          28
Rung 3:187
|                                                                      STAT_7B
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          30
Rung 3:188
|                                                                      STAT_8A
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          32
Rung 3:189
|                                                                      STAT_8B
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          34
Rung 3:190
|                                                                      STAT_9A
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          36
Rung 3:191
|                                                                      STAT_9B
|                                                                        B3
+-------------------------------------------------------------------------(L)---
|                                                                          38
```



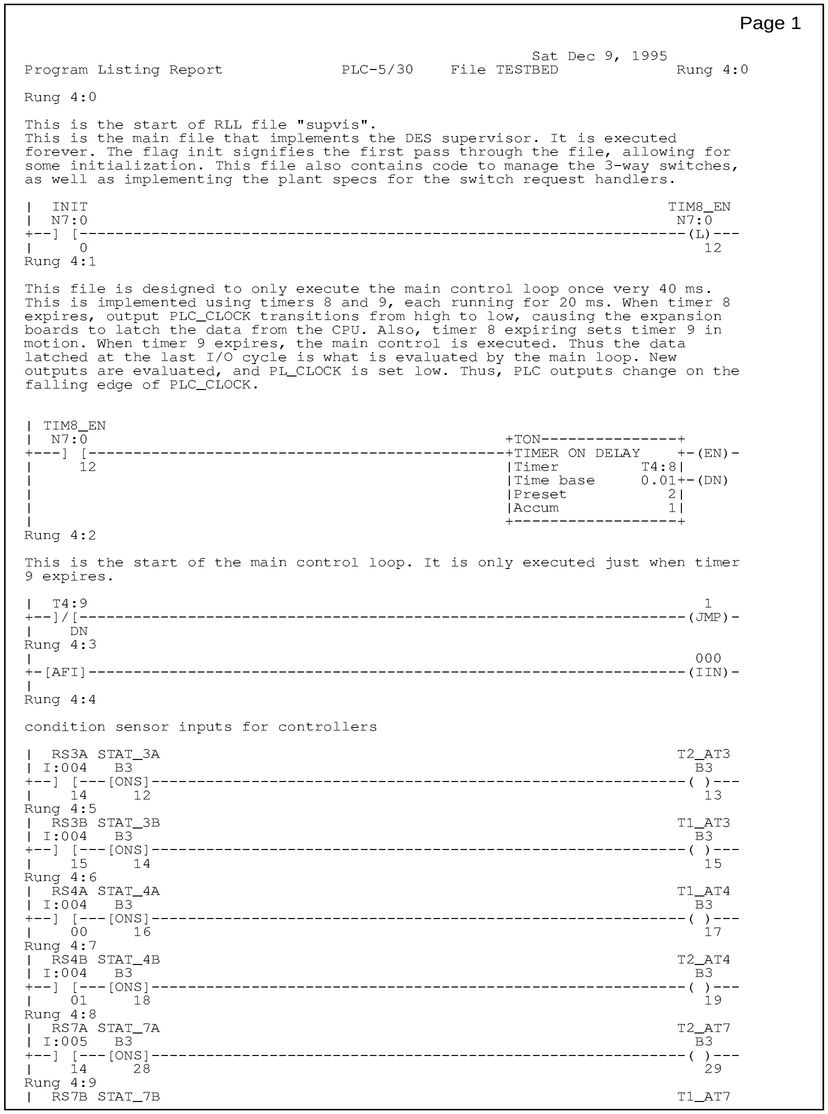
Page 2

```
Rung 3:192
|                                                                     STAT_12A
|                                                                        B3
+------------------------------------------------------------------------(L)----
|                                                                         48
Rung 3:193
|                                                                     STAT_12B
|                                                                        B3
+------------------------------------------------------------------------(L)----
|                                                                         50
Rung 3:194
|                                                                     STAT_16A
|                                                                        B3
+------------------------------------------------------------------------(L)----
|                                                                         64
Rung 3:195
|                                                                     STAT_16B
|                                                                        B3
+------------------------------------------------------------------------(L)----
|                                                                         66
Rung 3:196
|                                                                     STAT_25A
|                                                                        B3
+------------------------------------------------------------------------(L)----
|                                                                        100
Rung 3:197
|                                                                     STAT_25B
|                                                                        B3
+------------------------------------------------------------------------(L)----
|                                                                        102
Rung 3:198
|                                                                     STAT_27A
|                                                                        B3
+------------------------------------------------------------------------(L)----
|                                                                        108
Rung 3:199
|                                                                     STAT_27B
|                                                                        B3
+------------------------------------------------------------------------(L)----
|                                                                        110
Rung 3:200
|                                                                    PLC_CLOCK
|                                                                      O:005
+------------------------------------------------------------------------(U)----
|                                                                         06
Rung 3:201
|                                                                         005
+-------------------------------------------------------------------------(IOT)-
|
Rung 3:202
|   2    T4:6                                        +TON---------------+
+-[LBL]+-] [---+-------------------------------------+TIMER ON DELAY    +-(EN)-
|      |   DN  |                                     |Timer         T4:7|
|      | T4:7  |                                     |Time base     0.01+-(DN)
|      +-] [---+                                     |Preset           8|
|      |   TT  |                                     |Accum            0|
|      |TIM7_EN|                                     +------------------+
|      | N7:0  |
|      +--] [--+
|          11
Rung 3:203
|       START                                                          TIM6_EN
|  T4:6  N7:0                                                           N7:0
+--] [---]/[-------------------------------------------------------------( )---
|    TT    5                                                              10
Rung 3:204
|                                                                        START
|                                                                         N7:0
+--------------------------------------------------------------------------(U)--
|                                                                            5
Rung 3:205
|
+--------------------------------[END OF FILE]----------------------------------
|
```

1

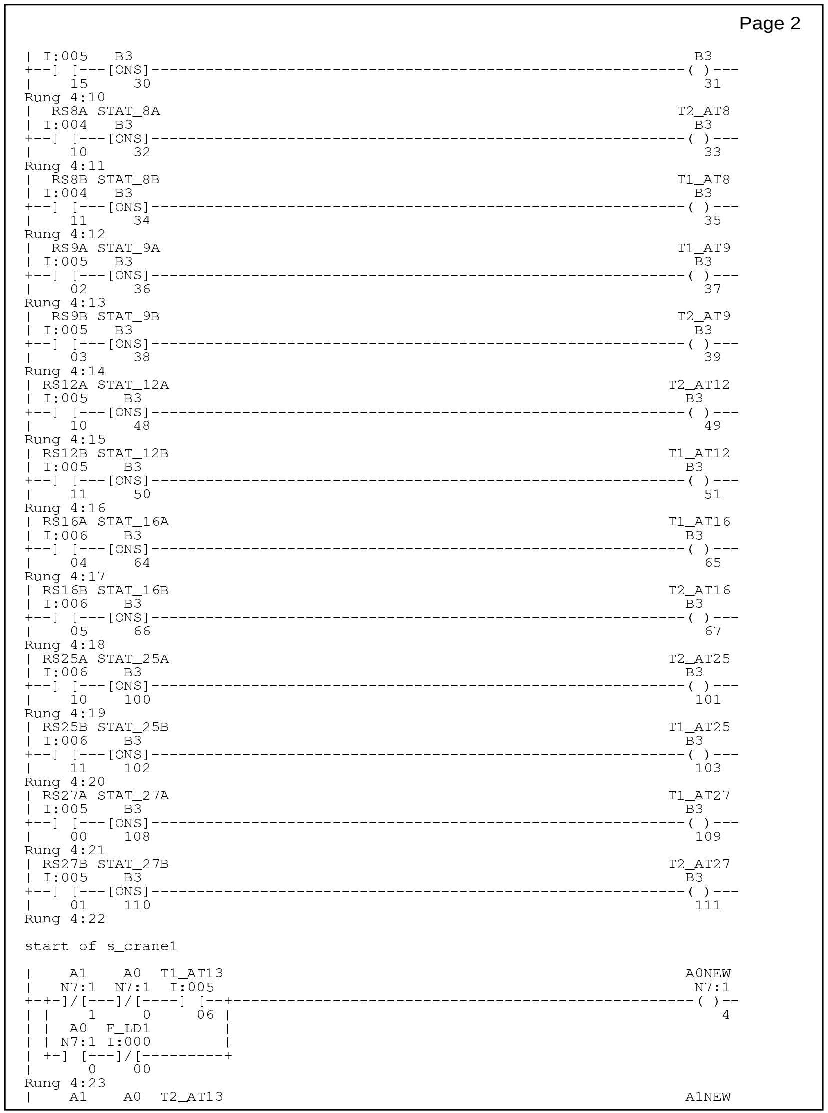
D.2.3 Supvis

The listing for the RLL program *supvis* is given below. This program emcompasses the main control loop that contains the state machines that implement the DES supervisors. This program executes forever.



Page 1

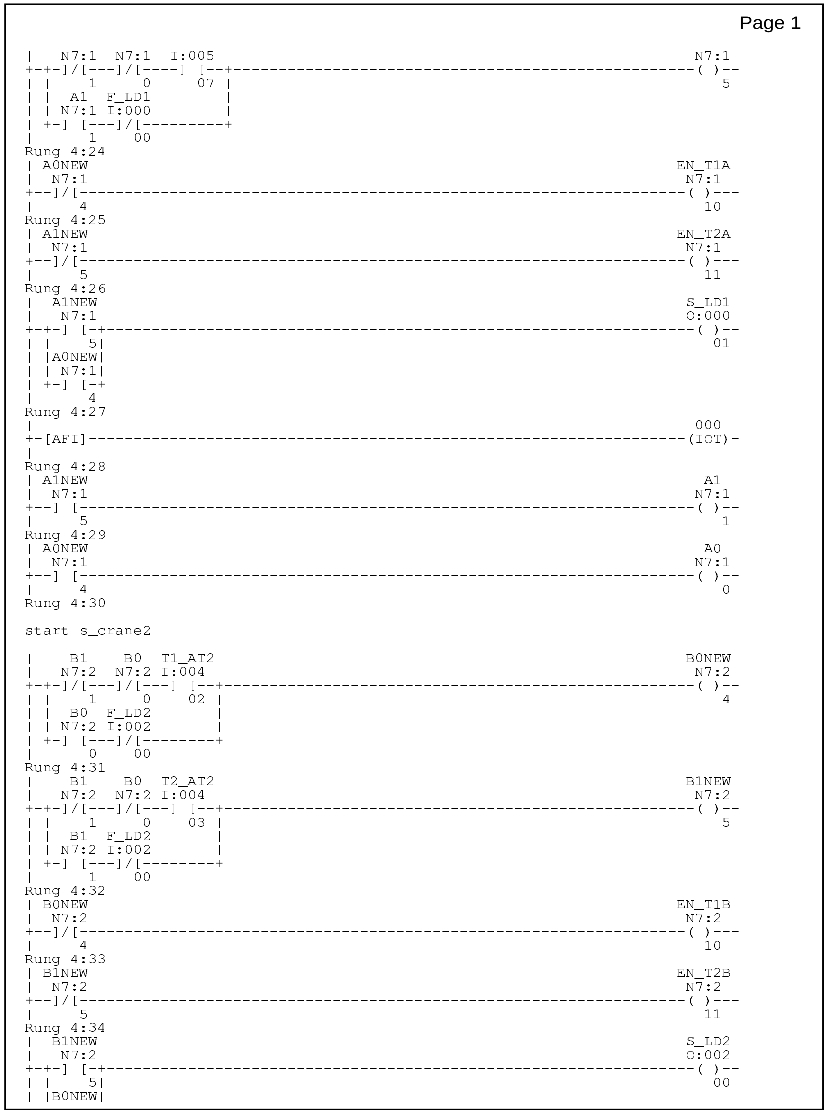
```
                                                        Sat Dec 9, 1995
Program Listing Report             PLC-5/30    File TESTBED             Rung 4:0

Rung 4:0

This is the start of RLL file "supvis".
This is the main file that implements the DES supervisor. It is executed
forever. The flag init signifies the first pass through the file, allowing for
some initialization. This file also contains code to manage the 3-way switches,
as well as implementing the plant specs for the switch request handlers.

|  INIT                                                                TIM8_EN
|  N7:0                                                                 N7:0
+--] [-----------------------------------------------------------------(L)---
|     0                                                                    12
Rung 4:1

This file is designed to only execute the main control loop once very 40 ms.
This is implemented using timers 8 and 9, each running for 20 ms. When timer 8
expires, output PLC_CLOCK transitions from high to low, causing the expansion
boards to latch the data from the CPU. Also, timer 8 expiring sets timer 9 in
motion. When timer 9 expires, the main control is executed. Thus the data
latched at the last I/O cycle is what is evaluated by the main loop. New
outputs are evaluated, and PL_CLOCK is set low. Thus, PLC outputs change on the
falling edge of PLC_CLOCK.


| TIM8_EN
|  N7:0                                                +TON---------------+
+---] [------------------------------------------------+TIMER ON DELAY    +-(EN)-
|      12                                              |Timer         T4:8|
|                                                      |Time base     0.01+-(DN)
|                                                      |Preset           2|
|                                                      |Accum            1|
|                                                      +------------------+
Rung 4:2

This is the start of the main control loop. It is only executed just when timer
9 expires.

|  T4:9                                                                    1
+--]/[-------------------------------------------------------------------(JMP)-
|    DN
Rung 4:3
|                                                                         000
+-[AFI]------------------------------------------------------------------(IIN)-
|
Rung 4:4

condition sensor inputs for controllers

|  RS3A STAT_3A                                                         T2_AT3
| I:004   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    14     12                                                             13
Rung 4:5
|  RS3B STAT_3B                                                         T1_AT3
| I:004   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    15     14                                                             15
Rung 4:6
|  RS4A STAT_4A                                                         T1_AT4
| I:004   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    00     16                                                             17
Rung 4:7
|  RS4B STAT_4B                                                         T2_AT4
| I:004   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    01     18                                                             19
Rung 4:8
|  RS7A STAT_7A                                                         T2_AT7
| I:005   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    14     28                                                             29
Rung 4:9
|  RS7B STAT_7B                                                         T1_AT7
```



Page 2

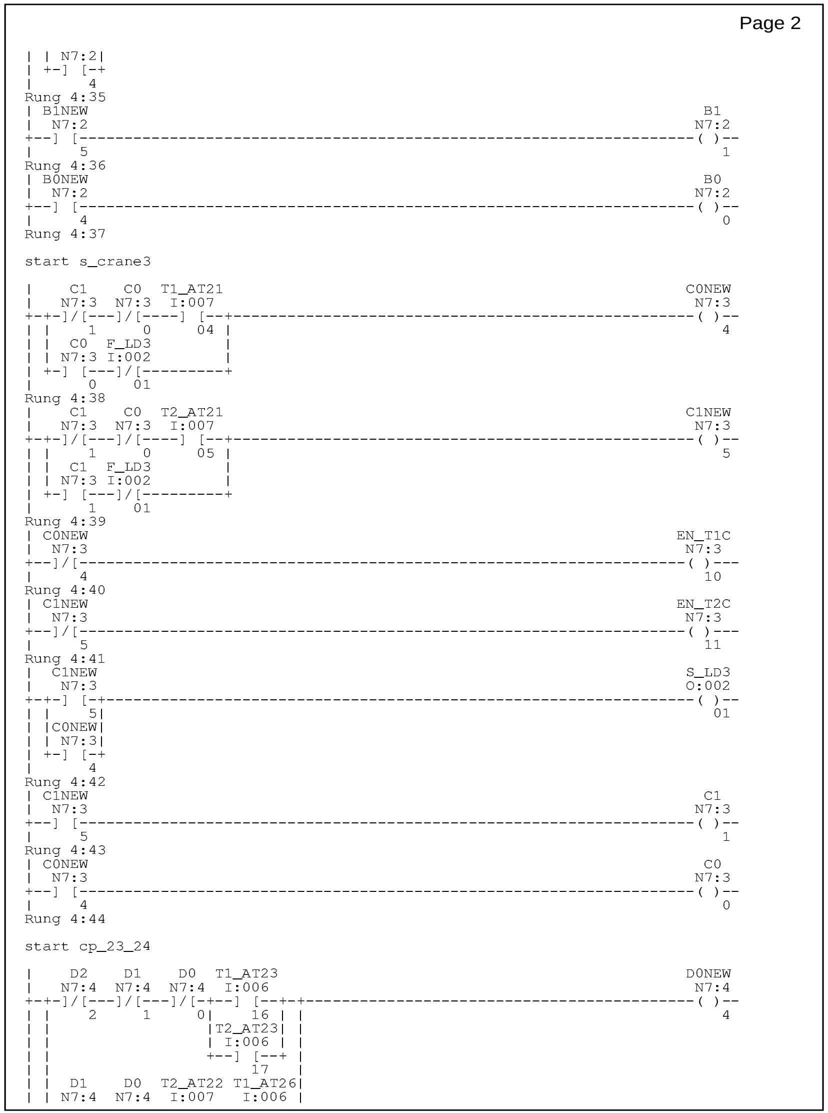
```
| I:005   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    15     30                                                             31
Rung 4:10
|  RS8A STAT_8A                                                         T2_AT8
| I:004   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    10     32                                                             33
Rung 4:11
|  RS8B STAT_8B                                                         T1_AT8
| I:004   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    11     34                                                             35
Rung 4:12
|  RS9A STAT_9A                                                         T1_AT9
| I:005   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    02     36                                                             37
Rung 4:13
|  RS9B STAT_9B                                                         T2_AT9
| I:005   B3                                                              B3
+--] [---[ONS]---------------------------------------------------------( )---
|    03     38                                                             39
Rung 4:14
| RS12A STAT_12A                                                       T2_AT12
| I:005    B3                                                            B3
+--] [---[ONS]---------------------------------------------------------( )---
|    10     48                                                             49
Rung 4:15
| RS12B STAT_12B                                                       T1_AT12
| I:005    B3                                                            B3
+--] [---[ONS]---------------------------------------------------------( )---
|    11     50                                                             51
Rung 4:16
| RS16A STAT_16A                                                       T1_AT16
| I:006    B3                                                            B3
+--] [---[ONS]---------------------------------------------------------( )---
|    04     64                                                             65
Rung 4:17
| RS16B STAT_16B                                                       T2_AT16
| I:006    B3                                                            B3
+--] [---[ONS]---------------------------------------------------------( )---
|    05     66                                                             67
Rung 4:18
| RS25A STAT_25A                                                       T2_AT25
| I:006    B3                                                            B3
+--] [---[ONS]---------------------------------------------------------( )---
|    10    100                                                            101
Rung 4:19
| RS25B STAT_25B                                                       T1_AT25
| I:006    B3                                                            B3
+--] [---[ONS]---------------------------------------------------------( )---
|    11    102                                                            103
Rung 4:20
| RS27A STAT_27A                                                       T1_AT27
| I:005    B3                                                            B3
+--] [---[ONS]---------------------------------------------------------( )---
|    00    108                                                            109
Rung 4:21
| RS27B STAT_27B                                                       T2_AT27
| I:005    B3                                                            B3
+--] [---[ONS]---------------------------------------------------------( )---
|    01    110                                                            111
Rung 4:22

start of s_crane1

|    A1    A0  T1_AT13                                                   A0NEW
|   N7:1  N7:1  I:005                                                     N7:1
+-+-]/[---]/[----] [--+-------------------------------------------------( )--
| |    1     0     06 |                                                      4
| |  A0  F_LD1        |
| | N7:1 I:000        |
| +-] [---]/[---------+
|      0    00
Rung 4:23
|    A1    A0  T2_AT13                                                   A1NEW
```

1



Page 1

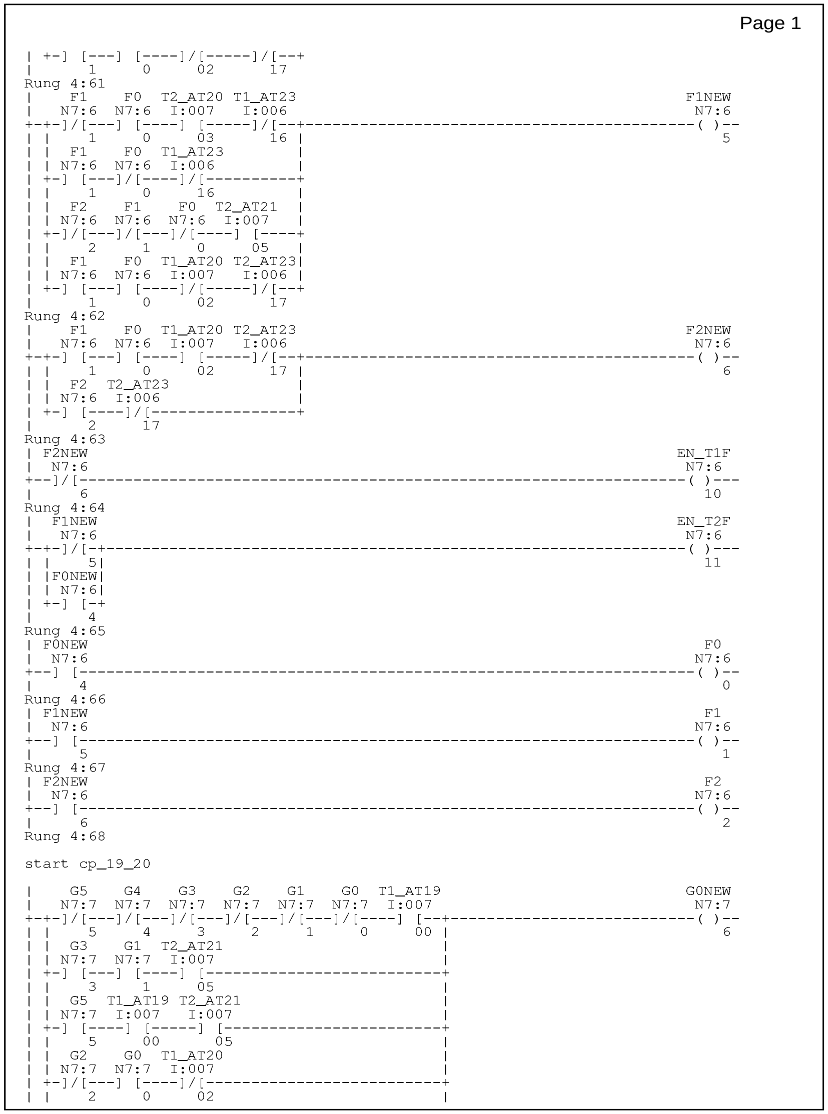
```
|   N7:1  N7:1  I:005                                                      N7:1
+-+-]/[---]/[----] [--+----------------------------------------------------( )--
| |    1     0     07 |                                                       5
| |  A1  F_LD1        |
| | N7:1 I:000        |
| +-] [---]/[---------+
|       1    00
Rung 4:24
| A0NEW                                                                  EN_T1A
|  N7:1                                                                   N7:1
+--]/[--------------------------------------------------------------------( )---
|     4                                                                     10
Rung 4:25
| A1NEW                                                                  EN_T2A
|  N7:1                                                                   N7:1
+--]/[--------------------------------------------------------------------( )---
|     5                                                                     11
Rung 4:26
|  A1NEW                                                                  S_LD1
|   N7:1                                                                  O:000
+-+-] [-+------------------------------------------------------------------( )--
| |    5|                                                                    01
| |A0NEW|
| | N7:1|
| +-] [-+
|       4
Rung 4:27
|                                                                          000
+-[AFI]-------------------------------------------------------------------(IOT)-
|
Rung 4:28
| A1NEW                                                                     A1
|  N7:1                                                                    N7:1
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:29
| A0NEW                                                                     A0
|  N7:1                                                                    N7:1
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:30

start s_crane2

|    B1    B0  T1_AT2                                                     B0NEW
|   N7:2  N7:2 I:004                                                       N7:2
+-+-]/[---]/[---] [--+-----------------------------------------------------( )--
| |    1     0    02 |                                                        4
| |  B0  F_LD2       |
| | N7:2 I:002       |
| +-] [---]/[--------+
|       0    00
Rung 4:31
|    B1    B0  T2_AT2                                                     B1NEW
|   N7:2  N7:2 I:004                                                       N7:2
+-+-]/[---]/[---] [--+-----------------------------------------------------( )--
| |    1     0    03 |                                                        5
| |  B1  F_LD2       |
| | N7:2 I:002       |
| +-] [---]/[--------+
|       1    00
Rung 4:32
| B0NEW                                                                  EN_T1B
|  N7:2                                                                   N7:2
+--]/[--------------------------------------------------------------------( )---
|     4                                                                     10
Rung 4:33
| B1NEW                                                                  EN_T2B
|  N7:2                                                                   N7:2
+--]/[--------------------------------------------------------------------( )---
|     5                                                                     11
Rung 4:34
|  B1NEW                                                                  S_LD2
|   N7:2                                                                  O:002
+-+-] [-+------------------------------------------------------------------( )--
| |    5|                                                                    00
| |B0NEW|
```



Page 2

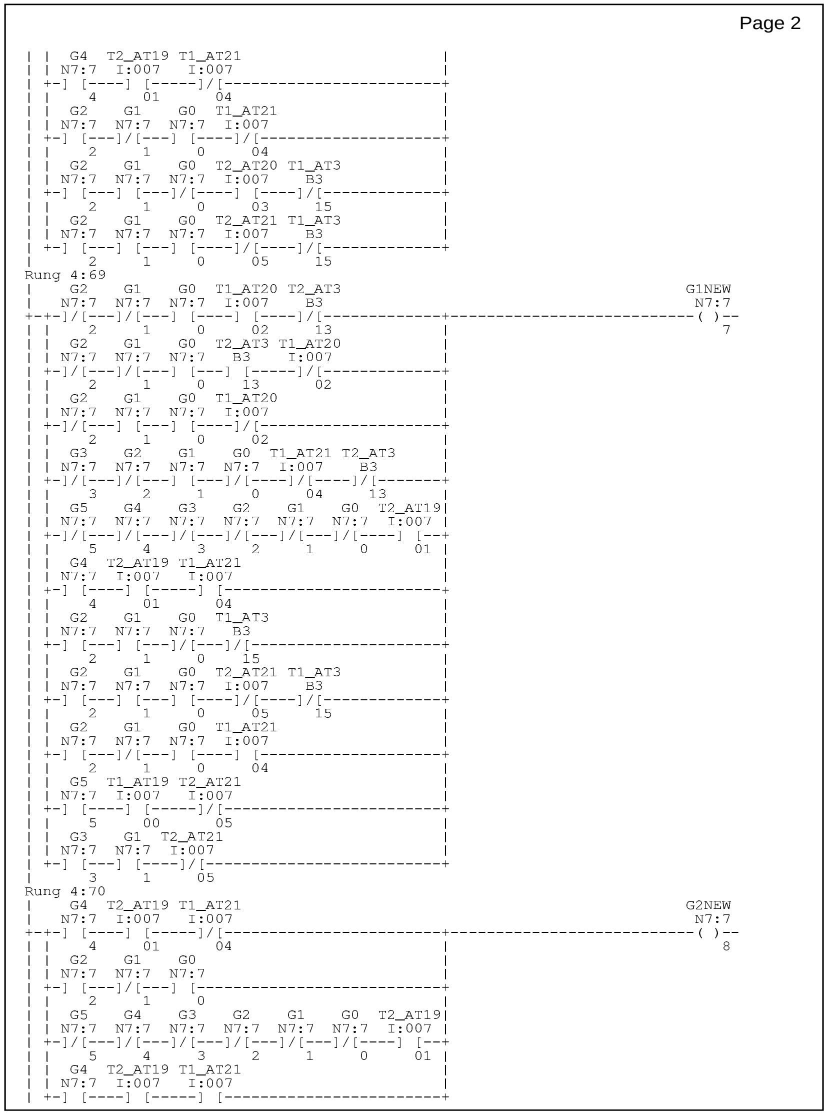
```
| | N7:2|
| +-] [-+
|       4
Rung 4:35
| B1NEW                                                                     B1
|  N7:2                                                                    N7:2
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:36
| B0NEW                                                                     B0
|  N7:2                                                                    N7:2
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:37

start s_crane3

|    C1    C0  T1_AT21                                                    C0NEW
|   N7:3  N7:3  I:007                                                      N7:3
+-+-]/[---]/[----] [--+----------------------------------------------------( )--
| |    1     0     04 |                                                       4
| |  C0  F_LD3        |
| | N7:3 I:002        |
| +-] [---]/[---------+
|       0    01
Rung 4:38
|    C1    C0  T2_AT21                                                    C1NEW
|   N7:3  N7:3  I:007                                                      N7:3
+-+-]/[---]/[----] [--+----------------------------------------------------( )--
| |    1     0     05 |                                                       5
| |  C1  F_LD3        |
| | N7:3 I:002        |
| +-] [---]/[---------+
|       1    01
Rung 4:39
| C0NEW                                                                  EN_T1C
|  N7:3                                                                   N7:3
+--]/[--------------------------------------------------------------------( )---
|     4                                                                     10
Rung 4:40
| C1NEW                                                                  EN_T2C
|  N7:3                                                                   N7:3
+--]/[--------------------------------------------------------------------( )---
|     5                                                                     11
Rung 4:41
|  C1NEW                                                                  S_LD3
|   N7:3                                                                  O:002
+-+-] [-+------------------------------------------------------------------( )--
| |    5|                                                                    01
| |C0NEW|
| | N7:3|
| +-] [-+
|       4
Rung 4:42
| C1NEW                                                                     C1
|  N7:3                                                                    N7:3
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:43
| C0NEW                                                                     C0
|  N7:3                                                                    N7:3
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:44

start cp_23_24

|    D2    D1    D0  T1_AT23                                              D0NEW
|   N7:4  N7:4  N7:4  I:006                                                N7:4
+-+-]/[---]/[---]/[-+--] [--+-+--------------------------------------------( )--
| |    2     1     0|    16 | |                                               4
| |                 |T2_AT23| |
| |                 | I:006 | |
| |                 +--] [--+ |
| |                      17   |
| |  D1    D0  T2_AT22 T1_AT26|
| | N7:4  N7:4  I:007   I:006 |
| +-]/[---] [----]/[-----]/[--+
| |    1     0    06      13 |
| |  D1    D0  T1_AT22 T2_AT26|
| | N7:4  N7:4  I:007   I:006 |
| +-] [---] [----]/[-----]/[--+
|      1     0    07      12
Rung 4:45
|    D1    D0  T2_AT22 T1_AT26                                          D1NEW
|   N7:4  N7:4  I:007   I:006                                            N7:4
+-+-]/[---] [----] [-----]/[--+-------------------------------------------( )--
| |    1     0    06      13 |                                              5
| |  D1    D0  T1_AT26        |
| | N7:4  N7:4  I:006         |
| +-] [---]/[----]/[----------+
| |    1     0    13          |
| |  D2    D1    D0  T2_AT23  |
| | N7:4  N7:4  N7:4  I:006   |
| +-]/[---]/[---]/[----] [----+
| |    2     1     0     17   |
| |  D1    D0  T1_AT22 T2_AT26|
| | N7:4  N7:4  I:007   I:006 |
| +-] [---] [----]/[-----]/[--+
|      1     0    07      12
Rung 4:46
|    D1    D0  T1_AT22 T2_AT26                                          D2NEW
|   N7:4  N7:4  I:007   I:006                                            N7:4
+-+-] [---] [----] [-----]/[--+-------------------------------------------( )--
| |    1     0    07      12 |                                              6
| |  D2  T2_AT26              |
| | N7:4  I:006               |
| +-] [----]/[----------------+
|      2     12
Rung 4:47
| D2NEW                                                                 EN_T1D
|  N7:4                                                                  N7:4
+--]/[-------------------------------------------------------------------( )---
|     6                                                                   10
Rung 4:48
|  D1NEW                                                                EN_T2D
|   N7:4                                                                 N7:4
+-+-]/[-+----------------------------------------------------------------( )---
| |    5|                                                                 11
| |D0NEW|
| | N7:4|
| +-] [-+
|      4
Rung 4:49
| D0NEW                                                                    D0
|  N7:4                                                                   N7:4
+--] [--------------------------------------------------------------------( )--
|     4                                                                     0
Rung 4:50
| D1NEW                                                                    D1
|  N7:4                                                                   N7:4
+--] [--------------------------------------------------------------------( )--
|     5                                                                     1
Rung 4:51
| D2NEW                                                                    D2
|  N7:4                                                                   N7:4
+--] [--------------------------------------------------------------------( )--
|     6                                                                     2
Rung 4:52

start cp_25_26

|    E2    E1    E0  T1_AT26                                            E0NEW
|   N7:5  N7:5  N7:5  I:006                                              N7:5
+-+-]/[---]/[---]/[-+--] [--+-+-------------------------------------------( )--
| |    2     1     0|    13 | |                                             4
| |                 |T2_AT26| |
| |                 | I:006 | |
| |                 +--] [--+ |
| |                      12   |
| |  E1    E0  T2_AT24 T1_AT14|
| | N7:5  N7:5  I:006   I:006 |
| +-]/[---] [----]/[-----]/[--+
| |    1     0    14      02 |
```

```
| |  E1    E0  T1_AT24 T2_AT14|
| | N7:5  N7:5  I:006   I:006 |
| +-] [---] [----]/[-----]/[--+
|      1     0    15      03
Rung 4:53
|    E1    E0  T2_AT24 T1_AT14                                          E1NEW
|   N7:5  N7:5  I:006   I:006                                            N7:5
+-+-]/[---] [----] [-----]/[--+-------------------------------------------( )--
| |    1     0    14      02 |                                              5
| |  E1    E0  T1_AT14        |
| | N7:5  N7:5  I:006         |
| +-] [---]/[----]/[----------+
| |    1     0    02          |
| |  E2    E1    E0  T2_AT26  |
| | N7:5  N7:5  N7:5  I:006   |
| +-]/[---]/[---]/[----] [----+
| |    2     1     0     12   |
| |  E1    E0  T1_AT24 T2_AT14|
| | N7:5  N7:5  I:006   I:006 |
| +-] [---] [----]/[-----]/[--+
|      1     0    15      03
Rung 4:54
|    E1    E0  T1_AT24 T2_AT14                                          E2NEW
|   N7:5  N7:5  I:006   I:006                                            N7:5
+-+-] [---] [----] [-----]/[--+-------------------------------------------( )--
| |    1     0    15      03 |                                              6
| |  E2  T2_AT14              |
| | N7:5  I:006               |
| +-] [----]/[----------------+
|      2     03
Rung 4:55
| E2NEW                                                                 EN_T1E
|  N7:5                                                                  N7:5
+--]/[-------------------------------------------------------------------( )---
|     6                                                                   10
Rung 4:56
|  E1NEW                                                                EN_T2E
|   N7:5                                                                 N7:5
+-+-]/[-+----------------------------------------------------------------( )---
| |    5|                                                                 11
| |E0NEW|
| | N7:5|
| +-] [-+
|      4
Rung 4:57
| E0NEW                                                                    E0
|  N7:5                                                                   N7:5
+--] [--------------------------------------------------------------------( )--
|     4                                                                     0
Rung 4:58
| E1NEW                                                                    E1
|  N7:5                                                                   N7:5
+--] [--------------------------------------------------------------------( )--
|     5                                                                     1
Rung 4:59
| E2NEW                                                                    E2
|  N7:5                                                                   N7:5
+--] [--------------------------------------------------------------------( )--
|     6                                                                     2
Rung 4:60

start cp_21_22

|    F2    F1    F0  T1_AT21                                            F0NEW
|   N7:6  N7:6  N7:6  I:007                                              N7:6
+-+-]/[---]/[---]/[-+--] [--+-+-------------------------------------------( )--
| |    2     1     0|    04 | |                                             4
| |                 |T2_AT21| |
| |                 | I:007 | |
| |                 +--] [--+ |
| |                      05   |
| |  F1    F0  T2_AT20 T1_AT23|
| | N7:6  N7:6  I:007   I:006 |
| +-]/[---] [----]/[-----]/[--+
| |    1     0    03      16 |
| |  F1    F0  T1_AT20 T2_AT23|
| | N7:6  N7:6  I:007   I:006 |
```



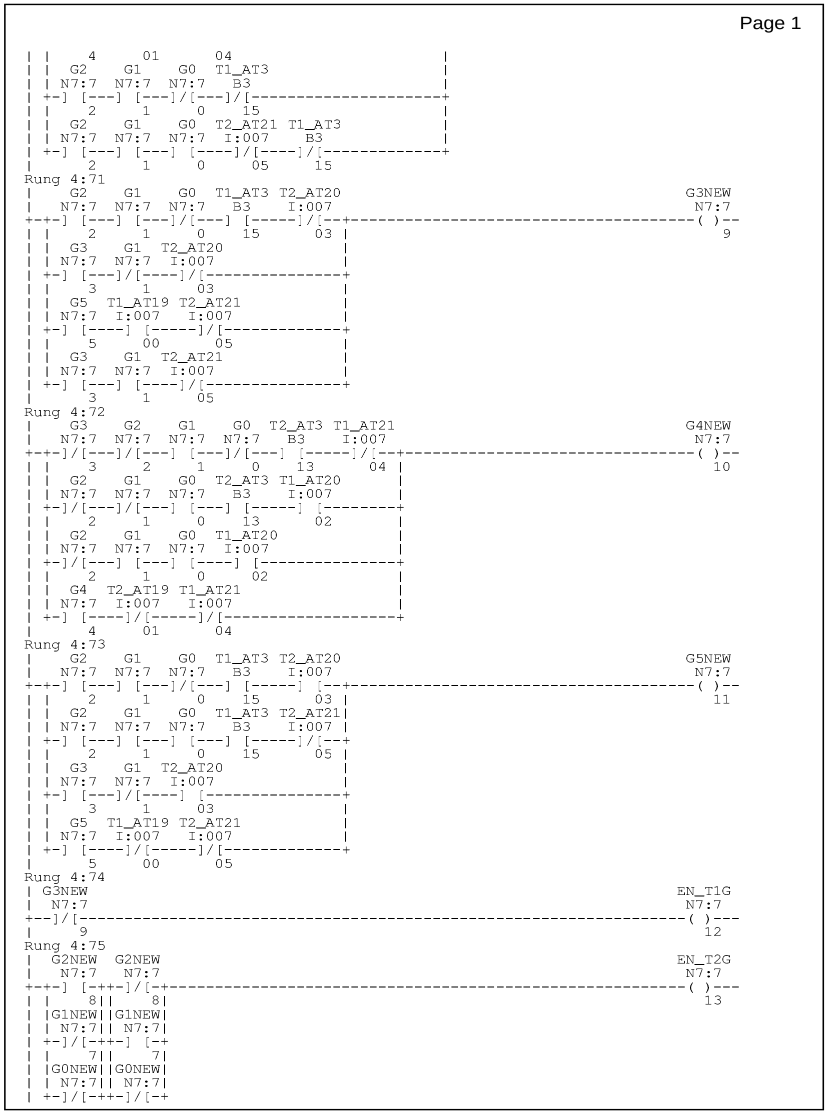
Page 1

```
| +-] [---] [----]/[-----]/[--+
|      1     0     02      17
Rung 4:61
|    F1    F0  T2_AT20 T1_AT23                                           F1NEW
|   N7:6  N7:6  I:007   I:006                                             N7:6
+-+-]/[---] [----] [-----]/[--+-------------------------------------------( )--
| |     1     0     03      16 |                                              5
| |  F1    F0  T1_AT23        |
| | N7:6  N7:6  I:006         |
| +-] [---]/[----]/[----------+
| |     1     0     16         |
| |  F2    F1    F0  T2_AT21  |
| | N7:6  N7:6  N7:6  I:007   |
| +-]/[---]/[---]/[----] [----+
| |     2     1     0     05   |
| |  F1    F0  T1_AT20 T2_AT23|
| | N7:6  N7:6  I:007   I:006 |
| +-] [---] [----]/[-----]/[--+
|      1     0     02      17
Rung 4:62
|    F1    F0  T1_AT20 T2_AT23                                           F2NEW
|   N7:6  N7:6  I:007   I:006                                             N7:6
+-+-] [---] [----] [-----]/[--+-------------------------------------------( )--
| |     1     0     02      17 |                                              6
| |  F2  T2_AT23              |
| | N7:6  I:006               |
| +-] [----]/[----------------+
|      2     17
Rung 4:63
| F2NEW                                                                 EN_T1F
|  N7:6                                                                  N7:6
+--]/[-------------------------------------------------------------------( )---
|     6                                                                     10
Rung 4:64
|  F1NEW                                                                EN_T2F
|   N7:6                                                                 N7:6
+-+-]/[-+----------------------------------------------------------------( )---
| |     5|                                                                  11
| |F0NEW|
| | N7:6|
| +-] [-+
|      4
Rung 4:65
| F0NEW                                                                    F0
|  N7:6                                                                   N7:6
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:66
| F1NEW                                                                    F1
|  N7:6                                                                   N7:6
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:67
| F2NEW                                                                    F2
|  N7:6                                                                   N7:6
+--] [--------------------------------------------------------------------( )--
|     6                                                                      2
Rung 4:68

start cp_19_20

|    G5    G4    G3    G2    G1    G0  T1_AT19                           G0NEW
|   N7:7  N7:7  N7:7  N7:7  N7:7  N7:7  I:007                             N7:7
+-+-]/[---]/[---]/[---]/[---]/[---]/[----] [--+---------------------------( )--
| |     5     4     3     2     1     0     00 |                              6
| |  G3    G1  T2_AT21                        |
| | N7:7  N7:7  I:007                         |
| +-] [---] [----] [--------------------------+
| |     3     1     05                        |
| |  G5  T1_AT19 T2_AT21                      |
| | N7:7  I:007   I:007                       |
| +-] [----] [-----] [------------------------+
| |     5     00      05                      |
| |  G2    G0  T1_AT20                        |
| | N7:7  N7:7  I:007                         |
| +-]/[---] [----]/[--------------------------+
| |     2     0     02                        |
```



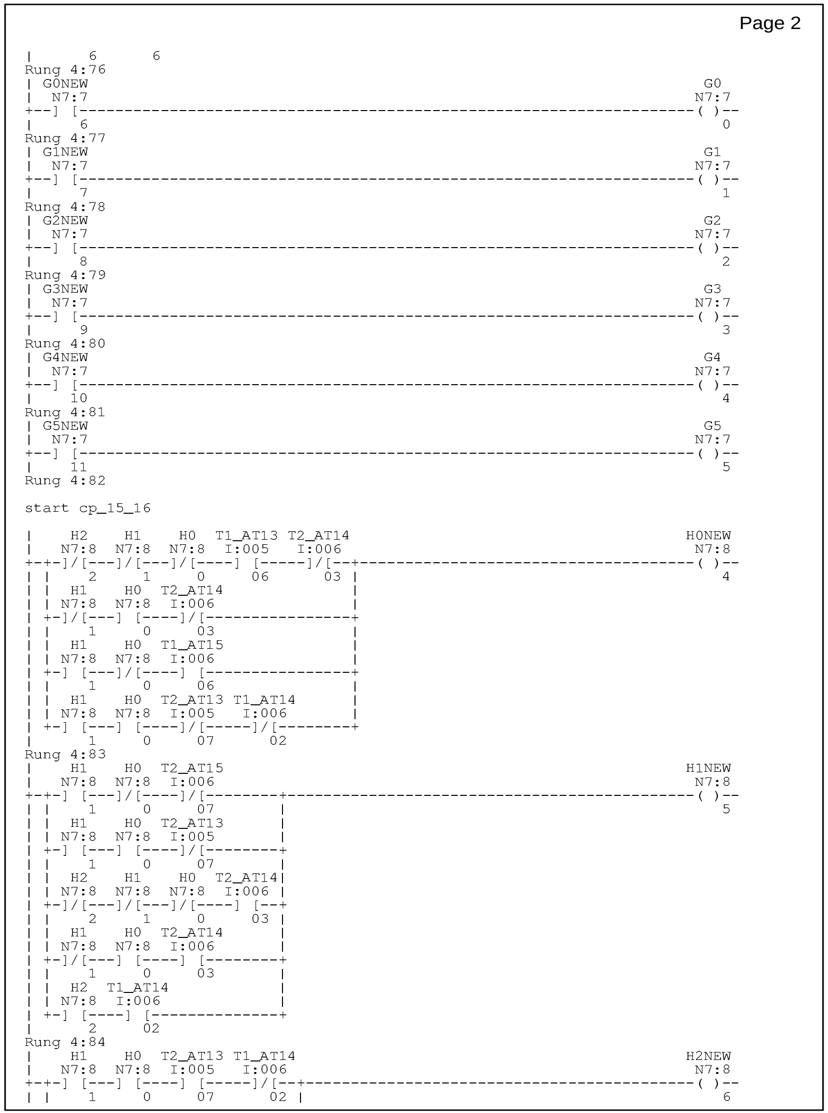
Page 2

```
| |  G4  T2_AT19 T1_AT21                      |
| | N7:7  I:007   I:007                       |
| +-] [----] [-----]/[------------------------+
| |     4     01      04                      |
| |  G2    G1    G0  T1_AT21                  |
| | N7:7  N7:7  N7:7  I:007                   |
| +-] [---]/[---] [----]/[--------------------+
| |     2     1     0     04                  |
| |  G2    G1    G0  T2_AT20 T1_AT3           |
| | N7:7  N7:7  N7:7  I:007    B3             |
| +-] [---] [---]/[----] [----]/[-------------+
| |     2     1     0     03     15            |
| |  G2    G1    G0  T2_AT21 T1_AT3           |
| | N7:7  N7:7  N7:7  I:007    B3             |
| +-] [---] [---] [----]/[----]/[-------------+
|      2     1     0     05     15
Rung 4:69
|    G2    G1    G0  T1_AT20 T2_AT3                                      G1NEW
|   N7:7  N7:7  N7:7  I:007    B3                                         N7:7
+-+-]/[---]/[---] [----] [----]/[-------------+---------------------------( )--
| |     2     1     0     02     13            |                              7
| |  G2    G1    G0  T2_AT3 T1_AT20           |
| | N7:7  N7:7  N7:7   B3    I:007            |
| +-]/[---]/[---] [---] [-----]/[-------------+
| |     2     1     0    13     02            |
| |  G2    G1    G0  T1_AT20                  |
| | N7:7  N7:7  N7:7  I:007                   |
| +-]/[---] [---] [----]/[--------------------+
| |     2     1     0     02                  |
| |  G3    G2    G1    G0  T1_AT21 T2_AT3     |
| | N7:7  N7:7  N7:7  N7:7  I:007    B3       |
| +-]/[---]/[---] [---]/[----]/[----]/[-------+
| |     3     2     1     0     04     13      |
| |  G5    G4    G3    G2    G1    G0  T2_AT19|
| | N7:7  N7:7  N7:7  N7:7  N7:7  N7:7  I:007 |
| +-]/[---]/[---]/[---]/[---]/[---]/[----] [--+
| |     5     4     3     2     1     0     01 |
| |  G4  T2_AT19 T1_AT21                      |
| | N7:7  I:007   I:007                       |
| +-] [----] [-----] [------------------------+
| |     4     01      04                      |
| |  G2    G1    G0  T1_AT3                   |
| | N7:7  N7:7  N7:7   B3                     |
| +-] [---] [---]/[---]/[---------------------+
| |     2     1     0    15                   |
| |  G2    G1    G0  T2_AT21 T1_AT3           |
| | N7:7  N7:7  N7:7  I:007    B3             |
| +-] [---] [---] [----]/[----]/[-------------+
| |     2     1     0     05     15            |
| |  G2    G1    G0  T1_AT21                  |
| | N7:7  N7:7  N7:7  I:007                   |
| +-] [---]/[---] [----] [--------------------+
| |     2     1     0     04                  |
| |  G5  T1_AT19 T2_AT21                      |
| | N7:7  I:007   I:007                       |
| +-] [----] [-----]/[------------------------+
| |     5     00      05                      |
| |  G3    G1  T2_AT21                        |
| | N7:7  N7:7  I:007                         |
| +-] [---] [----]/[--------------------------+
|      3     1     05
Rung 4:70
|    G4  T2_AT19 T1_AT21                                                 G2NEW
|   N7:7  I:007   I:007                                                   N7:7
+-+-] [----] [-----]/[------------------------+---------------------------( )--
| |     4     01      04                      |                              8
| |  G2    G1    G0                           |
| | N7:7  N7:7  N7:7                          |
| +-] [---]/[---] [---------------------------+
| |     2     1     0                         |
| |  G5    G4    G3    G2    G1    G0  T2_AT19|
| | N7:7  N7:7  N7:7  N7:7  N7:7  N7:7  I:007 |
| +-]/[---]/[---]/[---]/[---]/[---]/[----] [--+
| |     5     4     3     2     1     0     01 |
| |  G4  T2_AT19 T1_AT21                      |
| | N7:7  I:007   I:007                       |
| +-] [----] [-----] [------------------------+
```



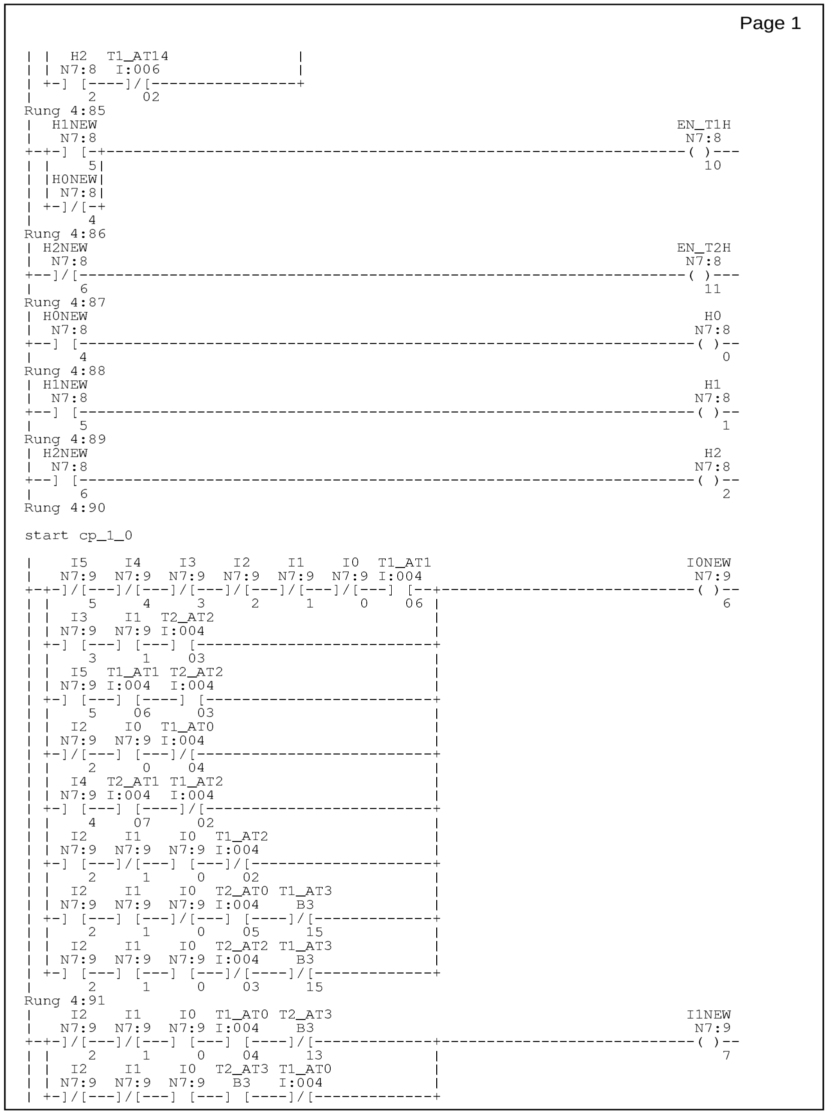
Page 1

```
| |    4     01      04                        |
| |  G2    G1    G0  T1_AT3                    |
| | N7:7  N7:7  N7:7   B3                      |
| +-] [---] [---]/[---]/[----------------------+
| |    2     1     0    15                     |
| |  G2    G1    G0  T2_AT21 T1_AT3            |
| | N7:7  N7:7  N7:7  I:007    B3              |
| +-] [---] [---] [----]/[----]/[--------------+
|      2     1     0     05     15
Rung 4:71
|    G2    G1    G0  T1_AT3 T2_AT20                                       G3NEW
|   N7:7  N7:7  N7:7   B3    I:007                                         N7:7
+-+-] [---] [---]/[---] [-----]/[--+---------------------------------------( )--
| |    2     1     0    15      03 |                                         9
| |  G3    G1  T2_AT20             |
| | N7:7  N7:7  I:007              |
| +-] [---]/[----]/[---------------+
| |    3     1     03              |
| |  G5  T1_AT19 T2_AT21           |
| | N7:7  I:007   I:007            |
| +-] [----] [-----]/[-------------+
| |    5     00      05            |
| |  G3    G1  T2_AT21             |
| | N7:7  N7:7  I:007              |
| +-] [---] [----]/[---------------+
|      3     1     05
Rung 4:72
|    G3    G2    G1    G0   T2_AT3 T1_AT21                                G4NEW
|   N7:7  N7:7  N7:7  N7:7   B3    I:007                                  N7:7
+-+-]/[---]/[---] [---]/[---] [------]/[--+--------------------------------( )--
| |    3     2     1     0    13       04 |                                 10
| |  G2    G1    G0  T2_AT3 T1_AT20       |
| | N7:7  N7:7  N7:7   B3    I:007        |
| +-]/[---]/[---] [---] [-----] [---------+
| |    2     1     0    13      02        |
| |  G2    G1    G0  T1_AT20              |
| | N7:7  N7:7  N7:7  I:007               |
| +-]/[---] [---] [----] [----------------+
| |    2     1     0     02               |
| |  G4  T2_AT19 T1_AT21                  |
| | N7:7  I:007   I:007                   |
| +-] [----]/[-----]/[--------------------+
|      4     01      04
Rung 4:73
|    G2    G1    G0  T1_AT3 T2_AT20                                       G5NEW
|   N7:7  N7:7  N7:7   B3    I:007                                         N7:7
+-+-] [---] [---]/[---] [-----] [--+---------------------------------------( )--
| |    2     1     0    15      03 |                                        11
| |  G2    G1    G0  T1_AT3 T2_AT21|
| | N7:7  N7:7  N7:7   B3    I:007 |
| +-] [---] [---] [---] [-----]/[--+
| |    2     1     0    15      05 |
| |  G3    G1  T2_AT20             |
| | N7:7  N7:7  I:007              |
| +-] [---]/[----] [---------------+
| |    3     1     03              |
| |  G5  T1_AT19 T2_AT21           |
| | N7:7  I:007   I:007            |
| +-] [----]/[-----]/[-------------+
|      5     00      05
Rung 4:74
| G3NEW                                                                  EN_T1G
|  N7:7                                                                   N7:7
+--]/[--------------------------------------------------------------------( )---
|     9                                                                     12
Rung 4:75
|  G2NEW  G2NEW                                                          EN_T2G
|   N7:7   N7:7                                                           N7:7
+-+-] [-++-]/[-+----------------------------------------------------------( )---
| |    8||    8|                                                            13
| |G1NEW||G1NEW|
| | N7:7|| N7:7|
| +-]/[-++-] [-+
| |    7||    7|
| |G0NEW||G0NEW|
| | N7:7|| N7:7|
| +-]/[-++-]/[-+
```



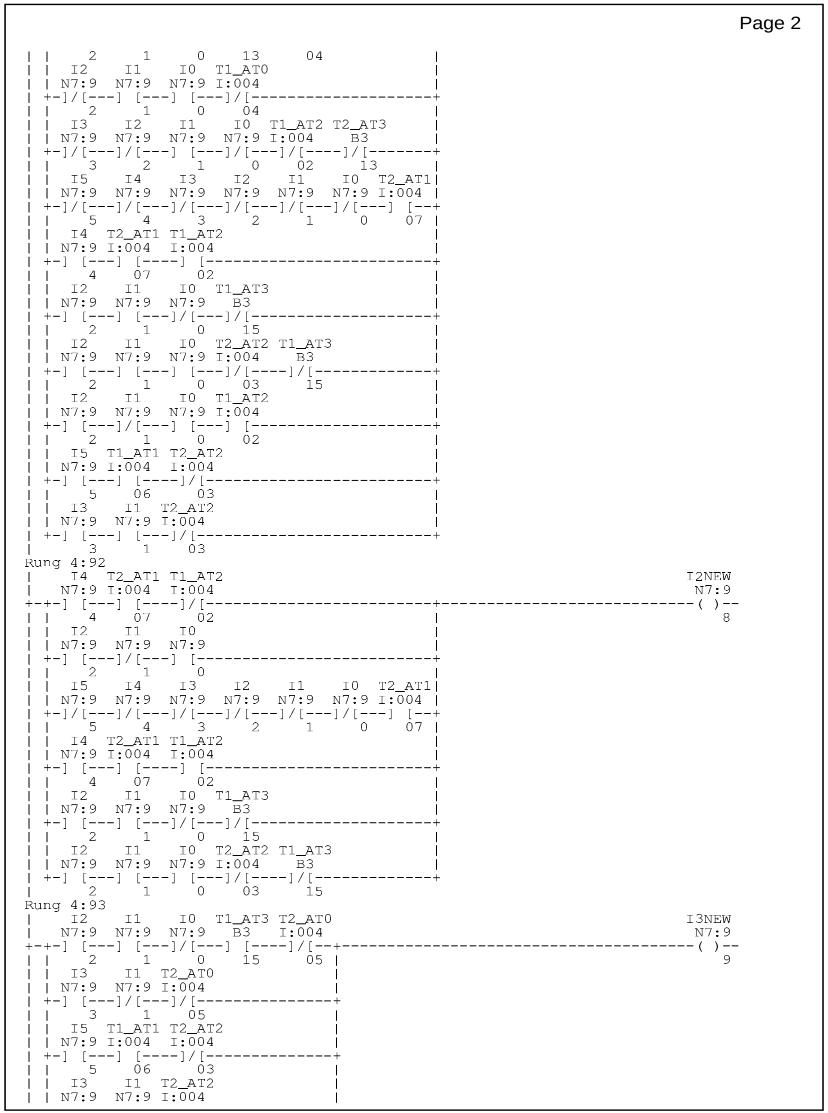
Page 2

```
|      6      6
Rung 4:76
| G0NEW                                                                     G0
|  N7:7                                                                    N7:7
+--] [---------------------------------------------------------------------( )--
|     6                                                                      0
Rung 4:77
| G1NEW                                                                     G1
|  N7:7                                                                    N7:7
+--] [---------------------------------------------------------------------( )--
|     7                                                                      1
Rung 4:78
| G2NEW                                                                     G2
|  N7:7                                                                    N7:7
+--] [---------------------------------------------------------------------( )--
|     8                                                                      2
Rung 4:79
| G3NEW                                                                     G3
|  N7:7                                                                    N7:7
+--] [---------------------------------------------------------------------( )--
|     9                                                                      3
Rung 4:80
| G4NEW                                                                     G4
|  N7:7                                                                    N7:7
+--] [---------------------------------------------------------------------( )--
|    10                                                                      4
Rung 4:81
| G5NEW                                                                     G5
|  N7:7                                                                    N7:7
+--] [---------------------------------------------------------------------( )--
|    11                                                                      5
Rung 4:82

start cp_15_16

|    H2    H1    H0  T1_AT13 T2_AT14                                     H0NEW
|   N7:8  N7:8  N7:8  I:005   I:006                                       N7:8
+-+-]/[---]/[---]/[----] [-----]/[--+--------------------------------------( )--
| |    2     1     0     06      03 |                                        4
| |  H1    H0  T2_AT14              |
| | N7:8  N7:8  I:006               |
| +-]/[---] [----]/[----------------+
| |    1     0     03               |
| |  H1    H0  T1_AT15              |
| | N7:8  N7:8  I:006               |
| +-] [---]/[----] [----------------+
| |    1     0     06               |
| |  H1    H0  T2_AT13 T1_AT14      |
| | N7:8  N7:8  I:005   I:006       |
| +-] [---] [----]/[-----]/[--------+
|      1     0     07      02
Rung 4:83
|    H1    H0  T2_AT15                                                   H1NEW
|   N7:8  N7:8  I:006                                                     N7:8
+-+-] [---]/[----]/[--------+----------------------------------------------( )--
| |    1     0     07        |                                               5
| |  H1    H0  T2_AT13       |
| | N7:8  N7:8  I:005        |
| +-] [---] [----]/[--------+
| |    1     0     07        |
| |  H2    H1    H0  T2_AT14|
| | N7:8  N7:8  N7:8  I:006 |
| +-]/[---]/[---]/[----] [--+
| |    2     1     0     03 |
| |  H1    H0  T2_AT14       |
| | N7:8  N7:8  I:006        |
| +-]/[---] [----] [--------+
| |    1     0     03        |
| |  H2  T1_AT14             |
| | N7:8  I:006              |
| +-] [----] [--------------+
|      2     02
Rung 4:84
|    H1    H0  T2_AT13 T1_AT14                                           H2NEW
|   N7:8  N7:8  I:005   I:006                                             N7:8
+-+-] [---] [----] [-----]/[--+--------------------------------------------( )--
| |    1     0     07      02 |                                              6
```



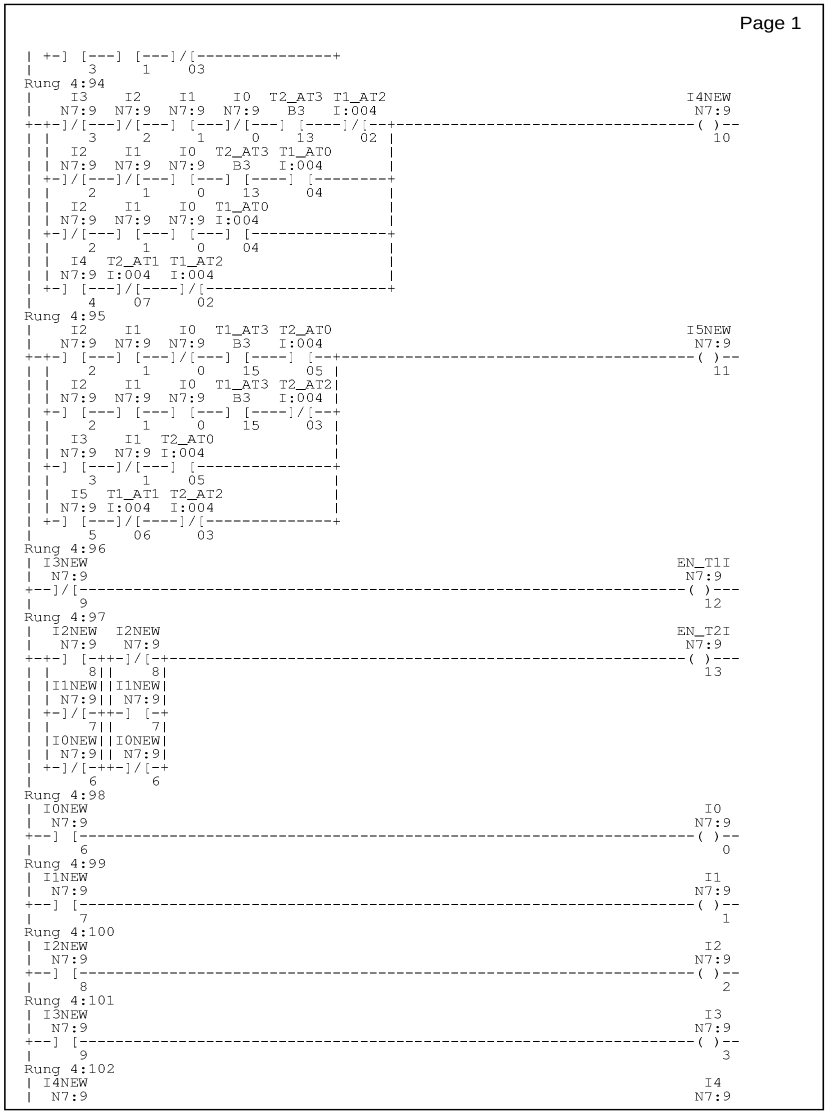
Page 1

```
| |  H2  T1_AT14              |
| | N7:8  I:006               |
| +-] [----]/[----------------+
|      2     02
Rung 4:85
|  H1NEW                                                               EN_T1H
|   N7:8                                                                N7:8
+-+-] [-+----------------------------------------------------------------( )---
| |    5|                                                                 10
| |H0NEW|
| | N7:8|
| +-]/[-+
|      4
Rung 4:86
| H2NEW                                                                EN_T2H
|  N7:8                                                                 N7:8
+--]/[-------------------------------------------------------------------( )---
|     6                                                                   11
Rung 4:87
| H0NEW                                                                   H0
|  N7:8                                                                  N7:8
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:88
| H1NEW                                                                   H1
|  N7:8                                                                  N7:8
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:89
| H2NEW                                                                   H2
|  N7:8                                                                  N7:8
+--] [--------------------------------------------------------------------( )--
|     6                                                                      2
Rung 4:90

start cp_1_0

|    I5    I4    I3    I2    I1    I0  T1_AT1                           I0NEW
|   N7:9  N7:9  N7:9  N7:9  N7:9  N7:9 I:004                             N7:9
+-+-]/[---]/[---]/[---]/[---]/[---]/[---] [--+----------------------------( )--
| |    5     4     3     2     1     0    06 |                               6
| |  I3    I1  T2_AT2                        |
| | N7:9  N7:9 I:004                         |
| +-] [---] [---] [--------------------------+
| |    3     1     03                        |
| |  I5  T1_AT1 T2_AT2                       |
| | N7:9 I:004  I:004                        |
| +-] [---] [----] [-------------------------+
| |    5    06     03                        |
| |  I2    I0  T1_AT0                        |
| | N7:9  N7:9 I:004                         |
| +-]/[---] [---]/[--------------------------+
| |    2     0    04                         |
| |  I4  T2_AT1 T1_AT2                       |
| | N7:9 I:004  I:004                        |
| +-] [---] [----]/[-------------------------+
| |    4    07     02                        |
| |  I2    I1    I0  T1_AT2                  |
| | N7:9  N7:9  N7:9 I:004                   |
| +-] [---]/[---] [---]/[--------------------+
| |    2     1     0    02                   |
| |  I2    I1    I0  T2_AT0 T1_AT3           |
| | N7:9  N7:9  N7:9 I:004    B3             |
| +-] [---] [---]/[---] [----]/[-------------+
| |    2     1     0    05      15           |
| |  I2    I1    I0  T2_AT2 T1_AT3           |
| | N7:9  N7:9  N7:9 I:004    B3             |
| +-] [---] [---] [---]/[----]/[-------------+
|      2     1     0    03      15
Rung 4:91
|    I2    I1    I0  T1_AT0 T2_AT3                                      I1NEW
|   N7:9  N7:9  N7:9 I:004    B3                                         N7:9
+-+-]/[---]/[---] [---] [----]/[-------------+----------------------------( )--
| |    2     1     0    04      13           |                               7
| |  I2    I1    I0  T2_AT3 T1_AT0           |
| | N7:9  N7:9  N7:9   B3   I:004            |
| +-]/[---]/[---] [---] [----]/[-------------+
```



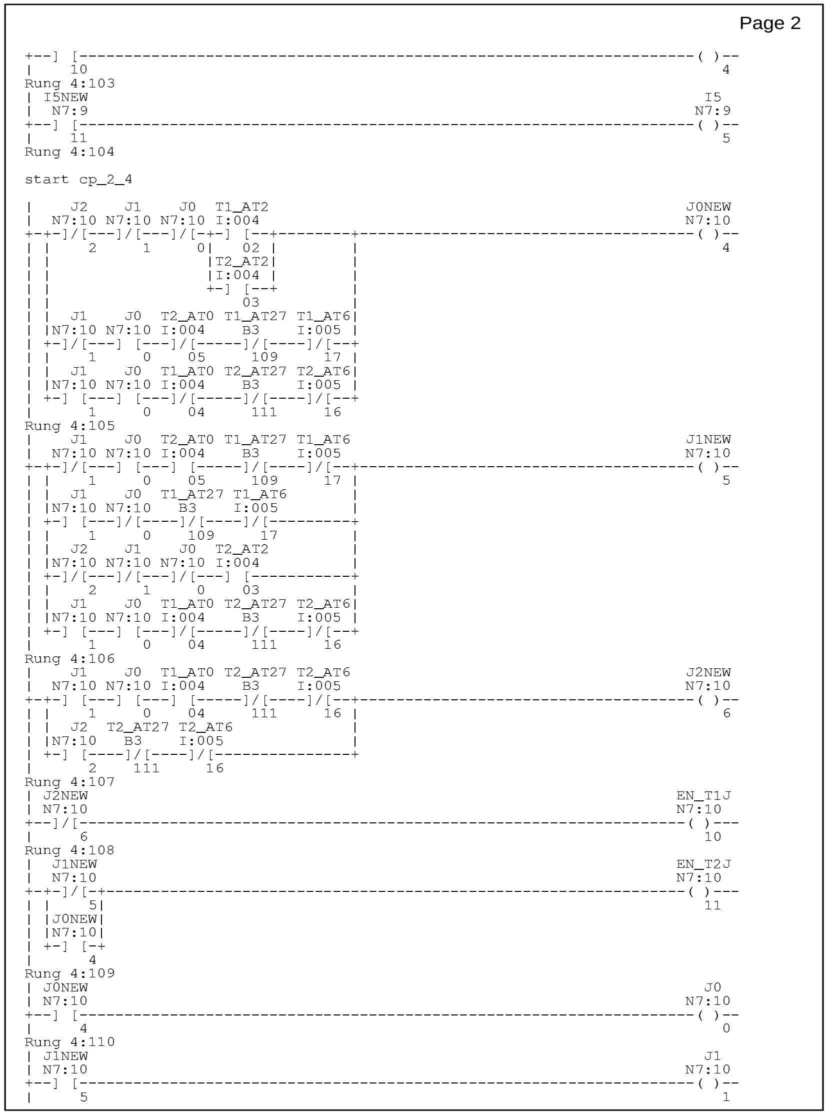
Page 2

```
| |    2     1     0    13      04           |
| |  I2    I1    I0  T1_AT0                  |
| | N7:9  N7:9  N7:9 I:004                   |
| +-]/[---] [---] [---]/[--------------------+
| |    2     1     0    04                   |
| |  I3    I2    I1    I0  T1_AT2 T2_AT3     |
| | N7:9  N7:9  N7:9  N7:9 I:004    B3       |
| +-]/[---]/[---] [---]/[---]/[----]/[-------+
| |    3     2     1     0    02      13     |
| |  I5    I4    I3    I2    I1    I0  T2_AT1|
| | N7:9  N7:9  N7:9  N7:9  N7:9  N7:9 I:004 |
| +-]/[---]/[---]/[---]/[---]/[---]/[---] [--+
| |    5     4     3     2     1     0    07 |
| |  I4  T2_AT1 T1_AT2                       |
| | N7:9 I:004  I:004                        |
| +-] [---] [----] [-------------------------+
| |    4    07     02                        |
| |  I2    I1    I0  T1_AT3                  |
| | N7:9  N7:9  N7:9   B3                    |
| +-] [---] [---]/[---]/[--------------------+
| |    2     1     0    15                   |
| |  I2    I1    I0  T2_AT2 T1_AT3           |
| | N7:9  N7:9  N7:9 I:004    B3             |
| +-] [---] [---] [---]/[----]/[-------------+
| |    2     1     0    03      15           |
| |  I2    I1    I0  T1_AT2                  |
| | N7:9  N7:9  N7:9 I:004                   |
| +-] [---]/[---] [---] [--------------------+
| |    2     1     0    02                   |
| |  I5  T1_AT1 T2_AT2                       |
| | N7:9 I:004  I:004                        |
| +-] [---] [----]/[-------------------------+
| |    5    06     03                        |
| |  I3    I1  T2_AT2                        |
| | N7:9  N7:9 I:004                         |
| +-] [---] [---]/[--------------------------+
|      3     1    03
Rung 4:92
|    I4  T2_AT1 T1_AT2                                                  I2NEW
|   N7:9 I:004  I:004                                                    N7:9
+-+-] [---] [----]/[-------------------------+----------------------------( )--
| |    4    07     02                        |                               8
| |  I2    I1    I0                          |
| | N7:9  N7:9  N7:9                         |
| +-] [---]/[---] [--------------------------+
| |    2     1     0                         |
| |  I5    I4    I3    I2    I1    I0  T2_AT1|
| | N7:9  N7:9  N7:9  N7:9  N7:9  N7:9 I:004 |
| +-]/[---]/[---]/[---]/[---]/[---]/[---] [--+
| |    5     4     3     2     1     0    07 |
| |  I4  T2_AT1 T1_AT2                       |
| | N7:9 I:004  I:004                        |
| +-] [---] [----] [-------------------------+
| |    4    07     02                        |
| |  I2    I1    I0  T1_AT3                  |
| | N7:9  N7:9  N7:9   B3                    |
| +-] [---] [---]/[---]/[--------------------+
| |    2     1     0    15                   |
| |  I2    I1    I0  T2_AT2 T1_AT3           |
| | N7:9  N7:9  N7:9 I:004    B3             |
| +-] [---] [---] [---]/[----]/[-------------+
|      2     1     0    03      15
Rung 4:93
|    I2    I1    I0  T1_AT3 T2_AT0                                      I3NEW
|   N7:9  N7:9  N7:9   B3   I:004                                        N7:9
+-+-] [---] [---]/[---] [----]/[--+---------------------------------------( )--
| |    2     1     0    15      05 |                                         9
| |  I3    I1  T2_AT0             |
| | N7:9  N7:9 I:004              |
| +-] [---]/[---]/[---------------+
| |    3     1    05              |
| |  I5  T1_AT1 T2_AT2            |
| | N7:9 I:004  I:004             |
| +-] [---] [----]/[--------------+
| |    5    06     03             |
| |  I3    I1  T2_AT2             |
| | N7:9  N7:9 I:004              |
```



Page 1

```
| +-] [---] [---]/[----------------+
|      3     1    03
Rung 4:94
|    I3    I2    I1    I0  T2_AT3 T1_AT2                                 I4NEW
|   N7:9  N7:9  N7:9  N7:9   B3   I:004                                  N7:9
+-+-]/[---]/[---] [---]/[---] [----]/[--+---------------------------------( )--
| |    3     2     1     0    13     02 |                                  10
| |  I2    I1    I0  T2_AT3 T1_AT0      |
| | N7:9  N7:9  N7:9   B3   I:004       |
| +-]/[---]/[---] [---] [----] [--------+
| |    2     1     0    13     04       |
| |  I2    I1    I0  T1_AT0             |
| | N7:9  N7:9  N7:9 I:004              |
| +-]/[---] [---] [---] [---------------+
| |    2     1     0    04              |
| |  I4  T2_AT1 T1_AT2                  |
| | N7:9 I:004  I:004                   |
| +-] [---]/[----]/[--------------------+
|      4    07     02
Rung 4:95
|    I2    I1    I0  T1_AT3 T2_AT0                                       I5NEW
|   N7:9  N7:9  N7:9   B3   I:004                                        N7:9
+-+-] [---] [---]/[---] [----] [--+---------------------------------------( )--
| |    2     1     0    15     05 |                                        11
| |  I2    I1    I0  T1_AT3 T2_AT2|
| | N7:9  N7:9  N7:9   B3   I:004 |
| +-] [---] [---] [---] [----]/[--+
| |    2     1     0    15     03 |
| |  I3    I1  T2_AT0             |
| | N7:9  N7:9 I:004              |
| +-] [---]/[---] [---------------+
| |    3     1    05              |
| |  I5  T1_AT1 T2_AT2            |
| | N7:9 I:004  I:004             |
| +-] [---]/[----]/[--------------+
|      5    06     03
Rung 4:96
| I3NEW                                                                 EN_T1I
|  N7:9                                                                  N7:9
+--]/[-------------------------------------------------------------------( )---
|     9                                                                   12
Rung 4:97
|  I2NEW  I2NEW                                                         EN_T2I
|   N7:9   N7:9                                                          N7:9
+-+-] [-++-]/[-+---------------------------------------------------------( )---
| |    8||    8|                                                          13
| |I1NEW||I1NEW|
| | N7:9|| N7:9|
| +-]/[-++-] [-+
| |    7||    7|
| |I0NEW||I0NEW|
| | N7:9|| N7:9|
| +-]/[-++-]/[-+
|      6      6
Rung 4:98
| I0NEW                                                                    I0
|  N7:9                                                                  N7:9
+--] [--------------------------------------------------------------------( )--
|     6                                                                     0
Rung 4:99
| I1NEW                                                                    I1
|  N7:9                                                                  N7:9
+--] [--------------------------------------------------------------------( )--
|     7                                                                     1
Rung 4:100
| I2NEW                                                                    I2
|  N7:9                                                                  N7:9
+--] [--------------------------------------------------------------------( )--
|     8                                                                     2
Rung 4:101
| I3NEW                                                                    I3
|  N7:9                                                                  N7:9
+--] [--------------------------------------------------------------------( )--
|     9                                                                     3
Rung 4:102
| I4NEW                                                                    I4
|  N7:9                                                                  N7:9
```



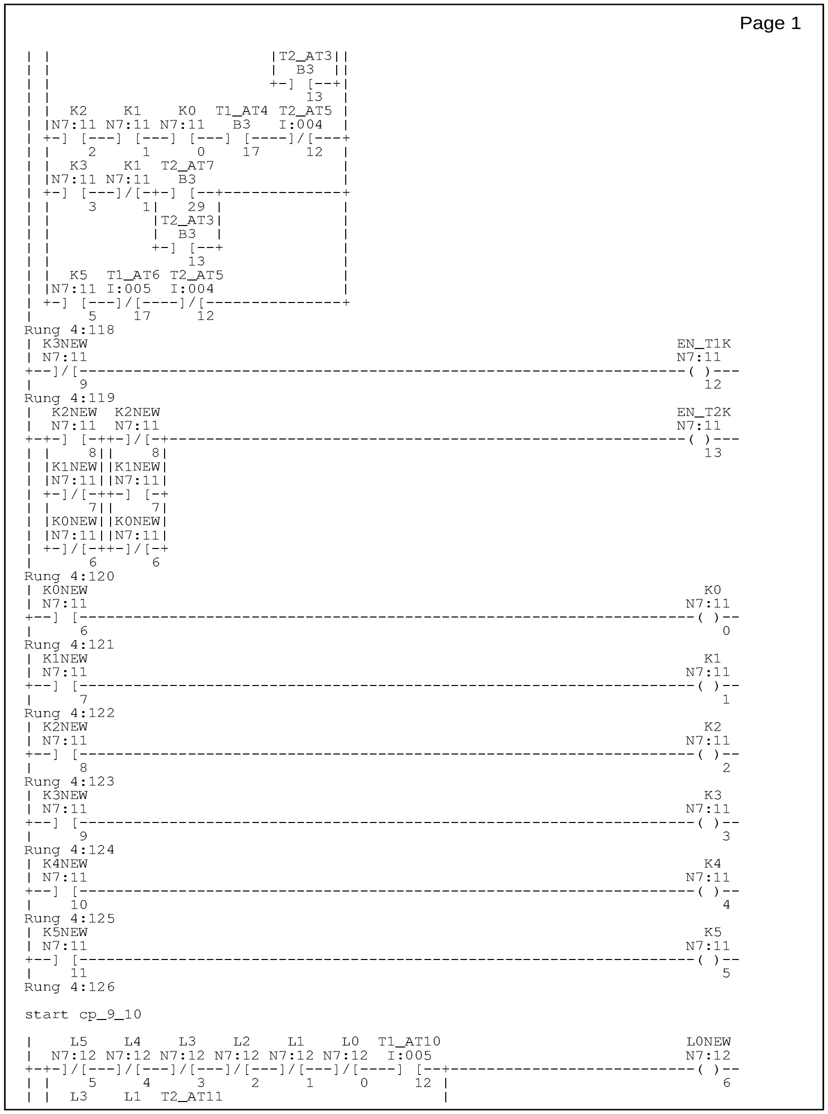
Page 2

```
+--] [--------------------------------------------------------------------( )--
|    10                                                                     4
Rung 4:103
| I5NEW                                                                    I5
|  N7:9                                                                  N7:9
+--] [--------------------------------------------------------------------( )--
|    11                                                                     5
Rung 4:104

start cp_2_4

|    J2    J1    J0  T1_AT2                                             J0NEW
|  N7:10 N7:10 N7:10 I:004                                              N7:10
+-+-]/[---]/[---]/[-+-] [--+---------+------------------------------------( )--
| |    2     1     0|  02 |         |                                       4
| |                 |T2_AT2|        |
| |                 |I:004 |        |
| |                 +-] [--+        |
| |                    03           |
| |  J1    J0  T2_AT0 T1_AT27 T1_AT6|
| |N7:10 N7:10 I:004    B3    I:005 |
| +-]/[---] [---]/[-----]/[----]/[--+
| |    1     0    05     109     17 |
| |  J1    J0  T1_AT0 T2_AT27 T2_AT6|
| |N7:10 N7:10 I:004    B3    I:005 |
| +-] [---] [---]/[-----]/[----]/[--+
|      1     0    04     111     16
Rung 4:105
|    J1    J0  T2_AT0 T1_AT27 T1_AT6                                    J1NEW
|  N7:10 N7:10 I:004    B3    I:005                                     N7:10
+-+-]/[---] [---] [-----]/[----]/[--+-------------------------------------( )--
| |    1     0    05     109     17 |                                       5
| |  J1    J0  T1_AT27 T1_AT6       |
| |N7:10 N7:10   B3    I:005        |
| +-] [---]/[----]/[----]/[---------+
| |    1     0    109     17        |
| |  J2    J1    J0  T2_AT2         |
| |N7:10 N7:10 N7:10 I:004          |
| +-]/[---]/[---]/[---] [-----------+
| |    2     1     0    03          |
| |  J1    J0  T1_AT0 T2_AT27 T2_AT6|
| |N7:10 N7:10 I:004    B3    I:005 |
| +-] [---] [---]/[-----]/[----]/[--+
|      1     0    04     111     16
Rung 4:106
|    J1    J0  T1_AT0 T2_AT27 T2_AT6                                    J2NEW
|  N7:10 N7:10 I:004    B3    I:005                                     N7:10
+-+-] [---] [---] [-----]/[----]/[--+-------------------------------------( )--
| |    1     0    04     111     16 |                                       6
| |  J2  T2_AT27 T2_AT6             |
| |N7:10   B3    I:005              |
| +-] [----]/[----]/[---------------+
|      2    111     16
Rung 4:107
| J2NEW                                                                 EN_T1J
| N7:10                                                                 N7:10
+--]/[-------------------------------------------------------------------( )---
|     6                                                                   10
Rung 4:108
|  J1NEW                                                                EN_T2J
|  N7:10                                                                N7:10
+-+-]/[-+----------------------------------------------------------------( )---
| |    5|                                                                 11
| |J0NEW|
| |N7:10|
| +-] [-+
|      4
Rung 4:109
| J0NEW                                                                    J0
| N7:10                                                                 N7:10
+--] [--------------------------------------------------------------------( )--
|     4                                                                     0
Rung 4:110
| J1NEW                                                                    J1
| N7:10                                                                 N7:10
+--] [--------------------------------------------------------------------( )--
|     5                                                                     1
```

1

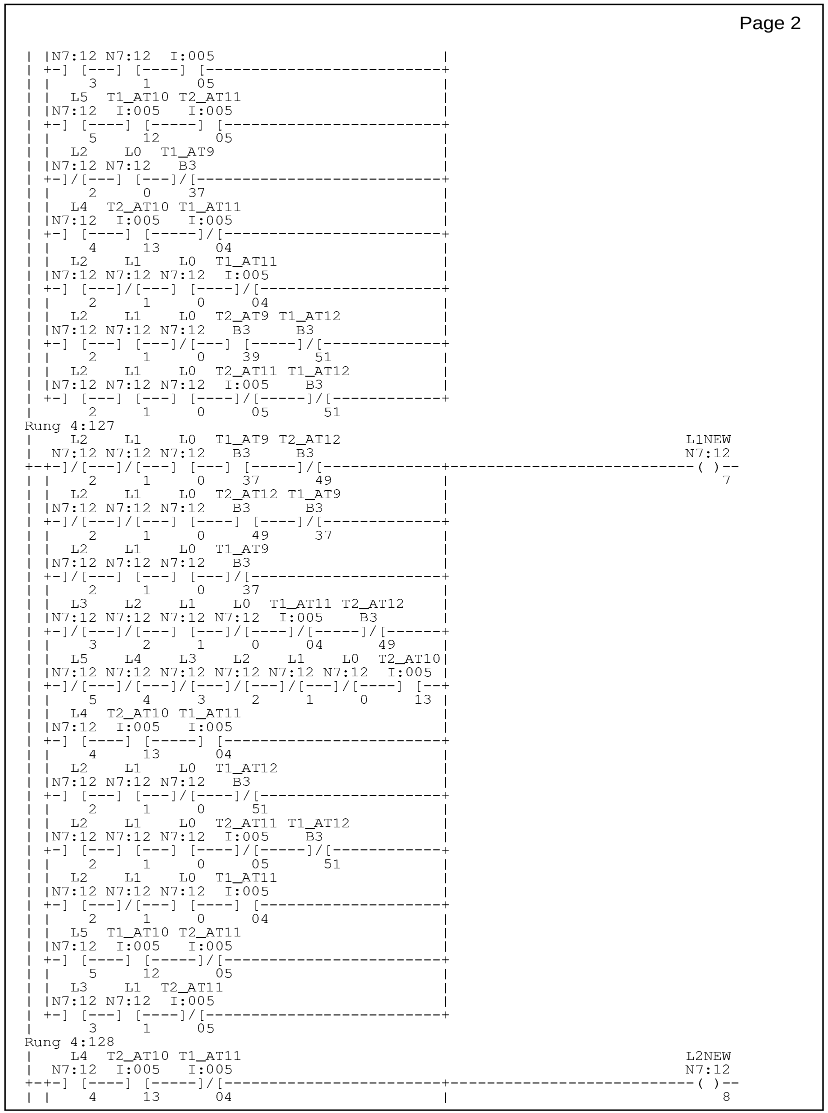
```
Rung 4:111
| J2NEW                                                                    J2
| N7:10                                                                  N7:10
+--] [--------------------------------------------------------------------( )--
|     6                                                                      2
Rung 4:112

start cp_6_7

|    K5    K4    K3    K2    K1    K0  T1_AT6                            K0NEW
|  N7:11 N7:11 N7:11 N7:11 N7:11 N7:11 I:005                             N7:11
+-+-]/[---]/[---]/[---]/[---]/[---]/[---] [--+----------------------------( )--
| |     5     4     3     2     1     0    17 |                               6
| |  K3    K1  T2_AT5                        |
| |N7:11 N7:11 I:004                         |
| +-] [---] [---] [--------------------------+
| |     3     1    12                         |
| |  K5  T1_AT6 T2_AT5                       |
| |N7:11 I:005  I:004                        |
| +-] [---] [----] [-------------------------+
| |     5    17     12                        |
| |  K2    K0  T1_AT7                        |
| |N7:11 N7:11   B3                          |
| +-]/[---] [---]/[--------------------------+
| |     2     0    31                         |
| |  K4  T2_AT6 T1_AT5                       |
| |N7:11 I:005  I:004                        |
| +-] [---] [----]/[-------------------------+
| |     4    16     13                        |
| |  K2    K1    K0  T1_AT5                  |
| |N7:11 N7:11 N7:11 I:004                   |
| +-] [---]/[---] [---]/[--------------------+
| |     2     1     0    13                   |
| |  K2    K1    K0  T2_AT7 T1_AT4           |
| |N7:11 N7:11 N7:11   B3     B3             |
| +-] [---] [---]/[-+-] [--+-]/[-------------+
| |     2     1     0|   29 |    17            |
| |                 |T2_AT3|                 |
| |                 |  B3  |                 |
| |                 +-] [--+                 |
| |                    13                   |
| |  K2    K1    K0  T2_AT5 T1_AT4           |
| |N7:11 N7:11 N7:11 I:004    B3             |
| +-] [---] [---] [---]/[----]/[-------------+
|      2     1     0    12     17
Rung 4:113
|    K2    K1    K0  T1_AT7 T2_AT4                                       K1NEW
|  N7:11 N7:11 N7:11   B3     B3                                         N7:11
+-+-]/[---]/[---] [---] [----]/[-------------+----------------------------( )--
| |     2     1     0    31     19            |                               7
| |  K2    K1    K0  T2_AT4 T1_AT7           |
| |N7:11 N7:11 N7:11   B3     B3             |
| +-]/[---]/[---] [---] [----]/[-------------+
| |     2     1     0    19     31            |
| |  K2    K1    K0  T1_AT7                  |
| |N7:11 N7:11 N7:11   B3                    |
| +-]/[---] [---] [---]/[--------------------+
| |     2     1     0    31                   |
| |  K3    K2    K1    K0  T1_AT5 T2_AT4     |
| |N7:11 N7:11 N7:11 N7:11 I:004    B3       |
| +-]/[---]/[---] [---]/[---]/[----]/[-------+
| |     3     2     1     0    13     19      |
| |  K5    K4    K3    K2    K1    K0  T2_AT6|
| |N7:11 N7:11 N7:11 N7:11 N7:11 N7:11 I:005 |
| +-]/[---]/[---]/[---]/[---]/[---]/[---] [--+
| |     5     4     3     2     1     0    16 |
| |  K4  T2_AT6 T1_AT5                       |
| |N7:11 I:005  I:004                        |
| +-] [---] [----] [-------------------------+
| |     4    16     13                        |
| |  K2    K1    K0  T1_AT4                  |
| |N7:11 N7:11 N7:11   B3                    |
| +-] [---] [---]/[---]/[--------------------+
| |     2     1     0    17                   |
| |  K2    K1    K0  T2_AT5 T1_AT4           |
| |N7:11 N7:11 N7:11 I:004    B3             |
| +-] [---] [---] [---]/[----]/[-------------+
```

```
| |     2     1     0    12     17            |
| |  K2    K1    K0  T1_AT5                  |
| |N7:11 N7:11 N7:11 I:004                   |
| +-] [---]/[---] [---] [--------------------+
| |     2     1     0    13                   |
| |  K5  T1_AT6 T2_AT5                       |
| |N7:11 I:005  I:004                        |
| +-] [---] [----]/[-------------------------+
| |     5    17     12                        |
| |  K3    K1  T2_AT5                        |
| |N7:11 N7:11 I:004                         |
| +-] [---] [---]/[--------------------------+
|      3     1    12
Rung 4:114
|    K4  T2_AT6 T1_AT5                                                   K2NEW
|  N7:11 I:005  I:004                                                    N7:11
+-+-] [---] [----]/[-------------------------+----------------------------( )--
| |     4    16     13                        |                               8
| |  K2    K1    K0                          |
| |N7:11 N7:11 N7:11                         |
| +-] [---]/[---] [--------------------------+
| |     2     1     0                         |
| |  K5    K4    K3    K2    K1    K0  T2_AT6|
| |N7:11 N7:11 N7:11 N7:11 N7:11 N7:11 I:005 |
| +-]/[---]/[---]/[---]/[---]/[---]/[---] [--+
| |     5     4     3     2     1     0    16 |
| |  K4  T2_AT6 T1_AT5                       |
| |N7:11 I:005  I:004                        |
| +-] [---] [----] [-------------------------+
| |     4    16     13                        |
| |  K2    K1    K0  T1_AT4                  |
| |N7:11 N7:11 N7:11   B3                    |
| +-] [---] [---]/[---]/[--------------------+
| |     2     1     0    17                   |
| |  K2    K1    K0  T2_AT5 T1_AT4           |
| |N7:11 N7:11 N7:11 I:004    B3             |
| +-] [---] [---] [---]/[----]/[-------------+
|      2     1     0    12     17
Rung 4:115
|    K2    K1    K0  T1_AT4 T2_AT7 T2_AT3                                K3NEW
|  N7:11 N7:11 N7:11   B3     B3     B3                                  N7:11
+-+-] [---] [---]/[---] [----]/[----]/[--+--------------------------------( )--
| |     2     1     0    17     29     13 |                                   9
| |  K3    K1  T2_AT7 T2_AT3             |
| |N7:11 N7:11   B3     B3               |
| +-] [---]/[---]/[----]/[---------------+
| |     3     1    29     13              |
| |  K5  T1_AT6 T2_AT5                   |
| |N7:11 I:005  I:004                    |
| +-] [---] [----]/[---------------------+
| |     5    17     12                    |
| |  K3    K1  T2_AT5                    |
| |N7:11 N7:11 I:004                     |
| +-] [---] [---]/[----------------------+
|      3     1    12
Rung 4:116
|    K3    K2    K1    K0  T2_AT4 T1_AT5                                 K4NEW
|  N7:11 N7:11 N7:11 N7:11   B3   I:004                                  N7:11
+-+-]/[---]/[---] [---]/[---] [----]/[--+---------------------------------( )--
| |     3     2     1     0    19     13 |                                   10
| |  K2    K1    K0  T2_AT4 T1_AT7      |
| |N7:11 N7:11 N7:11   B3     B3        |
| +-]/[---]/[---] [---] [----] [--------+
| |     2     1     0    19     31       |
| |  K2    K1    K0  T1_AT7             |
| |N7:11 N7:11 N7:11   B3               |
| +-]/[---] [---] [---] [---------------+
| |     2     1     0    31              |
| |  K4  T2_AT6 T1_AT5                  |
| |N7:11 I:005  I:004                   |
| +-] [---]/[----]/[--------------------+
|      4    16     13
Rung 4:117
|    K2    K1    K0  T1_AT4 T2_AT7                                       K5NEW
|  N7:11 N7:11 N7:11   B3     B3                                         N7:11
+-+-] [---] [---]/[---] [--+-] [--++--------------------------------------( )--
| |     2     1     0    17 |   29 ||                                        11
```



Page 1

```
| |                        |T2_AT3||
| |                        |  B3  ||
| |                        +-] [--+|
| |                            13  |
| |  K2    K1    K0  T1_AT4 T2_AT5 |
| |N7:11 N7:11 N7:11   B3   I:004  |
| +-] [---] [---] [---] [----]/[---+
| |    2     1     0    17     12  |
| |  K3    K1  T2_AT7              |
| |N7:11 N7:11   B3                |
| +-] [---]/[-+-] [--+-------------+
| |    3     1|   29 |             |
| |           |T2_AT3|             |
| |           |  B3  |             |
| |           +-] [--+             |
| |               13               |
| |  K5  T1_AT6 T2_AT5             |
| |N7:11 I:005  I:004              |
| +-] [---]/[----]/[---------------+
|      5    17     12
Rung 4:118
| K3NEW                                                                 EN_T1K
| N7:11                                                                 N7:11
+--]/[-------------------------------------------------------------------( )---
|     9                                                                    12
Rung 4:119
|  K2NEW  K2NEW                                                         EN_T2K
|  N7:11  N7:11                                                         N7:11
+-+-] [-++-]/[-+---------------------------------------------------------( )---
| |    8||    8|                                                           13
| |K1NEW||K1NEW|
| |N7:11||N7:11|
| +-]/[-++-] [-+
| |    7||    7|
| |K0NEW||K0NEW|
| |N7:11||N7:11|
| +-]/[-++-]/[-+
|      6      6
Rung 4:120
| K0NEW                                                                    K0
| N7:11                                                                  N7:11
+--] [--------------------------------------------------------------------( )--
|     6                                                                      0
Rung 4:121
| K1NEW                                                                    K1
| N7:11                                                                  N7:11
+--] [--------------------------------------------------------------------( )--
|     7                                                                      1
Rung 4:122
| K2NEW                                                                    K2
| N7:11                                                                  N7:11
+--] [--------------------------------------------------------------------( )--
|     8                                                                      2
Rung 4:123
| K3NEW                                                                    K3
| N7:11                                                                  N7:11
+--] [--------------------------------------------------------------------( )--
|     9                                                                      3
Rung 4:124
| K4NEW                                                                    K4
| N7:11                                                                  N7:11
+--] [--------------------------------------------------------------------( )--
|    10                                                                      4
Rung 4:125
| K5NEW                                                                    K5
| N7:11                                                                  N7:11
+--] [--------------------------------------------------------------------( )--
|    11                                                                      5
Rung 4:126

start cp_9_10

|    L5    L4    L3    L2    L1    L0  T1_AT10                           L0NEW
|  N7:12 N7:12 N7:12 N7:12 N7:12 N7:12  I:005                            N7:12
+-+-]/[---]/[---]/[---]/[---]/[---]/[----] [--+---------------------------( )--
| |    5     4     3     2     1     0    12 |                              6
| |  L3    L1  T2_AT11                        |
```



Page 2

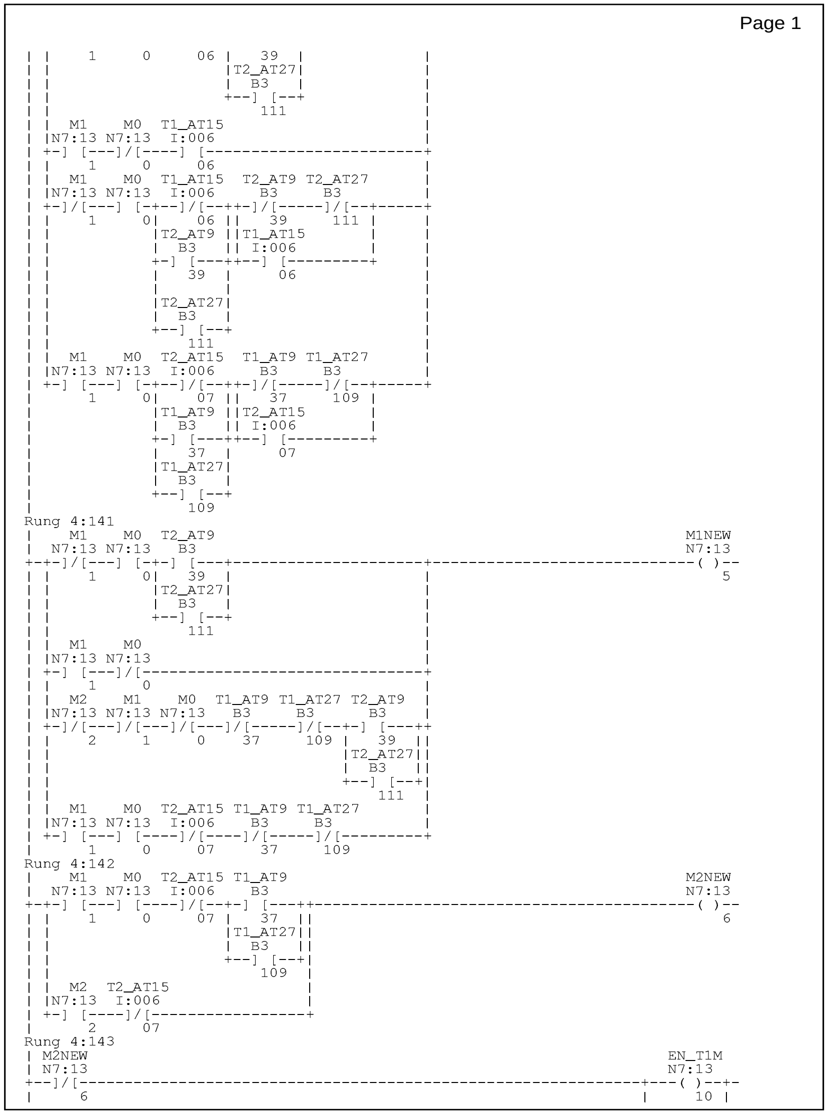
```
| |N7:12 N7:12  I:005                         |
| +-] [---] [----] [--------------------------+
| |    3     1     05                         |
| |  L5  T1_AT10 T2_AT11                      |
| |N7:12  I:005   I:005                       |
| +-] [----] [-----] [------------------------+
| |    5     12      05                       |
| |  L2    L0  T1_AT9                         |
| |N7:12 N7:12   B3                           |
| +-]/[---] [---]/[---------------------------+
| |    2     0    37                          |
| |  L4  T2_AT10 T1_AT11                      |
| |N7:12  I:005   I:005                       |
| +-] [----] [-----]/[------------------------+
| |    4     13      04                       |
| |  L2    L1    L0  T1_AT11                  |
| |N7:12 N7:12 N7:12  I:005                   |
| +-] [---]/[---] [----]/[--------------------+
| |    2     1     0     04                   |
| |  L2    L1    L0  T2_AT9 T1_AT12           |
| |N7:12 N7:12 N7:12   B3     B3              |
| +-] [---] [---]/[---] [-----]/[-------------+
| |    2     1     0    39      51            |
| |  L2    L1    L0  T2_AT11 T1_AT12          |
| |N7:12 N7:12 N7:12  I:005    B3             |
| +-] [---] [---] [----]/[-----]/[------------+
|      2     1     0     05      51
Rung 4:127
|    L2    L1    L0  T1_AT9 T2_AT12                                      L1NEW
|  N7:12 N7:12 N7:12   B3     B3                                         N7:12
+-+-]/[---]/[---] [---] [-----]/[-------------+---------------------------( )--
| |    2     1     0    37      49            |                              7
| |  L2    L1    L0  T2_AT12 T1_AT9           |
| |N7:12 N7:12 N7:12   B3      B3             |
| +-]/[---]/[---] [----] [----]/[-------------+
| |    2     1     0     49     37            |
| |  L2    L1    L0  T1_AT9                   |
| |N7:12 N7:12 N7:12   B3                     |
| +-]/[---] [---] [---]/[---------------------+
| |    2     1     0    37                    |
| |  L3    L2    L1    L0  T1_AT11 T2_AT12    |
| |N7:12 N7:12 N7:12 N7:12  I:005    B3       |
| +-]/[---]/[---] [---]/[----]/[-----]/[------+
| |    3     2     1     0     04      49     |
| |  L5    L4    L3    L2    L1    L0  T2_AT10|
| |N7:12 N7:12 N7:12 N7:12 N7:12 N7:12  I:005 |
| +-]/[---]/[---]/[---]/[---]/[---]/[----] [--+
| |    5     4     3     2     1     0     13 |
| |  L4  T2_AT10 T1_AT11                      |
| |N7:12  I:005   I:005                       |
| +-] [----] [-----] [------------------------+
| |    4     13      04                       |
| |  L2    L1    L0  T1_AT12                  |
| |N7:12 N7:12 N7:12   B3                     |
| +-] [---] [---]/[----]/[--------------------+
| |    2     1     0     51                   |
| |  L2    L1    L0  T2_AT11 T1_AT12          |
| |N7:12 N7:12 N7:12  I:005    B3             |
| +-] [---] [---] [----]/[-----]/[------------+
| |    2     1     0     05      51           |
| |  L2    L1    L0  T1_AT11                  |
| |N7:12 N7:12 N7:12  I:005                   |
| +-] [---]/[---] [----] [--------------------+
| |    2     1     0     04                   |
| |  L5  T1_AT10 T2_AT11                      |
| |N7:12  I:005   I:005                       |
| +-] [----] [-----]/[------------------------+
| |    5     12      05                       |
| |  L3    L1  T2_AT11                        |
| |N7:12 N7:12  I:005                         |
| +-] [---] [----]/[--------------------------+
|      3     1     05
Rung 4:128
|    L4  T2_AT10 T1_AT11                                                 L2NEW
|  N7:12  I:005   I:005                                                  N7:12
+-+-] [----] [-----]/[------------------------+---------------------------( )--
| |    4     13      04                       |                              8
```

Page 1

```
| |  L2    L1    L0                           |
| |N7:12 N7:12 N7:12                          |
| +-] [---]/[---] [---------------------------+
| |    2     1     0                          |
| |  L5    L4    L3    L2    L1    L0  T2_AT10|
| |N7:12 N7:12 N7:12 N7:12 N7:12 N7:12  I:005 |
| +-]/[---]/[---]/[---]/[---]/[---]/[----] [--+
| |    5     4     3     2     1     0     13 |
| |  L4  T2_AT10 T1_AT11                      |
| |N7:12  I:005   I:005                       |
| +-] [----] [-----] [------------------------+
| |    4     13      04                       |
| |  L2    L1    L0  T1_AT12                  |
| |N7:12 N7:12 N7:12   B3                     |
| +-] [---] [---]/[----]/[--------------------+
| |    2     1     0     51                   |
| |  L2    L1    L0  T2_AT11 T1_AT12          |
| |N7:12 N7:12 N7:12  I:005    B3             |
| +-] [---] [---] [----]/[-----]/[------------+
|      2     1     0     05      51
Rung 4:129
|    L2    L1    L0  T1_AT12 T2_AT9                                     L3NEW
|  N7:12 N7:12 N7:12   B3      B3                                       N7:12
+-+-] [---] [---]/[----] [----]/[--+--------------------------------------( )--
| |    2     1     0     51     39 |                                        9
| |  L3    L1  T2_AT9              |
| |N7:12 N7:12   B3                |
| +-] [---]/[---]/[----------------+
| |    3     1    39               |
| |  L5  T1_AT10 T2_AT11           |
| |N7:12  I:005   I:005            |
| +-] [----] [-----]/[-------------+
| |    5     12      05            |
| |  L3    L1  T2_AT11             |
| |N7:12 N7:12  I:005              |
| +-] [---] [----]/[---------------+
|      3     1     05
Rung 4:130
|    L3    L2    L1    L0  T2_AT12 T1_AT11                              L4NEW
|  N7:12 N7:12 N7:12 N7:12   B3     I:005                               N7:12
+-+-]/[---]/[---] [---]/[-----] [-----]/[--+-------------------------------( )--
| |    3     2     1     0     49      04 |                                10
| |  L2    L1    L0  T2_AT12 T1_AT9       |
| |N7:12 N7:12 N7:12   B3      B3         |
| +-]/[---]/[---] [----] [----] [---------+
| |    2     1     0     49     37        |
| |  L2    L1    L0  T1_AT9               |
| |N7:12 N7:12 N7:12   B3                 |
| +-]/[---] [---] [---] [-----------------+
| |    2     1     0    37                |
| |  L4  T2_AT10 T1_AT11                  |
| |N7:12  I:005   I:005                   |
| +-] [----]/[-----]/[--------------------+
|      4     13      04
Rung 4:131
|    L2    L1    L0  T1_AT12 T2_AT9                                     L5NEW
|  N7:12 N7:12 N7:12   B3      B3                                       N7:12
+-+-] [---] [---]/[----] [----] [---+-------------------------------------( )--
| |    2     1     0     51     39  |                                      11
| |  L2    L1    L0  T1_AT12 T2_AT11|
| |N7:12 N7:12 N7:12   B3     I:005 |
| +-] [---] [---] [----] [-----]/[--+
| |    2     1     0     51      05 |
| |  L3    L1  T2_AT9               |
| |N7:12 N7:12   B3                 |
| +-] [---]/[---] [-----------------+
| |    3     1    39                |
| |  L5  T1_AT10 T2_AT11            |
| |N7:12  I:005   I:005             |
| +-] [----]/[-----]/[--------------+
|      5     12      05
Rung 4:132
| L3NEW                                                               EN_T1L
| N7:12                                                               N7:12
+--]/[-------------------------------------------------------------------( )---
|     9                                                                  12
Rung 4:133
```

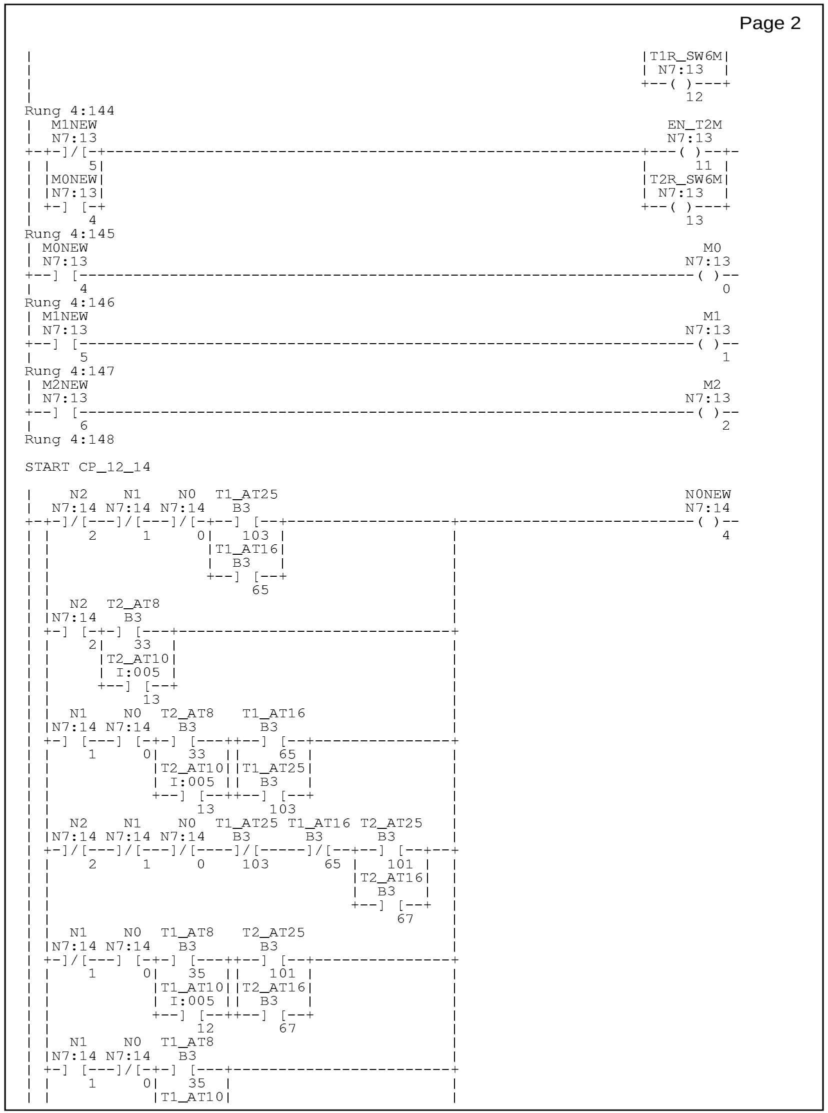
Page 2

```
|  L2NEW  L2NEW                                                        EN_T2L
|  N7:12  N7:12                                                        N7:12
+-+-] [-++-]/[-+----------------------------------------------------------( )---
| |    8||    8|                                                           13
| |L1NEW||L1NEW|
| |N7:12||N7:12|
| +-]/[-++-] [-+
| |    7||    7|
| |L0NEW||L0NEW|
| |N7:12||N7:12|
| +-]/[-++-]/[-+
|      6      6
Rung 4:134
| L0NEW                                                                     L0
| N7:12                                                                  N7:12
+--] [---------------------------------------------------------------------( )--
|     6                                                                       0
Rung 4:135
| L1NEW                                                                     L1
| N7:12                                                                  N7:12
+--] [---------------------------------------------------------------------( )--
|     7                                                                       1
Rung 4:136
| L2NEW                                                                     L2
| N7:12                                                                  N7:12
+--] [---------------------------------------------------------------------( )--
|     8                                                                       2
Rung 4:137
| L3NEW                                                                     L3
| N7:12                                                                  N7:12
+--] [---------------------------------------------------------------------( )--
|     9                                                                       3
Rung 4:138
| L4NEW                                                                     L4
| N7:12                                                                  N7:12
+--] [---------------------------------------------------------------------( )--
|    10                                                                       4
Rung 4:139
| L5NEW                                                                     L5
| N7:12                                                                  N7:12
+--] [---------------------------------------------------------------------( )--
|    11                                                                       5
Rung 4:140

START CP_11_13

|    M2    M1    M0  T1_AT9                                              M0NEW
|  N7:13 N7:13 N7:13   B3                                                N7:13
+-+-]/[---]/[---]/[-+-] [---+---------------+------------------------------( )--
| |    2     1     0|   37   |               |                                 4
| |                 |T1_AT27|               |
| |                 |  B3   |               |
| |                 +--] [--+               |
| |                      109                |
| |  M2  T2_AT15                            |
| |N7:13  I:006                             |
| +-] [----] [------------------------------+
| |    2     07                             |
| |  M1    M0  T2_AT15 T1_AT27              |
| |N7:13 N7:13  I:006    B3                 |
| +-] [---] [----] [--+--] [--+-------------+
| |    1     0     07 |   109 |             |
| |                   |T1_AT9 |             |
| |                   |  B3   |             |
| |                   +-] [---+             |
| |                        37               |
| |  M2    M1    M0  T1_AT9 T1_AT27 T2_AT9  |
| |N7:13 N7:13 N7:13   B3      B3      B3   |
| +-]/[---]/[---]/[---]/[-----]/[--+-] [---++
| |    2     1     0    37     109 |   39  ||
| |                                |T2_AT27||
| |                                |  B3   ||
| |                                +--] [--+|
| |                                    111  |
| |  M1    M0  T1_AT15 T2_AT9               |
| |N7:13 N7:13  I:006    B3                 |
| +-]/[---] [----] [--+-] [---+-------------+
```



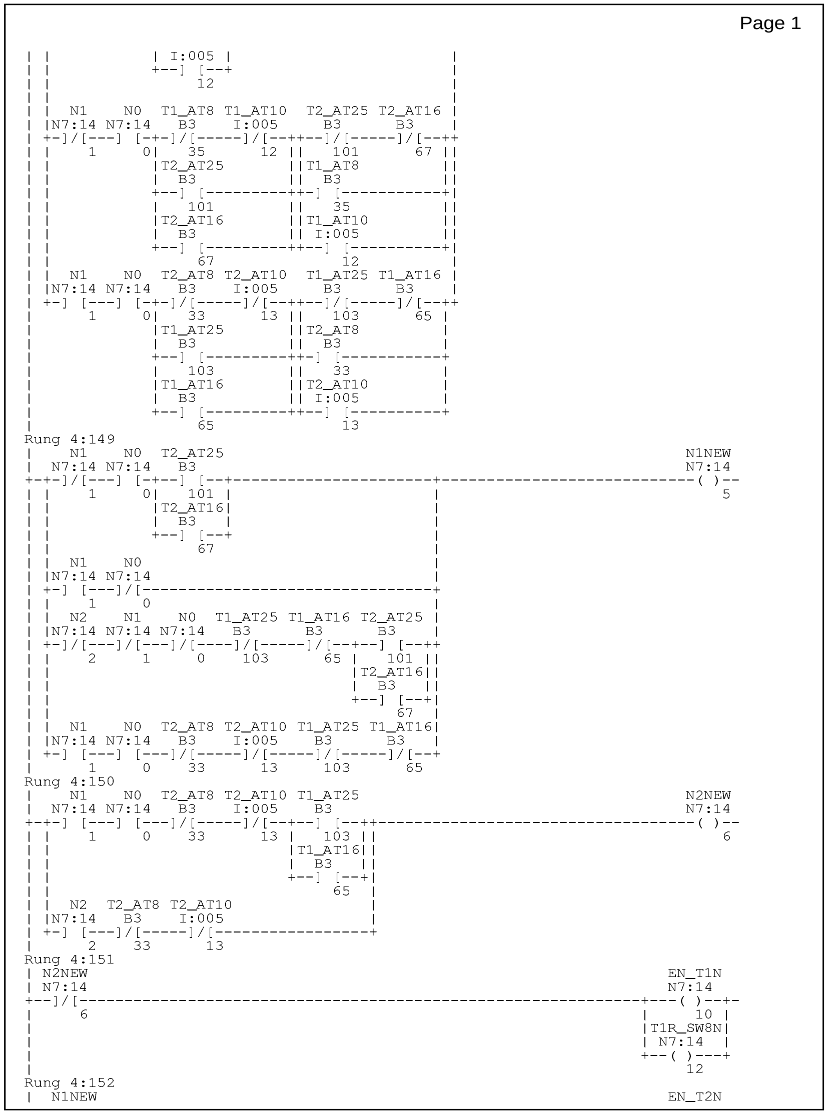
Page 1

```
| |    1     0     06 |   39  |             |
| |                   |T2_AT27|             |
| |                   |  B3   |             |
| |                   +--] [--+             |
| |                       111               |
| |  M1    M0  T1_AT15                      |
| |N7:13 N7:13  I:006                       |
| +-] [---]/[----] [------------------------+
| |    1     0     06                       |
| |  M1    M0  T1_AT15  T2_AT9 T2_AT27      |
| |N7:13 N7:13  I:006     B3     B3         |
| +-]/[---] [-+--]/[--++-]/[-----]/[--+-----+
| |    1     0|    06 ||   39     111 |     |
| |           |T2_AT9 ||T1_AT15       |     |
| |           |  B3   || I:006        |     |
| |           +-] [---++--] [---------+     |
| |           |   39  |     06              |
| |           |       |                     |
| |           |T2_AT27|                     |
| |           |  B3   |                     |
| |           +--] [--+                     |
| |               111                       |
| |  M1    M0  T2_AT15  T1_AT9 T1_AT27      |
| |N7:13 N7:13  I:006     B3     B3         |
| +-] [---] [-+--]/[--++-]/[-----]/[--+-----+
|      1     0|    07 ||   37     109 |
|             |T1_AT9 ||T2_AT15       |
|             |  B3   || I:006        |
|             +-] [---++--] [---------+
|             |   37  |     07
|             |T1_AT27|
|             |  B3   |
|             +--] [--+
|                 109
Rung 4:141
|    M1    M0  T2_AT9                                                    M1NEW
|  N7:13 N7:13   B3                                                      N7:13
+-+-]/[---] [-+-] [---+---------------------+-----------------------------( )--
| |    1     0|   39  |                     |                                5
| |           |T2_AT27|                     |
| |           |  B3   |                     |
| |           +--] [--+                     |
| |               111                       |
| |  M1    M0                               |
| |N7:13 N7:13                              |
| +-] [---]/[-------------------------------+
| |    1     0                              |
| |  M2    M1    M0  T1_AT9 T1_AT27 T2_AT9  |
| |N7:13 N7:13 N7:13   B3     B3      B3    |
| +-]/[---]/[---]/[---]/[-----]/[--+-] [---++
| |    2     1     0    37     109 |   39  ||
| |                                |T2_AT27||
| |                                |  B3   ||
| |                                +--] [--+|
| |                                    111  |
| |  M1    M0  T2_AT15 T1_AT9 T1_AT27       |
| |N7:13 N7:13  I:006    B3     B3          |
| +-] [---] [----]/[----]/[-----]/[---------+
|      1     0     07     37     109
Rung 4:142
|    M1    M0  T2_AT15 T1_AT9                                            M2NEW
|  N7:13 N7:13  I:006    B3                                              N7:13
+-+-] [---] [----]/[--+-] [---++------------------------------------------( )--
| |    1     0     07 |   37  ||                                             6
| |                   |T1_AT27||
| |                   |  B3   ||
| |                   +--] [--+|
| |                       109  |
| |  M2  T2_AT15               |
| |N7:13  I:006                |
| +-] [----]/[-----------------+
|      2     07
Rung 4:143
| M2NEW                                                                EN_T1M
| N7:13                                                                N7:13
+--]/[--------------------------------------------------------------+---( )--+-
|     6                                                             |     10 |
```



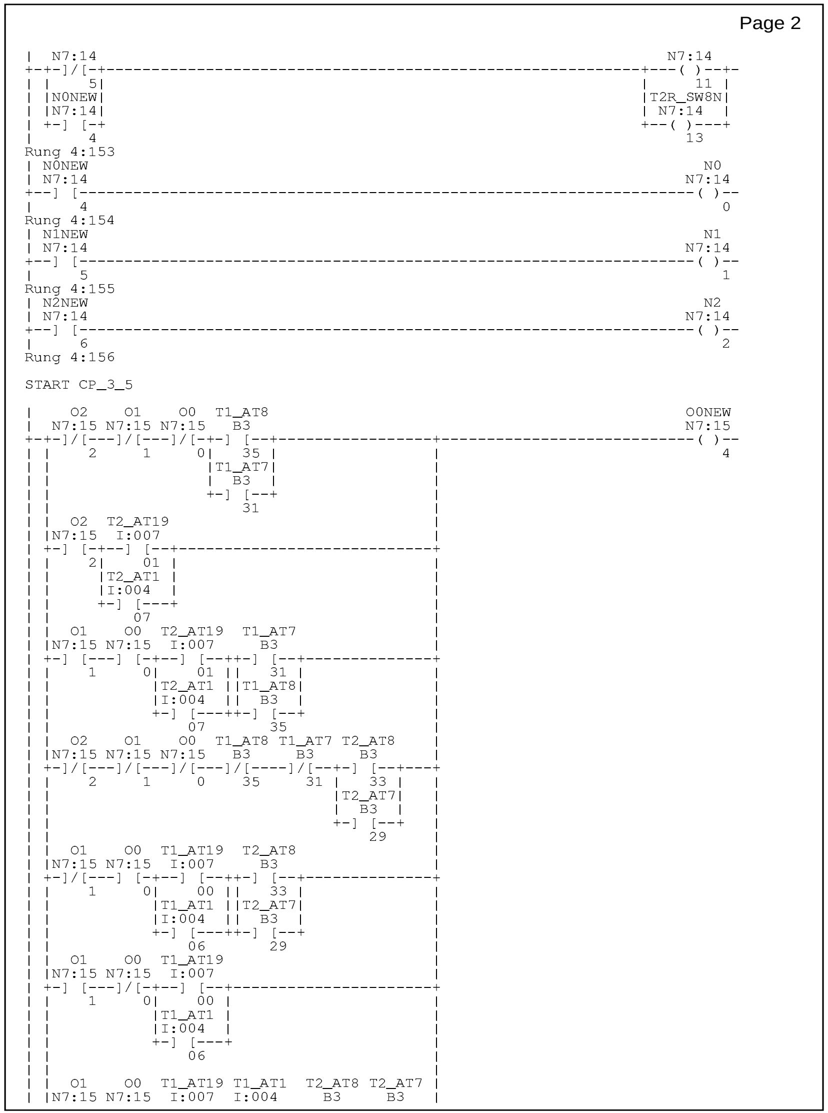
Page 2

```
|                                                                    |T1R_SW6M|
|                                                                    | N7:13  |
|                                                                    +--( )---+
|                                                                        12
Rung 4:144
|  M1NEW                                                                EN_T2M
|  N7:13                                                                N7:13
+-+-]/[-+------------------------------------------------------------+---( )--+-
| |    5|                                                            |     11 |
| |M0NEW|                                                            |T2R_SW6M|
| |N7:13|                                                            | N7:13  |
| +-] [-+                                                            +--( )---+
|      4                                                                  13
Rung 4:145
| M0NEW                                                                     M0
| N7:13                                                                   N7:13
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:146
| M1NEW                                                                     M1
| N7:13                                                                   N7:13
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:147
| M2NEW                                                                     M2
| N7:13                                                                   N7:13
+--] [---------------------------------------------------------------------( )--
|     6                                                                       2
Rung 4:148

START CP_12_14

|    N2    N1    N0  T1_AT25                                             N0NEW
|  N7:14 N7:14 N7:14   B3                                                N7:14
+-+-]/[---]/[---]/[-+--] [--+------------------+--------------------------( )--
| |    2     1     0|   103 |                  |                             4
| |                 |T1_AT16|                  |
| |                 |  B3   |                  |
| |                 +--] [--+                  |
| |                      65                    |
| |  N2  T2_AT8                                |
| |N7:14   B3                                  |
| +-] [-+-] [---+------------------------------+
| |    2|   33  |                              |
| |     |T2_AT10|                              |
| |     | I:005 |                              |
| |     +--] [--+                              |
| |          13                                |
| |  N1    N0  T2_AT8   T1_AT16                |
| |N7:14 N7:14   B3       B3                   |
| +-] [---] [-+-] [---++--] [--+---------------+
| |    1     0|   33  ||    65 |               |
| |           |T2_AT10||T1_AT25|               |
| |           | I:005 ||  B3   |               |
| |           +--] [--++--] [--+               |
| |                13      103                 |
| |  N2    N1    N0  T1_AT25 T1_AT16 T2_AT25   |
| |N7:14 N7:14 N7:14   B3      B3       B3     |
| +-]/[---]/[---]/[----]/[-----]/[--+--] [--+--+
| |    2     1     0    103      65 |   101 |  |
| |                                 |T2_AT16|  |
| |                                 |  B3   |  |
| |                                 +--] [--+  |
| |                                      67    |
| |  N1    N0  T1_AT8   T2_AT25                |
| |N7:14 N7:14   B3       B3                   |
| +-]/[---] [-+-] [---++--] [--+---------------+
| |    1     0|   35  ||   101 |               |
| |           |T1_AT10||T2_AT16|               |
| |           | I:005 ||  B3   |               |
| |           +--] [--++--] [--+               |
| |                12       67                 |
| |  N1    N0  T1_AT8                          |
| |N7:14 N7:14   B3                            |
| +-] [---]/[-+-] [---+------------------------+
| |    1     0|   35  |                        |
| |           |T1_AT10|                        |
```



Page 1

```
| |           | I:005 |                        |
| |           +--] [--+                        |
| |                12                          |
| |                                            |
| |  N1    N0  T1_AT8 T1_AT10  T2_AT25 T2_AT16 |
| |N7:14 N7:14   B3    I:005     B3      B3    |
| +-]/[---] [-+-]/[-----]/[--++--]/[-----]/[--++
| |    1     0|   35      12 ||   101      67 ||
| |           |T2_AT25       ||T1_AT8         ||
| |           |  B3          ||  B3           ||
| |           +--] [---------++-] [-----------+|
| |           |   101        ||   35          ||
| |           |T2_AT16       ||T1_AT10        ||
| |           |  B3          || I:005         ||
| |           +--] [---------++--] [----------+|
| |                67              12          |
| |  N1    N0  T2_AT8 T2_AT10  T1_AT25 T1_AT16 |
| |N7:14 N7:14   B3    I:005     B3      B3    |
| +-] [---] [-+-]/[-----]/[--++--]/[-----]/[--++
|      1     0|   33      13 ||   103      65 |
|             |T1_AT25       ||T2_AT8         |
|             |  B3          ||  B3           |
|             +--] [---------++-] [-----------+
|             |   103        ||   33          |
|             |T1_AT16       ||T2_AT10        |
|             |  B3          || I:005         |
|             +--] [---------++--] [----------+
|                  65              13
Rung 4:149
|    N1    N0  T2_AT25                                                   N1NEW
|  N7:14 N7:14   B3                                                      N7:14
+-+-]/[---] [-+--] [--+----------------------+----------------------------( )--
| |    1     0|   101 |                      |                               5
| |           |T2_AT16|                      |
| |           |  B3   |                      |
| |           +--] [--+                      |
| |                67                        |
| |  N1    N0                                |
| |N7:14 N7:14                               |
| +-] [---]/[--------------------------------+
| |    1     0                               |
| |  N2    N1    N0  T1_AT25 T1_AT16 T2_AT25 |
| |N7:14 N7:14 N7:14   B3      B3      B3    |
| +-]/[---]/[---]/[----]/[-----]/[--+--] [--++
| |    2     1     0    103       65 |   101 ||
| |                                 |T2_AT16||
| |                                 |  B3   ||
| |                                 +--] [--+|
| |                                      67  |
| |  N1    N0  T2_AT8 T2_AT10 T1_AT25 T1_AT16|
| |N7:14 N7:14   B3    I:005    B3      B3   |
| +-] [---] [---]/[-----]/[-----]/[-----]/[--+
|      1     0    33      13     103      65
Rung 4:150
|    N1    N0  T2_AT8 T2_AT10 T1_AT25                                    N2NEW
|  N7:14 N7:14   B3    I:005    B3                                       N7:14
+-+-] [---] [---]/[-----]/[--+--] [--++-----------------------------------( )--
| |    1     0    33      13 |   103 ||                                      6
| |                          |T1_AT16||
| |                          |  B3   ||
| |                          +--] [--+|
| |                               65  |
| |  N2  T2_AT8 T2_AT10               |
| |N7:14   B3    I:005                |
| +-] [---]/[-----]/[-----------------+
|      2    33      13
Rung 4:151
| N2NEW                                                               EN_T1N
| N7:14                                                               N7:14
+--]/[--------------------------------------------------------------+---( )--+-
|     6                                                             |     10 |
|                                                                   |T1R_SW8N|
|                                                                   | N7:14  |
|                                                                   +--( )---+
|                                                                        12
Rung 4:152
|  N1NEW                                                              EN_T2N
```



Page 2

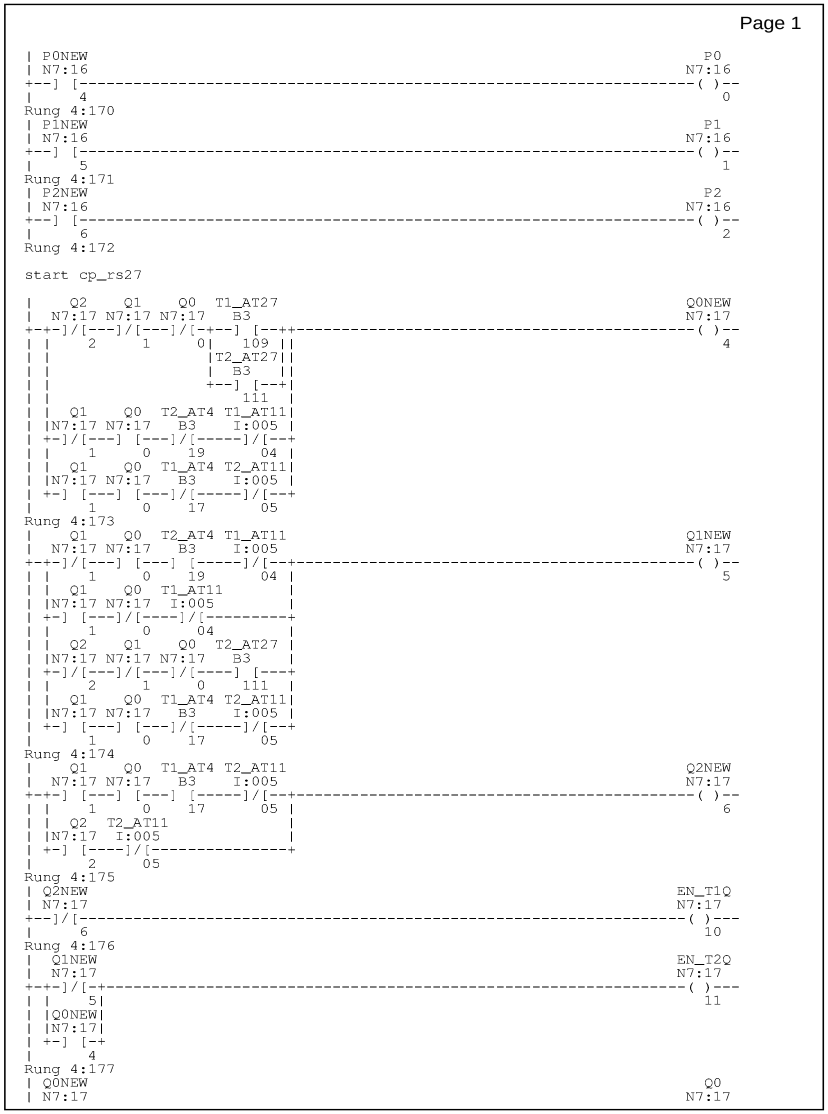
```
|  N7:14                                                               N7:14
+-+-]/[-+-----------------------------------------------------------+---( )--+-
| |    5|                                                           |     11 |
| |N0NEW|                                                           |T2R_SW8N|
| |N7:14|                                                           | N7:14  |
| +-] [-+                                                           +--( )---+
|      4                                                                 13
Rung 4:153
| N0NEW                                                                   N0
| N7:14                                                                N7:14
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:154
| N1NEW                                                                   N1
| N7:14                                                                N7:14
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:155
| N2NEW                                                                   N2
| N7:14                                                                N7:14
+--] [--------------------------------------------------------------------( )--
|     6                                                                      2
Rung 4:156

START CP_3_5

|    O2    O1    O0  T1_AT8                                              O0NEW
|  N7:15 N7:15 N7:15   B3                                                N7:15
+-+-]/[---]/[---]/[-+-] [--+-----------------+----------------------------( )--
| |    2     1     0|   35 |                 |                               4
| |                 |T1_AT7|                 |
| |                 |  B3  |                 |
| |                 +-] [--+                 |
| |                    31                    |
| |  O2  T2_AT19                             |
| |N7:15  I:007                              |
| +-] [-+--] [--+----------------------------+
| |    2|    01 |                            |
| |     |T2_AT1 |                            |
| |     |I:004  |                            |
| |     +-] [---+                            |
| |         07                               |
| |  O1    O0  T2_AT19  T1_AT7               |
| |N7:15 N7:15  I:007     B3                 |
| +-] [---] [-+--] [--++-] [--+--------------+
| |    1     0|    01 ||   31 |              |
| |           |T2_AT1 ||T1_AT8|              |
| |           |I:004  ||  B3  |              |
| |           +-] [---++-] [--+              |
| |               07       35                |
| |  O2    O1    O0  T1_AT8 T1_AT7 T2_AT8    |
| |N7:15 N7:15 N7:15   B3     B3     B3      |
| +-]/[---]/[---]/[---]/[----]/[--+-] [--+---+
| |    2     1     0    35     31 |   33 |   |
| |                               |T2_AT7|   |
| |                               |  B3  |   |
| |                               +-] [--+   |
| |                                   29     |
| |  O1    O0  T1_AT19  T2_AT8               |
| |N7:15 N7:15  I:007     B3                 |
| +-]/[---] [-+--] [--++-] [--+--------------+
| |    1     0|    00 ||   33 |              |
| |           |T1_AT1 ||T2_AT7|              |
| |           |I:004  ||  B3  |              |
| |           +-] [---++-] [--+              |
| |               06       29                |
| |  O1    O0  T1_AT19                       |
| |N7:15 N7:15  I:007                        |
| +-] [---]/[-+--] [--+----------------------+
| |    1     0|    00 |                      |
| |           |T1_AT1 |                      |
| |           |I:004  |                      |
| |           +-] [---+                      |
| |               06                         |
| |                                          |
| |  O1    O0  T1_AT19 T1_AT1  T2_AT8 T2_AT7 |
| |N7:15 N7:15  I:007  I:004     B3     B3    |
```

Page 1

```
| +-]/[---] [-+--]/[----]/[--++-]/[----]/[--++
| |    1     0|    00     06 ||   33     29 ||
| |           |T2_AT8        ||T1_AT19      ||
| |           |  B3          || I:007       ||
| |           +-] [----------++--] [--------+|
| |           |   33         ||    00       ||
| |           |T2_AT7        ||T1_AT1       ||
| |           |  B3          ||I:004        ||
| |           +-] [----------++-] [---------+|
| |               29              06         |
| |  O1    O0  T2_AT19 T2_AT1  T1_AT8 T1_AT7 |
| |N7:15 N7:15  I:007  I:004     B3     B3   |
| +-] [---] [-+--]/[----]/[--++-]/[----]/[--++
|      1     0|    01     07 ||   35     31 |
|             |T1_AT8        ||T2_AT19      |
|             |  B3          || I:007       |
|             +-] [----------++--] [--------+
|             |   35         ||    01       |
|             |T1_AT7        ||T2_AT1       |
|             |  B3          ||I:004        |
|             +-] [----------++-] [---------+
|                 31              07
Rung 4:157
|    O1    O0  T2_AT8                                                    O1NEW
|  N7:15 N7:15   B3                                                      N7:15
+-+-]/[---] [-+-] [--+---------------------+------------------------------( )--
| |    1     0|   33 |                     |                                 5
| |           |T2_AT7|                     |
| |           |  B3  |                     |
| |           +-] [--+                     |
| |               29                       |
| |  O1    O0                              |
| |N7:15 N7:15                             |
| +-] [---]/[------------------------------+
| |    1     0                             |
| |  O2    O1    O0  T1_AT8 T1_AT7 T2_AT8  |
| |N7:15 N7:15 N7:15   B3     B3     B3    |
| +-]/[---]/[---]/[---]/[----]/[--+-] [--+-+
| |    2     1     0    35     31 |   33 | |
| |                               |T2_AT7| |
| |                               |  B3  | |
| |                               +-] [--+ |
| |                                   29   |
| |  O1    O0  T2_AT19 T2_AT1 T1_AT8 T1_AT7|
| |N7:15 N7:15  I:007  I:004    B3     B3  |
| +-] [---] [----]/[----]/[----]/[----]/[--+
|      1     0     01     07     35     31
Rung 4:158
|    O1    O0  T2_AT19 T2_AT1 T1_AT8                                     O2NEW
|  N7:15 N7:15  I:007  I:004    B3                                       N7:15
+-+-] [---] [----]/[----]/[--+-] [--++------------------------------------( )--
| |    1     0     01     07 |   35 ||                                       6
| |                          |T1_AT7||
| |                          |  B3  ||
| |                          +-] [--+|
| |                              31  |
| |  O2  T2_AT19 T2_AT1              |
| |N7:15  I:007  I:004               |
| +-] [----]/[----]/[----------------+
|      2     01     07
Rung 4:159
| O2NEW                                                                EN_T1O
| N7:15                                                                N7:15
+--]/[--------------------------------------------------------------+---( )--+-
|     6                                                             |     10 |
|                                                                   |T1R_SW3O|
|                                                                   | N7:15  |
|                                                                   +--( )---+
|                                                                        12
Rung 4:160
|  O1NEW                                                               EN_T2O
|  N7:15                                                               N7:15
+-+-]/[-+-----------------------------------------------------------+---( )--+-
| |    5|                                                           |     11 |
| |O0NEW|                                                           |T2R_SW3O|
| |N7:15|                                                           | N7:15  |
| +-] [-+                                                           +--( )---+
```

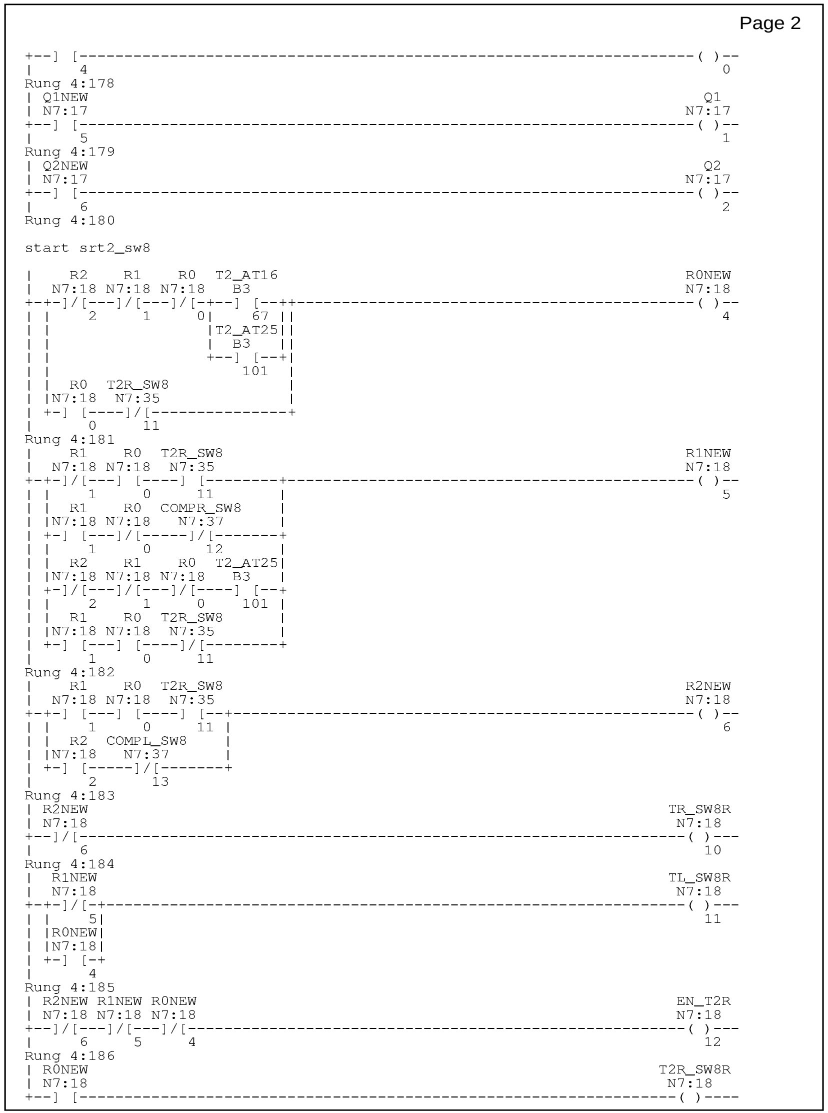
Page 2

```
|      4                                                                 13
Rung 4:161
| O0NEW                                                                    O0
| N7:15                                                                  N7:15
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:162
| O1NEW                                                                    O1
| N7:15                                                                  N7:15
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:163
| O2NEW                                                                    O2
| N7:15                                                                  N7:15
+--] [--------------------------------------------------------------------( )--
|     6                                                                      2
Rung 4:164

start cp_rs8

|    P2    P1    P0  T1_AT8                                              P0NEW
|  N7:16 N7:16 N7:16   B3                                                N7:16
+-+-]/[---]/[---]/[-+-] [--+-+--------------------------------------------( )--
| |    2     1     0|   35 | |                                               4
| |                 |T2_AT8| |
| |                 |  B3  | |
| |                 +-] [--+ |
| |                     33   |
| |  P1    P0  T2_AT12 T1_AT5|
| |N7:16 N7:16   B3    I:004 |
| +-]/[---] [----]/[----]/[--+
| |    1     0     49     13 |
| |  P1    P0  T1_AT12 T2_AT5|
| |N7:16 N7:16   B3    I:004 |
| +-] [---] [----]/[----]/[--+
|      1     0     51     12
Rung 4:165
|    P1    P0  T2_AT12 T1_AT5                                            P1NEW
|  N7:16 N7:16   B3    I:004                                             N7:16
+-+-]/[---] [----] [----]/[--+--------------------------------------------( )--
| |    1     0     49     13 |                                               5
| |  P1    P0  T1_AT5        |
| |N7:16 N7:16 I:004         |
| +-] [---]/[---]/[----------+
| |    1     0    13         |
| |  P2    P1    P0  T2_AT8  |
| |N7:16 N7:16 N7:16   B3    |
| +-]/[---]/[---]/[---] [----+
| |    2     1     0    33   |
| |  P1    P0  T1_AT12 T2_AT5|
| |N7:16 N7:16   B3    I:004 |
| +-] [---] [----]/[----]/[--+
|      1     0     51     12
Rung 4:166
|    P1    P0  T1_AT12 T2_AT5                                            P2NEW
|  N7:16 N7:16   B3    I:004                                             N7:16
+-+-] [---] [----] [----]/[--+--------------------------------------------( )--
| |    1     0     51     12 |                                               6
| |  P2  T2_AT5              |
| |N7:16 I:004               |
| +-] [---]/[----------------+
|      2    12
Rung 4:167
| P2NEW                                                                EN_T1P
| N7:16                                                                N7:16
+--]/[------------------------------------------------------------------( )---
|     6                                                                   10
Rung 4:168
|  P1NEW                                                               EN_T2P
|  N7:16                                                               N7:16
+-+-]/[-+---------------------------------------------------------------( )---
| |    5|                                                                 11
| |P0NEW|
| |N7:16|
| +-] [-+
|      4
Rung 4:169
```



Page 1

```
| P0NEW                                                                    P0
| N7:16                                                                  N7:16
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:170
| P1NEW                                                                    P1
| N7:16                                                                  N7:16
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:171
| P2NEW                                                                    P2
| N7:16                                                                  N7:16
+--] [--------------------------------------------------------------------( )--
|     6                                                                      2
Rung 4:172

start cp_rs27

|    Q2    Q1    Q0  T1_AT27                                             Q0NEW
|  N7:17 N7:17 N7:17   B3                                                N7:17
+-+-]/[---]/[---]/[-+--] [--++--------------------------------------------( )--
| |    2     1     0|   109 ||                                               4
| |                |T2_AT27||
| |                |  B3   ||
| |                +--] [--+|
| |                    111  |
| |  Q1    Q0  T2_AT4 T1_AT11|
| |N7:17 N7:17   B3    I:005 |
| +-]/[---] [---]/[-----]/[--+
| |    1     0    19      04 |
| |  Q1    Q0  T1_AT4 T2_AT11|
| |N7:17 N7:17   B3    I:005 |
| +-] [---] [---]/[-----]/[--+
|      1     0    17      05
Rung 4:173
|    Q1    Q0  T2_AT4 T1_AT11                                            Q1NEW
|  N7:17 N7:17   B3    I:005                                             N7:17
+-+-]/[---] [---] [-----]/[--+--------------------------------------------( )--
| |    1     0    19      04 |                                               5
| |  Q1    Q0  T1_AT11       |
| |N7:17 N7:17  I:005        |
| +-] [---]/[----]/[---------+
| |    1     0     04        |
| |  Q2    Q1    Q0  T2_AT27 |
| |N7:17 N7:17 N7:17   B3    |
| +-]/[---]/[---]/[----] [---+
| |    2     1     0    111  |
| |  Q1    Q0  T1_AT4 T2_AT11|
| |N7:17 N7:17   B3    I:005 |
| +-] [---] [---]/[-----]/[--+
|      1     0    17      05
Rung 4:174
|    Q1    Q0  T1_AT4 T2_AT11                                            Q2NEW
|  N7:17 N7:17   B3    I:005                                             N7:17
+-+-] [---] [---] [-----]/[--+--------------------------------------------( )--
| |    1     0    17      05 |                                               6
| |  Q2  T2_AT11             |
| |N7:17  I:005              |
| +-] [----]/[---------------+
|      2     05
Rung 4:175
| Q2NEW                                                                 EN_T1Q
| N7:17                                                                 N7:17
+--]/[-------------------------------------------------------------------( )---
|     6                                                                    10
Rung 4:176
|  Q1NEW                                                                EN_T2Q
|  N7:17                                                                N7:17
+-+-]/[-+----------------------------------------------------------------( )---
| |    5|                                                                  11
| |Q0NEW|
| |N7:17|
| +-] [-+
|      4
Rung 4:177
| Q0NEW                                                                    Q0
| N7:17                                                                  N7:17
```



Page 2

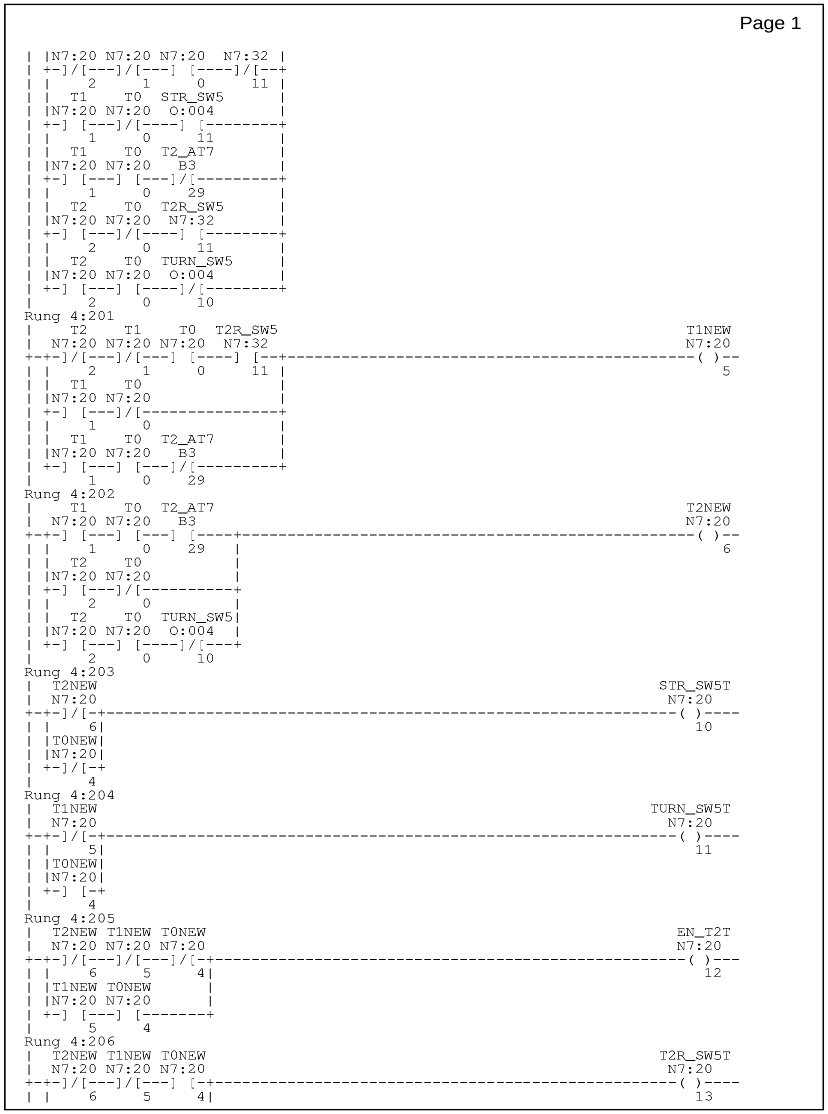
```
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:178
| Q1NEW                                                                    Q1
| N7:17                                                                  N7:17
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:179
| Q2NEW                                                                    Q2
| N7:17                                                                  N7:17
+--] [--------------------------------------------------------------------( )--
|     6                                                                      2
Rung 4:180

start srt2_sw8

|    R2    R1    R0  T2_AT16                                             R0NEW
|  N7:18 N7:18 N7:18   B3                                                N7:18
+-+-]/[---]/[---]/[-+--] [--++--------------------------------------------( )--
| |    2     1     0|    67 ||                                               4
| |                |T2_AT25||
| |                |  B3   ||
| |                +--] [--+|
| |                    101  |
| |  R0  T2R_SW8             |
| |N7:18  N7:35              |
| +-] [----]/[---------------+
|      0     11
Rung 4:181
|    R1    R0  T2R_SW8                                                   R1NEW
|  N7:18 N7:18  N7:35                                                    N7:18
+-+-]/[---] [----] [--------+---------------------------------------------( )--
| |    1     0     11       |                                                5
| |  R1    R0  COMPR_SW8    |
| |N7:18 N7:18   N7:37      |
| +-] [---]/[-----]/[-------+
| |    1     0      12      |
| |  R2    R1    R0  T2_AT25|
| |N7:18 N7:18 N7:18   B3   |
| +-]/[---]/[---]/[----] [--+
| |    2     1     0    101 |
| |  R1    R0  T2R_SW8      |
| |N7:18 N7:18  N7:35       |
| +-] [---] [----]/[--------+
|      1     0     11
Rung 4:182
|    R1    R0  T2R_SW8                                                   R2NEW
|  N7:18 N7:18  N7:35                                                    N7:18
+-+-] [---] [----] [--+---------------------------------------------------( )--
| |    1     0     11 |                                                      6
| |  R2  COMPL_SW8    |
| |N7:18   N7:37      |
| +-] [-----]/[-------+
|      2      13
Rung 4:183
| R2NEW                                                                TR_SW8R
| N7:18                                                                 N7:18
+--]/[-------------------------------------------------------------------( )---
|     6                                                                    10
Rung 4:184
|  R1NEW                                                               TL_SW8R
|  N7:18                                                                N7:18
+-+-]/[-+----------------------------------------------------------------( )---
| |    5|                                                                  11
| |R0NEW|
| |N7:18|
| +-] [-+
|      4
Rung 4:185
| R2NEW R1NEW R0NEW                                                     EN_T2R
| N7:18 N7:18 N7:18                                                     N7:18
+--]/[---]/[---]/[-------------------------------------------------------( )---
|     6     5     4                                                        12
Rung 4:186
| R0NEW                                                               T2R_SW8R
| N7:18                                                                N7:18
+--] [------------------------------------------------------------------( )----
```

Page 1

```
|     4                                                                   13
Rung 4:187
| R0NEW                                                                    R0
| N7:18                                                                  N7:18
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:188
| R1NEW                                                                    R1
| N7:18                                                                  N7:18
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:189
| R2NEW                                                                    R2
| N7:18                                                                  N7:18
+--] [--------------------------------------------------------------------( )--
|     6                                                                      2
Rung 4:190

start srt2_sw7

|    S2    S1    S0  T2_AT14                                             S0NEW
|  N7:19 N7:19 N7:19  I:006                                              N7:19
+-+-]/[---]/[---]/[----] [---+--------------------------------------------( )--
| |    2     1     0     03  |                                               4
| |  S2    S1    S0  T2_AT12 |
| |N7:19 N7:19 N7:19   B3    |
| +-]/[---]/[---] [----]/[---+
| |    2     1     0     49  |
| |  S2    S1    S0  T2R_SW7 |
| |N7:19 N7:19 N7:19  N7:34  |
| +-]/[---] [---]/[----] [---+
| |    2     1     0     11  |
| |  S2    S1    S0  TURN_SW7|
| |N7:19 N7:19 N7:19  O:004  |
| +-]/[---] [---] [----]/[---+
| |    2     1     0     14  |
| |  S2    S1    S0  T2_AT14 |
| |N7:19 N7:19 N7:19  I:006  |
| +-] [---]/[---]/[----] [---+
| |    2     1     0     03  |
| |  S2    S1    S0  T2_AT12 |
| |N7:19 N7:19 N7:19   B3    |
| +-] [---]/[---] [----]/[---+
| |    2     1     0     49  |
| |  S2    S1    S0  T2R_SW7 |
| |N7:19 N7:19 N7:19  N7:34  |
| +-] [---] [---]/[----] [---+
| |    2     1     0     11  |
| |  S2    S1    S0  STR_SW7 |
| |N7:19 N7:19 N7:19  O:004  |
| +-] [---] [---] [----]/[---+
|      2     1     0     17
Rung 4:191
|    S2    S1    S0  T2_AT12                                             S1NEW
|  N7:19 N7:19 N7:19   B3                                                N7:19
+-+-]/[---]/[---] [----] [---+--------------------------------------------( )--
| |    2     1     0     49  |                                               5
| |  S2    S1    S0          |
| |N7:19 N7:19 N7:19         |
| +-]/[---] [---]/[----------+
| |    2     1     0         |
| |  S2    S1    S0  TURN_SW7|
| |N7:19 N7:19 N7:19  O:004  |
| +-]/[---] [---] [----]/[---+
| |    2     1     0     14  |
| |  S2    S1    S0  T2_AT12 |
| |N7:19 N7:19 N7:19   B3    |
| +-] [---]/[---] [----] [---+
| |    2     1     0     49  |
| |  S2    S1    S0          |
| |N7:19 N7:19 N7:19         |
| +-] [---] [---]/[----------+
| |    2     1     0         |
| |  S2    S1    S0  STR_SW7 |
| |N7:19 N7:19 N7:19  O:004  |
| +-] [---] [---] [----]/[---+
|      2     1     0     17
```

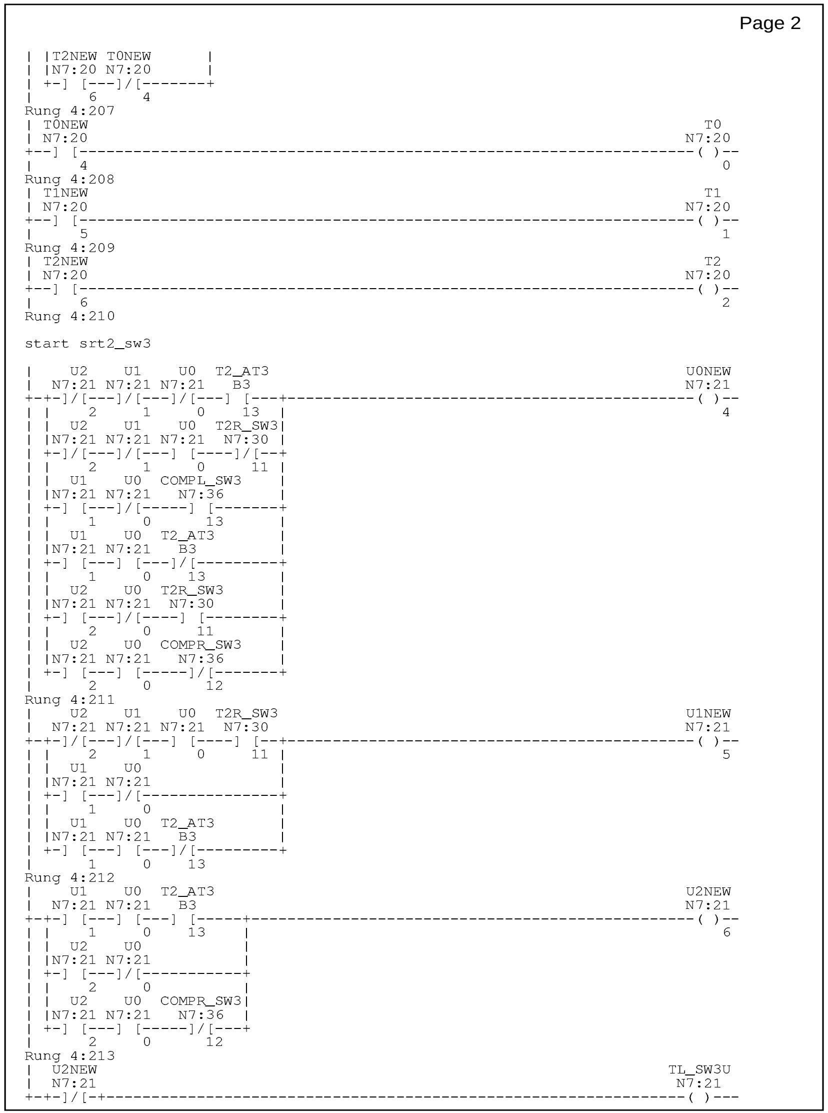
Page 2

```
Rung 4:192
|    S2    S1    S0  TURN_SW7                                             S2NEW
|  N7:19 N7:19 N7:19  O:004                                               N7:19
+-+-]/[---] [---] [----] [---+---------------------------------------------( )--
| |    2     1     0     14  |                                                6
| |  S2    S1                |
| |N7:19 N7:19               |
| +-] [-+-]/[-+--------------+
| |    2|    1|              |
| |     |  S0 |              |
| |     |N7:19|              |
| |     +-]/[-+              |
| |          0               |
| |  S2    S1    S0  STR_SW7 |
| |N7:19 N7:19 N7:19  O:004  |
| +-] [---] [---] [----]/[---+
|      2     1     0     17
Rung 4:193
|  S2NEW                                                               STR_SW7S
|  N7:19                                                                N7:19
+-+-] [-+----------------------------------------------------------------( )----
| |    6|                                                                  10
| |S1NEW|
| |N7:19|
| +-]/[-+
| |    5|
| |S0NEW|
| |N7:19|
| +-]/[-+
|      4
Rung 4:194
|  S2NEW                                                              TURN_SW7S
|  N7:19                                                                N7:19
+-+-]/[-+----------------------------------------------------------------( )----
| |    6|                                                                  11
| |S1NEW|
| |N7:19|
| +-]/[-+
| |    5|
| |S0NEW|
| |N7:19|
| +-]/[-+
|      4
Rung 4:195
| S1NEW                                                                  EN_T2S
| N7:19                                                                  N7:19
+--]/[--------------------------------------------------------------------( )---
|     5                                                                     12
Rung 4:196
| S1NEW S0NEW                                                         T2R_SW7S
| N7:19 N7:19                                                           N7:19
+--] [---]/[-------------------------------------------------------------( )----
|     5     4                                                              13
Rung 4:197
| S0NEW                                                                     S0
| N7:19                                                                   N7:19
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:198
| S1NEW                                                                     S1
| N7:19                                                                   N7:19
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:199
| S2NEW                                                                     S2
| N7:19                                                                   N7:19
+--] [---------------------------------------------------------------------( )--
|     6                                                                       2
Rung 4:200

start srt2_sw5

|    T2    T1    T0  T2_AT8                                               T0NEW
|  N7:20 N7:20 N7:20   B3                                                 N7:20
+-+-]/[---]/[---]/[---] [---+----------------------------------------------( )--
| |    2     1     0    33  |                                                 4
| |  T2    T1    T0  T2R_SW5|
```



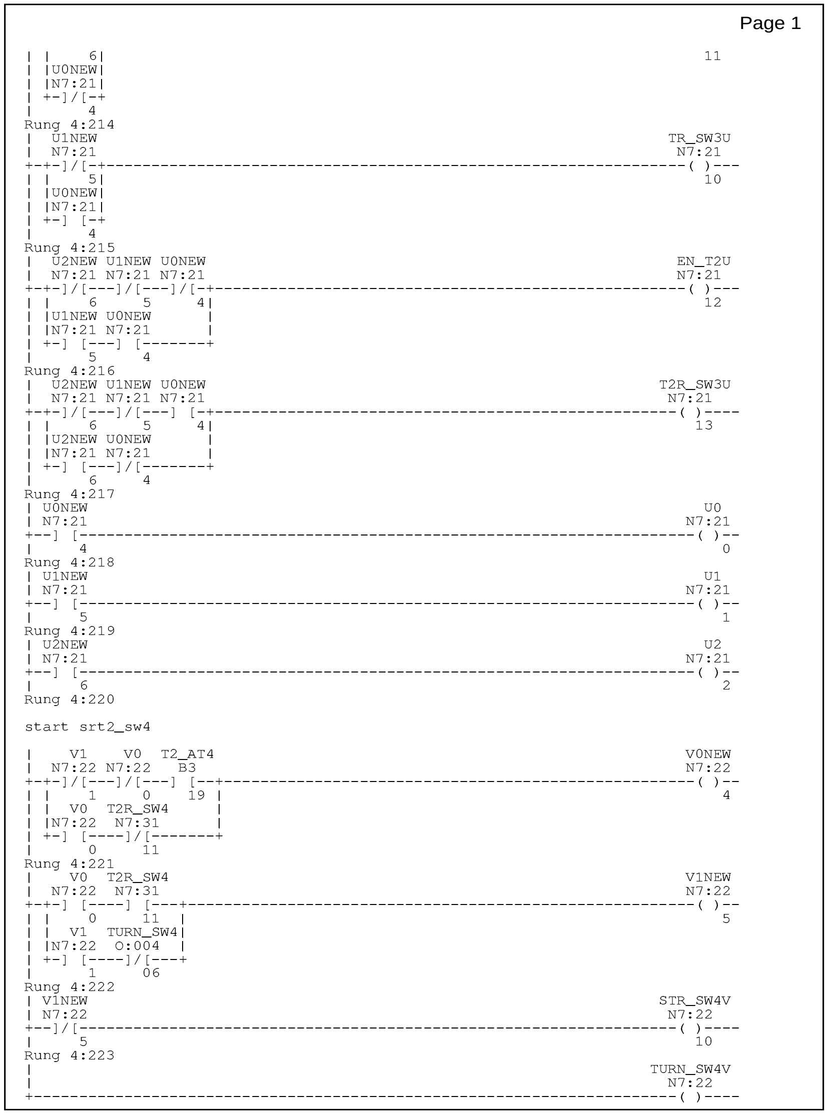
Page 1

```
| |N7:20 N7:20 N7:20  N7:32 |
| +-]/[---]/[---] [----]/[--+
| |    2     1     0     11 |
| |  T1    T0  STR_SW5      |
| |N7:20 N7:20  O:004       |
| +-] [---]/[----] [--------+
| |    1     0     11       |
| |  T1    T0  T2_AT7       |
| |N7:20 N7:20   B3         |
| +-] [---] [---]/[---------+
| |    1     0    29        |
| |  T2    T0  T2R_SW5      |
| |N7:20 N7:20  N7:32       |
| +-] [---]/[----] [--------+
| |    2     0     11       |
| |  T2    T0  TURN_SW5     |
| |N7:20 N7:20  O:004       |
| +-] [---] [----]/[--------+
|      2     0     10
Rung 4:201
|    T2    T1    T0  T2R_SW5                                             T1NEW
|  N7:20 N7:20 N7:20  N7:32                                              N7:20
+-+-]/[---]/[---] [----] [--+---------------------------------------------( )--
| |    2     1     0     11 |                                                5
| |  T1    T0               |
| |N7:20 N7:20              |
| +-] [---]/[---------------+
| |    1     0              |
| |  T1    T0  T2_AT7       |
| |N7:20 N7:20   B3         |
| +-] [---] [---]/[---------+
|      1     0    29
Rung 4:202
|    T1    T0  T2_AT7                                                    T2NEW
|  N7:20 N7:20   B3                                                      N7:20
+-+-] [---] [---] [----+--------------------------------------------------( )--
| |    1     0    29   |                                                     6
| |  T2    T0          |
| |N7:20 N7:20         |
| +-] [---]/[----------+
| |    2     0         |
| |  T2    T0  TURN_SW5|
| |N7:20 N7:20  O:004  |
| +-] [---] [----]/[---+
|      2     0     10
Rung 4:203
|  T2NEW                                                              STR_SW5T
|  N7:20                                                               N7:20
+-+-]/[-+---------------------------------------------------------------( )----
| |    6|                                                                 10
| |T0NEW|
| |N7:20|
| +-]/[-+
|      4
Rung 4:204
|  T1NEW                                                             TURN_SW5T
|  N7:20                                                               N7:20
+-+-]/[-+---------------------------------------------------------------( )----
| |    5|                                                                 11
| |T0NEW|
| |N7:20|
| +-] [-+
|      4
Rung 4:205
|  T2NEW T1NEW T0NEW                                                    EN_T2T
|  N7:20 N7:20 N7:20                                                    N7:20
+-+-]/[---]/[---]/[-+----------------------------------------------------( )---
| |    6     5     4|                                                      12
| |T1NEW T0NEW      |
| |N7:20 N7:20      |
| +-] [---] [-------+
|      5     4
Rung 4:206
|  T2NEW T1NEW T0NEW                                                  T2R_SW5T
|  N7:20 N7:20 N7:20                                                   N7:20
+-+-]/[---]/[---] [-+---------------------------------------------------( )----
| |    6     5     4|                                                     13
```



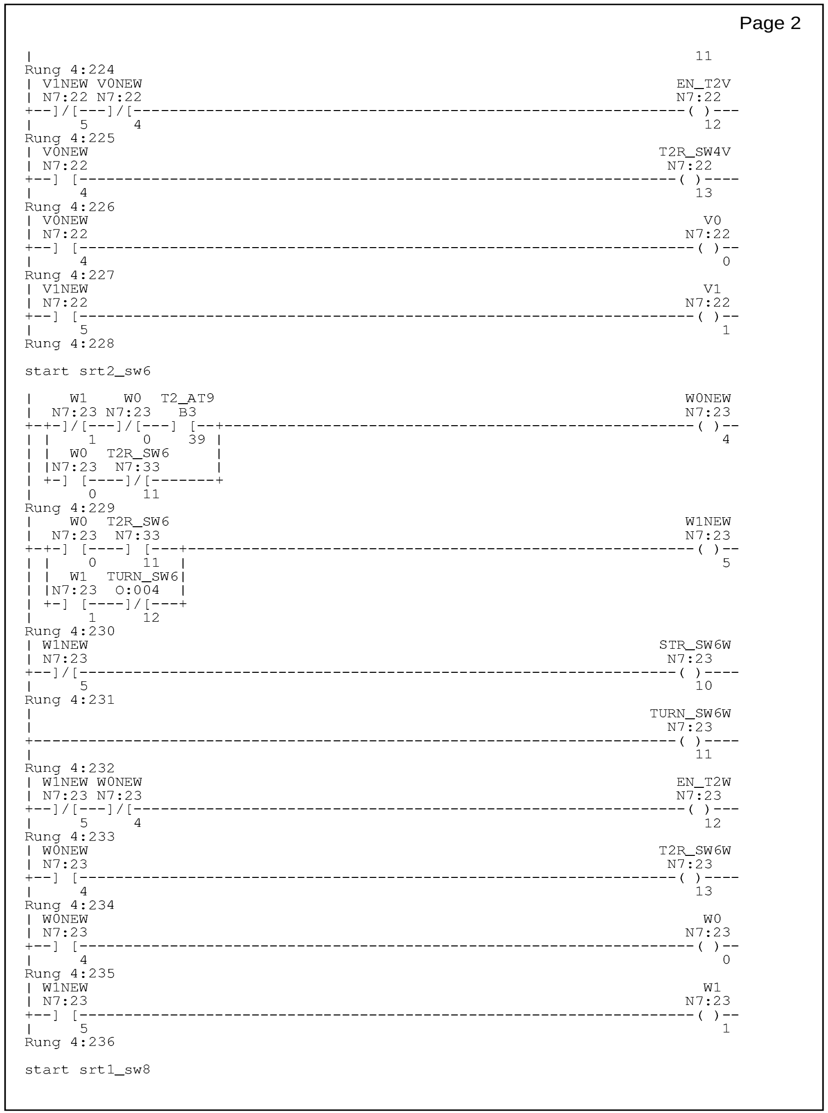
Page 2

```
| |T2NEW T0NEW      |
| |N7:20 N7:20      |
| +-] [---]/[-------+
|      6     4
Rung 4:207
| T0NEW                                                                    T0
| N7:20                                                                  N7:20
+--] [--------------------------------------------------------------------( )--
|    4                                                                       0
Rung 4:208
| T1NEW                                                                    T1
| N7:20                                                                  N7:20
+--] [--------------------------------------------------------------------( )--
|    5                                                                       1
Rung 4:209
| T2NEW                                                                    T2
| N7:20                                                                  N7:20
+--] [--------------------------------------------------------------------( )--
|    6                                                                       2
Rung 4:210

start srt2_sw3

|    U2    U1    U0  T2_AT3                                              U0NEW
|  N7:21 N7:21 N7:21   B3                                                N7:21
+-+-]/[---]/[---]/[---] [---+---------------------------------------------( )--
| |    2     1     0    13  |                                                4
| |  U2    U1    U0  T2R_SW3|
| |N7:21 N7:21 N7:21  N7:30 |
| +-]/[---]/[---] [----]/[--+
| |    2     1     0     11 |
| |  U1    U0  COMPL_SW3    |
| |N7:21 N7:21   N7:36      |
| +-] [---]/[-----] [-------+
| |    1     0      13      |
| |  U1    U0  T2_AT3       |
| |N7:21 N7:21   B3         |
| +-] [---] [---]/[---------+
| |    1     0    13        |
| |  U2    U0  T2R_SW3      |
| |N7:21 N7:21  N7:30       |
| +-] [---]/[----] [--------+
| |    2     0     11       |
| |  U2    U0  COMPR_SW3    |
| |N7:21 N7:21   N7:36      |
| +-] [---] [-----]/[-------+
|      2     0      12
Rung 4:211
|    U2    U1    U0  T2R_SW3                                             U1NEW
|  N7:21 N7:21 N7:21  N7:30                                              N7:21
+-+-]/[---]/[---] [----] [--+---------------------------------------------( )--
| |    2     1     0     11 |                                                5
| |  U1    U0               |
| |N7:21 N7:21              |
| +-] [---]/[---------------+
| |    1     0              |
| |  U1    U0  T2_AT3       |
| |N7:21 N7:21   B3         |
| +-] [---] [---]/[---------+
|      1     0    13
Rung 4:212
|    U1    U0  T2_AT3                                                    U2NEW
|  N7:21 N7:21   B3                                                      N7:21
+-+-] [---] [---] [-----+-------------------------------------------------( )--
| |    1     0    13    |                                                    6
| |  U2    U0           |
| |N7:21 N7:21          |
| +-] [---]/[-----------+
| |    2     0          |
| |  U2    U0  COMPR_SW3|
| |N7:21 N7:21   N7:36  |
| +-] [---] [-----]/[---+
|      2     0      12
Rung 4:213
|  U2NEW                                                               TL_SW3U
|  N7:21                                                                N7:21
+-+-]/[-+----------------------------------------------------------------( )---
```



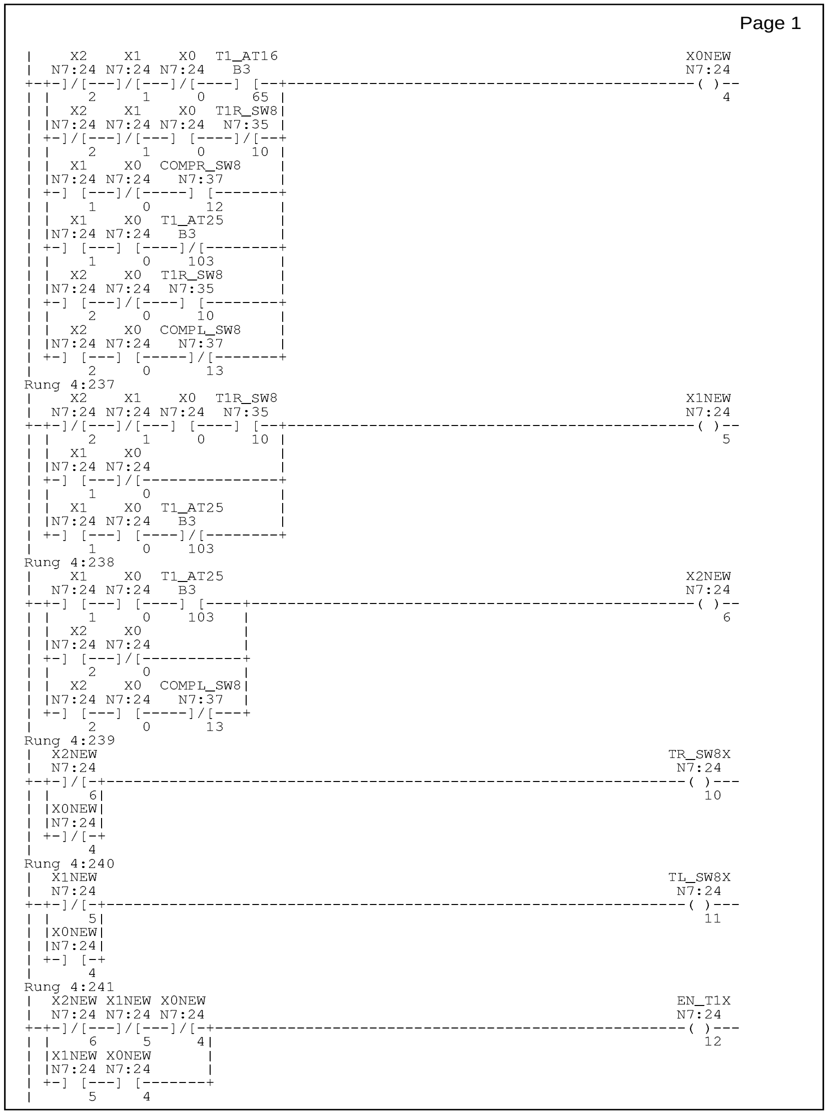
Page 1

```
| |    6|                                                                  11
| |U0NEW|
| |N7:21|
| +-]/[-+
|      4
Rung 4:214
|  U1NEW                                                               TR_SW3U
|  N7:21                                                                N7:21
+-+-]/[-+-----------------------------------------------------------------( )---
| |    5|                                                                  10
| |U0NEW|
| |N7:21|
| +-] [-+
|      4
Rung 4:215
|  U2NEW U1NEW U0NEW                                                    EN_T2U
|  N7:21 N7:21 N7:21                                                    N7:21
+-+-]/[---]/[---]/[-+-----------------------------------------------------( )---
| |    6     5     4|                                                      12
| |U1NEW U0NEW      |
| |N7:21 N7:21      |
| +-] [---] [-------+
|      5     4
Rung 4:216
|  U2NEW U1NEW U0NEW                                                  T2R_SW3U
|  N7:21 N7:21 N7:21                                                   N7:21
+-+-]/[---]/[---] [-+----------------------------------------------------( )----
| |    6     5     4|                                                     13
| |U2NEW U0NEW      |
| |N7:21 N7:21      |
| +-] [---]/[-------+
|      6     4
Rung 4:217
| U0NEW                                                                     U0
| N7:21                                                                   N7:21
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:218
| U1NEW                                                                     U1
| N7:21                                                                   N7:21
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:219
| U2NEW                                                                     U2
| N7:21                                                                   N7:21
+--] [---------------------------------------------------------------------( )--
|     6                                                                       2
Rung 4:220

start srt2_sw4

|    V1    V0  T2_AT4                                                     V0NEW
|  N7:22 N7:22   B3                                                       N7:22
+-+-]/[---]/[---] [--+-----------------------------------------------------( )--
| |    1     0    19 |                                                        4
| |  V0  T2R_SW4     |
| |N7:22  N7:31      |
| +-] [----]/[-------+
|      0     11
Rung 4:221
|    V0  T2R_SW4                                                          V1NEW
|  N7:22  N7:31                                                           N7:22
+-+-] [----] [---+---------------------------------------------------------( )--
| |    0     11  |                                                            5
| |  V1  TURN_SW4|
| |N7:22  O:004  |
| +-] [----]/[---+
|      1     06
Rung 4:222
| V1NEW                                                                STR_SW4V
| N7:22                                                                 N7:22
+--]/[-------------------------------------------------------------------( )----
|     5                                                                    10
Rung 4:223
|                                                                     TURN_SW4V
|                                                                       N7:22
+------------------------------------------------------------------------( )----
```



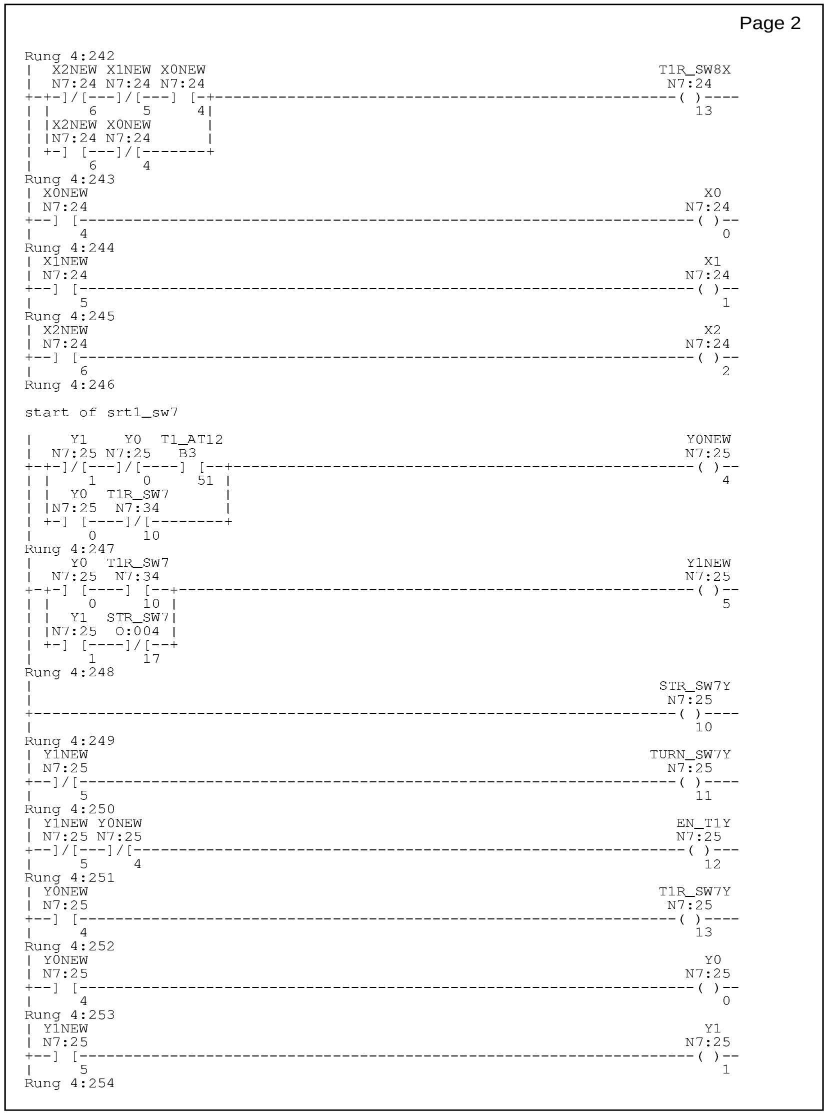
Page 2

```
|                                                                          11
Rung 4:224
| V1NEW V0NEW                                                           EN_T2V
| N7:22 N7:22                                                           N7:22
+--]/[---]/[--------------------------------------------------------------( )---
|     5     4                                                               12
Rung 4:225
| V0NEW                                                                T2R_SW4V
| N7:22                                                                 N7:22
+--] [-------------------------------------------------------------------( )----
|     4                                                                    13
Rung 4:226
| V0NEW                                                                     V0
| N7:22                                                                   N7:22
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:227
| V1NEW                                                                     V1
| N7:22                                                                   N7:22
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:228

start srt2_sw6

|    W1    W0  T2_AT9                                                     W0NEW
|  N7:23 N7:23   B3                                                       N7:23
+-+-]/[---]/[---] [--+-----------------------------------------------------( )--
| |    1     0    39 |                                                        4
| |  W0  T2R_SW6     |
| |N7:23  N7:33      |
| +-] [----]/[-------+
|      0     11
Rung 4:229
|    W0  T2R_SW6                                                          W1NEW
|  N7:23  N7:33                                                           N7:23
+-+-] [----] [---+---------------------------------------------------------( )--
| |    0     11  |                                                            5
| |  W1  TURN_SW6|
| |N7:23  O:004  |
| +-] [----]/[---+
|      1     12
Rung 4:230
| W1NEW                                                                STR_SW6W
| N7:23                                                                 N7:23
+--]/[-------------------------------------------------------------------( )----
|     5                                                                    10
Rung 4:231
|                                                                     TURN_SW6W
|                                                                       N7:23
+------------------------------------------------------------------------( )----
|                                                                          11
Rung 4:232
| W1NEW W0NEW                                                           EN_T2W
| N7:23 N7:23                                                           N7:23
+--]/[---]/[--------------------------------------------------------------( )---
|     5     4                                                               12
Rung 4:233
| W0NEW                                                                T2R_SW6W
| N7:23                                                                 N7:23
+--] [-------------------------------------------------------------------( )----
|     4                                                                    13
Rung 4:234
| W0NEW                                                                     W0
| N7:23                                                                   N7:23
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:235
| W1NEW                                                                     W1
| N7:23                                                                   N7:23
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:236

start srt1_sw8
```



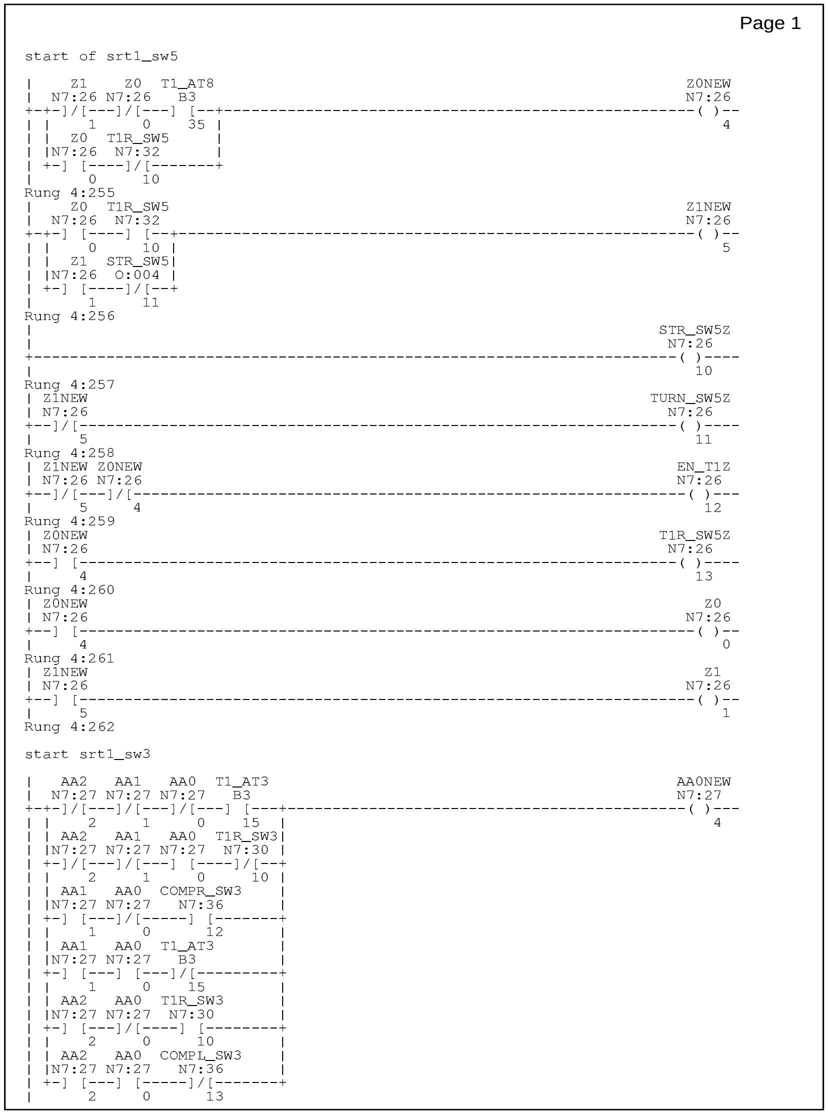
Page 1

```
|    X2    X1    X0  T1_AT16                                              X0NEW
|  N7:24 N7:24 N7:24   B3                                                 N7:24
+-+-]/[---]/[---]/[----] [--+----------------------------------------------( )--
| |    2     1     0     65 |                                                 4
| |  X2    X1    X0  T1R_SW8|
| |N7:24 N7:24 N7:24  N7:35 |
| +-]/[---]/[---] [----]/[--+
| |    2     1     0     10 |
| |  X1    X0  COMPR_SW8    |
| |N7:24 N7:24   N7:37      |
| +-] [---]/[-----] [-------+
| |    1     0      12      |
| |  X1    X0  T1_AT25      |
| |N7:24 N7:24   B3         |
| +-] [---] [----]/[--------+
| |    1     0    103       |
| |  X2    X0  T1R_SW8      |
| |N7:24 N7:24  N7:35       |
| +-] [---]/[----] [--------+
| |    2     0     10       |
| |  X2    X0  COMPL_SW8    |
| |N7:24 N7:24   N7:37      |
| +-] [---] [-----]/[-------+
|      2     0      13
Rung 4:237
|    X2    X1    X0  T1R_SW8                                              X1NEW
|  N7:24 N7:24 N7:24  N7:35                                               N7:24
+-+-]/[---]/[---] [----] [--+----------------------------------------------( )--
| |    2     1     0     10 |                                                 5
| |  X1    X0               |
| |N7:24 N7:24              |
| +-] [---]/[---------------+
| |    1     0              |
| |  X1    X0  T1_AT25      |
| |N7:24 N7:24   B3         |
| +-] [---] [----]/[--------+
|      1     0    103
Rung 4:238
|    X1    X0  T1_AT25                                                    X2NEW
|  N7:24 N7:24   B3                                                       N7:24
+-+-] [---] [----] [----+--------------------------------------------------( )--
| |    1     0    103   |                                                     6
| |  X2    X0           |
| |N7:24 N7:24          |
| +-] [---]/[-----------+
| |    2     0          |
| |  X2    X0  COMPL_SW8|
| |N7:24 N7:24   N7:37  |
| +-] [---] [-----]/[---+
|      2     0      13
Rung 4:239
|  X2NEW                                                               TR_SW8X
|  N7:24                                                                N7:24
+-+-]/[-+----------------------------------------------------------------( )---
| |    6|                                                                  10
| |X0NEW|
| |N7:24|
| +-]/[-+
|      4
Rung 4:240
|  X1NEW                                                               TL_SW8X
|  N7:24                                                                N7:24
+-+-]/[-+----------------------------------------------------------------( )---
| |    5|                                                                  11
| |X0NEW|
| |N7:24|
| +-] [-+
|      4
Rung 4:241
|  X2NEW X1NEW X0NEW                                                    EN_T1X
|  N7:24 N7:24 N7:24                                                    N7:24
+-+-]/[---]/[---]/[-+----------------------------------------------------( )---
| |    6     5     4|                                                      12
| |X1NEW X0NEW      |
| |N7:24 N7:24      |
| +-] [---] [-------+
|      5     4
```



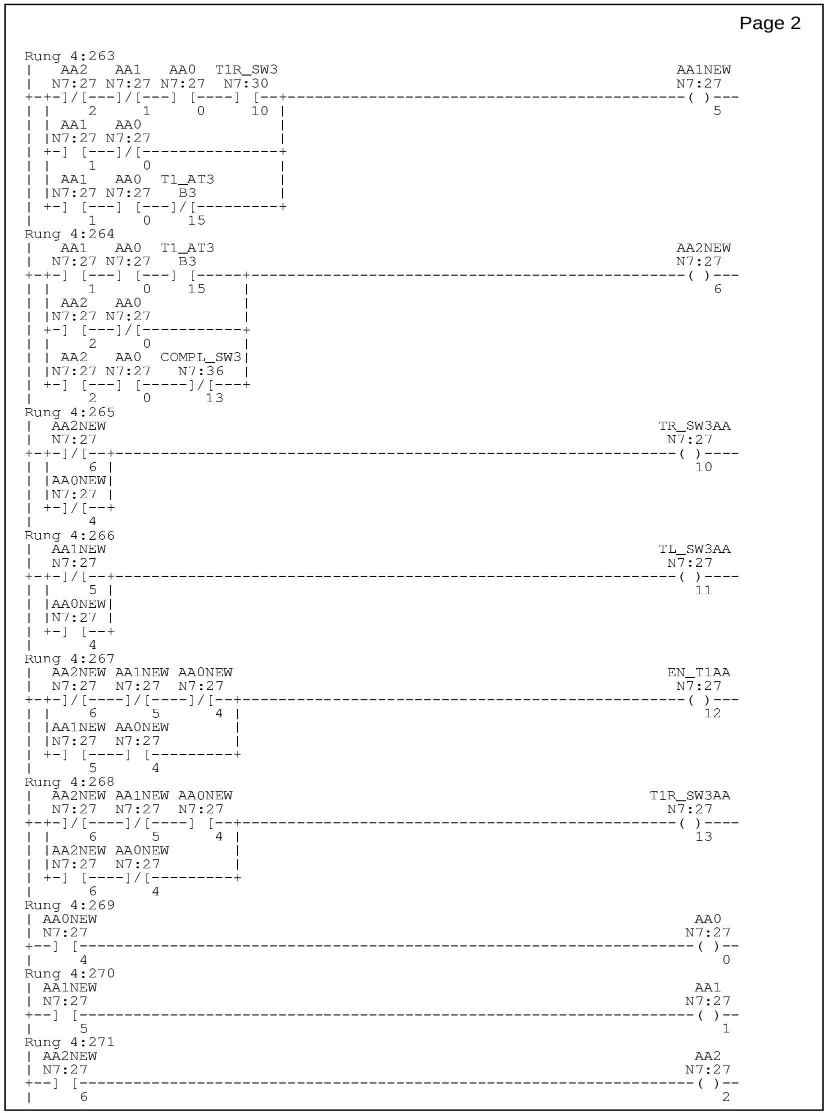
Page 2

```
Rung 4:242
|  X2NEW X1NEW X0NEW                                                   T1R_SW8X
|  N7:24 N7:24 N7:24                                                    N7:24
+-+-]/[---]/[---] [-+----------------------------------------------------( )----
| |    6     5     4|                                                      13
| |X2NEW X0NEW      |
| |N7:24 N7:24      |
| +-] [---]/[-------+
|      6     4
Rung 4:243
| X0NEW                                                                     X0
| N7:24                                                                   N7:24
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:244
| X1NEW                                                                     X1
| N7:24                                                                   N7:24
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:245
| X2NEW                                                                     X2
| N7:24                                                                   N7:24
+--] [---------------------------------------------------------------------( )--
|     6                                                                       2
Rung 4:246

start of srt1_sw7

|    Y1    Y0  T1_AT12                                                    Y0NEW
|  N7:25 N7:25   B3                                                       N7:25
+-+-]/[---]/[----] [--+----------------------------------------------------( )--
| |    1     0     51 |                                                       4
| |  Y0  T1R_SW7      |
| |N7:25  N7:34       |
| +-] [----]/[--------+
|      0     10
Rung 4:247
|    Y0  T1R_SW7                                                          Y1NEW
|  N7:25  N7:34                                                           N7:25
+-+-] [----] [--+----------------------------------------------------------( )--
| |    0     10 |                                                             5
| |  Y1  STR_SW7|
| |N7:25  O:004 |
| +-] [----]/[--+
|      1     17
Rung 4:248
|                                                                      STR_SW7Y
|                                                                       N7:25
+------------------------------------------------------------------------( )----
|                                                                          10
Rung 4:249
| Y1NEW                                                               TURN_SW7Y
| N7:25                                                                 N7:25
+--]/[-------------------------------------------------------------------( )----
|     5                                                                    11
Rung 4:250
| Y1NEW Y0NEW                                                           EN_T1Y
| N7:25 N7:25                                                           N7:25
+--]/[---]/[--------------------------------------------------------------( )---
|     5     4                                                               12
Rung 4:251
| Y0NEW                                                                T1R_SW7Y
| N7:25                                                                 N7:25
+--] [-------------------------------------------------------------------( )----
|     4                                                                    13
Rung 4:252
| Y0NEW                                                                     Y0
| N7:25                                                                   N7:25
+--] [---------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:253
| Y1NEW                                                                     Y1
| N7:25                                                                   N7:25
+--] [---------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:254
```



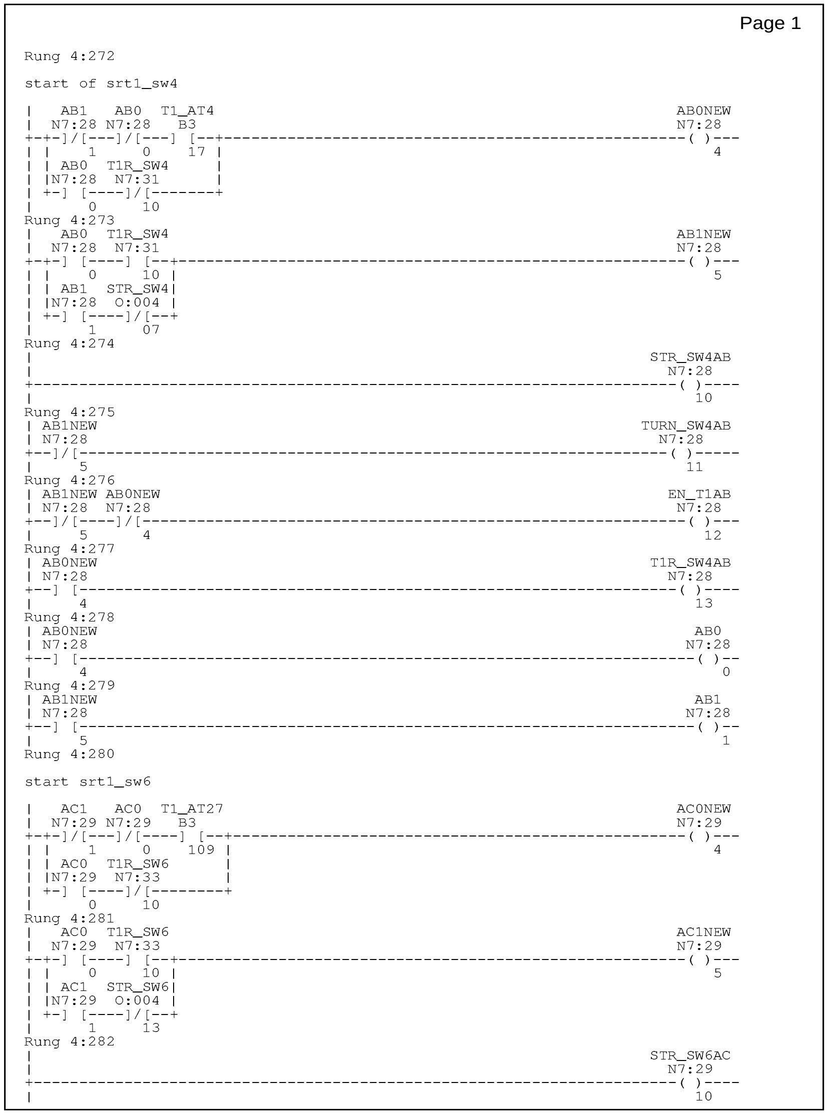
Page 1

```
start of srt1_sw5

|    Z1    Z0  T1_AT8                                                Z0NEW
|  N7:26 N7:26   B3                                                  N7:26
+-+-]/[---]/[---] [--+--------------------------------------------------( )--
| |    1     0   35 |                                                    4
| |  Z0  T1R_SW5    |
| |N7:26  N7:32     |
| +-] [----]/[-------+
|      0     10
Rung 4:255
|    Z0  T1R_SW5                                                     Z1NEW
|  N7:26  N7:32                                                      N7:26
+-+-] [----] [--+-------------------------------------------------------( )--
| |    0     10 |                                                        5
| |  Z1  STR_SW5|
| |N7:26  O:004 |
| +-] [----]/[--+
|      1     11
Rung 4:256
|                                                                 STR_SW5Z
|                                                                  N7:26
+---------------------------------------------------------------------( )----
|                                                                    10
Rung 4:257
| Z1NEW                                                          TURN_SW5Z
| N7:26                                                            N7:26
+--]/[----------------------------------------------------------------( )----
|     5                                                              11
Rung 4:258
| Z1NEW Z0NEW                                                      EN_T1Z
| N7:26 N7:26                                                      N7:26
+--]/[---]/[-----------------------------------------------------------( )---
|     5     4                                                        12
Rung 4:259
| Z0NEW                                                           T1R_SW5Z
| N7:26                                                             N7:26
+--] [----------------------------------------------------------------( )----
|     4                                                              13
Rung 4:260
| Z0NEW                                                                 Z0
| N7:26                                                               N7:26
+--] [-----------------------------------------------------------------( )--
|     4                                                                  0
Rung 4:261
| Z1NEW                                                                 Z1
| N7:26                                                               N7:26
+--] [-----------------------------------------------------------------( )--
|     5                                                                  1
Rung 4:262

start srt1_sw3

|   AA2   AA1   AA0  T1_AT3                                        AA0NEW
|  N7:27 N7:27 N7:27   B3                                          N7:27
+-+-]/[---]/[---]/[---] [---+------------------------------------------( )---
| |    2     1     0    15  |                                          4
| | AA2   AA1   AA0  T1R_SW3|
| |N7:27 N7:27 N7:27  N7:30 |
| +-]/[---]/[---] [----]/[--+
| |    2     1     0     10 |
| | AA1   AA0  COMPR_SW3    |
| |N7:27 N7:27   N7:36      |
| +-] [---]/[-----] [-------+
| |    1     0      12      |
| | AA1   AA0  T1_AT3       |
| |N7:27 N7:27   B3         |
| +-] [---] [---]/[---------+
| |    1     0    15        |
| | AA2   AA0  T1R_SW3      |
| |N7:27 N7:27  N7:30       |
| +-] [---]/[----] [--------+
| |    2     0     10       |
| | AA2   AA0  COMPL_SW3    |
| |N7:27 N7:27   N7:36      |
| +-] [---] [-----]/[-------+
|      2     0      13
```



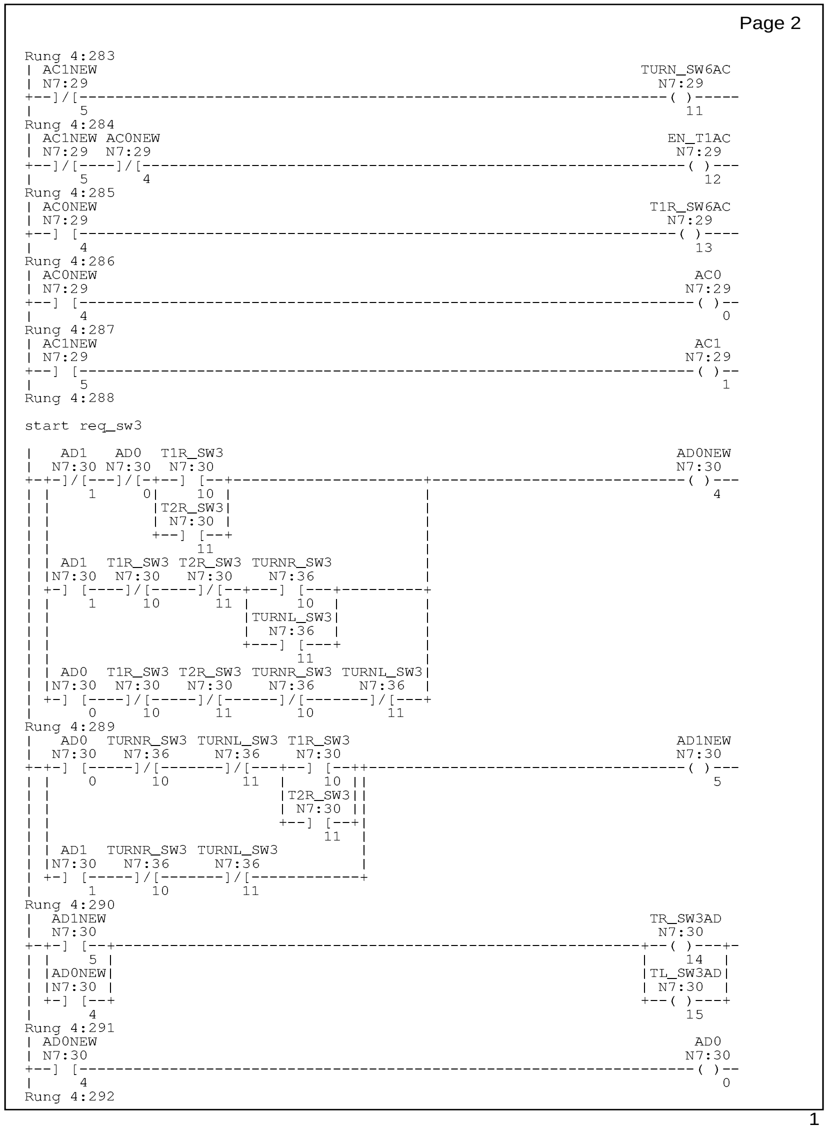
Page 2

```
Rung 4:263
|   AA2   AA1   AA0  T1R_SW3                                          AA1NEW
|  N7:27 N7:27 N7:27  N7:30                                           N7:27
+-+-]/[---]/[---] [----] [--+---------------------------------------------( )---
| |    2     1     0     10 |                                             5
| | AA1   AA0               |
| |N7:27 N7:27              |
| +-] [---]/[---------------+
| |    1     0              |
| | AA1   AA0  T1_AT3       |
| |N7:27 N7:27   B3         |
| +-] [---] [---]/[---------+
|      1     0    15
Rung 4:264
|   AA1   AA0  T1_AT3                                                 AA2NEW
|  N7:27 N7:27   B3                                                   N7:27
+-+-] [---] [---] [-----+-------------------------------------------------( )---
| |    1     0    15    |                                                 6
| | AA2   AA0           |
| |N7:27 N7:27          |
| +-] [---]/[-----------+
| |    2     0          |
| | AA2   AA0  COMPL_SW3|
| |N7:27 N7:27   N7:36  |
| +-] [---] [-----]/[---+
|      2     0      13
Rung 4:265
|  AA2NEW                                                            TR_SW3AA
|  N7:27                                                              N7:27
+-+-]/[--+---------------------------------------------------------------( )----
| |    6 |                                                             10
| |AA0NEW|
| |N7:27 |
| +-]/[--+
|      4
Rung 4:266
|  AA1NEW                                                            TL_SW3AA
|  N7:27                                                              N7:27
+-+-]/[--+---------------------------------------------------------------( )----
| |    5 |                                                             11
| |AA0NEW|
| |N7:27 |
| +-] [--+
|      4
Rung 4:267
|  AA2NEW AA1NEW AA0NEW                                               EN_T1AA
|  N7:27  N7:27  N7:27                                                 N7:27
+-+-]/[----]/[----]/[--+--------------------------------------------------( )---
| |    6      5      4 |                                                 12
| |AA1NEW AA0NEW       |
| |N7:27  N7:27        |
| +-] [----] [---------+
|      5      4
Rung 4:268
|  AA2NEW AA1NEW AA0NEW                                             T1R_SW3AA
|  N7:27  N7:27  N7:27                                                N7:27
+-+-]/[----]/[----] [--+-------------------------------------------------( )----
| |    6      5      4 |                                               13
| |AA2NEW AA0NEW       |
| |N7:27  N7:27        |
| +-] [----]/[---------+
|      6      4
Rung 4:269
| AA0NEW                                                                 AA0
| N7:27                                                                N7:27
+--] [-------------------------------------------------------------------( )--
|     4                                                                   0
Rung 4:270
| AA1NEW                                                                 AA1
| N7:27                                                                N7:27
+--] [-------------------------------------------------------------------( )--
|     5                                                                   1
Rung 4:271
| AA2NEW                                                                 AA2
| N7:27                                                                N7:27
+--] [-------------------------------------------------------------------( )--
|     6                                                                   2
```



Page 1

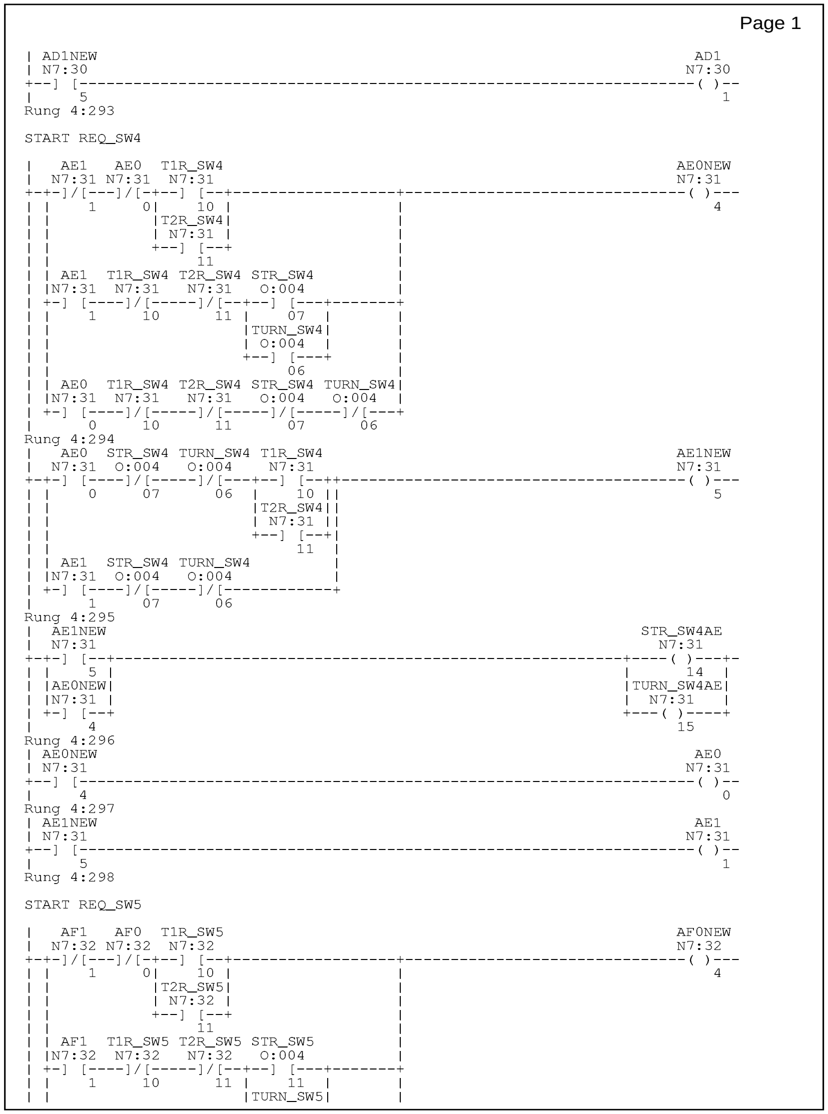
```
Rung 4:272

start of srt1_sw4

|   AB1   AB0  T1_AT4                                                   AB0NEW
|  N7:28 N7:28   B3                                                     N7:28
+-+-]/[---]/[---] [--+---------------------------------------------------( )---
| |    1     0    17 |                                                      4
| | AB0  T1R_SW4     |
| |N7:28  N7:31      |
| +-] [----]/[-------+
|      0     10
Rung 4:273
|   AB0  T1R_SW4                                                        AB1NEW
|  N7:28  N7:31                                                         N7:28
+-+-] [----] [--+--------------------------------------------------------( )---
| |    0     10 |                                                           5
| | AB1  STR_SW4|
| |N7:28  O:004 |
| +-] [----]/[--+
|      1     07
Rung 4:274
|                                                                    STR_SW4AB
|                                                                      N7:28
+-----------------------------------------------------------------------( )----
|                                                                        10
Rung 4:275
| AB1NEW                                                            TURN_SW4AB
| N7:28                                                               N7:28
+--]/[-----------------------------------------------------------------( )-----
|     5                                                                  11
Rung 4:276
| AB1NEW AB0NEW                                                        EN_T1AB
| N7:28  N7:28                                                          N7:28
+--]/[----]/[------------------------------------------------------------( )---
|     5      4                                                             12
Rung 4:277
| AB0NEW                                                             T1R_SW4AB
| N7:28                                                                N7:28
+--] [------------------------------------------------------------------( )----
|     4                                                                  13
Rung 4:278
| AB0NEW                                                                  AB0
| N7:28                                                                  N7:28
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:279
| AB1NEW                                                                  AB1
| N7:28                                                                  N7:28
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:280

start srt1_sw6

|   AC1   AC0  T1_AT27                                                  AC0NEW
|  N7:29 N7:29   B3                                                     N7:29
+-+-]/[---]/[----] [--+--------------------------------------------------( )---
| |    1     0    109 |                                                     4
| | AC0  T1R_SW6      |
| |N7:29  N7:33       |
| +-] [----]/[--------+
|      0     10
Rung 4:281
|   AC0  T1R_SW6                                                        AC1NEW
|  N7:29  N7:33                                                         N7:29
+-+-] [----] [--+--------------------------------------------------------( )---
| |    0     10 |                                                           5
| | AC1  STR_SW6|
| |N7:29  O:004 |
| +-] [----]/[--+
|      1     13
Rung 4:282
|                                                                    STR_SW6AC
|                                                                      N7:29
+-----------------------------------------------------------------------( )----
|                                                                        10
```



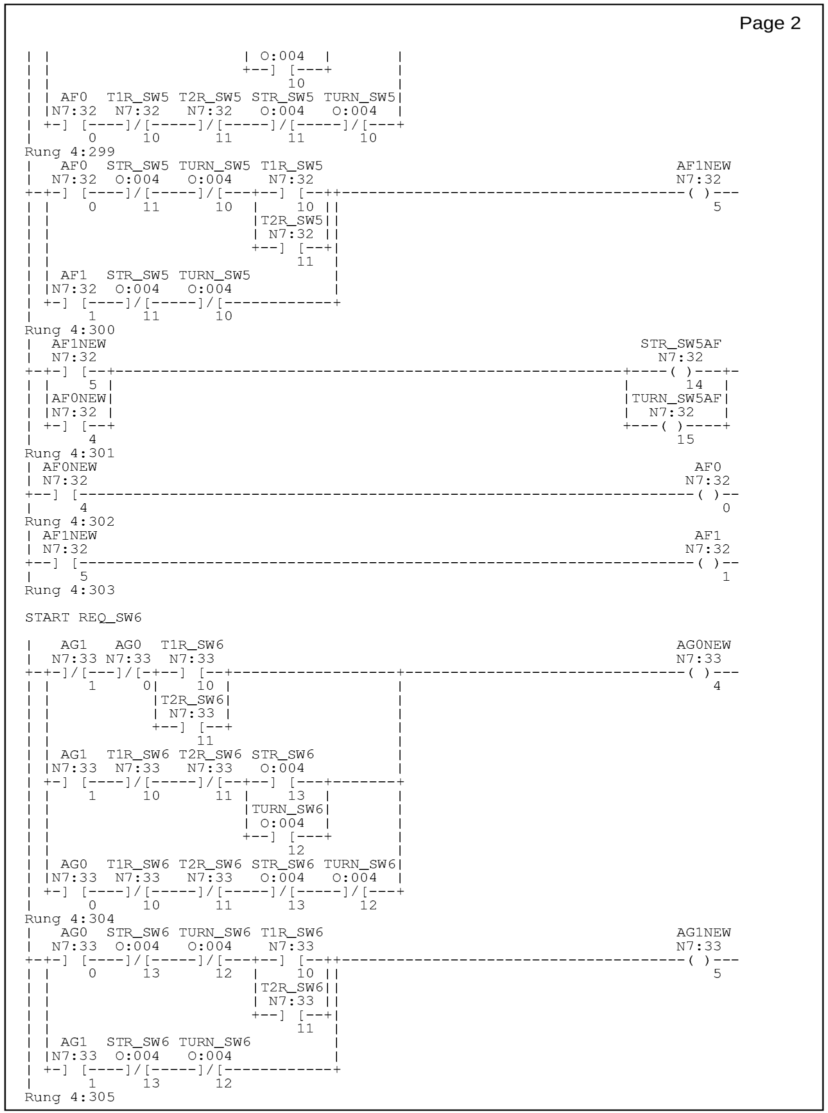
Page 2

```
Rung 4:283
| AC1NEW                                                            TURN_SW6AC
| N7:29                                                               N7:29
+--]/[-----------------------------------------------------------------( )-----
|     5                                                                  11
Rung 4:284
| AC1NEW AC0NEW                                                        EN_T1AC
| N7:29  N7:29                                                          N7:29
+--]/[----]/[------------------------------------------------------------( )---
|     5      4                                                             12
Rung 4:285
| AC0NEW                                                             T1R_SW6AC
| N7:29                                                                N7:29
+--] [------------------------------------------------------------------( )----
|     4                                                                  13
Rung 4:286
| AC0NEW                                                                  AC0
| N7:29                                                                  N7:29
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:287
| AC1NEW                                                                  AC1
| N7:29                                                                  N7:29
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:288

start req_sw3

|   AD1   AD0  T1R_SW3                                                  AD0NEW
|  N7:30 N7:30  N7:30                                                   N7:30
+-+-]/[---]/[-+--] [--+---------------------+----------------------------( )---
| |    1     0|    10 |                     |                               4
| |           |T2R_SW3|                     |
| |           | N7:30 |                     |
| |           +--] [--+                     |
| |                11                       |
| | AD1  T1R_SW3 T2R_SW3 TURNR_SW3          |
| |N7:30  N7:30   N7:30    N7:36            |
| +-] [----]/[-----]/[--+---] [---+---------+
| |    1     10      11 |     10  |         |
| |                     |TURNL_SW3|         |
| |                     |  N7:36  |         |
| |                     +---] [---+         |
| |                          11             |
| | AD0  T1R_SW3 T2R_SW3 TURNR_SW3 TURNL_SW3|
| |N7:30  N7:30   N7:30    N7:36     N7:36  |
| +-] [----]/[-----]/[------]/[-------]/[---+
|      0     10      11       10        11
Rung 4:289
|   AD0  TURNR_SW3 TURNL_SW3 T1R_SW3                                    AD1NEW
|  N7:30   N7:36     N7:36    N7:30                                     N7:30
+-+-] [-----]/[-------]/[---+--] [--++-----------------------------------( )---
| |    0      10        11  |    10 ||                                      5
| |                       |T2R_SW3||
| |                       | N7:30 ||
| |                       +--] [--+|
| |                            11  |
| | AD1  TURNR_SW3 TURNL_SW3         |
| |N7:30   N7:36     N7:36           |
| +-] [-----]/[-------]/[------------+
|      1      10        11
Rung 4:290
|  AD1NEW                                                            TR_SW3AD
|  N7:30                                                               N7:30
+-+-] [--+----------------------------------------------------------+--( )---+-
| |    5 |                                                          |    14  |
| |AD0NEW|                                                          |TL_SW3AD|
| |N7:30 |                                                          | N7:30  |
| +-] [--+                                                          +--( )---+
|      4                                                                 15
Rung 4:291
| AD0NEW                                                                  AD0
| N7:30                                                                  N7:30
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:292
```

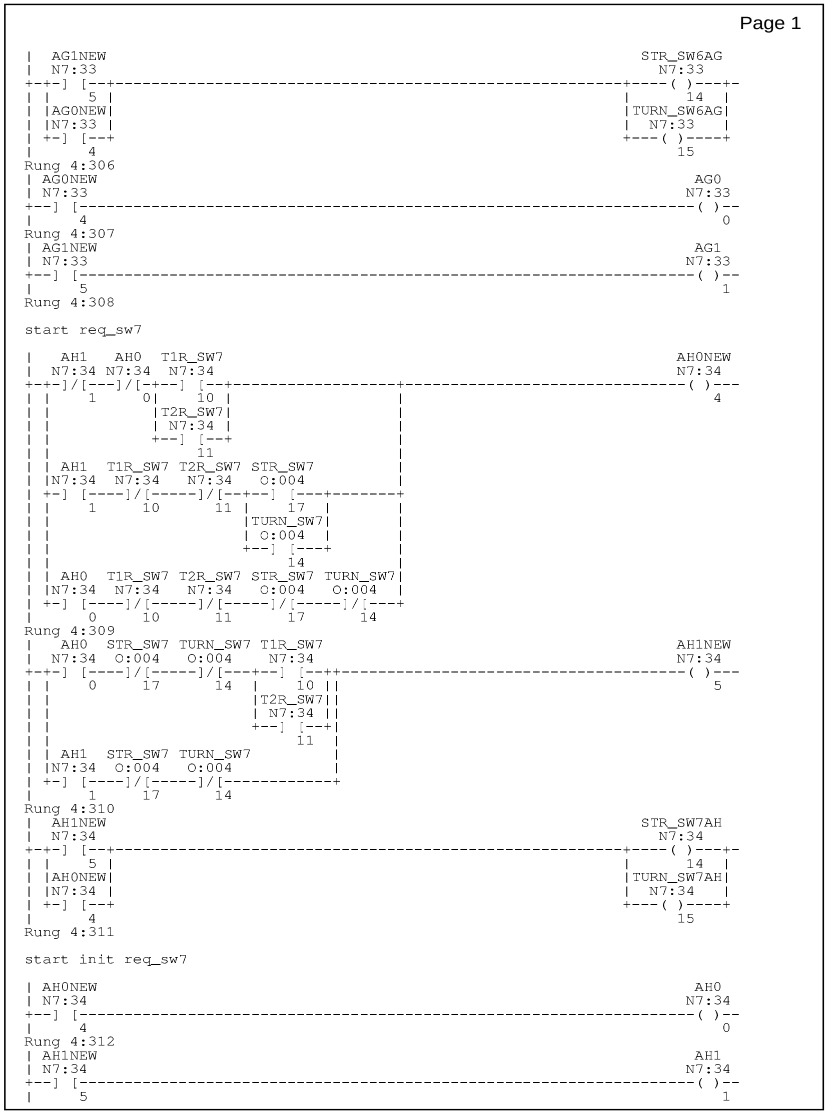
1



```
| AD1NEW                                                                  AD1
| N7:30                                                                  N7:30
+--] [--------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:293

START REQ_SW4

|   AE1   AE0  T1R_SW4                                                  AE0NEW
|  N7:31 N7:31  N7:31                                                   N7:31
+-+-]/[---]/[-+--] [--+------------------+-------------------------------( )---
| |    1     0|    10 |                  |                                  4
| |           |T2R_SW4|                  |
| |           | N7:31 |                  |
| |           +--] [--+                  |
| |               11                     |
| | AE1  T1R_SW4 T2R_SW4 STR_SW4         |
| |N7:31  N7:31   N7:31   O:004          |
| +-] [----]/[-----]/[--+--] [---+-------+
| |    1     10      11 |    07  |       |
| |                     |TURN_SW4|       |
| |                     | O:004  |       |
| |                     +--] [---+       |
| |                          06          |
| | AE0  T1R_SW4 T2R_SW4 STR_SW4 TURN_SW4|
| |N7:31  N7:31   N7:31   O:004   O:004  |
| +-] [----]/[-----]/[-----]/[-----]/[---+
|      0     10      11      07      06
Rung 4:294
|   AE0  STR_SW4 TURN_SW4 T1R_SW4                                       AE1NEW
|  N7:31  O:004   O:004    N7:31                                        N7:31
+-+-] [----]/[-----]/[---+--] [--++--------------------------------------( )---
| |    0     07      06  |    10 ||                                         5
| |                      |T2R_SW4||
| |                      | N7:31 ||
| |                      +--] [--+|
| |                           11  |
| | AE1  STR_SW4 TURN_SW4         |
| |N7:31  O:004   O:004           |
| +-] [----]/[-----]/[------------+
|      1     07      06
Rung 4:295
|  AE1NEW                                                           STR_SW4AE
|  N7:31                                                              N7:31
+-+-] [--+--------------------------------------------------------+----( )---+-
| |    5 |                                                        |      14  |
| |AE0NEW|                                                        |TURN_SW4AE|
| |N7:31 |                                                        |  N7:31   |
| +-] [--+                                                        +---( )----+
|      4                                                                15
Rung 4:296
| AE0NEW                                                                  AE0
| N7:31                                                                  N7:31
+--] [--------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:297
| AE1NEW                                                                  AE1
| N7:31                                                                  N7:31
+--] [--------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:298

START REQ_SW5

|   AF1   AF0  T1R_SW5                                                  AF0NEW
|  N7:32 N7:32  N7:32                                                   N7:32
+-+-]/[---]/[-+--] [--+------------------+-------------------------------( )---
| |    1     0|    10 |                  |                                  4
| |           |T2R_SW5|                  |
| |           | N7:32 |                  |
| |           +--] [--+                  |
| |               11                     |
| | AF1  T1R_SW5 T2R_SW5 STR_SW5         |
| |N7:32  N7:32   N7:32   O:004          |
| +-] [----]/[-----]/[--+--] [---+-------+
| |    1     10      11 |    11  |       |
| |                     |TURN_SW5|       |
```

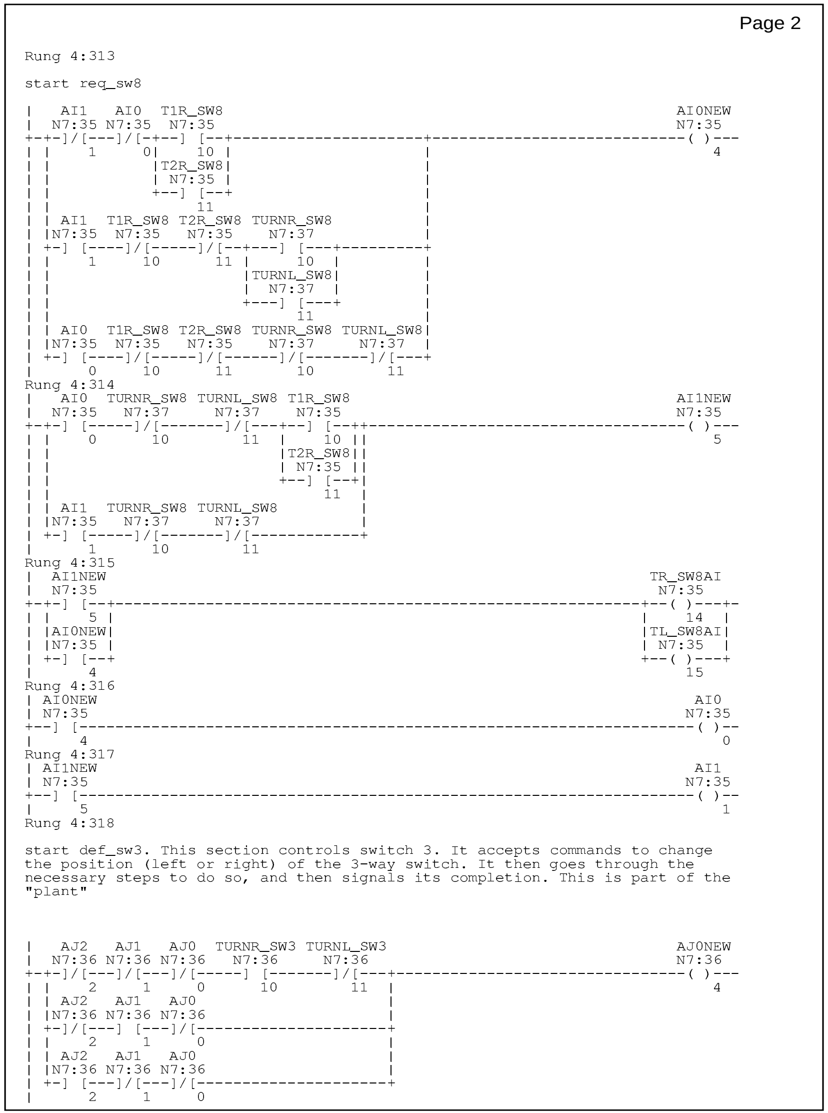


```
| |                     | O:004  |       |
| |                     +--] [---+       |
| |                          10          |
| | AF0  T1R_SW5 T2R_SW5 STR_SW5 TURN_SW5|
| |N7:32  N7:32   N7:32   O:004   O:004  |
| +-] [----]/[-----]/[-----]/[-----]/[---+
|      0     10      11      11      10
Rung 4:299
|   AF0  STR_SW5 TURN_SW5 T1R_SW5                                       AF1NEW
|  N7:32  O:004   O:004    N7:32                                        N7:32
+-+-] [----]/[-----]/[---+--] [--++--------------------------------------( )---
| |    0     11      10  |    10 ||                                         5
| |                      |T2R_SW5||
| |                      | N7:32 ||
| |                      +--] [--+|
| |                           11  |
| | AF1  STR_SW5 TURN_SW5         |
| |N7:32  O:004   O:004           |
| +-] [----]/[-----]/[------------+
|      1     11      10
Rung 4:300
|  AF1NEW                                                           STR_SW5AF
|  N7:32                                                              N7:32
+-+-] [--+--------------------------------------------------------+----( )---+-
| |    5 |                                                        |      14  |
| |AF0NEW|                                                        |TURN_SW5AF|
| |N7:32 |                                                        |  N7:32   |
| +-] [--+                                                        +---( )----+
|      4                                                                15
Rung 4:301
| AF0NEW                                                                  AF0
| N7:32                                                                  N7:32
+--] [--------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:302
| AF1NEW                                                                  AF1
| N7:32                                                                  N7:32
+--] [--------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:303

START REQ_SW6

|   AG1   AG0  T1R_SW6                                                  AG0NEW
|  N7:33 N7:33  N7:33                                                   N7:33
+-+-]/[---]/[-+--] [--+------------------+-------------------------------( )---
| |    1     0|    10 |                  |                                  4
| |           |T2R_SW6|                  |
| |           | N7:33 |                  |
| |           +--] [--+                  |
| |               11                     |
| | AG1  T1R_SW6 T2R_SW6 STR_SW6         |
| |N7:33  N7:33   N7:33   O:004          |
| +-] [----]/[-----]/[--+--] [---+-------+
| |    1     10      11 |    13  |       |
| |                     |TURN_SW6|       |
| |                     | O:004  |       |
| |                     +--] [---+       |
| |                          12          |
| | AG0  T1R_SW6 T2R_SW6 STR_SW6 TURN_SW6|
| |N7:33  N7:33   N7:33   O:004   O:004  |
| +-] [----]/[-----]/[-----]/[-----]/[---+
|      0     10      11      13      12
Rung 4:304
|   AG0  STR_SW6 TURN_SW6 T1R_SW6                                       AG1NEW
|  N7:33  O:004   O:004    N7:33                                        N7:33
+-+-] [----]/[-----]/[---+--] [--++--------------------------------------( )---
| |    0     13      12  |    10 ||                                         5
| |                      |T2R_SW6||
| |                      | N7:33 ||
| |                      +--] [--+|
| |                           11  |
| | AG1  STR_SW6 TURN_SW6         |
| |N7:33  O:004   O:004           |
| +-] [----]/[-----]/[------------+
|      1     13      12
Rung 4:305
```



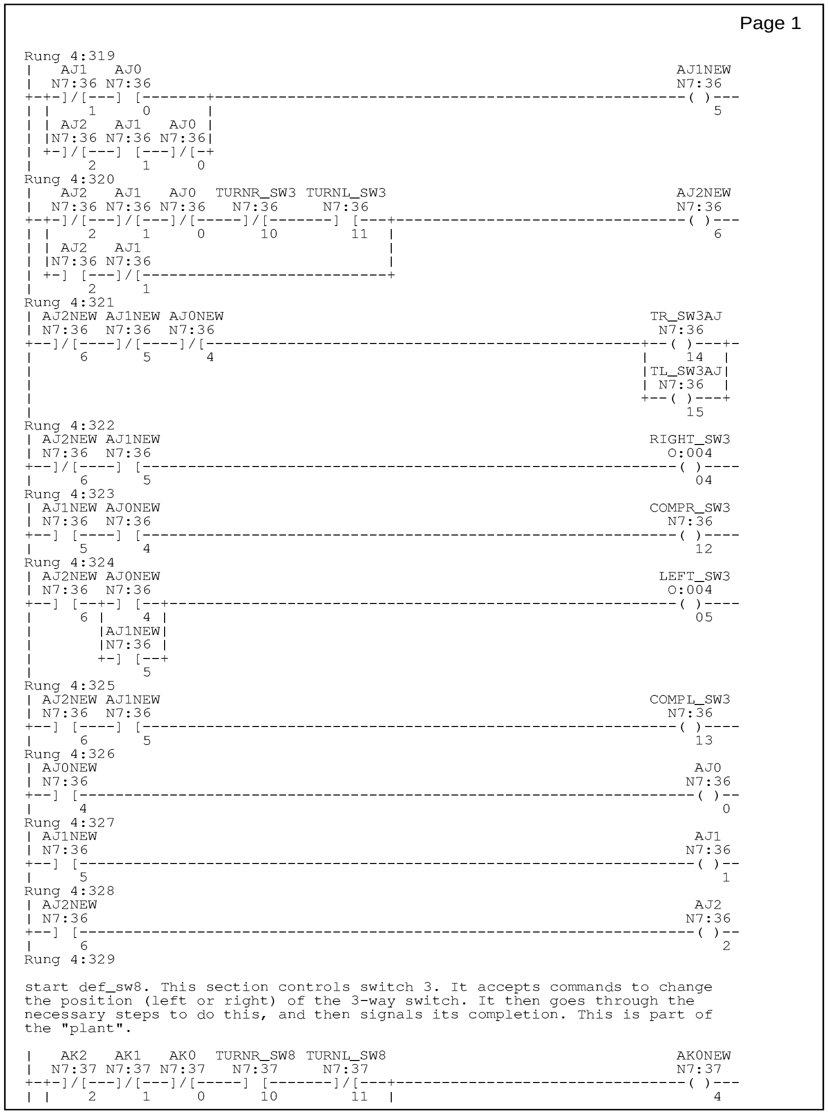
Page 1

```
|  AG1NEW                                                           STR_SW6AG
|  N7:33                                                              N7:33
+-+-] [--+--------------------------------------------------------+----( )---+-
| |    5 |                                                        |      14  |
| |AG0NEW|                                                        |TURN_SW6AG|
| |N7:33 |                                                        |  N7:33   |
| +-] [--+                                                        +---( )----+
|      4                                                                15
Rung 4:306
| AG0NEW                                                                  AG0
| N7:33                                                                  N7:33
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:307
| AG1NEW                                                                  AG1
| N7:33                                                                  N7:33
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:308

start req_sw7

|   AH1   AH0  T1R_SW7                                                  AH0NEW
|  N7:34 N7:34  N7:34                                                   N7:34
+-+-]/[---]/[-+--] [--+------------------+-------------------------------( )---
| |    1     0|    10 |                  |                                  4
| |           |T2R_SW7|                  |
| |           | N7:34 |                  |
| |           +--] [--+                  |
| |                11                    |
| | AH1  T1R_SW7 T2R_SW7 STR_SW7         |
| |N7:34  N7:34   N7:34   O:004          |
| +-] [----]/[-----]/[--+--] [---+-------+
| |    1     10      11 |    17  |       |
| |                     |TURN_SW7|       |
| |                     | O:004  |       |
| |                     +--] [---+       |
| |                          14          |
| | AH0  T1R_SW7 T2R_SW7 STR_SW7 TURN_SW7|
| |N7:34  N7:34   N7:34   O:004   O:004  |
| +-] [----]/[-----]/[-----]/[-----]/[---+
|      0     10      11      17      14
Rung 4:309
|   AH0  STR_SW7 TURN_SW7 T1R_SW7                                       AH1NEW
|  N7:34  O:004   O:004    N7:34                                        N7:34
+-+-] [----]/[-----]/[---+--] [--++--------------------------------------( )---
| |    0     17      14   |     10 ||                                       5
| |                       |T2R_SW7||
| |                       | N7:34 ||
| |                       +--] [--+|
| |                             11 |
| | AH1  STR_SW7 TURN_SW7          |
| |N7:34  O:004   O:004            |
| +-] [----]/[-----]/[------------+
|      1     17      14
Rung 4:310
|  AH1NEW                                                           STR_SW7AH
|  N7:34                                                              N7:34
+-+-] [--+--------------------------------------------------------+----( )---+-
| |    5 |                                                        |      14  |
| |AH0NEW|                                                        |TURN_SW7AH|
| |N7:34 |                                                        |  N7:34   |
| +-] [--+                                                        +---( )----+
|      4                                                                15
Rung 4:311

start init req_sw7

| AH0NEW                                                                  AH0
| N7:34                                                                  N7:34
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:312
| AH1NEW                                                                  AH1
| N7:34                                                                  N7:34
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
```



Page 2

```
Rung 4:313

start req_sw8

|   AI1   AI0  T1R_SW8                                                  AI0NEW
|  N7:35 N7:35  N7:35                                                   N7:35
+-+-]/[---]/[-+--] [--+---------------------+----------------------------( )---
| |    1     0|    10 |                     |                               4
| |           |T2R_SW8|                     |
| |           | N7:35 |                     |
| |           +--] [--+                     |
| |                11                       |
| | AI1  T1R_SW8 T2R_SW8 TURNR_SW8          |
| |N7:35  N7:35   N7:35    N7:37            |
| +-] [----]/[-----]/[--+---] [---+---------+
| |    1     10      11 |     10  |         |
| |                     |TURNL_SW8|         |
| |                     |  N7:37  |         |
| |                     +---] [---+         |
| |                           11            |
| | AI0  T1R_SW8 T2R_SW8 TURNR_SW8 TURNL_SW8|
| |N7:35  N7:35   N7:35    N7:37     N7:37  |
| +-] [----]/[-----]/[------]/[-------]/[---+
|      0     10      11       10        11
Rung 4:314
|   AI0  TURNR_SW8 TURNL_SW8 T1R_SW8                                    AI1NEW
|  N7:35   N7:37     N7:37    N7:35                                     N7:35
+-+-] [-----]/[-------]/[---+--] [--++-----------------------------------( )---
| |    0      10        11  |     10 ||                                     5
| |                         |T2R_SW8||
| |                         | N7:35 ||
| |                         +--] [--+|
| |                              11  |
| | AI1  TURNR_SW8 TURNL_SW8         |
| |N7:35   N7:37     N7:37           |
| +-] [-----]/[-------]/[------------+
|      1      10        11
Rung 4:315
|  AI1NEW                                                            TR_SW8AI
|  N7:35                                                              N7:35
+-+-] [--+----------------------------------------------------------+--( )---+-
| |    5 |                                                          |    14  |
| |AI0NEW|                                                          |TL_SW8AI|
| |N7:35 |                                                          | N7:35  |
| +-] [--+                                                          +--( )---+
|      4                                                                 15
Rung 4:316
| AI0NEW                                                                  AI0
| N7:35                                                                  N7:35
+--] [--------------------------------------------------------------------( )--
|     4                                                                      0
Rung 4:317
| AI1NEW                                                                  AI1
| N7:35                                                                  N7:35
+--] [--------------------------------------------------------------------( )--
|     5                                                                      1
Rung 4:318

start def_sw3. This section controls switch 3. It accepts commands to change
the position (left or right) of the 3-way switch. It then goes through the
necessary steps to do so, and then signals its completion. This is part of the
"plant"


|   AJ2   AJ1   AJ0  TURNR_SW3 TURNL_SW3                                AJ0NEW
|  N7:36 N7:36 N7:36   N7:36     N7:36                                  N7:36
+-+-]/[---]/[---]/[-----] [-------]/[---+--------------------------------( )---
| |    2     1     0      10        11  |                                   4
| | AJ2   AJ1   AJ0                     |
| |N7:36 N7:36 N7:36                    |
| +-]/[---] [---]/[---------------------+
| |    2     1     0                    |
| | AJ2   AJ1   AJ0                     |
| |N7:36 N7:36 N7:36                    |
| +-] [---]/[---]/[---------------------+
|      2     1     0
```

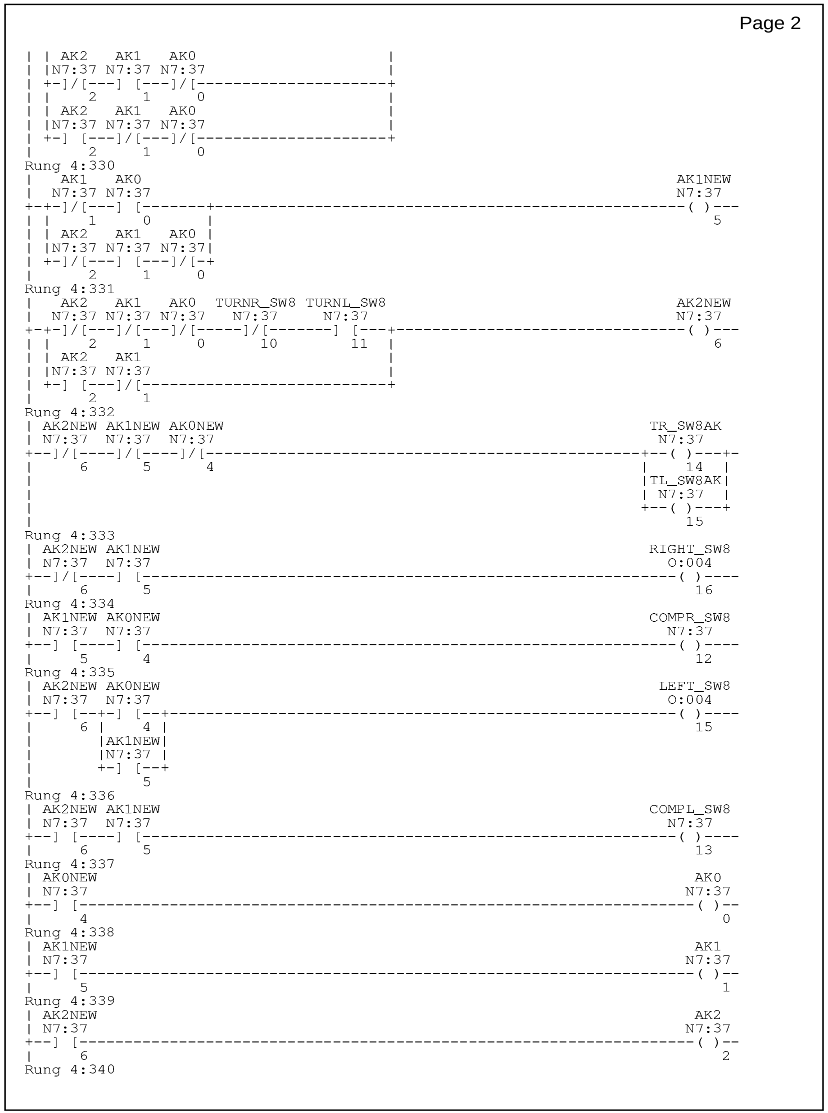


Page 1

```
Rung 4:319
|   AJ1   AJ0                                                           AJ1NEW
|  N7:36 N7:36                                                          N7:36
+-+-]/[---] [-------+----------------------------------------------------( )---
| |    1     0      |                                                       5
| | AJ2   AJ1   AJ0 |
| |N7:36 N7:36 N7:36|
| +-]/[---] [---]/[-+
|      2     1     0
Rung 4:320
|   AJ2   AJ1   AJ0  TURNR_SW3 TURNL_SW3                                AJ2NEW
|  N7:36 N7:36 N7:36   N7:36     N7:36                                  N7:36
+-+-]/[---]/[---]/[-----]/[-------] [---+--------------------------------( )---
| |    2     1     0      10        11  |                                   6
| | AJ2   AJ1                           |
| |N7:36 N7:36                          |
| +-] [---]/[---------------------------+
|      2     1
Rung 4:321
| AJ2NEW AJ1NEW AJ0NEW                                               TR_SW3AJ
| N7:36  N7:36  N7:36                                                 N7:36
+--]/[----]/[----]/[------------------------------------------------+--( )---+-
|      6      5      4                                               |    14  |
|                                                                    |TL_SW3AJ|
|                                                                    | N7:36  |
|                                                                    +--( )---+
|                                                                         15
Rung 4:322
| AJ2NEW AJ1NEW                                                      RIGHT_SW3
| N7:36  N7:36                                                         O:004
+--]/[----] [-----------------------------------------------------------( )----
|      6      5                                                           04
Rung 4:323
| AJ1NEW AJ0NEW                                                      COMPR_SW3
| N7:36  N7:36                                                         N7:36
+--] [----] [-----------------------------------------------------------( )----
|      5      4                                                           12
Rung 4:324
| AJ2NEW AJ0NEW                                                       LEFT_SW3
| N7:36  N7:36                                                         O:004
+--] [--+-] [--+--------------------------------------------------------( )----
|      6 |    4 |                                                         05
|       |AJ1NEW|
|       |N7:36 |
|       +-] [--+
|            5
Rung 4:325
| AJ2NEW AJ1NEW                                                      COMPL_SW3
| N7:36  N7:36                                                         N7:36
+--] [----] [-----------------------------------------------------------( )----
|      6      5                                                           13
Rung 4:326
| AJ0NEW                                                                  AJ0
| N7:36                                                                  N7:36
+--] [--------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:327
| AJ1NEW                                                                  AJ1
| N7:36                                                                  N7:36
+--] [--------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:328
| AJ2NEW                                                                  AJ2
| N7:36                                                                  N7:36
+--] [--------------------------------------------------------------------( )--
|     6                                                                       2
Rung 4:329
```

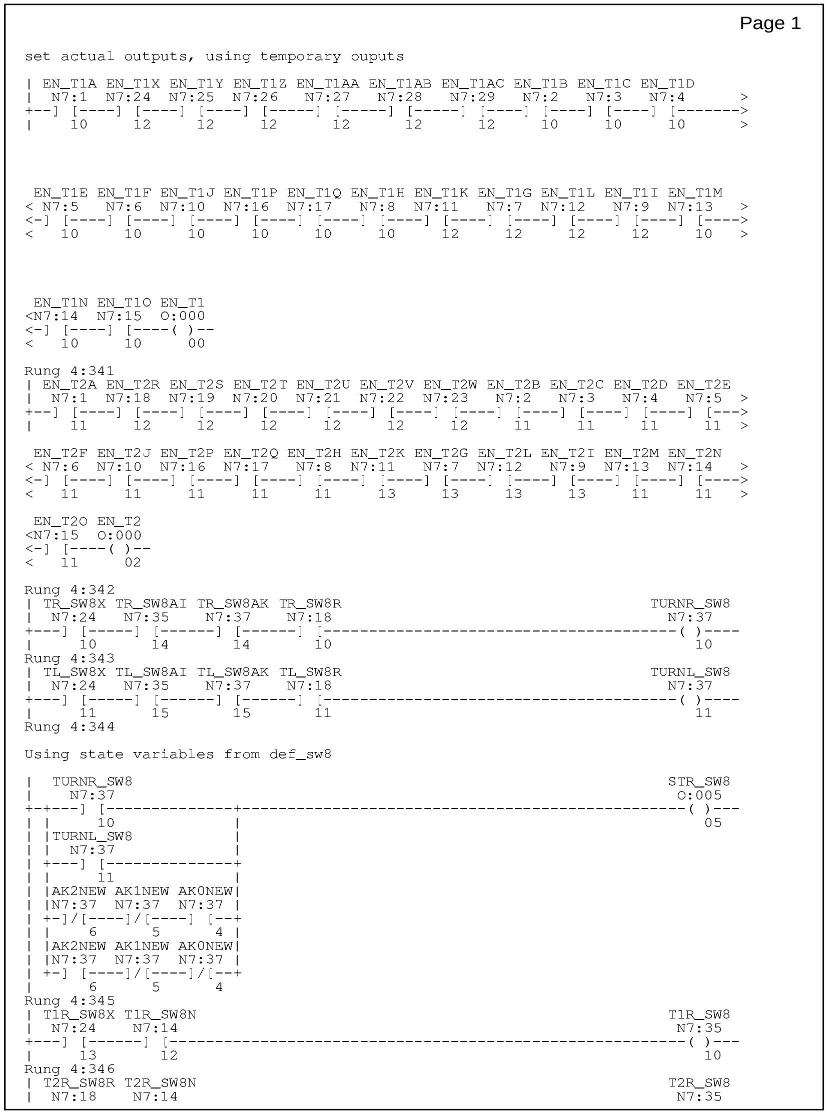
start def_sw8. This section controls switch 3. It accepts commands to change the position (left or right) of the 3-way switch. It then goes through the necessary steps to do this, and then signals its completion. This is part of the "plant".

```
|   AK2   AK1   AK0  TURNR_SW8 TURNL_SW8                                AK0NEW
|  N7:37 N7:37 N7:37   N7:37     N7:37                                  N7:37
+-+-]/[---]/[---]/[-----] [-------]/[---+--------------------------------( )---
| |    2     1     0      10        11  |                                   4
```



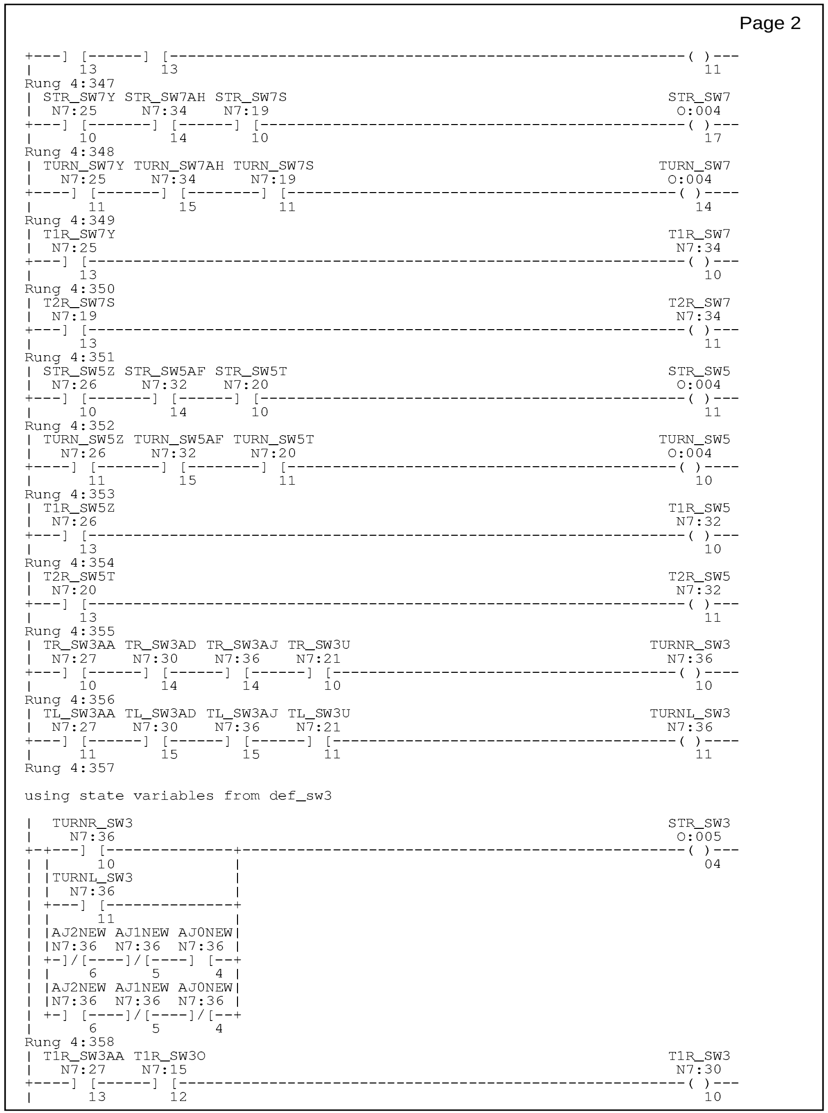
Page 2

```
| | AK2   AK1   AK0                     |
| |N7:37 N7:37 N7:37                    |
| +-]/[---] [---]/[---------------------+
| |    2     1     0                    |
| | AK2   AK1   AK0                     |
| |N7:37 N7:37 N7:37                    |
| +-] [---]/[---]/[---------------------+
|      2     1     0
Rung 4:330
|   AK1   AK0                                                           AK1NEW
|  N7:37 N7:37                                                          N7:37
+-+-]/[---] [-------+----------------------------------------------------( )---
| |    1     0      |                                                       5
| | AK2   AK1   AK0 |
| |N7:37 N7:37 N7:37|
| +-]/[---] [---]/[-+
|      2     1     0
Rung 4:331
|   AK2   AK1   AK0  TURNR_SW8 TURNL_SW8                                AK2NEW
|  N7:37 N7:37 N7:37   N7:37     N7:37                                  N7:37
+-+-]/[---]/[---]/[-----]/[-------] [---+--------------------------------( )---
| |    2     1     0      10        11  |                                   6
| | AK2   AK1                           |
| |N7:37 N7:37                          |
| +-] [---]/[---------------------------+
|      2     1
Rung 4:332
| AK2NEW AK1NEW AK0NEW                                               TR_SW8AK
| N7:37  N7:37  N7:37                                                 N7:37
+--]/[----]/[----]/[------------------------------------------------+--( )---+-
|      6      5      4                                               |    14  |
|                                                                    |TL_SW8AK|
|                                                                    | N7:37  |
|                                                                    +--( )---+
|                                                                         15
Rung 4:333
| AK2NEW AK1NEW                                                      RIGHT_SW8
| N7:37  N7:37                                                         O:004
+--]/[----] [-----------------------------------------------------------( )----
|      6      5                                                           16
Rung 4:334
| AK1NEW AK0NEW                                                      COMPR_SW8
| N7:37  N7:37                                                         N7:37
+--] [----] [-----------------------------------------------------------( )----
|      5      4                                                           12
Rung 4:335
| AK2NEW AK0NEW                                                       LEFT_SW8
| N7:37  N7:37                                                         O:004
+--] [--+-] [--+--------------------------------------------------------( )----
|      6 |    4 |                                                         15
|       |AK1NEW|
|       |N7:37 |
|       +-] [--+
|            5
Rung 4:336
| AK2NEW AK1NEW                                                      COMPL_SW8
| N7:37  N7:37                                                         N7:37
+--] [----] [-----------------------------------------------------------( )----
|      6      5                                                           13
Rung 4:337
| AK0NEW                                                                  AK0
| N7:37                                                                  N7:37
+--] [--------------------------------------------------------------------( )--
|     4                                                                       0
Rung 4:338
| AK1NEW                                                                  AK1
| N7:37                                                                  N7:37
+--] [--------------------------------------------------------------------( )--
|     5                                                                       1
Rung 4:339
| AK2NEW                                                                  AK2
| N7:37                                                                  N7:37
+--] [--------------------------------------------------------------------( )--
|     6                                                                       2
Rung 4:340
```



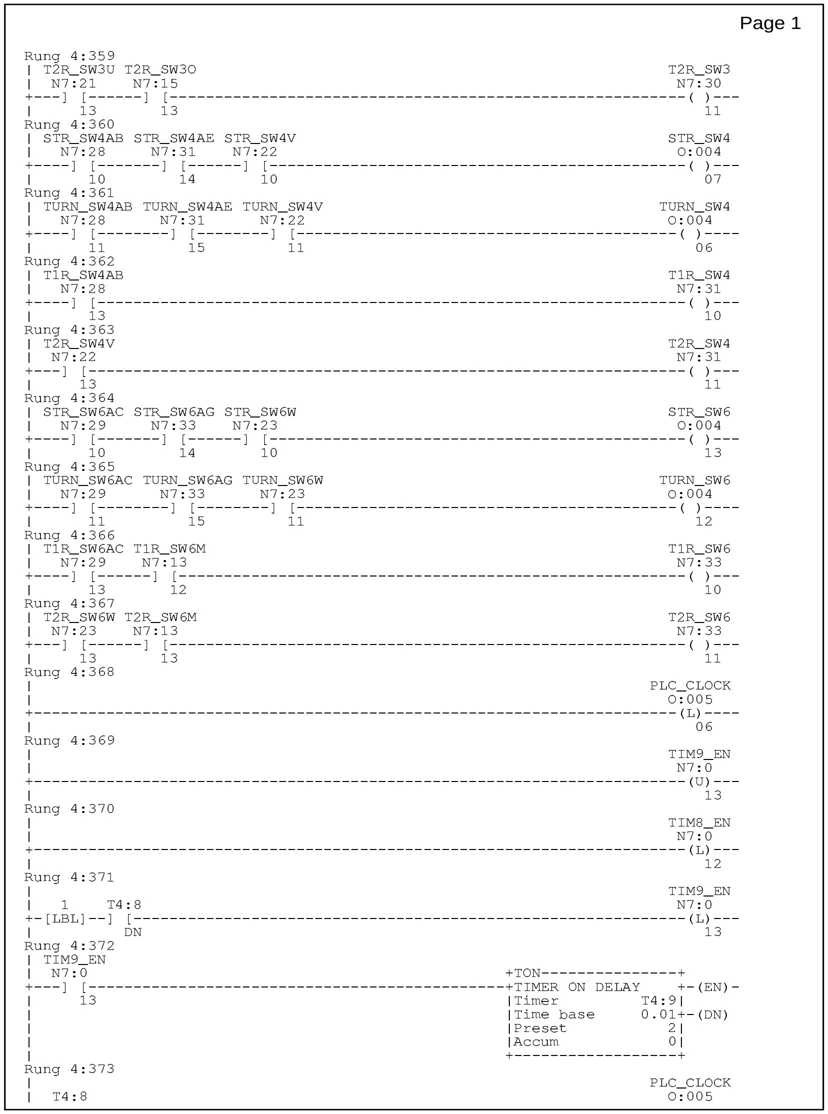
Page 1

```
set actual outputs, using temporary ouputs

| EN_T1A EN_T1X EN_T1Y EN_T1Z EN_T1AA EN_T1AB EN_T1AC EN_T1B EN_T1C EN_T1D
|  N7:1  N7:24  N7:25  N7:26   N7:27   N7:28   N7:29   N7:2   N7:3   N7:4      >
+--] [----] [----] [----] [-----] [-----] [-----] [----] [----] [----] [------->
|    10     12     12     12      12      12      12     10     10     10      >



 EN_T1E EN_T1F EN_T1J EN_T1P EN_T1Q EN_T1H EN_T1K EN_T1G EN_T1L EN_T1I EN_T1M
< N7:5   N7:6  N7:10  N7:16  N7:17   N7:8  N7:11   N7:7  N7:12   N7:9  N7:13   >
<-] [----] [----] [----] [----] [----] [----] [----] [----] [----] [----] [---->
<   10     10     10     10     10     10     12     12     12     12     10   >



 EN_T1N EN_T1O EN_T1
<N7:14  N7:15  O:000
<-] [----] [----( )--
<   10     10     00

Rung 4:341
| EN_T2A EN_T2R EN_T2S EN_T2T EN_T2U EN_T2V EN_T2W EN_T2B EN_T2C EN_T2D EN_T2E
|  N7:1  N7:18  N7:19  N7:20  N7:21  N7:22  N7:23   N7:2   N7:3   N7:4   N7:5  >
+--] [----] [----] [----] [----] [----] [----] [----] [----] [----] [----] [--->
|    11     12     12     12     12     12     12     11     11     11     11  >

 EN_T2F EN_T2J EN_T2P EN_T2Q EN_T2H EN_T2K EN_T2G EN_T2L EN_T2I EN_T2M EN_T2N
< N7:6  N7:10  N7:16  N7:17   N7:8  N7:11   N7:7  N7:12   N7:9  N7:13  N7:14   >
<-] [----] [----] [----] [----] [----] [----] [----] [----] [----] [----] [---->
<   11     11     11     11     11     13     13     13     13     11     11   >

 EN_T2O EN_T2
<N7:15  O:000
<-] [----( )--
<   11     02

Rung 4:342
| TR_SW8X TR_SW8AI TR_SW8AK TR_SW8R                                  TURNR_SW8
|  N7:24   N7:35    N7:37    N7:18                                     N7:37
+---] [-----] [------] [------] [---------------------------------------( )----
|     10      14       14       10                                       10
Rung 4:343
| TL_SW8X TL_SW8AI TL_SW8AK TL_SW8R                                  TURNL_SW8
|  N7:24   N7:35    N7:37    N7:18                                     N7:37
+---] [-----] [------] [------] [---------------------------------------( )----
|     11      15       15       11                                       11
Rung 4:344

Using state variables from def_sw8

|  TURNR_SW8                                                           STR_SW8
|    N7:37                                                              O:005
+-+---] [--------------+---------------------------------------------------( )---
| |      10            |                                                   05
| |TURNL_SW8           |
| |  N7:37             |
| +---] [--------------+
| |      11            |
| |AK2NEW AK1NEW AK0NEW|
| |N7:37  N7:37  N7:37 |
| +-]/[----]/[----] [--+
| |    6      5      4 |
| |AK2NEW AK1NEW AK0NEW|
| |N7:37  N7:37  N7:37 |
| +-] [----]/[----]/[--+
|      6      5      4
Rung 4:345
| T1R_SW8X T1R_SW8N                                                    T1R_SW8
|  N7:24    N7:14                                                       N7:35
+---] [------] [---------------------------------------------------------( )---
|     13       12                                                        10
Rung 4:346
| T2R_SW8R T2R_SW8N                                                    T2R_SW8
|  N7:18    N7:14                                                       N7:35
```



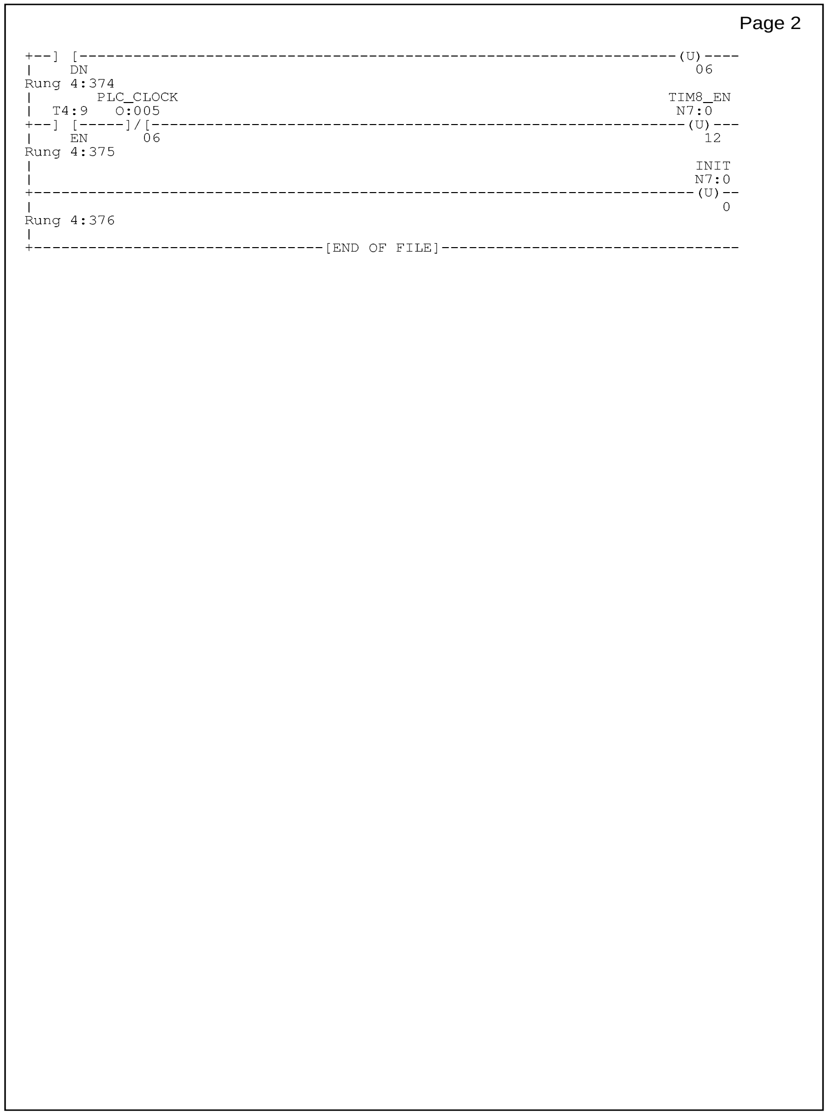
Page 2

```
+---] [------] [--------------------------------------------------------( )---
|     13       13                                                        11
Rung 4:347
| STR_SW7Y STR_SW7AH STR_SW7S                                          STR_SW7
|  N7:25     N7:34    N7:19                                             O:004
+---] [-------] [------] [----------------------------------------------( )---
|     10        14       10                                              17
Rung 4:348
| TURN_SW7Y TURN_SW7AH TURN_SW7S                                      TURN_SW7
|   N7:25     N7:34      N7:19                                         O:004
+----] [-------] [--------] [------------------------------------------( )----
|      11        15         11                                         14
Rung 4:349
| T1R_SW7Y                                                             T1R_SW7
|  N7:25                                                                N7:34
+---] [-----------------------------------------------------------------( )---
|     13                                                                 10
Rung 4:350
| T2R_SW7S                                                             T2R_SW7
|  N7:19                                                                N7:34
+---] [-----------------------------------------------------------------( )---
|     13                                                                 11
Rung 4:351
| STR_SW5Z STR_SW5AF STR_SW5T                                          STR_SW5
|  N7:26     N7:32    N7:20                                             O:004
+---] [-------] [------] [----------------------------------------------( )---
|     10        14       10                                              11
Rung 4:352
| TURN_SW5Z TURN_SW5AF TURN_SW5T                                      TURN_SW5
|   N7:26     N7:32      N7:20                                         O:004
+----] [-------] [--------] [------------------------------------------( )----
|      11        15         11                                         10
Rung 4:353
| T1R_SW5Z                                                             T1R_SW5
|  N7:26                                                                N7:32
+---] [-----------------------------------------------------------------( )---
|     13                                                                 10
Rung 4:354
| T2R_SW5T                                                             T2R_SW5
|  N7:20                                                                N7:32
+---] [-----------------------------------------------------------------( )---
|     13                                                                 11
Rung 4:355
| TR_SW3AA TR_SW3AD TR_SW3AJ TR_SW3U                                 TURNR_SW3
|  N7:27    N7:30    N7:36    N7:21                                    N7:36
+---] [------] [------] [------] [--------------------------------------( )----
|     10       14       14       10                                      10
Rung 4:356
| TL_SW3AA TL_SW3AD TL_SW3AJ TL_SW3U                                 TURNL_SW3
|  N7:27    N7:30    N7:36    N7:21                                    N7:36
+---] [------] [------] [------] [--------------------------------------( )----
|     11       15       15       11                                      11
Rung 4:357

using state variables from def_sw3

|  TURNR_SW3                                                           STR_SW3
|    N7:36                                                              O:005
+-+---] [--------------+---------------------------------------------------( )---
| |      10            |                                                   04
| |TURNL_SW3           |
| |  N7:36             |
| +---] [--------------+
| |      11            |
| |AJ2NEW AJ1NEW AJ0NEW|
| |N7:36  N7:36  N7:36 |
| +-]/[----]/[----] [--+
| |    6      5      4 |
| |AJ2NEW AJ1NEW AJ0NEW|
| |N7:36  N7:36  N7:36 |
| +-] [----]/[----]/[--+
|      6      5      4
Rung 4:358
| T1R_SW3AA T1R_SW3O                                                   T1R_SW3
|   N7:27    N7:15                                                      N7:30
+----] [------] [-------------------------------------------------------( )---
|      13       12                                                       10
```



Page 1

```
Rung 4:359
| T2R_SW3U T2R_SW3O                                                   T2R_SW3
|  N7:21    N7:15                                                      N7:30
+---] [------] [--------------------------------------------------------( )---
|     13       13                                                         11
Rung 4:360
| STR_SW4AB STR_SW4AE STR_SW4V                                        STR_SW4
|   N7:28    N7:31    N7:22                                            O:004
+----] [-------] [------] [---------------------------------------------( )---
|       10        14       10                                             07
Rung 4:361
| TURN_SW4AB TURN_SW4AE TURN_SW4V                                    TURN_SW4
|   N7:28      N7:31      N7:22                                       O:004
+----] [--------] [--------] [-----------------------------------------( )----
|       11          15         11                                        06
Rung 4:362
| T1R_SW4AB                                                           T1R_SW4
|   N7:28                                                              N7:31
+----] [----------------------------------------------------------------( )---
|       13                                                                10
Rung 4:363
| T2R_SW4V                                                            T2R_SW4
|  N7:22                                                               N7:31
+---] [-----------------------------------------------------------------( )---
|     13                                                                  11
Rung 4:364
| STR_SW6AC STR_SW6AG STR_SW6W                                        STR_SW6
|   N7:29    N7:33    N7:23                                            O:004
+----] [-------] [------] [---------------------------------------------( )---
|       10        14       10                                             13
Rung 4:365
| TURN_SW6AC TURN_SW6AG TURN_SW6W                                    TURN_SW6
|   N7:29      N7:33      N7:23                                       O:004
+----] [--------] [--------] [-----------------------------------------( )----
|       11          15         11                                        12
Rung 4:366
| T1R_SW6AC T1R_SW6M                                                  T1R_SW6
|   N7:29   N7:13                                                      N7:33
+----] [------] [-------------------------------------------------------( )---
|       13       12                                                       10
Rung 4:367
| T2R_SW6W T2R_SW6M                                                   T2R_SW6
|  N7:23    N7:13                                                      N7:33
+---] [------] [--------------------------------------------------------( )---
|     13       13                                                         11
Rung 4:368
|                                                                   PLC_CLOCK
|                                                                     O:005
+----------------------------------------------------------------------(L)----
|                                                                        06
Rung 4:369
|                                                                     TIM9_EN
|                                                                      N7:0
+-----------------------------------------------------------------------(U)---
|                                                                         13
Rung 4:370
|                                                                     TIM8_EN
|                                                                      N7:0
+-----------------------------------------------------------------------(L)---
|                                                                         12
Rung 4:371
|                                                                     TIM9_EN
|   1    T4:8                                                          N7:0
+-[LBL]--] [------------------------------------------------------------(L)---
|          DN                                                             13
Rung 4:372
| TIM9_EN
|  N7:0                                             +TON----------------+
+---] [---------------------------------------------+TIMER ON DELAY     +-(EN)-
|     13                                            |Timer          T4:9|
|                                                   |Time base      0.01+-(DN)
|                                                   |Preset            2|
|                                                   |Accum             0|
|                                                   +-------------------+
Rung 4:373
|                                                                   PLC_CLOCK
|  T4:8                                                               O:005
```



Page 2

```
+--] [-----------------------------------------------------------------(U)----
|    DN                                                                 06
Rung 4:374
|       PLC_CLOCK                                                    TIM8_EN
|  T4:9   O:005                                                       N7:0
+--] [-----]/[----------------------------------------------------------(U)---
|    EN      06                                                          12
Rung 4:375
|                                                                       INIT
|                                                                       N7:0
+------------------------------------------------------------------------(U)--
|                                                                          0
Rung 4:376
|
+--------------------------------[END OF FILE]--------------------------------
```

Appendix E

PAL Programs and MC68332 Source Code

This appendix contains the listings of the PAL programs[1] used on the expansion boards, and the MC68332 programs[2] that provide the PLC interface and configure the processors chip selects.

E.1 PAL One

This section contains the ABEL program that defines PAL one (U8) on the expansion boards. The listing is given below:

```
module  PAL1_program

title   'Combinational logic for PAL 1'

"Combinational logic for expansion board for the DES test bed
"for Pal 1, chip U8

declarations

PAL1    device  'P22V10';    "ACN, TI

"inputs
R_L_W_B         pin 1;
```

[1] The PAL programs are written in ABEL ([12]).

[2] The MC68332 programs are written in 68000 asembly language ([13]).

```
L_CS3_B         pin 2;
L_CS5_B         pin 3;
L_CS7_B         pin 4;
L_CS8_B         pin 5;
L_CS9_B         pin 6;
L_CS10_B        pin 7;
L_DS_B          pin 8;
Siz1_B          pin 9;
Siz0_B          pin 10;
A0_B            pin 11;
A4_B            pin 13;

"outputs
L_Duart_Sel     pin 14;
L_CSB4_L        pin 15;
L_CSB4_U        pin 16;
L_CSB3_L        pin 17;
L_CSB3_U        pin 18;
L_CSB2_L        pin 19;
L_CSB2_U        pin 20;
L_CSB1_L        pin 21;
L_CSB1_U        pin 22;
Direction       pin 23;

X = .x.;                        ".x. indicates don’t care condition

equations

@ALTERNATE              "allow * and + etc in eqns
Direction = !R_L_W_B + (L_CS3_B*L_CS5_B*L_CS7_B*L_CS8_B*L_CS9_B*L_CS10_B);
L_CSB1_U = L_CS3_B + A0_B + L_DS_B;
L_CSB1_L = L_CS3_B + (!A0_B*!Siz1_B* Siz0_B) + L_DS_B;
L_CSB2_U = L_CS5_B + A0_B + L_DS_B;
L_CSB2_L = L_CS5_B + (!A0_B*!Siz1_B* Siz0_B) + L_DS_B;
L_CSB3_U = L_CS7_B + A0_B + L_DS_B;
L_CSB3_L = L_CS7_B + (!A0_B*!Siz1_B* Siz0_B) + L_DS_B;
L_CSB4_U = L_CS8_B + A0_B + L_DS_B;
L_CSB4_L = L_CS8_B + (!A0_B*!Siz1_B* Siz0_B) + L_DS_B;
L_Duart_Sel = L_CS9_B + L_DS_B + A4_B;

test_vectors
([R_L_W_B,L_CS3_B,L_CS5_B,L_CS7_B,L_CS8_B,L_CS9_B,L_CS10_B] -> [Direction])
[0,X,X,X,X,X,X] -> 1;
[X,1,1,1,1,1,1] -> 1;
[1,0,X,X,X,X,X] -> 0;
```

```
[1,X,X,X,0,X,X] -> 0;


end
```

E.2 PAL Two

This section contains the ABEL program that defines PAL two (U9) on the expansion boards. The listing is given below:

```
module  PAL2_program

title   'Combinational logic for PAL 2'

"Combinational logic for expansion board for the DES test bed
"for Pal 2, chip U9
declarations

PAL2    device  'P22V10';    "ACN, TI

"inputs
L_CS10_B        pin 1;
R_L_W_B         pin 2;
L_DS_B          pin 3;
A1_B            pin 4;
L_CS9_B         pin 5;
A2_B            pin 6;
A4_B            pin 7;
A0_B            pin 8;

"outputs
L_PLCi1_EN      pin 14;
L_PLCi0_EN      pin 15;
Crane0_CL       pin 16;
PLCRo1_CL       pin 17;
PLCRo0_CL       pin 18;
L_R_L_W_B       pin 19;

X = .x.;                        ".x. indicates don't care condition

equations

@ALTERNATE              "allow * and + etc in eqns
PLCRo0_CL = !L_CS10_B*!R_L_W_B*!A1_B*!L_DS_B;
PLCRo1_CL = !L_CS10_B*!R_L_W_B*A1_B*!L_DS_B;
Crane0_CL = !L_CS9_B*!R_L_W_B*!A1_B*A4_B*!L_DS_B;
L_PLCi0_EN = L_CS10_B +!R_L_W_B + A1_B + L_DS_B;
L_PLCi1_EN = L_CS10_B +!R_L_W_B + !A1_B + L_DS_B;
L_R_L_W_B = !R_L_W_B;
```

```
test_vectors
([L_CS10_B,R_L_W_B,A1_B,L_DS_B] -> [PLCRo0_CL])
[0,0,0,0] -> 1;
[1,X,X,X] -> 0;

end
```

E.3 Csinit.s

This section contains the listing of the MC68332 program, *csinit.s*. This program configures the chip selects on a processor so that it can access its expansion board. The listing is given below:

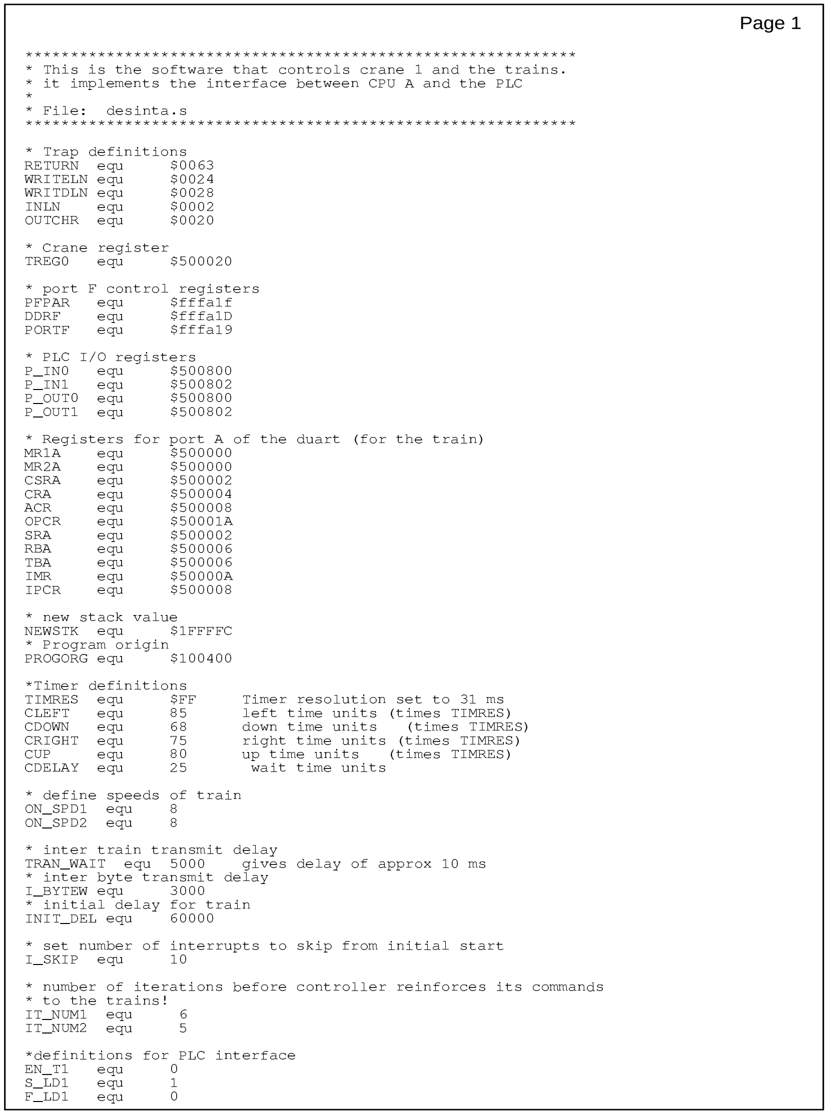
```
*******************************************************************
*
* Name: csinit.s
*
* This program reprograms chip select registers to access
* the CPU's expansion boards
*
*******************************************************************

* Chip select registers
CSPAR0  equu    $fffa44
CSPAR1  equ     $fffa46
CSBAR3  equ     $fffa58
CSBAR5  equ     $fffa60
CSBAR6  equ     $fffa64
CSBAR7  equ     $fffa68
CSBAR8  equ     $fffa6c
CSBAR9  equ     $fffa70
CSBAR10 equ     $fffa74
CSOR3   equ     $fffa5a
CSOR5   equ     $fffa62
CSOR6   equ     $fffa66
CSOR7   equ     $fffa6a
CSOR8   equ     $fffa6e
CSOR9   equ     $fffa72
CSOR10   equ    $fffa76
* trap definition
RETURN  equ     $0063
* Crane register
CREG    equ     $500020

****************************************************
* Start program
****************************************************

org     $5000

* configure pin assignment registers
ori.w   #$3300,CSPAR0
```

```
        ori.w   #$3fc,CSPAR1
* configure base address registers
        move.w  #$1005,CSBAR3
        move.w  #$1405,CSBAR5
        move.w  #$1805,CSBAR7
        move.w  #$1c05,CSBAR8
        move.w  #$5000,CSBAR9
        move.w  #$5008,CSBAR10
* configure chip select option registers
        move.w  #$f87e,CSOR3
        move.w  #$f87e,CSOR5
        move.w  #$f87e,CSOR7
        move.w  #$f87e,CSOR8
        move.w  #$f87e,CSOR9
        move.w  #$f87e,CSOR10

* initialize crane registers to zero
        move.w  #0,CREG

* exit prog
        trap    #15
        dc.w    RETURN
```

E.4 Desinta.s

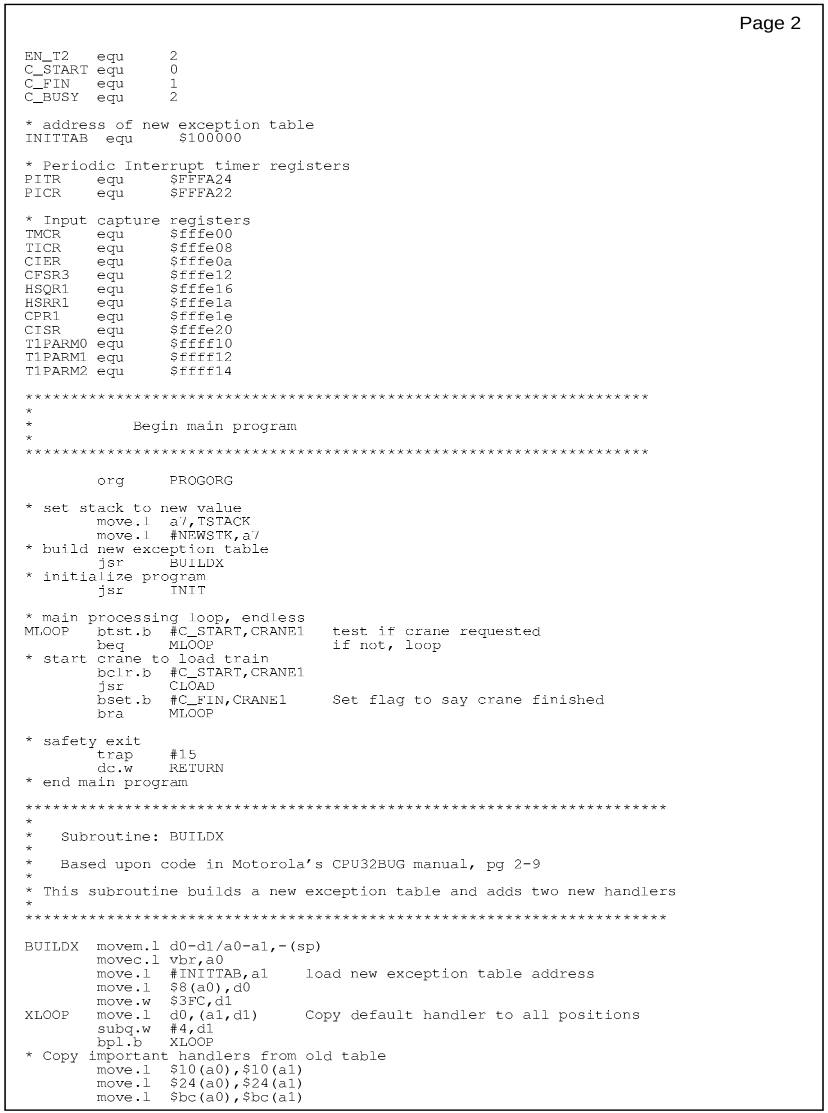
This section contains the listing of the MC68332 program, *desinta.s*. This program provides an interrupt driven interface to the PLC, allowing the PLC remote access to crane 1 and the trains. The listing is given below.

NOTE: Please excuse the fact that each set of pages is numbered “1” and “2”. This is an uncorrectable quirk of the method used to generate the program listing in the two page format.



Page 1

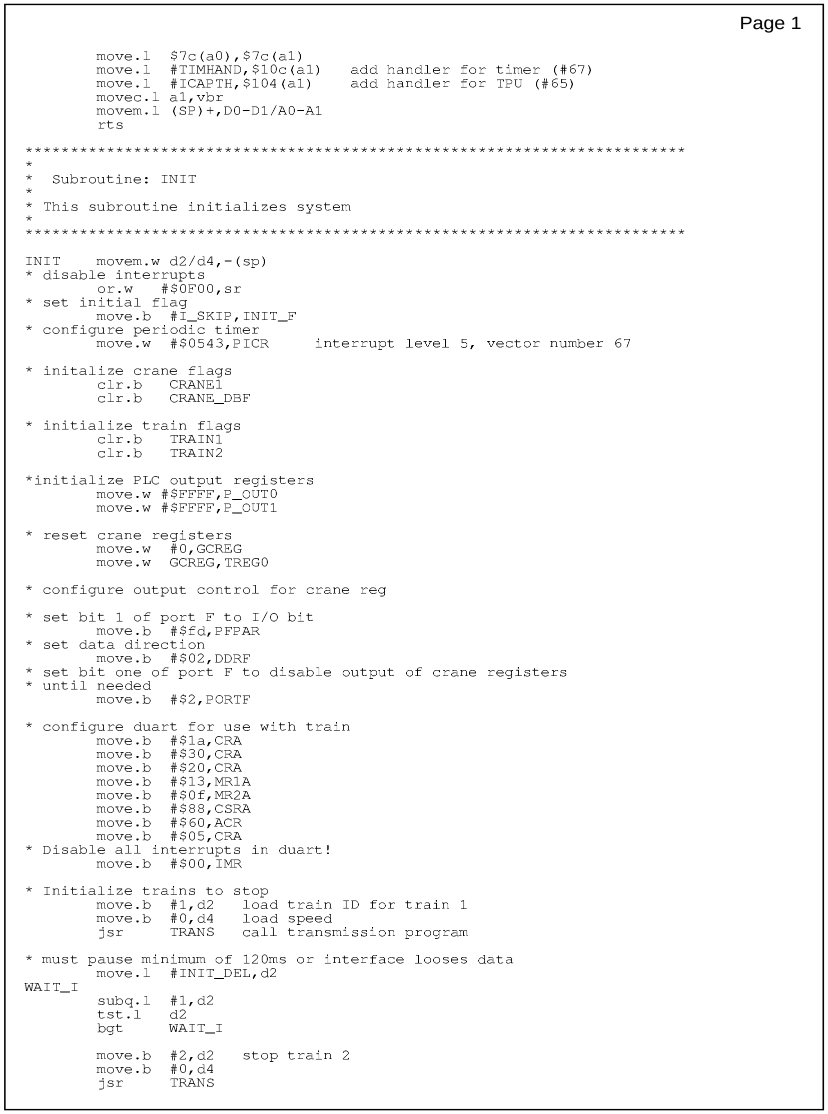
```
**************************************************************
* This is the software that controls crane 1 and the trains.
* it implements the interface between CPU A and the PLC
*
* File:  desinta.s
**************************************************************

* Trap definitions
RETURN  equ     $0063
WRITELN equ     $0024
WRITDLN equ     $0028
INLN    equ     $0002
OUTCHR  equ     $0020

* Crane register
TREG0   equ     $500020

* port F control registers
PFPAR   equ     $fffa1f
DDRF    equ     $fffa1D
PORTF   equ     $fffa19

* PLC I/O registers
P_IN0   equ     $500800
P_IN1   equ     $500802
P_OUT0  equ     $500800
P_OUT1  equ     $500802

* Registers for port A of the duart (for the train)
MR1A    equ     $500000
MR2A    equ     $500000
CSRA    equ     $500002
CRA     equ     $500004
ACR     equ     $500008
OPCR    equ     $50001A
SRA     equ     $500002
RBA     equ     $500006
TBA     equ     $500006
IMR     equ     $50000A
IPCR    equ     $500008

* new stack value
NEWSTK  equ     $1FFFFC
* Program origin
PROGORG equ     $100400

*Timer definitions
TIMRES  equ     $FF     Timer resolution set to 31 ms
CLEFT   equ     85      left time units (times TIMRES)
CDOWN   equ     68      down time units   (times TIMRES)
CRIGHT  equ     75      right time units (times TIMRES)
CUP     equ     80      up time units   (times TIMRES)
CDELAY  equ     25       wait time units

* define speeds of train
ON_SPD1  equ    8
ON_SPD2  equ    8

* inter train transmit delay
TRAN_WAIT  equ  5000    gives delay of approx 10 ms
* inter byte transmit delay
I_BYTEW equ     3000
* initial delay for train
INIT_DEL equ    60000

* set number of interrupts to skip from initial start
I_SKIP  equ     10

* number of iterations before controller reinforces its commands
* to the trains!
IT_NUM1  equ     6
IT_NUM2  equ     5

*definitions for PLC interface
EN_T1   equ     0
S_LD1   equ     1
F_LD1   equ     0
```



Page 2

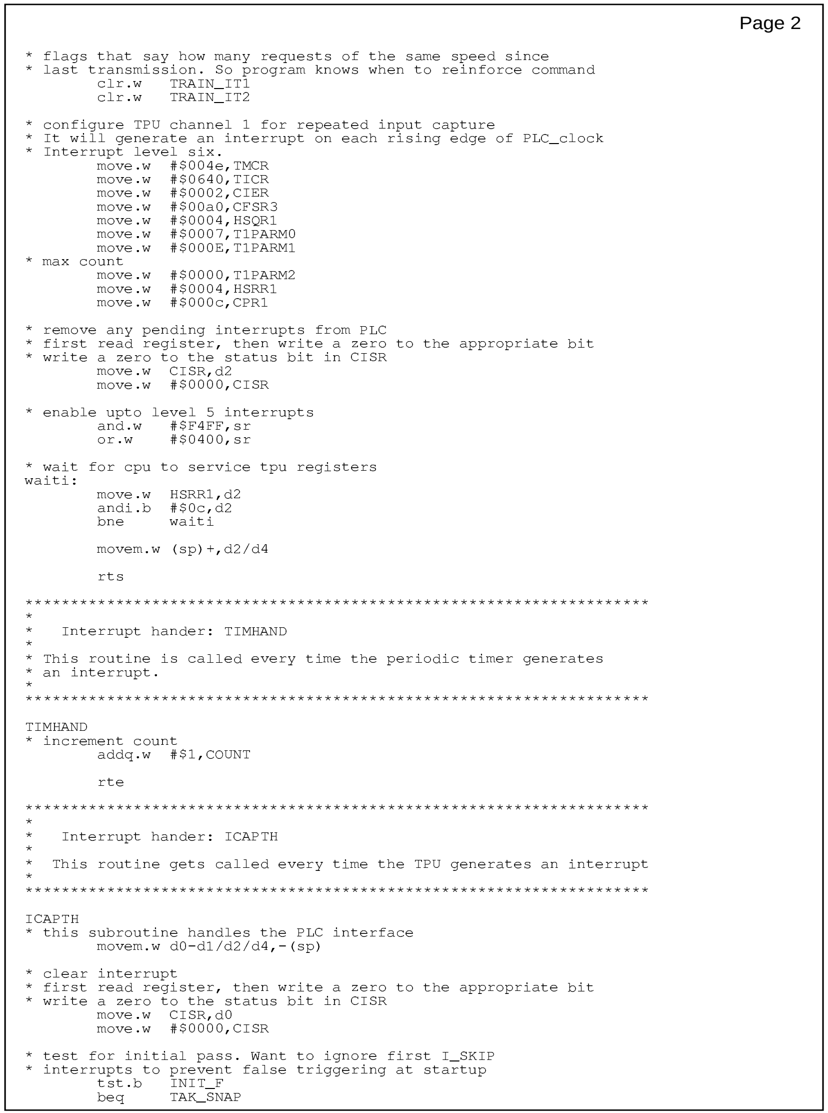
```
EN_T2   equ     2
C_START equ     0
C_FIN   equ     1
C_BUSY  equ     2

* address of new exception table
INITTAB  equ     $100000

* Periodic Interrupt timer registers
PITR    equ     $FFFA24
PICR    equ     $FFFA22

* Input capture registers
TMCR    equ     $fffe00
TICR    equ     $fffe08
CIER    equ     $fffe0a
CFSR3   equ     $fffe12
HSQR1   equ     $fffe16
HSRR1   equ     $fffe1a
CPR1    equ     $fffe1e
CISR    equ     $fffe20
T1PARM0 equ     $ffff10
T1PARM1 equ     $ffff12
T1PARM2 equ     $ffff14

**********************************************************************
*
*           Begin main program
*
**********************************************************************

        org     PROGORG

* set stack to new value
        move.l  a7,TSTACK
        move.l  #NEWSTK,a7
* build new exception table
        jsr     BUILDX
* initialize program
        jsr     INIT

* main processing loop, endless
MLOOP   btst.b  #C_START,CRANE1   test if crane requested
        beq     MLOOP             if not, loop
* start crane to load train
        bclr.b  #C_START,CRANE1
        jsr     CLOAD
        bset.b  #C_FIN,CRANE1     Set flag to say crane finished
        bra     MLOOP

* safety exit
        trap    #15
        dc.w    RETURN
* end main program

***********************************************************************
*
*   Subroutine: BUILDX
*
*   Based upon code in Motorola’s CPU32BUG manual, pg 2-9
*
* This subroutine builds a new exception table and adds two new handlers
*
***********************************************************************

BUILDX  movem.l d0-d1/a0-a1,-(sp)
        movec.l vbr,a0
        move.l  #INITTAB,a1    load new exception table address
        move.l  $8(a0),d0
        move.w  $3FC,d1
XLOOP   move.l  d0,(a1,d1)     Copy default handler to all positions
        subq.w  #4,d1
        bpl.b   XLOOP
* Copy important handlers from old table
        move.l  $10(a0),$10(a1)
        move.l  $24(a0),$24(a1)
        move.l  $bc(a0),$bc(a1)
```

1



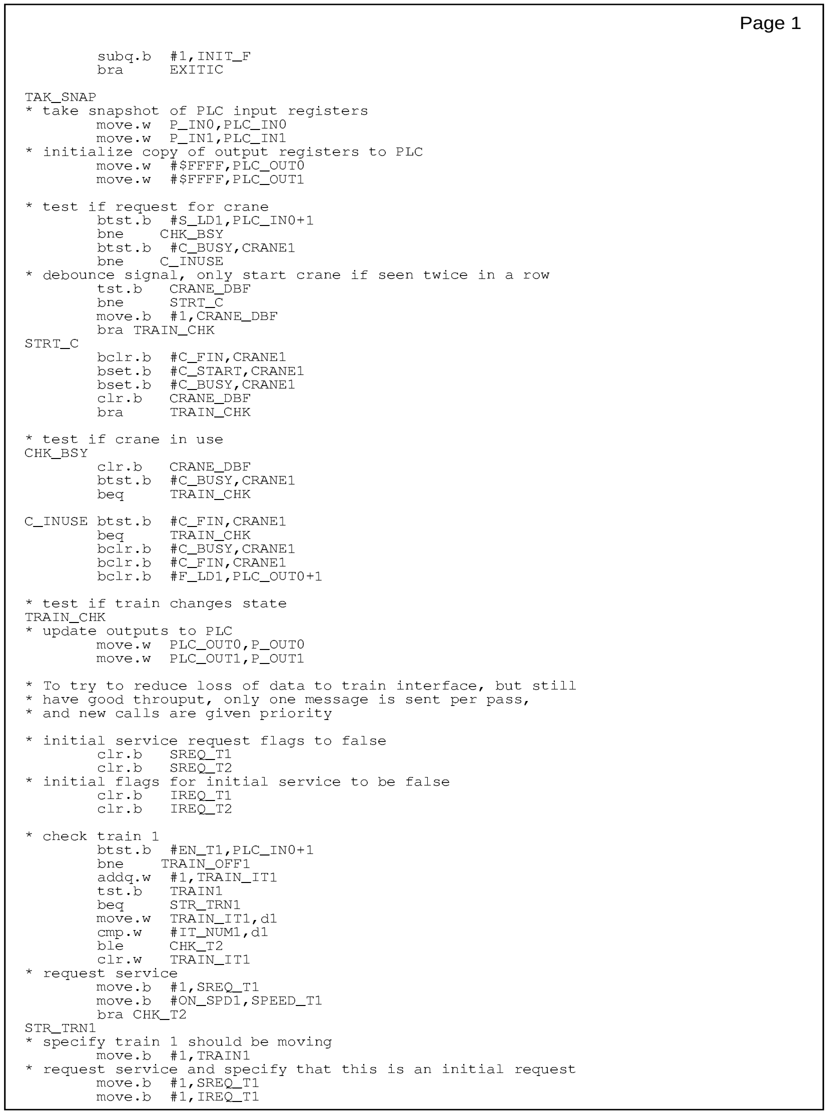
Page 1

```
        move.l  $7c(a0),$7c(a1)
        move.l  #TIMHAND,$10c(a1)   add handler for timer (#67)
        move.l  #ICAPTH,$104(a1)    add handler for TPU (#65)
        movec.l a1,vbr
        movem.l (SP)+,D0-D1/A0-A1
        rts

*************************************************************************
*
*  Subroutine: INIT
*
* This subroutine initializes system
*
*************************************************************************

INIT    movem.w d2/d4,-(sp)
* disable interrupts
        or.w    #$0F00,sr
* set initial flag
        move.b  #I_SKIP,INIT_F
* configure periodic timer
        move.w  #$0543,PICR     interrupt level 5, vector number 67

* initalize crane flags
        clr.b   CRANE1
        clr.b   CRANE_DBF

* initialize train flags
        clr.b   TRAIN1
        clr.b   TRAIN2

*initialize PLC output registers
        move.w #$FFFF,P_OUT0
        move.w #$FFFF,P_OUT1

* reset crane registers
        move.w  #0,GCREG
        move.w  GCREG,TREG0

* configure output control for crane reg

* set bit 1 of port F to I/O bit
        move.b  #$fd,PFPAR
* set data direction
        move.b  #$02,DDRF
* set bit one of port F to disable output of crane registers
* until needed
        move.b  #$2,PORTF

* configure duart for use with train
        move.b  #$1a,CRA
        move.b  #$30,CRA
        move.b  #$20,CRA
        move.b  #$13,MR1A
        move.b  #$0f,MR2A
        move.b  #$88,CSRA
        move.b  #$60,ACR
        move.b  #$05,CRA
* Disable all interrupts in duart!
        move.b  #$00,IMR

* Initialize trains to stop
        move.b  #1,d2   load train ID for train 1
        move.b  #0,d4   load speed
        jsr     TRANS   call transmission program

* must pause minimum of 120ms or interface looses data
        move.l  #INIT_DEL,d2
WAIT_I
        subq.l  #1,d2
        tst.l   d2
        bgt     WAIT_I

        move.b  #2,d2   stop train 2
        move.b  #0,d4
        jsr     TRANS
```



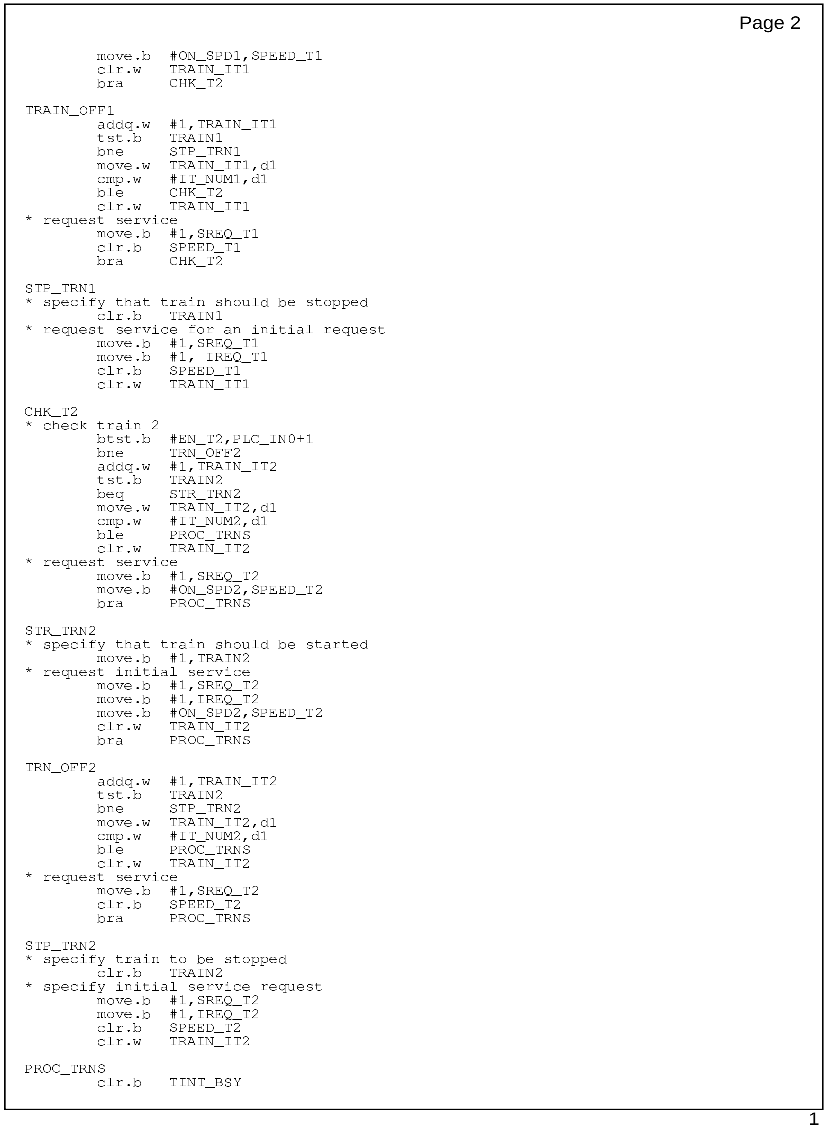
Page 2

```
* flags that say how many requests of the same speed since
* last transmission. So program knows when to reinforce command
        clr.w   TRAIN_IT1
        clr.w   TRAIN_IT2

* configure TPU channel 1 for repeated input capture
* It will generate an interrupt on each rising edge of PLC_clock
* Interrupt level six.
        move.w  #$004e,TMCR
        move.w  #$0640,TICR
        move.w  #$0002,CIER
        move.w  #$00a0,CFSR3
        move.w  #$0004,HSQR1
        move.w  #$0007,T1PARM0
        move.w  #$000E,T1PARM1
* max count
        move.w  #$0000,T1PARM2
        move.w  #$0004,HSRR1
        move.w  #$000c,CPR1

* remove any pending interrupts from PLC
* first read register, then write a zero to the appropriate bit
* write a zero to the status bit in CISR
        move.w  CISR,d2
        move.w  #$0000,CISR

* enable upto level 5 interrupts
        and.w   #$F4FF,sr
        or.w    #$0400,sr

* wait for cpu to service tpu registers
waiti:
        move.w  HSRR1,d2
        andi.b  #$0c,d2
        bne     waiti

        movem.w (sp)+,d2/d4

        rts

*********************************************************************
*
*   Interrupt hander: TIMHAND
*
* This routine is called every time the periodic timer generates
* an interrupt.
*
*********************************************************************

TIMHAND
* increment count
        addq.w  #$1,COUNT

        rte

*********************************************************************
*
*   Interrupt hander: ICAPTH
*
*  This routine gets called every time the TPU generates an interrupt
*
*********************************************************************

ICAPTH
* this subroutine handles the PLC interface
        movem.w d0-d1/d2/d4,-(sp)

* clear interrupt
* first read register, then write a zero to the appropriate bit
* write a zero to the status bit in CISR
        move.w  CISR,d0
        move.w  #$0000,CISR

* test for initial pass. Want to ignore first I_SKIP
* interrupts to prevent false triggering at startup
        tst.b   INIT_F
        beq     TAK_SNAP
```



Page 1

```
        subq.b  #1,INIT_F
        bra     EXITIC

TAK_SNAP
* take snapshot of PLC input registers
        move.w  P_IN0,PLC_IN0
        move.w  P_IN1,PLC_IN1
* initialize copy of output registers to PLC
        move.w  #$FFFF,PLC_OUT0
        move.w  #$FFFF,PLC_OUT1

* test if request for crane
        btst.b  #S_LD1,PLC_IN0+1
        bne    CHK_BSY
        btst.b  #C_BUSY,CRANE1
        bne    C_INUSE
* debounce signal, only start crane if seen twice in a row
        tst.b   CRANE_DBF
        bne     STRT_C
        move.b  #1,CRANE_DBF
        bra TRAIN_CHK
STRT_C
        bclr.b  #C_FIN,CRANE1
        bset.b  #C_START,CRANE1
        bset.b  #C_BUSY,CRANE1
        clr.b   CRANE_DBF
        bra     TRAIN_CHK

* test if crane in use
CHK_BSY
        clr.b   CRANE_DBF
        btst.b  #C_BUSY,CRANE1
        beq     TRAIN_CHK

C_INUSE btst.b  #C_FIN,CRANE1
        beq     TRAIN_CHK
        bclr.b  #C_BUSY,CRANE1
        bclr.b  #C_FIN,CRANE1
        bclr.b  #F_LD1,PLC_OUT0+1

* test if train changes state
TRAIN_CHK
* update outputs to PLC
        move.w  PLC_OUT0,P_OUT0
        move.w  PLC_OUT1,P_OUT1

* To try to reduce loss of data to train interface, but still
* have good throuput, only one message is sent per pass,
* and new calls are given priority

* initial service request flags to false
        clr.b   SREQ_T1
        clr.b   SREQ_T2
* initial flags for initial service to be false
        clr.b   IREQ_T1
        clr.b   IREQ_T2

* check train 1
        btst.b  #EN_T1,PLC_IN0+1
        bne    TRAIN_OFF1
        addq.w  #1,TRAIN_IT1
        tst.b   TRAIN1
        beq     STR_TRN1
        move.w  TRAIN_IT1,d1
        cmp.w   #IT_NUM1,d1
        ble     CHK_T2
        clr.w   TRAIN_IT1
* request service
        move.b  #1,SREQ_T1
        move.b  #ON_SPD1,SPEED_T1
        bra CHK_T2
STR_TRN1
* specify train 1 should be moving
        move.b  #1,TRAIN1
* request service and specify that this is an initial request
        move.b  #1,SREQ_T1
        move.b  #1,IREQ_T1
```



Page 2

```
        move.b  #ON_SPD1,SPEED_T1
        clr.w   TRAIN_IT1
        bra     CHK_T2

TRAIN_OFF1
        addq.w  #1,TRAIN_IT1
        tst.b   TRAIN1
        bne     STP_TRN1
        move.w  TRAIN_IT1,d1
        cmp.w   #IT_NUM1,d1
        ble     CHK_T2
        clr.w   TRAIN_IT1
* request service
        move.b  #1,SREQ_T1
        clr.b   SPEED_T1
        bra     CHK_T2

STP_TRN1
* specify that train should be stopped
        clr.b   TRAIN1
* request service for an initial request
        move.b  #1,SREQ_T1
        move.b  #1, IREQ_T1
        clr.b   SPEED_T1
        clr.w   TRAIN_IT1

CHK_T2
* check train 2
        btst.b  #EN_T2,PLC_IN0+1
        bne     TRN_OFF2
        addq.w  #1,TRAIN_IT2
        tst.b   TRAIN2
        beq     STR_TRN2
        move.w  TRAIN_IT2,d1
        cmp.w   #IT_NUM2,d1
        ble     PROC_TRNS
        clr.w   TRAIN_IT2
* request service
        move.b  #1,SREQ_T2
        move.b  #ON_SPD2,SPEED_T2
        bra     PROC_TRNS

STR_TRN2
* specify that train should be started
        move.b  #1,TRAIN2
* request initial service
        move.b  #1,SREQ_T2
        move.b  #1,IREQ_T2
        move.b  #ON_SPD2,SPEED_T2
        clr.w   TRAIN_IT2
        bra     PROC_TRNS

TRN_OFF2
        addq.w  #1,TRAIN_IT2
        tst.b   TRAIN2
        bne     STP_TRN2
        move.w  TRAIN_IT2,d1
        cmp.w   #IT_NUM2,d1
        ble     PROC_TRNS
        clr.w   TRAIN_IT2
* request service
        move.b  #1,SREQ_T2
        clr.b   SPEED_T2
        bra     PROC_TRNS

STP_TRN2
* specify train to be stopped
        clr.b   TRAIN2
* specify initial service request
        move.b  #1,SREQ_T2
        move.b  #1,IREQ_T2
        clr.b   SPEED_T2
        clr.w   TRAIN_IT2

PROC_TRNS
        clr.b   TINT_BSY
```

1

```
* check if TRAIN 1 requires initial service
        tst.b   IREQ_T1
        bne     TRAN_T1

* if train 2 requires service, if so skip train 1
        tst.b   SREQ_T2
        bne     TRAN_T2

* check if train 1 requires service
        tst.b   SREQ_T1
        beq     EXITIC

TRAN_T1
        move.b  #1,TINT_BSY
        move.b  #1,d2
        move.b  SPEED_T1,d4
        jsr     TRANS

* test if TRAIN 2 requires initial service
        tst.b   IREQ_T2
        beq     EXITIC

TRAN_T2
* test if just transmitted to train 1.  If so, delay

        tst.b   TINT_BSY
        beq     BG_T2

        move.w  #TRAN_WAIT,d0
WAIT_T2
        subq.w  #1,d0
        tst.w   d0
        bgt     WAIT_T2

BG_T2
        move.b  #2,d2
        move.b  SPEED_T2,d4
        jsr     TRANS

EXITIC
        movem.w (sp)+,D0-d1/d2/d4

        rte

********************************************************************
*
*  Subroutine: CLOAD
*
*  This routine takes the crane through the paces of loading.
*  The controller is open loop (timer driven) and doesn't actually
*  try to load anything.
*
********************************************************************

CLOAD   movem.l d0-d1,-(sp)
* load crane register ghost into d0
        move.w  GCREG,d0   ghost is used since crane is read only
        clr.w   d0
        move.w  d0,TREG0

* clear bit 1 of port F to enable output of crane registers
        move.b  #0,PORTF

* setup and start timer
        clr.w   COUNT
        move.w  #CLEFT,d1       load horizontal time
        move.w  #TIMRES,PITR

* start crane moving to left
        or.w    #$0200,d0
        and.w   #$f2ff,d0
        move.w  d0,TREG0

LEFT    cmp.w   COUNT,d1        loop until count reached
        bgt     LEFT

* stop crane
```

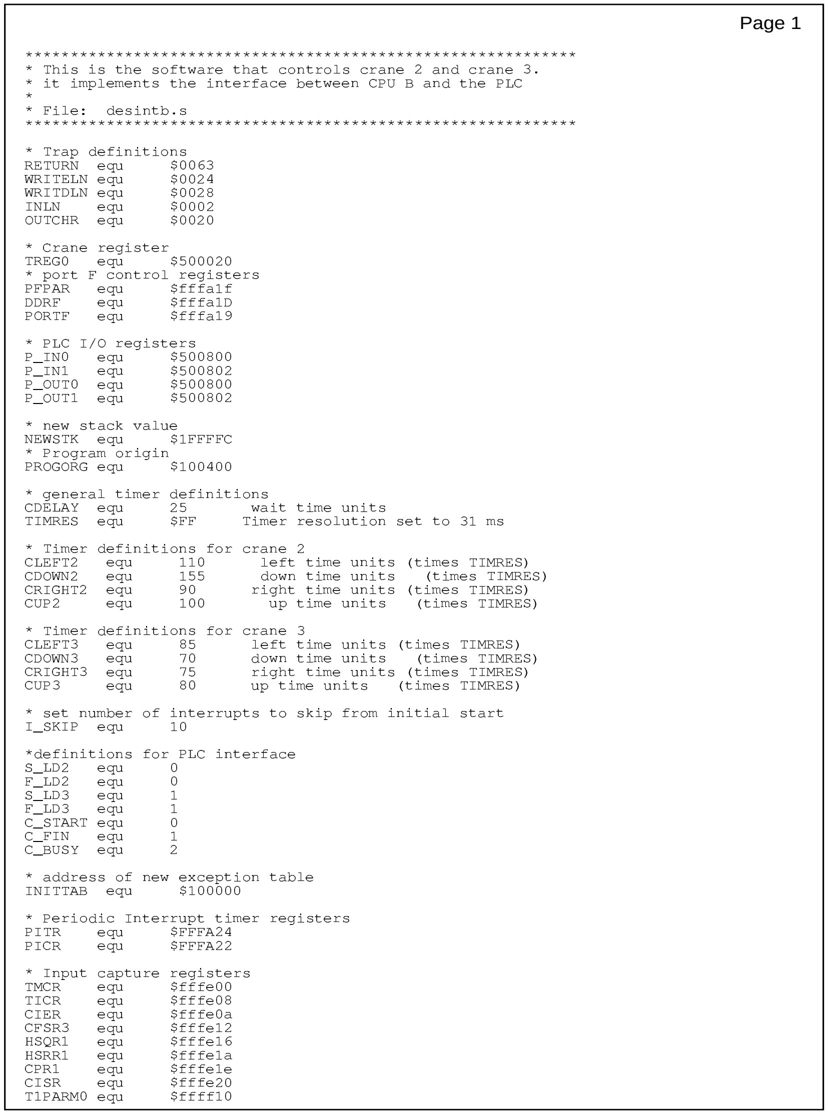
```
        and.w   #$f0ff,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY1  cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY1

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDOWN,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* lower magnet
        or.w    #$0400,d0
        and.w   #$f4ff,d0
        move.w  d0,TREG0

DOWN1   cmp.w   COUNT,d1        loop until count reached
        bgt     DOWN1

* stop crane
        and.w   #$f0ff,d0
        move.w  d0,TREG0

* turn magnet on
        or.w    #$1000,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY2  cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY2

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CUP,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* raise magnet
        or.w    #$0800,d0
        and.w   #$f8ff,d0
        move.w  d0,TREG0

UP1     cmp.w   COUNT,d1        loop until count reached
        bgt     UP1

* stop crane
        and.w   #$f0ff,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY3  cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY3

* stop timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CRIGHT,d1        load horizontal time
* restart timer
        move.w  #TIMRES,PITR

* start crane moving to right
        or.w    #$0100,d0
        and.w   #$f1ff,d0
        move.w  d0,TREG0

RIGHT   cmp.w   COUNT,d1         loop until count reached
        bgt     RIGHT

* stop crane
        and.w   #$f0ff,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1        load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY4  cmp.w   COUNT,d1         loop until count reached
        bgt     DELAY4

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDOWN,d1        load vertical time
* restart timer
        move.w  #TIMRES,PITR

* lower magnet
        or.w    #$0400,d0
        and.w   #$f4ff,d0
        move.w  d0,TREG0

DOWN2   cmp.w   COUNT,d1         loop until count reached
        bgt     DOWN2

* stop crane
        and.w   #$f0ff,d0
        move.w  d0,TREG0

* turn magnet off
        and.w   #$efff,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1        load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY5  cmp.w   COUNT,d1         loop until count reached
        bgt     DELAY5

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CUP,d1        load vertical time
* restart timer
        move.w  #TIMRES,PITR

* raise magnet
        or.w    #$0800,d0
        and.w   #$f8ff,d0
        move.w  d0,TREG0

UP2     cmp.w   COUNT,d1         loop until count reached
        bgt     UP2

* stop crane
        and.w   #$f0ff,d0
```

```
        move.w  d0,TREG0

* stop timer
        move.w  #0,PITR

*restore ghost register to memory
        move.w  d0,GCREG

* set bit 1 of port F to disable output of crane registers
        move.b  #$2,PORTF

        movem.l (sp)+,d0-d1

        rts

************************************************************************
*
* Subroutine: TRANS
*
* This routine transmits commands to the train interface
************************************************************************

TRANS:  movem.l d0,-(sp)

*transmit to train
TSTTX1  btst.b  #2,SRA
        beq     TSTTX1
        move.b  d4,TBA   transmit speed

        move.w  #I_BYTEW,d0
WAITTRAN
        subq.w  #1,d0
        tst.w   d0
        bgt     WAITTRAN

TSTTX2  btst.b  #2,SRA
        beq     TSTTX2
        move.b  d2,TBA   transmit train ID

EX
        movem.l (sp)+,d0

        rts

PROMPT          dc.b    14, 'Enter Command:'
PROMPT1         dc.b    18, 'train stopped:    '
PROMPT2         dc.b    18, 'train started:    '
PROMPT3         dc.b    18, 'crane started:    '
TRAIN1          ds.b    1
BUFFER          ds.b    256
CRANE1          ds.b    1
ALIGN           even
COUNT           ds.w    1
GCREG           ds.w    1
PLC_OUT0        ds.w    1
PLC_OUT1        ds.w    1
PLC_IN0         ds.w    1
PLC_IN1         ds.w    1
TSTACK          ds.l    1
TRAIN_IT1       ds.w    1
TRAIN_IT2       ds.w    1
TMP             ds.w    1
TMP2            ds.w    1
TRAIN2          ds.b    1
TINT_BSY        ds.b    1
CNT_CRN         ds.b    1
INIT_F          ds.b    1
CRANE_DBF       ds.b    1
SREQ_T1         ds.b    1
SREQ_T2         ds.b    1
IREQ_T1         ds.b    1
IREQ_T2         ds.b    1
SPEED_T1        ds.b    1
SPEED_T2        ds.b    1
```

E.5 Desintb.s

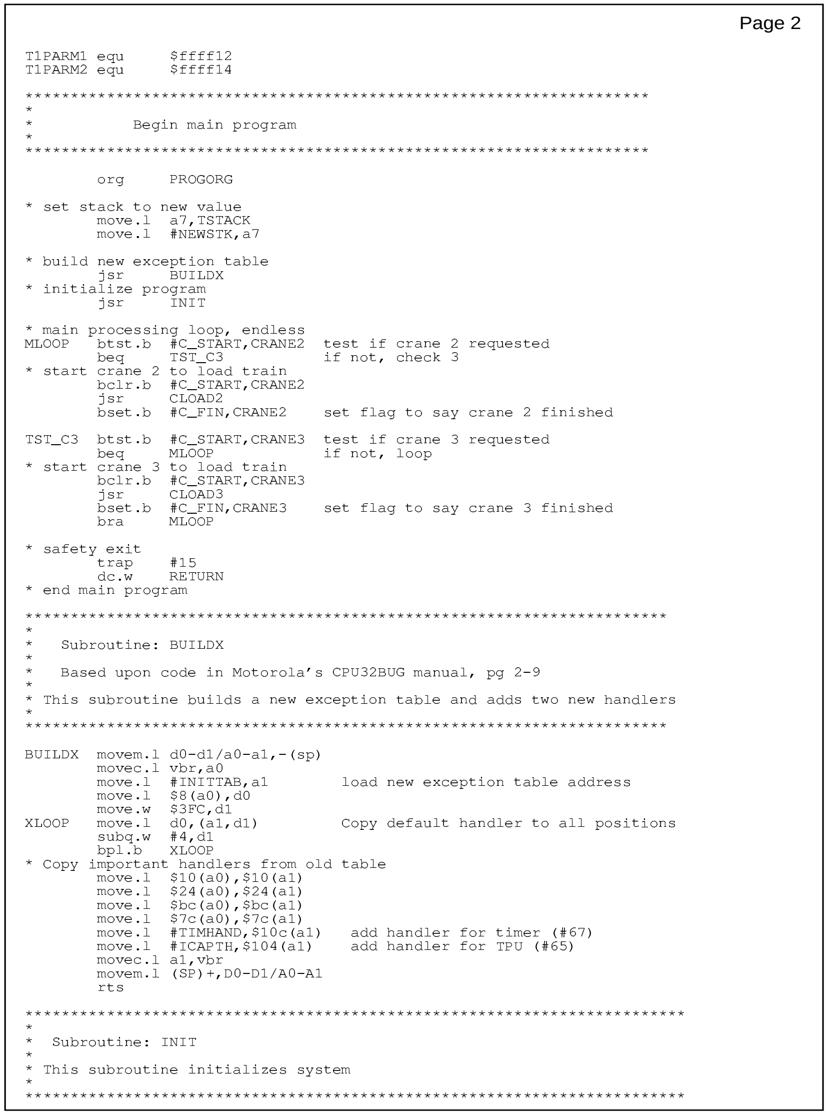
This section contains the listing of the MC68332 program, *desintb.s.* This program provides an interrupt driven interface to the PLC, allowing the PLC remote access to cranes 2 and 3. The listing is given below.

NOTE: Please excuse the fact that each set of pages is numbered "1" and "2". This is an uncorrectable quirk of the method used to generate the program listing in the two page format.



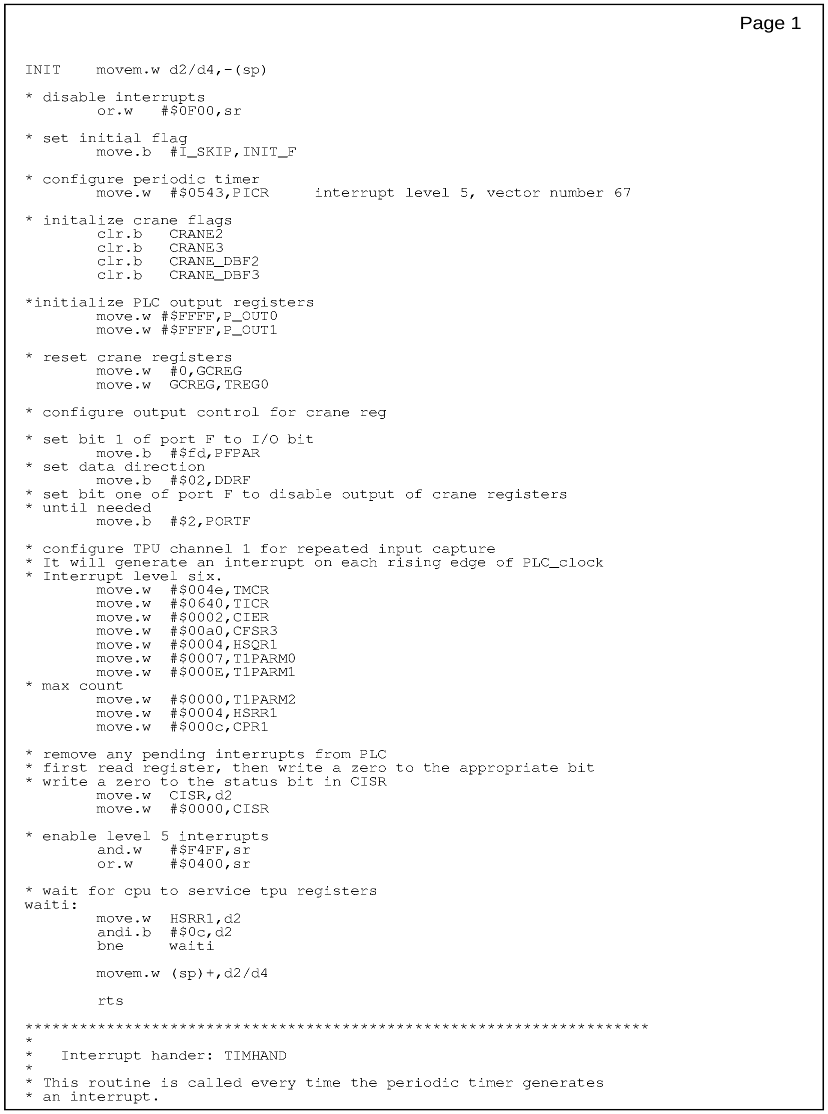
Page 1

```
**************************************************************
* This is the software that controls crane 2 and crane 3.
* it implements the interface between CPU B and the PLC
*
* File:  desintb.s
**************************************************************

* Trap definitions
RETURN  equu     $0063
WRITELN equ     $0024
WRITDLN equ     $0028
INLN    equ     $0002
OUTCHR  equ     $0020

* Crane register
TREG0   equ     $500020
* port F control registers
PFPAR   equ     $fffa1f
DDRF    equ     $fffa1D
PORTF   equ     $fffa19

* PLC I/O registers
P_IN0   equ     $500800
P_IN1   equ     $500802
P_OUT0  equ     $500800
P_OUT1  equ     $500802

* new stack value
NEWSTK  equ     $1FFFFC
* Program origin
PROGORG equ     $100400

* general timer definitions
CDELAY  equ     25       wait time units
TIMRES  equ     $FF     Timer resolution set to 31 ms

* Timer definitions for crane 2
CLEFT2   equ     110      left time units (times TIMRES)
CDOWN2   equ     155      down time units   (times TIMRES)
CRIGHT2  equ     90      right time units (times TIMRES)
CUP2     equ     100       up time units   (times TIMRES)

* Timer definitions for crane 3
CLEFT3   equ     85      left time units (times TIMRES)
CDOWN3   equ     70      down time units   (times TIMRES)
CRIGHT3  equ     75      right time units (times TIMRES)
CUP3     equ     80      up time units   (times TIMRES)

* set number of interrupts to skip from initial start
I_SKIP  equ     10

*definitions for PLC interface
S_LD2   equ     0
F_LD2   equ     0
S_LD3   equ     1
F_LD3   equ     1
C_START equ     0
C_FIN   equ     1
C_BUSY  equ     2

* address of new exception table
INITTAB  equ     $100000

* Periodic Interrupt timer registers
PITR    equ     $FFFA24
PICR    equ     $FFFA22

* Input capture registers
TMCR    equ     $fffe00
TICR    equ     $fffe08
CIER    equ     $fffe0a
CFSR3   equ     $fffe12
HSQR1   equ     $fffe16
HSRR1   equ     $fffe1a
CPR1    equ     $fffe1e
CISR    equ     $fffe20
T1PARM0 equ     $ffff10
```



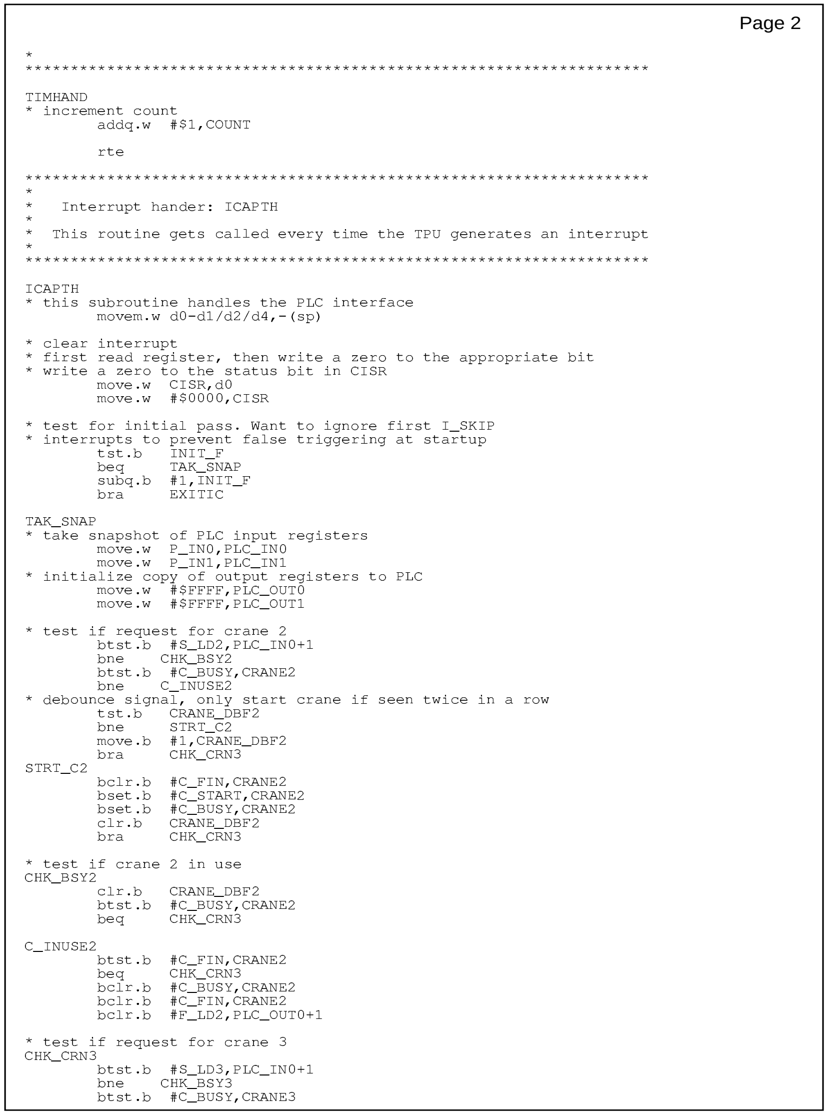
Page 2

```
T1PARM1 equ     $ffff12
T1PARM2 equ     $ffff14

**********************************************************************
*
*           Begin main program
*
**********************************************************************

        org     PROGORG

* set stack to new value
        move.l  a7,TSTACK
        move.l  #NEWSTK,a7

* build new exception table
        jsr     BUILDX
* initialize program
        jsr     INIT

* main processing loop, endless
MLOOP   btst.b  #C_START,CRANE2  test if crane 2 requested
        beq     TST_C3           if not, check 3
* start crane 2 to load train
        bclr.b  #C_START,CRANE2
        jsr     CLOAD2
        bset.b  #C_FIN,CRANE2    set flag to say crane 2 finished

TST_C3  btst.b  #C_START,CRANE3  test if crane 3 requested
        beq     MLOOP            if not, loop
* start crane 3 to load train
        bclr.b  #C_START,CRANE3
        jsr     CLOAD3
        bset.b  #C_FIN,CRANE3    set flag to say crane 3 finished
        bra     MLOOP

* safety exit
        trap    #15
        dc.w    RETURN
* end main program

************************************************************************
*
*   Subroutine: BUILDX
*
*   Based upon code in Motorola's CPU32BUG manual, pg 2-9
*
* This subroutine builds a new exception table and adds two new handlers
*
************************************************************************

BUILDX  movem.l d0-d1/a0-a1,-(sp)
        movec.l vbr,a0
        move.l  #INITTAB,a1        load new exception table address
        move.l  $8(a0),d0
        move.w  $3FC,d1
XLOOP   move.l  d0,(a1,d1)         Copy default handler to all positions
        subq.w  #4,d1
        bpl.b   XLOOP
* Copy important handlers from old table
        move.l  $10(a0),$10(a1)
        move.l  $24(a0),$24(a1)
        move.l  $bc(a0),$bc(a1)
        move.l  $7c(a0),$7c(a1)
        move.l  #TIMHAND,$10c(a1)   add handler for timer (#67)
        move.l  #ICAPTH,$104(a1)    add handler for TPU (#65)
        movec.l a1,vbr
        movem.l (SP)+,D0-D1/A0-A1
        rts

**************************************************************************
*
*  Subroutine: INIT
*
* This subroutine initializes system
*
**************************************************************************
```



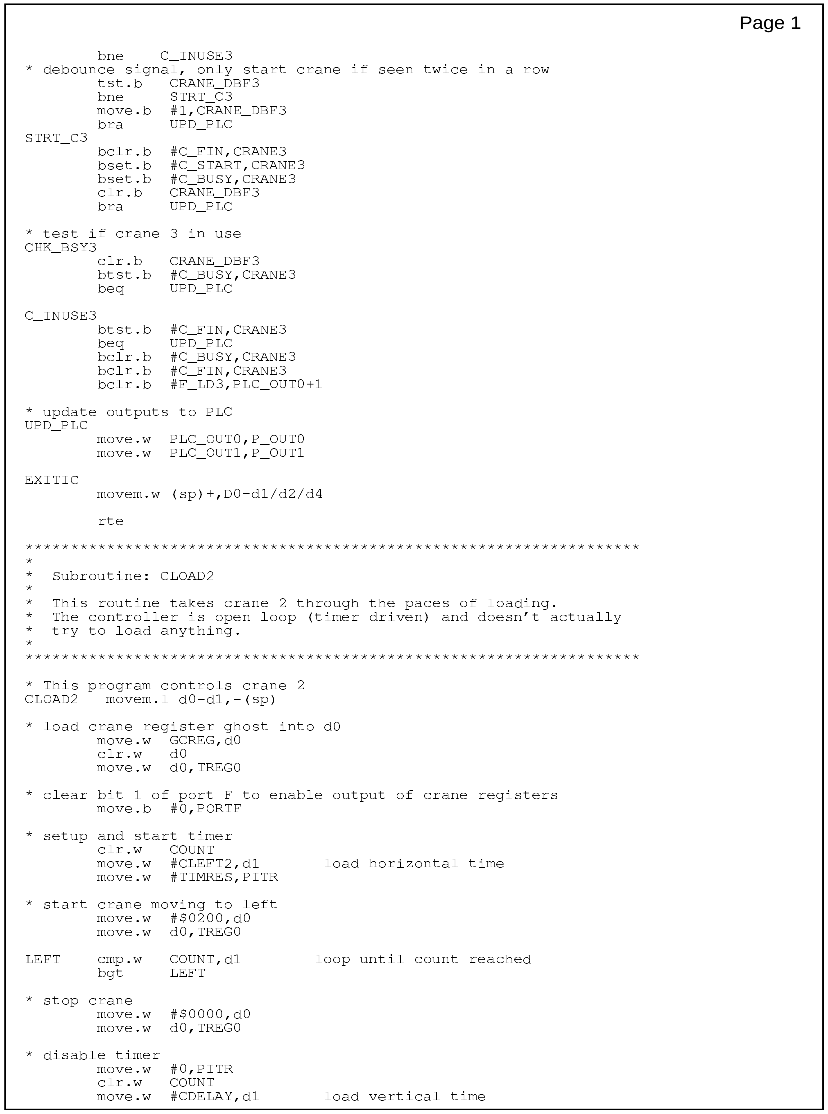
Page 1

```
INIT    movem.w d2/d4,-(sp)

* disable interrupts
        or.w   #$0F00,sr

* set initial flag
        move.b  #I_SKIP,INIT_F

* configure periodic timer
        move.w  #$0543,PICR     interrupt level 5, vector number 67

* initalize crane flags
        clr.b   CRANE2
        clr.b   CRANE3
        clr.b   CRANE_DBF2
        clr.b   CRANE_DBF3

*initialize PLC output registers
        move.w #$FFFF,P_OUT0
        move.w #$FFFF,P_OUT1

* reset crane registers
        move.w  #0,GCREG
        move.w  GCREG,TREG0

* configure output control for crane reg

* set bit 1 of port F to I/O bit
        move.b  #$fd,PFPAR
* set data direction
        move.b  #$02,DDRF
* set bit one of port F to disable output of crane registers
* until needed
        move.b  #$2,PORTF

* configure TPU channel 1 for repeated input capture
* It will generate an interrupt on each rising edge of PLC_clock
* Interrupt level six.
        move.w  #$004e,TMCR
        move.w  #$0640,TICR
        move.w  #$0002,CIER
        move.w  #$00a0,CFSR3
        move.w  #$0004,HSQR1
        move.w  #$0007,T1PARM0
        move.w  #$000E,T1PARM1
* max count
        move.w  #$0000,T1PARM2
        move.w  #$0004,HSRR1
        move.w  #$000c,CPR1

* remove any pending interrupts from PLC
* first read register, then write a zero to the appropriate bit
* write a zero to the status bit in CISR
        move.w  CISR,d2
        move.w  #$0000,CISR

* enable level 5 interrupts
        and.w   #$F4FF,sr
        or.w    #$0400,sr

* wait for cpu to service tpu registers
waiti:
        move.w  HSRR1,d2
        andi.b  #$0c,d2
        bne     waiti

        movem.w (sp)+,d2/d4

        rts

**********************************************************************
*
*   Interrupt hander: TIMHAND
*
* This routine is called every time the periodic timer generates
* an interrupt.
```



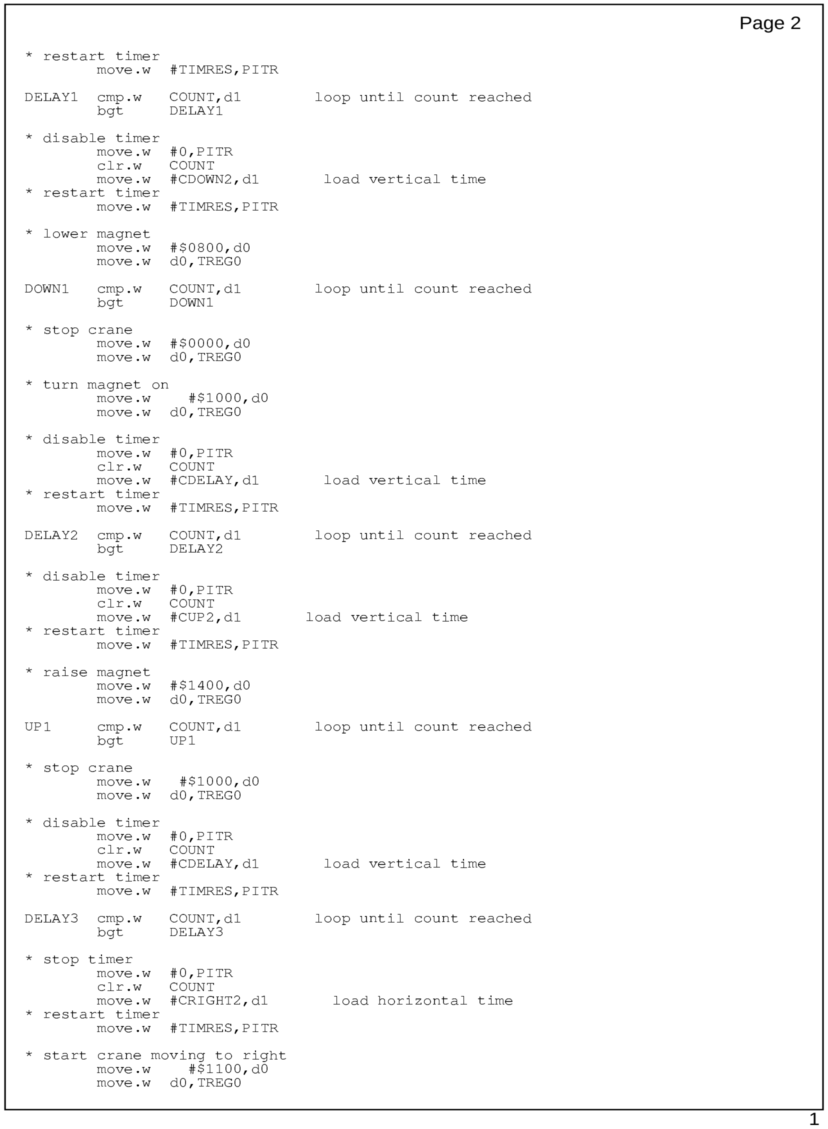
Page 2

```
*
**********************************************************************

TIMHAND
* increment count
        addq.w  #$1,COUNT

        rte

**********************************************************************
*
*   Interrupt hander: ICAPTH
*
*  This routine gets called every time the TPU generates an interrupt
*
**********************************************************************

ICAPTH
* this subroutine handles the PLC interface
        movem.w d0-d1/d2/d4,-(sp)

* clear interrupt
* first read register, then write a zero to the appropriate bit
* write a zero to the status bit in CISR
        move.w  CISR,d0
        move.w  #$0000,CISR

* test for initial pass. Want to ignore first I_SKIP
* interrupts to prevent false triggering at startup
        tst.b   INIT_F
        beq     TAK_SNAP
        subq.b  #1,INIT_F
        bra     EXITIC

TAK_SNAP
* take snapshot of PLC input registers
        move.w  P_IN0,PLC_IN0
        move.w  P_IN1,PLC_IN1
* initialize copy of output registers to PLC
        move.w  #$FFFF,PLC_OUT0
        move.w  #$FFFF,PLC_OUT1

* test if request for crane 2
        btst.b  #S_LD2,PLC_IN0+1
        bne    CHK_BSY2
        btst.b  #C_BUSY,CRANE2
        bne    C_INUSE2
* debounce signal, only start crane if seen twice in a row
        tst.b   CRANE_DBF2
        bne     STRT_C2
        move.b  #1,CRANE_DBF2
        bra     CHK_CRN3
STRT_C2
        bclr.b  #C_FIN,CRANE2
        bset.b  #C_START,CRANE2
        bset.b  #C_BUSY,CRANE2
        clr.b   CRANE_DBF2
        bra     CHK_CRN3

* test if crane 2 in use
CHK_BSY2
        clr.b   CRANE_DBF2
        btst.b  #C_BUSY,CRANE2
        beq     CHK_CRN3

C_INUSE2
        btst.b  #C_FIN,CRANE2
        beq     CHK_CRN3
        bclr.b  #C_BUSY,CRANE2
        bclr.b  #C_FIN,CRANE2
        bclr.b  #F_LD2,PLC_OUT0+1

* test if request for crane 3
CHK_CRN3
        btst.b  #S_LD3,PLC_IN0+1
        bne    CHK_BSY3
        btst.b  #C_BUSY,CRANE3
```



Page 1

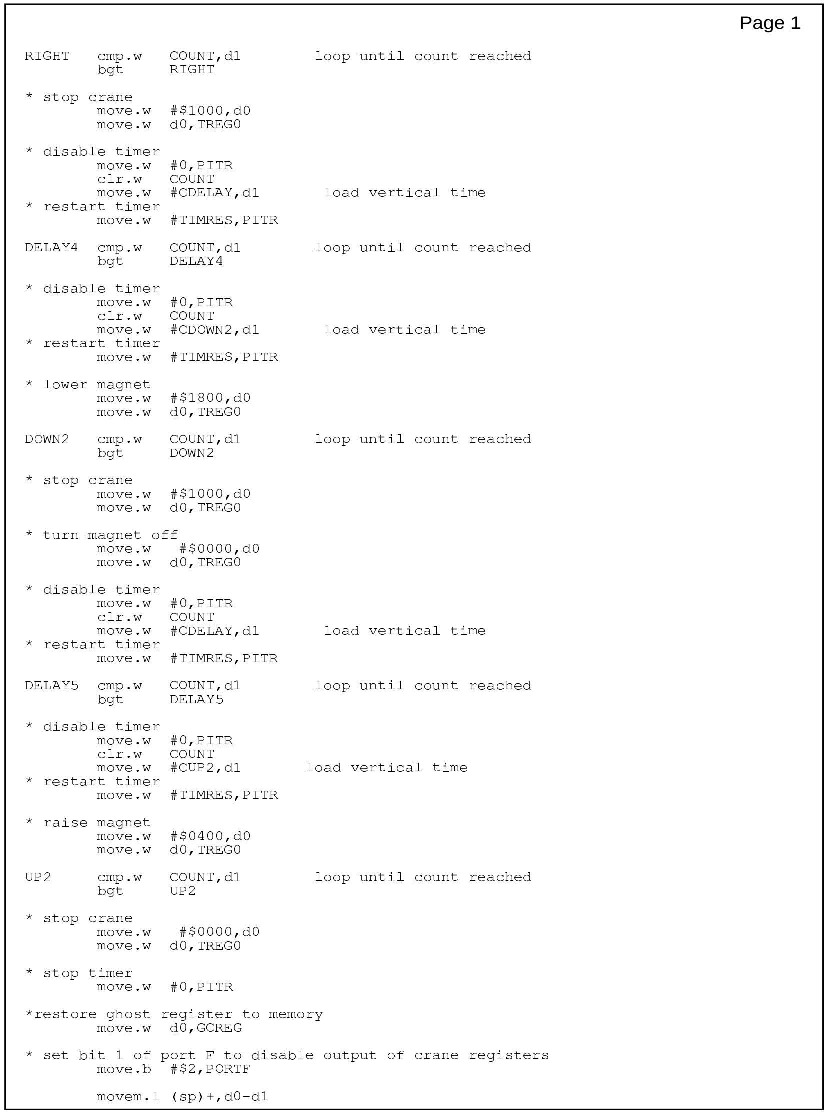
```
        bne    C_INUSE3
* debounce signal, only start crane if seen twice in a row
        tst.b   CRANE_DBF3
        bne     STRT_C3
        move.b  #1,CRANE_DBF3
        bra     UPD_PLC
STRT_C3
        bclr.b  #C_FIN,CRANE3
        bset.b  #C_START,CRANE3
        bset.b  #C_BUSY,CRANE3
        clr.b   CRANE_DBF3
        bra     UPD_PLC

* test if crane 3 in use
CHK_BSY3
        clr.b   CRANE_DBF3
        btst.b  #C_BUSY,CRANE3
        beq     UPD_PLC

C_INUSE3
        btst.b  #C_FIN,CRANE3
        beq     UPD_PLC
        bclr.b  #C_BUSY,CRANE3
        bclr.b  #C_FIN,CRANE3
        bclr.b  #F_LD3,PLC_OUT0+1

* update outputs to PLC
UPD_PLC
        move.w  PLC_OUT0,P_OUT0
        move.w  PLC_OUT1,P_OUT1

EXITIC
        movem.w (sp)+,D0-d1/d2/d4

        rte

********************************************************************
*
*  Subroutine: CLOAD2
*
*  This routine takes crane 2 through the paces of loading.
*  The controller is open loop (timer driven) and doesn't actually
*  try to load anything.
*
********************************************************************

* This program controls crane 2
CLOAD2   movem.l d0-d1,-(sp)

* load crane register ghost into d0
        move.w  GCREG,d0
        clr.w   d0
        move.w  d0,TREG0

* clear bit 1 of port F to enable output of crane registers
        move.b  #0,PORTF

* setup and start timer
        clr.w   COUNT
        move.w  #CLEFT2,d1       load horizontal time
        move.w  #TIMRES,PITR

* start crane moving to left
        move.w  #$0200,d0
        move.w  d0,TREG0

LEFT    cmp.w   COUNT,d1        loop until count reached
        bgt     LEFT

* stop crane
        move.w  #$0000,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
```



Page 2

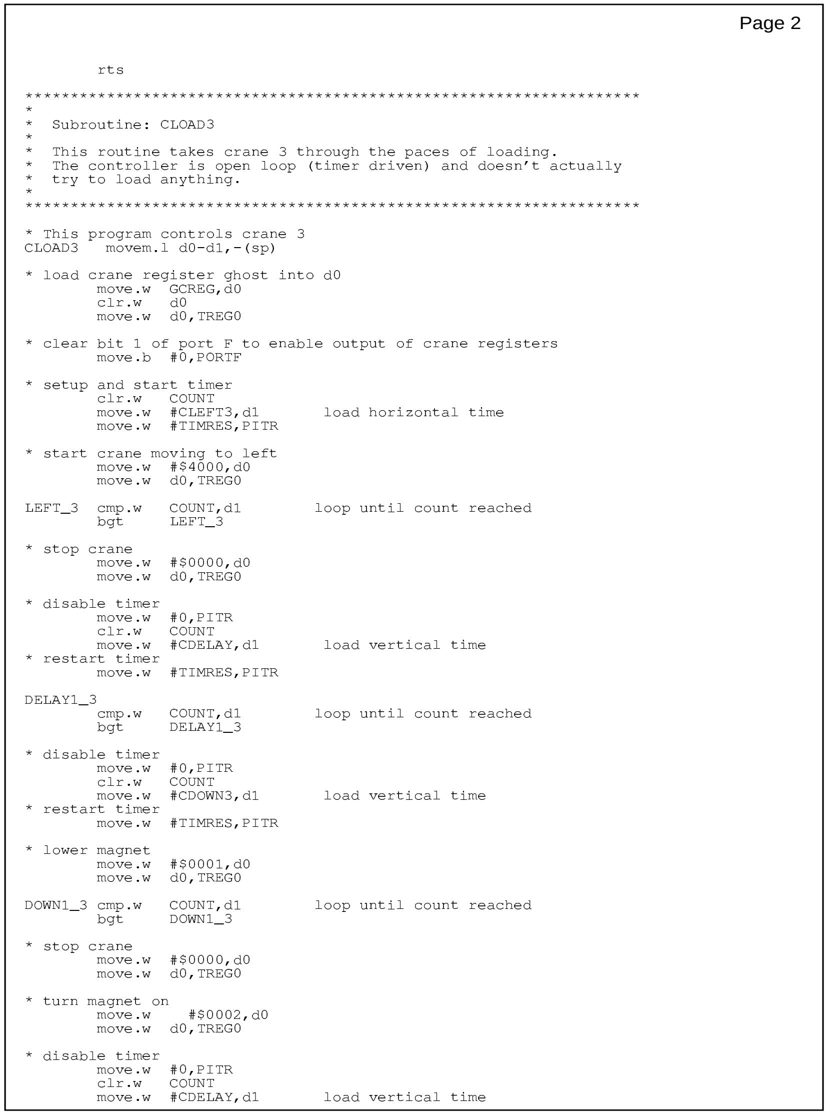
```
* restart timer
        move.w  #TIMRES,PITR

DELAY1  cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY1

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDOWN2,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* lower magnet
        move.w  #$0800,d0
        move.w  d0,TREG0

DOWN1   cmp.w   COUNT,d1        loop until count reached
        bgt     DOWN1

* stop crane
        move.w  #$0000,d0
        move.w  d0,TREG0

* turn magnet on
        move.w    #$1000,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY2  cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY2

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CUP2,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* raise magnet
        move.w  #$1400,d0
        move.w  d0,TREG0

UP1     cmp.w   COUNT,d1        loop until count reached
        bgt     UP1

* stop crane
        move.w   #$1000,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY3  cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY3

* stop timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CRIGHT2,d1       load horizontal time
* restart timer
        move.w  #TIMRES,PITR

* start crane moving to right
        move.w    #$1100,d0
        move.w  d0,TREG0
```

1



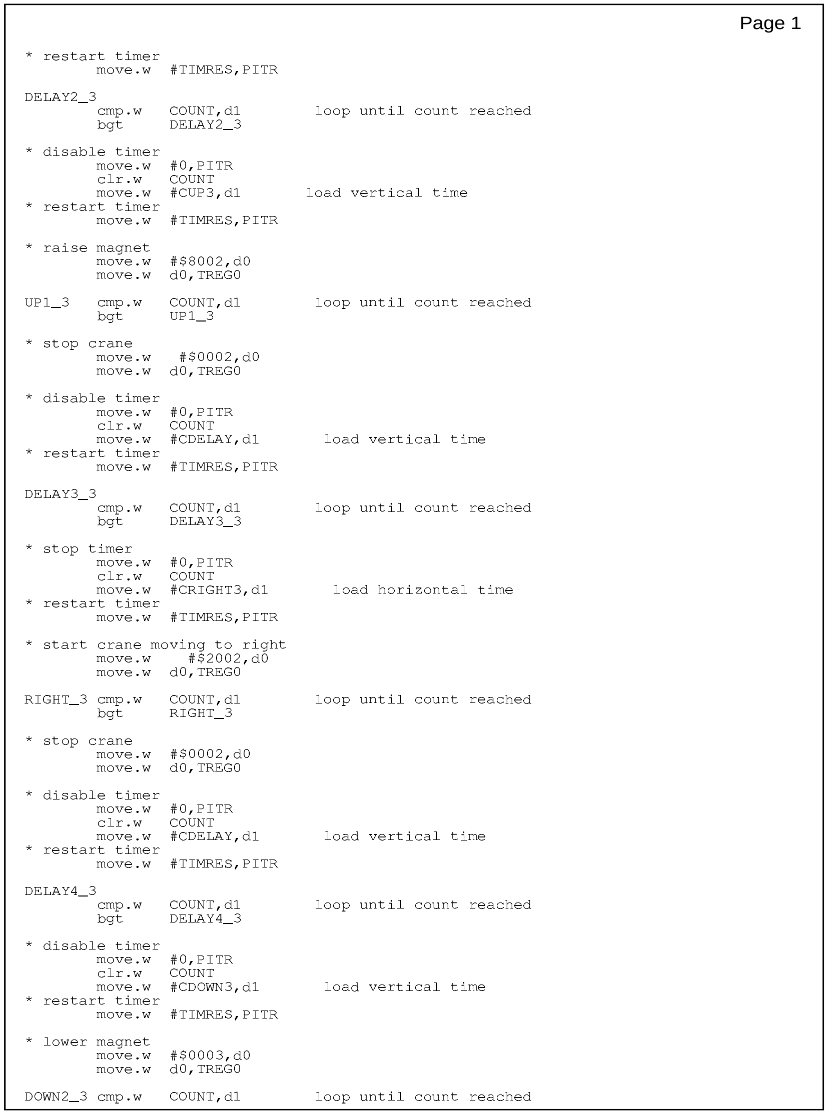
Page 1

```
RIGHT   cmp.w   COUNT,d1        loop until count reached
        bgt     RIGHT

* stop crane
        move.w  #$1000,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY4  cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY4

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDOWN2,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* lower magnet
        move.w  #$1800,d0
        move.w  d0,TREG0

DOWN2   cmp.w   COUNT,d1        loop until count reached
        bgt     DOWN2

* stop crane
        move.w  #$1000,d0
        move.w  d0,TREG0

* turn magnet off
        move.w   #$0000,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY5  cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY5

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CUP2,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* raise magnet
        move.w  #$0400,d0
        move.w  d0,TREG0

UP2     cmp.w   COUNT,d1        loop until count reached
        bgt     UP2

* stop crane
        move.w   #$0000,d0
        move.w  d0,TREG0

* stop timer
        move.w  #0,PITR

*restore ghost register to memory
        move.w  d0,GCREG

* set bit 1 of port F to disable output of crane registers
        move.b  #$2,PORTF

        movem.l (sp)+,d0-d1
```



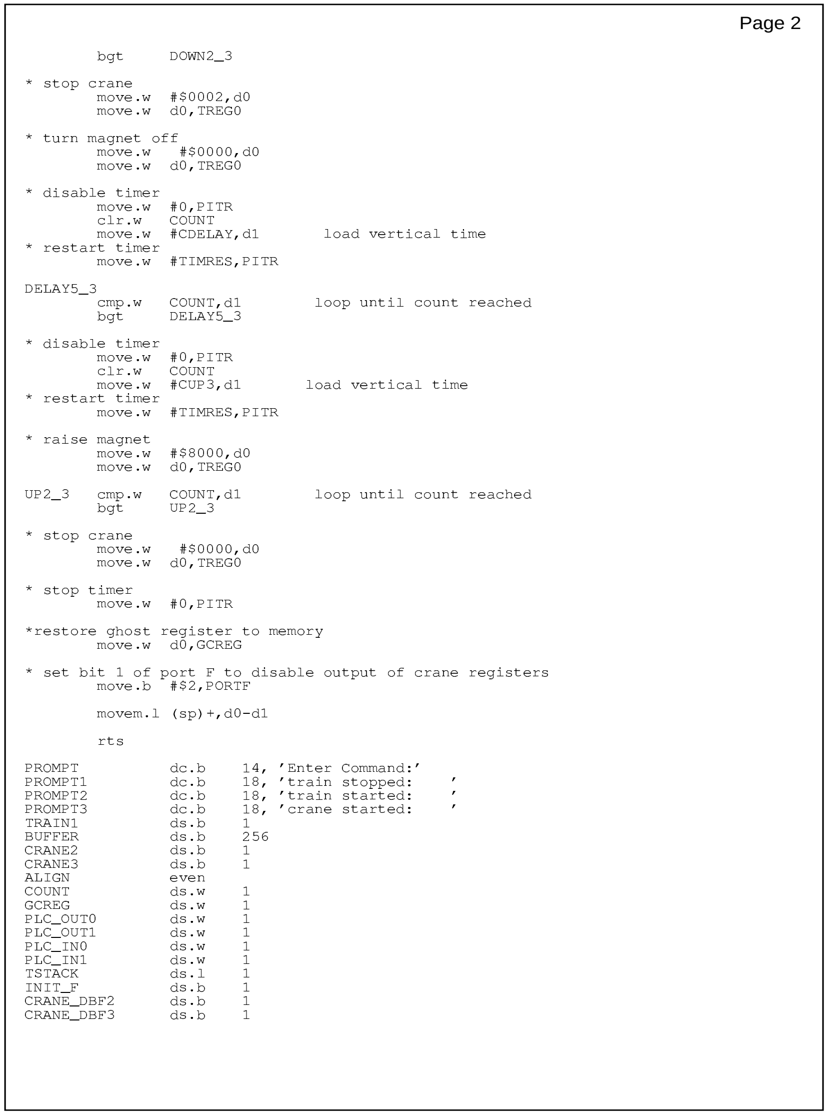
Page 2

```
        rts

*******************************************************************
*
*  Subroutine: CLOAD3
*
*  This routine takes crane 3 through the paces of loading.
*  The controller is open loop (timer driven) and doesn't actually
*  try to load anything.
*
*******************************************************************

* This program controls crane 3
CLOAD3   movem.l d0-d1,-(sp)

* load crane register ghost into d0
        move.w  GCREG,d0
        clr.w   d0
        move.w  d0,TREG0

* clear bit 1 of port F to enable output of crane registers
        move.b  #0,PORTF

* setup and start timer
        clr.w   COUNT
        move.w  #CLEFT3,d1       load horizontal time
        move.w  #TIMRES,PITR

* start crane moving to left
        move.w  #$4000,d0
        move.w  d0,TREG0

LEFT_3  cmp.w   COUNT,d1        loop until count reached
        bgt     LEFT_3

* stop crane
        move.w  #$0000,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY1_3
        cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY1_3

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDOWN3,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* lower magnet
        move.w  #$0001,d0
        move.w  d0,TREG0

DOWN1_3 cmp.w   COUNT,d1        loop until count reached
        bgt     DOWN1_3

* stop crane
        move.w  #$0000,d0
        move.w  d0,TREG0

* turn magnet on
        move.w    #$0002,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
```



Page 1

```
* restart timer
        move.w  #TIMRES,PITR

DELAY2_3
        cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY2_3

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CUP3,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* raise magnet
        move.w  #$8002,d0
        move.w  d0,TREG0

UP1_3   cmp.w   COUNT,d1        loop until count reached
        bgt     UP1_3

* stop crane
        move.w   #$0002,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY3_3
        cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY3_3

* stop timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CRIGHT3,d1       load horizontal time
* restart timer
        move.w  #TIMRES,PITR

* start crane moving to right
        move.w    #$2002,d0
        move.w  d0,TREG0

RIGHT_3 cmp.w   COUNT,d1        loop until count reached
        bgt     RIGHT_3

* stop crane
        move.w  #$0002,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY4_3
        cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY4_3

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDOWN3,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* lower magnet
        move.w  #$0003,d0
        move.w  d0,TREG0

DOWN2_3 cmp.w   COUNT,d1        loop until count reached
```



Page 2

```
        bgt     DOWN2_3

* stop crane
        move.w  #$0002,d0
        move.w  d0,TREG0

* turn magnet off
        move.w   #$0000,d0
        move.w  d0,TREG0

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CDELAY,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

DELAY5_3
        cmp.w   COUNT,d1        loop until count reached
        bgt     DELAY5_3

* disable timer
        move.w  #0,PITR
        clr.w   COUNT
        move.w  #CUP3,d1       load vertical time
* restart timer
        move.w  #TIMRES,PITR

* raise magnet
        move.w  #$8000,d0
        move.w  d0,TREG0

UP2_3   cmp.w   COUNT,d1        loop until count reached
        bgt     UP2_3

* stop crane
        move.w   #$0000,d0
        move.w  d0,TREG0

* stop timer
        move.w  #0,PITR

*restore ghost register to memory
        move.w  d0,GCREG

* set bit 1 of port F to disable output of crane registers
        move.b  #$2,PORTF

        movem.l (sp)+,d0-d1

        rts

PROMPT          dc.b    14, 'Enter Command:'
PROMPT1         dc.b    18, 'train stopped:    '
PROMPT2         dc.b    18, 'train started:    '
PROMPT3         dc.b    18, 'crane started:    '
TRAIN1          ds.b    1
BUFFER          ds.b    256
CRANE2          ds.b    1
CRANE3          ds.b    1
ALIGN           even
COUNT           ds.w    1
GCREG           ds.w    1
PLC_OUT0        ds.w    1
PLC_OUT1        ds.w    1
PLC_IN0         ds.w    1
PLC_IN1         ds.w    1
TSTACK          ds.l    1
INIT_F          ds.b    1
CRANE_DBF2      ds.b    1
CRANE_DBF3      ds.b    1
```

1