3-3-2008

Automatic Generation of PLC Code Based on Net Condition Event Systems

Natalia Sandberg
University of South Florida

Scholar Commons Citation

Sandberg, Natalia, "Automatic Generation of PLC Code Based on Net Condition Event Systems" (2008). *Graduate Theses and Dissertations.*
http://scholarcommons.usf.edu/etd/3771

Automatic Generation of PLC Code Based on Net Condition Event Systems

by

Natalia Sandberg

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Industrial Engineering
Department of Industrial and Management Systems Engineering
College of Engineering
University of South Florida

Major Professor: Ali Yalcin, Ph.D.
Marilyn Barger, Ph.D.
Richard Gilbert, Ph.D.
William Miller, Ph.D.

Date of Approval:
March 3, 2008

Keywords: Automated Manufacturing Systems, Supervisory Control Theory, Petri Nets, Discrete Events Systems, Ladder Logic Diagram

Table of Contents

List of Tables

List of Figures

Automatic Generation of PLC Code Based on Net Condition Event Systems

Natalia Sandberg

ABSTRACT

An important consideration in discrete event dynamic systems control theory is the selection of a suitable modeling formalism that can capture the complex characteristics of the system and the capability to automatically synthesize a controller based on the system model. Net condition event systems are well suited for modeling complex discrete event dynamic systems owing to their input and output structure, which effectively captures the behavior of the physical devices to be monitored and/or controlled. To date, net condition event systems control models have not been extensively applied to highly automated manufacturing systems and there are few guidelines on how to automatically generate Programmable Logic Controller programming languages from net condition event systems models. This research automatically converted net condition event systems control models into Programmable Logic Controller programming language and evaluated the applicability of the proposed methodology in highly automated manufacturing systems using HAS-200 as a test bed.

Chapter 1: Introduction

Discrete event dynamic systems (DEDS) are asynchronous and non-deterministic systems in which state changes take place by the occurrence of events rather than time. Such systems include manufacturing, robotics, and communications systems.

As DEDS become more complex, the control and coordination of the physical devices that compose them becomes more important. Therefore, in DEDS control theory the selection of a suitable modeling formalism that can capture the complex characteristics of the system is critical. Several modeling formalisms have been introduced to model and control discrete event dynamic systems. Net condition event systems (NCES) are well suited for modeling complex DEDS because they possess the following characteristics:

- Good graphical interface which facilitates ease of understanding of the system.
- Strong mathematical foundation for logical analysis.
- Ease of modification and maintenance compared with Ladder Logic Diagrams.
- An input/output structure that allows the modeling of the physical devices usually found in an automated manufacturing environment.

- Representation of the system behavior including concurrency, asynchronous behavior, mutual exclusion, etc.

Furthermore NCES can be automatically transformed into a Programmable Logic Controller (PLC) programming languages. In addition, the desirable logical properties and correctness of these types of PLC programming language can be verified. To date, net condition event systems control models have not been extensively applied to highly automated manufacturing systems. In addition, there are few guidelines on how to automatically generate Programmable Logic Controller programming languages based on NCES control models.

1.1 Research Goal and Objectives

The goal of this research is to automatically generate a PLC programming language for a complex manufacturing system control model. The objectives are as follows:

- Develop an algorithm to generate Ladder Logic Diagram from NCES models.
- Develop a NCES control model of the HAS-200 system [34] focusing on the container filling sequence.
- Convert the NCES control model to Ladder Logic Diagram and evaluate the applicability of the conversion methodology by verifying the correctness of the PLC programming language obtained from the algorithm.

1.2 Organization of the Thesis

The rest of this chapter introduces the fundamental concepts on petri nets (PN) and NCES. A tank filling and draining example is used to illustrate the NCES modeling process. The remainder of this thesis is organized into seven chapters. Chapter 2 reviews the literature on NCES and DEDS modeling formalism's used for the automatic generation of PLC programming language. In Chapter 3, the fundamentals of supervisory control theory are introduced along with a NCES control model of the tank filling and draining example of Chapter 1. Chapter 4 presents the analysis tools used to verify the correctness of the tank filling and draining NCES control model from Chapter 3. Chapter 5 introduces a preliminary algorithm to convert NCES model into a PLC programming languages. Furthermore, the tank filling and draining NCES control model from Chapter 1 is transformed into a PLC programming language using the algorithm. Chapter 6 provides a brief introduction to the HAS-200 system along with a NCES control model for the HAS-200. The NCES control model is analyzed for correctness and transformed into a PLC programming language using the algorithm developed in Chapter 5. Chapter 7 discusses the final conclusions and contributions of this thesis, as well as future areas of research.

1.3 Petri Nets

A PN [3] is identified as a particular kind of bipartite graph populated by three types of objects. These objects are places (circles), transitions (boxes), and directed arcs (arrows) connecting places to transitions and transitions to places. A place is an input place of a transition if there is a directed arc connecting this place to a transition. A place is an output place of a transition if there is a directed arc connecting the transition to the place. Figure 1.1 shows an example of a PN with two places, two transitions and four directed arcs. In this PN p1 is an input place and p2 is an output place for transition t1.

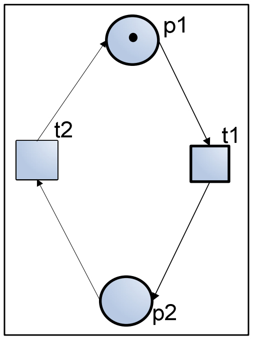


Figure 1.1: Petri Net

For Figure 1.1, the input and output places are defined in a matrix form as follows:

$$I = \begin{array}{c|cc|} & t1 & t2 \\ p1 & 1 & 0 \\ p2 & 0 & 1 \end{array} \qquad O = \begin{array}{c|cc|} & t1 & t2 \\ p1 & 0 & 1 \\ p2 & 1 & 0 \end{array}$$

Figure 1.2: Petri Net I/O Matrix

Notice that the I/O matrix show a one to represent the existence of an arc connecting a place (transition) to a transition (place) and zero otherwise.

The places and transitions are used to represent various aspects of the modeled system. For instance, an input place may represent the availability of a resource, the transitions the resource change from available to occupied, and the output place the resource utilization. Another example is that the places and transition represent the status of a device in a manufacturing process, such as a conveyor belt. If Figure 1.1 represents a conveyor belt status, then p1 means that the conveyor belt is off, t1 is the transition of the conveyor belt from off to on, p2 means that the conveyor belt is on, and t2 is the transition of the conveyor belt from on to off.

A PN is defined by a four tuple $\mathcal{PN} = (\mathcal{P}, \mathcal{T}, I, O)$ where:

- $\mathcal{P}$ is a set of n places, where $p \in \mathcal{P}$
- $\mathcal{T}$ is a set of m transitions, where $t \in \mathcal{T}$
- $I: \mathcal{P} \times \mathcal{T} \rightarrow \mathcal{N}$ is an input function that defines directed arcs from places to transitions, where $\mathcal{N}$ is a set of nonnegative integers.
- $O: \mathcal{P} \times \mathcal{T} \rightarrow \mathcal{N}$ is an output function that defines directed arcs from transitions to places, where $\mathcal{N}$ is a set of nonnegative integers.

1.3.1 Petri Nets Marking

A PN marking is the number of tokens in each of the net places at any given time. Graphically, a token is represented by a small black dot as the one shown in p1 in Figure 1.1. The distribution of tokens in places defines the current state of the modeled system. Each place may potentially hold either no tokens or a positive number of tokens. The presence or absence of a token in a place can indicate whether a condition associated with this place is true or false.

A marking of a PN with n places is represented by an **(n x 1)** vector μ. The elements of this vector are denoted as $\mu(p)$ and are nonnegative integers representing the number of tokens in the corresponding places. In a PN, μ_0 represents the initial marking. For example, in the PN model shown in Figure 1.1, μ_0 = (1, 0).

If $I(p_j, t_i) = \mathcal{K}$ $(O(p_j, t_i) = \mathcal{K})$, then there exist $\mathcal{K}$ directed arcs connecting place p_j to transition t_i (transition t_i to place p_j). If $I(p_j, t_i) = 0$ $(O(p_j, t_i) = 0)$, then there exists no directed arcs connecting place p_j, to transition t_i (transition t_i to place p_j,). The direct arcs that connect places (transition) to transitions (places) are labeled with weight $\mathcal{K}$ as shown in Figure 1.3. The arc weight controls the number of tokens that can travel along the arc. However, if the arc weight is one, then the weight label is omitted. A PN is called ordinary [4] if all of its arc weights are one. The PN shown in Figure 1.1 is an ordinary PN.

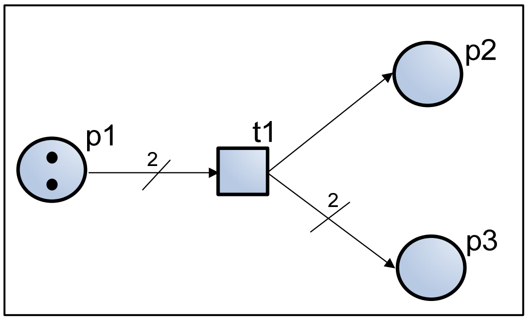


Figure 1.3: Arc Weight

1.3.2 Petri Nets Enabling and Firing Rule

Tokens reside in places, travel along arcs and their movement is regulated by transitions. The transition enabling rule states that a transition t_i is said to be enabled if each input place p_j of t_i contains at least the number of tokens equal to the weight of the directed arc. For example in Figure 1.1, for the token to move from p1 to p2, t1 must be enabled. Transition t1 is enabled because the input place p1 contains one token and the directed arc weight is one.

If enabled in a marking, transition t_i may or may not fire depending on additional interpretation. When an enabled transition t_i fires, the number of tokens equal to the weight of the directed arc connecting p_j to t_i are removed from input places p_j and then deposited in output places p_n. The number of tokens deposited in the output places p_n should equal the weight of the directed arc connecting t_i to p_n. Therefore, the firing of transition t_i will generate a new marking μ'. The new marking is given by:

$$\mu'(\mathrm{p}) = \left\{ \begin{array}{lll} \mu(\mathrm{p}) - I(\mathrm{p_j}, \mathrm{t_i}) & \text{if} & \mathrm{p} \in \mathrm{P} : I(\mathrm{p_j}, \mathrm{t_i}) > 0 \\ \mu(\mathrm{p}) + \mathrm{O}(\mathrm{p_j}, \mathrm{t_i}) & \text{if} & \mathrm{p} \in \mathrm{P} : \mathrm{O}(\mathrm{p_j}, \mathrm{t_i}) > 0 \\ \mu(\mathrm{p}) & \text{otherwise} & \end{array} \right\}$$

Figure 1.4: New Marking Equations

For example in Figure 1.3, the firing of transition t1 removes two tokens from input place p1 and deposits two tokens in p3 and one in p2. The marking for the PN shown in Figure 1.3 is $\mu = (2, 0, 0)$. After t1 fires, the new marking is $\mu' = (0, 1, 2)$ as shown in Figure 1.5.

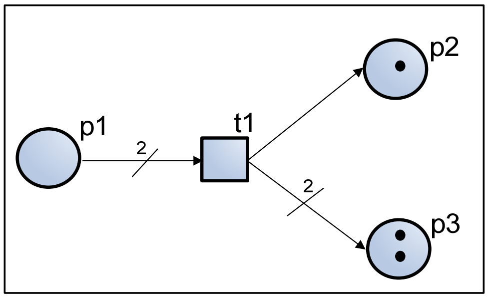


Figure 1.5: Transition Fire

1.3.3 Petri Net Properties

The importance of modeling a system using PN is the analysis of its properties. PN properties allow one to study the dynamics of the modeled system, in terms of its states and state changes. There are two types of properties that can be identified in a PN model; the properties that depend on the initial marking and are called behavioral properties and the properties that do not depend on the initial marking and are called structural properties. For the purpose of this thesis, only six behavioral properties will be considered namely reachability, reversibility, boundedness, safeness, conservativeness, and liveness [4].

1.3.3.1 Petri Net Reachability

Reachability is used to determine if the modeled system can reach a specific state. From the previous section, one knows that the firing of a transition will change the marking of a PN. Therefore, in order to determine if a system will reach a specific state, it is necessary to find the sequence of transition firings that will lead to the desired marking. A marking μ_i is said to be reachable from marking μ_o, if there exist a sequence of transitions firings that transform μ_o to μ_i. A firing sequence is denoted by $\sigma = t_1, t_2, \ldots t_i$. The set of all possible firing sequences from μ_o is denoted by $\mathcal{L}(\mu_o)$. The set of all possible markings reachable from μ_o is called the reachability set and is denoted by $\mathcal{R}(\mu_o)$. For example, in the case of the PN shown in Figure 1.6, the firing sequence $\sigma = (t1, t2)$

will transform μ_0 = (1, 0, 0) into μ_2 = (0, 0, 1), hence μ_2 is said to be reachable from μ_0.

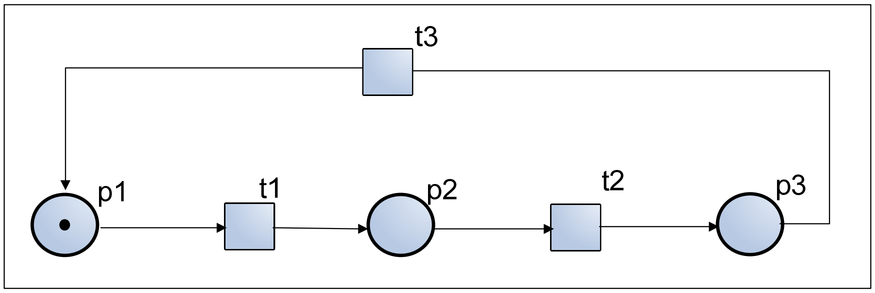


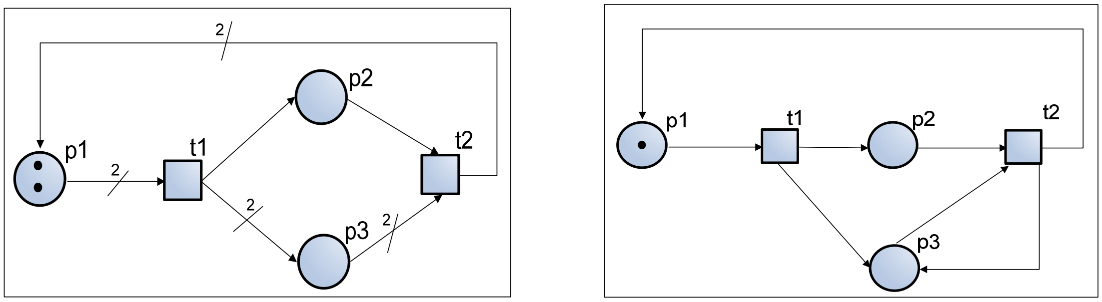
Figure 1.6: Reachable Marking

1.3.3.2 Petri Net Reversibility

In some manufacturing applications it is necessary that a system returns to its initial state, such cases could be a machine failure or an error. A PN is said to be reversible if for each marking μ_i in $\mathcal{R}(\mu_0)$, μ_0 is reachable from μ_i. An example of a reversible PN is shown in Figure 1.7a, where the initial marking μ_0 = (2, 0, 0) is reachable from all the markings (μ_1 = (0, 1, 0); μ_2 = (0, 0, 1)).

1.3.3.3 Petri Net Boundedness

A PN is said to be $\mathcal{K}$-bounded if the number of tokens in any place p_j, is always less or equal to $\mathcal{K}$, where $\mathcal{K}$ is a nonnegative integer number. For example, the PN shown in Figure 1.7(a) is a 2-bounded PN, which means that in any reachable marking, p1, p2 and p3 holds two tokens or less. On the other hand, the PN shown in Figure 1.7(b) is unbounded, because p3 can hold an arbitrarily large number of tokens. Verifying that a PN is bounded will guarantee that the modeled system will have no overflows regardless of what firing sequence is executed.



(a) (b)

Figure 1.7: (a) 2-Bounded Petri Net (b) Unbounded Petri Net

1.3.3.4 Petri Net Safeness

A PN is safe if it is 1-bounded, which means that in any reachable marking the number of tokens in each place is one or zero. Notice the difference between safe and ordinary PNs. A safe PN is ordinary, but an Ordinary PN is not always safe. For example, the unbounded PN shown in Figure 1.7b is ordinary; all of its arc weights are equal to one. However, the PN in Figure 1.7b is not safe. On the other hand, the PN shown in Figure 1.8 is ordinary and safe. In this net, no place can contain more than one token at any reachable marking and all weights are one.

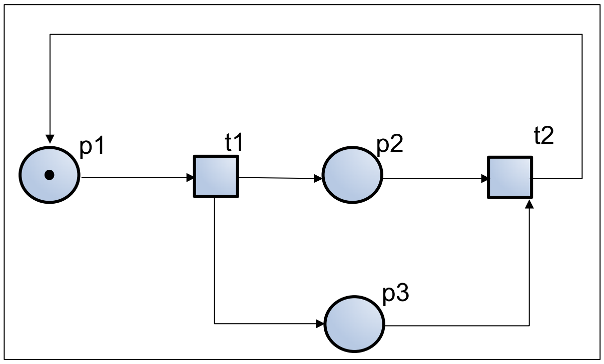


Figure 1.8: Safe Petri Net

1.3.3.5 Petri Net Conservativeness

A PN is said to be conservative if the number of tokens remains the same for all markings reachable $\mathcal{R}(\mu_0)$ from the initial marking μ_i. However, conservativeness can also depend on a weighted vector for cases in which resources need to be combined together for a task and later separated after the task is completed. A PN is said to be conservative if there exist a vector $w = \{w_1, w_2,\ldots,w_n\}$, where n is the number of places and $w(p) > 0$ for each $p \epsilon \mathcal{P}$, such that the weighted sum of tokens remains the same for each marking μ_i reachable from the initial marking μ_0. The PN shown in Figure 1.19 is conservative with respect to vector w = {2,1,1,1,1,2}.

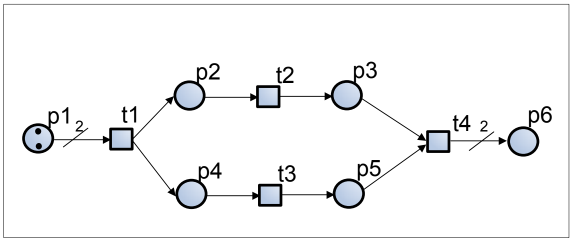


Figure 1.9: Petri Net Conservativeness

1.3.3.6 Petri Net Liveness

A PN model is said to be live if all markings μ_i reachable from the initial marking μ_0, are able to fire any transition by progressing through some firing sequence. The existence of liveness in a PN model guarantees a deadlock free system no matter what firing sequence is selected. There are four different levels of liveness for a transition t_i:

- $\mathcal{L}0$-*live* (dead): if there is no firing sequence in $\mathcal{L}(\mu_0)$ for which t_i can fire.
- $\mathcal{L}1$-*live*: if t_i can be fired at least once in some firing sequence in $\mathcal{L}(\mu_0)$.

- $\mathcal{L}2$-*live*: if t_i can be fired at least k times in some firing sequence in $\mathcal{L}(\mu_0)$, given that k is a positive integer.
- $\mathcal{L}3$-*live*: if t_i can be fired infinitely in some firing sequence in $\mathcal{L}(\mu_0)$.
- $\mathcal{L}4$-*live*: if t_i is $\mathcal{L}1$-*live* in every marking in $\mathcal{R}(\mu_0)$.

The transitions in the PN model shown in Figure 1.10 have different levels of liveness. Transition t1, t2, t3, and t4 are $\mathcal{L}3$, $\mathcal{L}1$, $\mathcal{L}2$, and $\mathcal{L}2$ respectively.

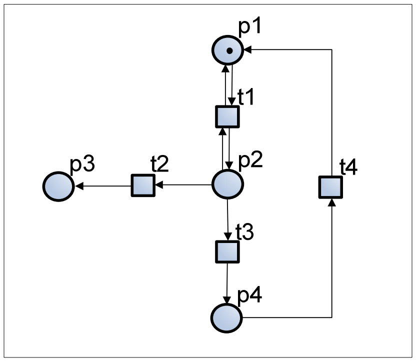


Figure 1.10: Petri Net Liveness

1.3.4 Petri Net Analysis Methods

In the previous section, several properties of PNs were introduced. The identification of those properties in a PN model is necessary, because it will establish a relationship with the functional properties of the real system. Nevertheless, the use of analysis methods such as reachability tree and the incidence matrix can study the presence or absence of PN properties. An overview of the two fundamental methods of analysis will be presented in this section.

1.3.4.1 Reachability Tree or Graph

The reachability tree or graph (RG) illustrates all the possible markings of a PN in a tree representation. The RG starts from the initial marking and obtains all the possible new markings from all the enabled transitions. Then, from each of the new marking it obtains the next reachable marking. The markings are represented by nodes and the transitions firings by arcs.

A reachability tree can become unbounded for two reasons:

- The existence of duplicate markings
- Unbounded PNs

To eliminate duplicate markings one must determine if the current marking μ' is identical to a previous marking μ. If true, μ' is a duplicate marking and becomes a terminal node. A duplicate marking indicates that all possible markings reachable from μ' have already been added to the tree. For unbounded PNs the tree will grow infinitely large. The symbol w is introduced to keep the tree finite. It has the property that for each integer n, $w > n$, $w \pm n = w$ and $w \geq w$. To construct the RG of a PN the following algorithm can be used:

Step 1: Label the initial marking μ_o as the root and tag it "new".

Step 2: While "new" markings exists, do the following:

Step 2.1: Select a new marking $\mu_{i.}$

Step 2.2: If $\mu_{i.}$i is identical to a marking on the path from the root to $\mu_{i.}$, then tag $\mu_{i.}$ "old" and go to another new marking.

Step 2.3: If no transitions are enabled at $\mu_{i.}$, tag $\mu_{i.}$ "dead end."

Step 2.4: While there exists enabled transitions at $\mu_{i.}$, do the following for each enabled transitions t at $\mu_{i.}$:

Step 2.4.1: Obtain the marking μ' that results from firing t at $\mu_{i.}$.

Step 2.4.2: On the path from the root to μ_i if there exists a marking μ'' such that $\mu'(p) \geq \mu''(p)$ for each place p_j and $\mu' \neq \mu''$, then replace $\mu'(p)$ by w for each p_j such that $\mu'(p) > \mu''(p)$.

Step 2.4.3: Introduce μ' as a node. Draw an arc with label t from μ_i to μ', and tag μ' "new."

For example the reachability tree for Figure 1.11 is shown in Figure 1.12.

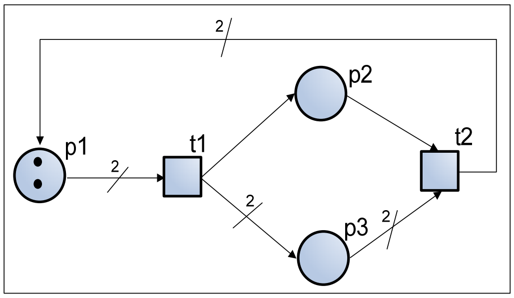


Figure 1.11: Petri Net Reachability

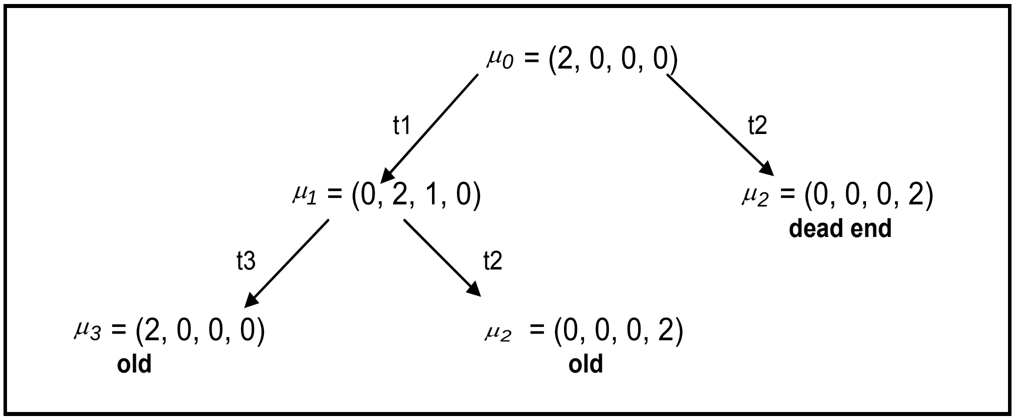


Figure 1.12: Reachability Tree

1.3.4.2 Petri Nets Incidence Matrix and State Equation

The incidence matrix is a method to represent and analyze the dynamic behavior of PN by using algebraic equations. The incidence matrix defines all the possible connections between the places and transitions of a PN. The incidence matrix $\mathcal{A}$ = $[a_{ij}]$ is an n **x** m matrix, where n is the number of transitions and m is the number of places. The entries are defined as follows:

$$a_{ij} = a_{ij}^{+} - a_{ij}^{-}$$

Where a_{ij}^{+} is the weight of the arc from transition i to its output place j and a_{ij}^{-} is the weight of the arc from its input place j to transition i. In other words, when transition t_i fires, a_{ij}^{+} represents the number of tokens deposited on its output place p_j, a_{ij}^{-} represents the number of tokens removed from its output place p_n. In order to make sure the incidence matrix properly reflects the structure of a PN, the net must be pure. A PN is said to be pure if it has no self loops. A self loop means that no transition is both an input and an output of the same place. For example, the incidence matrix of Figure 1.13 is shown in Figure 1.14.

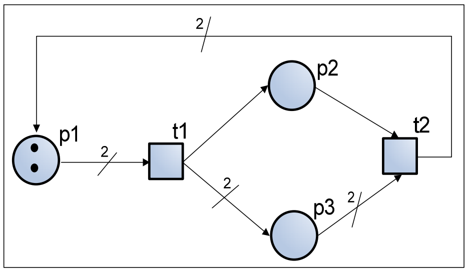


Figure 1.13: Petri Net Incidence Matrix

$$\mathcal{A} \quad = \quad \begin{array}{c|ccc|} & p1 & p2 & p3 \\ t1 & -2 & 1 & 2 \\ t2 & 2 & -1 & -2 \end{array}$$

Figure 1.14: Incidence Matrix

The state equation of a PN represents a change of the distribution of tokens as result of a transitions firing (marking, section 1.3.1). This equation is defined as follows:

$$\mu_{k} = \mu_{k-1} + \mathcal{A}^{T} M_{K} \text{ where k= 1, 2, ….}$$

μ_k is a m x 1 column vector representing a marking μ_k immediately reachable from marking μ_{k-1} after firing a transition t_i. The k-th firing vector M_K, an n x 1 column vector, has only one nonzero entry. This nonzero entry is a 1 in the i-th position that indicates the firing of transition t_i in the k-th firing. This 1 entry corresponds to the i-th row of the incidence matrix and indicates the change of the marking. For example, Figure 1.15 illustrates the use of the state equation to obtain the new marking (μ'= (0, 1, 2)) after transition t1 in Figure 1.13 fires. Notice that the state equation uses the transpose of the incidence matrix instead of the incidence matrix.

$$\begin{vmatrix} 0 \\ 1 \\ 2 \end{vmatrix} = \begin{vmatrix} 2 \\ 0 \\ 0 \end{vmatrix} + \begin{vmatrix} -2 & 2 \\ 1 & -1 \\ 2 & -2 \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix}$$

Figure 1.15: Petri Net State Equation

1.4 An Introduction to Net Condition Event System

Net condition event systems (NCES) were developed by Hanisch and Rausch [1] based on the work in Condition Event System by Sreenivas and Krogh [2]. NCES are based on an ordinary safe PN extended with an input/output structure. This input/output structure provides modularity to the uncontrolled system model, because the design includes a set of predefined modules for physical devices in an automated manufacturing environment such as actuators, pumps, valves, sensors, and stoppers.

1.4.1 Condition and Event Signals

Condition Event Systems provide a modular modeling formalism for discrete event dynamic systems. The modules of each of the devices are interconnected by means of their input/output behavior to form the uncontrolled system model. The input/output behavior consists of two signals: condition signals and event signals [5, 8].

Condition signals provide state (place) information to a transition. Condition signals are a piecewise constant signal, because they keep transmitting information whether the condition is true or false. A condition signal is true, if there is a token in the place related to that condition. A condition signal is false, when there is not a token in the place related to that condition. For example, a manufacturing process with a valve that opens or closes depending on the level of the liquid in the tank. A condition signal can be used to provide information of the state of the level sensor. If p2 in Figure 1.16 represents that the level sensor is active; then the condition signal from place p2 to transition t3 will sent a true signal when the level sensor alarm is active and a false signal when the level sensor is passive.

Condition arcs are the arcs that carry a condition signal. Condition arcs connect a place p_i in one module to a transition t_i in another module. Condition arcs are graphically represented by an arc with a black dot at its end instead of an arrow head. Condition arcs can be classified as condition inputs and condition outputs. Condition inputs and outputs are graphically represented by a small box at the border of the module as illustrated in Figure 1.16. Condition outputs are associated with places and will have an incoming condition arc to the box; meanwhile condition inputs are associated with transitions and will have an outgoing condition arc from the box as depicted in Figure 1.16.

Event signals provide information on state transition and are null except at a discrete points in time. In other words, an event signal is only true in the instance that the transition is fired. The rest of the time an event signal value is null. Following the tank example previously explained, one can conclude that t2 in Figure 1.16 signifies that the transition of the level sensor from passive to active. Therefore, the event signal from transition t2 to transition t3 is only true when t2 is fired.

Event arcs are the arcs that carry an event signal and connect a transition t_j in one module to a transition t_m in another module. An event arc is graphically represented by an arc with a zigzag symbol in the middle. Event arcs can be classified as event inputs and event outputs. Event outputs and inputs are graphically represented by a small diamond at the border of the module. Event outputs have an incoming event arc towards the diamond as shown in Figure 1.16. The event inputs have an outgoing event arc from the diamond as illustrated in Figure 1.16.

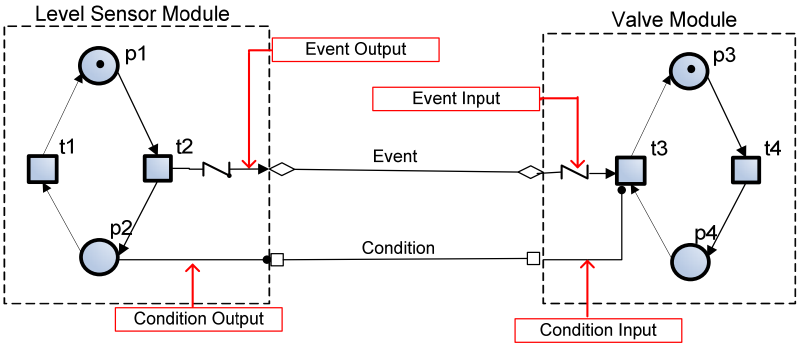


Figure 1.16: Condition and Event Signals

1.4.2 Net Condition Event Systems

Condition and event signals are useful because they are able to characterize the interaction between the physical components of the system. The model obtained by interconnecting the modules of the physical components by means of condition and event signals is known as net condition event system (NCES) [7]. The NCES consist of a four tuple structure as follows:

NCES = $\{\mathcal{PN}, \Psi, \mathcal{CN}, \mathcal{EN}\}$ where:

- $\mathcal{PN}$ is a PN
- Ψ is the input/output structure
- $\mathcal{C}_{\mathcal{N}}$ is the condition signal matrix
- $\mathcal{E}_{\mathcal{N}}$ is the event signal matrix

The input/output structure is defined as follows:

$\Psi = \{C_{in}, \mathcal{E}_{in}, C_{out}, \mathcal{E}_{out}, \mathcal{B}_c, \mathcal{B}_e, C_s, \mathcal{D}_t\}$ where:

- C_{in} is a set of r condition inputs
- $\mathcal{E}_{in}$ is a set of s events inputs
- C_{out} is a set of p condition outputs
- $\mathcal{E}_{out}$ is a set of q events outputs
- $\mathcal{B}_c \in \{0,1\}^{r \times m}$ is the condition input matrix
- $\mathcal{B}_e \in \{0,1\}^{s \times m}$ is the event input matrix
- $C_s \in \{0,1\}^{n \times p}$ is the condition output matrix
- $\mathcal{D}_t \in \{0,1\}^{m \times q}$ is the event output matrix

1.4.3 Net Condition Event System Enabling and Firing Rule

In NCESs, unlike PNs, there are three enabling rules to consider before a transition t is enabled.

- Marking enabled: A transition $t_j \in \mathcal{T}$ is marking enabled, if $\min(\mu - \mathcal{F}m(\cdot, j)) \geq 0$. Transition t_j is said to be marking enabled if each input place p_i of t_j contains at least the number of tokens equal to the weight of the directed arc.

The marking enabled and firing rule for NCES follows the same principles as the marking enabled and firing rule for PNs (Section 1.3.1 and 1.3.2).

- Condition enabled: A transition $t_j \in \mathcal{T}$ is condition enabled, if $\min(\mu - C_{\mathcal{N}}(\cdot, j)) \geq 0$. Transition t_j is said to be condition enabled when each of its condition inputs places (if any) are marked with a token. However the firing of transition t_j will not change the marking of the condition input place.

Transitions with an incoming condition arc and no event input are known as spontaneous transitions. Spontaneous transition can only be enabled or disabled by condition signals, but they cannot be forced to fire. For example in Figure 1.17, transition t3 in module 2 is a spontaneous transition. Therefore, transition t3 can only fire if it is marking and condition enabled. Transition t3 is not marking enabled because there is no token in p4. However, transition t3 is condition enabled because there is a token in p2, which makes the condition signal true. Notice that the condition signal from p2 is not forcing t3 to fire. Also, notice that the condition signal will remain true (constant signal) as long as the token remains in p2. Transition t1 can fire because it is marking enabled and the condition signal in p2 doesn't affect the firing of t1. If the initial marking of Figure 1.12 is μ_o = (0, 1, 0, 1), then the reachable marking after firing t1 will be μ' = (1, 0, 0, 1). Notice, that p3 and p4 are not affected by the firing of t1. Now, if t3 becomes marking enabled (token in p4) and condition enabled (token in p2), then the marking is μ = (0, 1, 1, 0). If t3 fires, a token is remove from p3 and deposit in p4. The token in p2 remains there. So, the new marking is μ' = (0, 1, 0, 1). Take into consideration that condition arcs can only carry condition signals and no tokens.

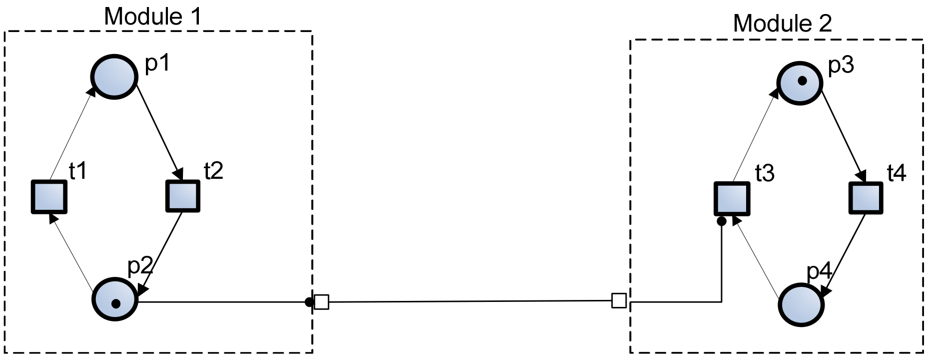


Figure 1.17: Spontaneous Transitions

- Event enabled: The set T_e contains all transitions which are connected with t_j by an incoming event arc at t_j. $\mathcal{T}_e(t_j) = \{t_m / \mathcal{E}_\mathcal{N}(m, j) > 0\}$. Transition t_j is said to be event enabled if there are no event inputs, or if all the transitions $t_m \epsilon \mathcal{T}_e(t_j)$ are marking, condition and event enabled.

An event signal can force a transition to fire if enabled. A transition with an incoming event arc is known as a forced transition. Furthermore, all forced transitions occur at the same time instant as the event signal which forces the transition to fire. Hence, incoming event signals force transitions to fire if they are marking and condition enabled and the transitions must fire immediately. For example in Figure 1.18, transition t3 in module 2 is a forced transition, which means that firing transition t2 will simultaneously fire transition t3 if enabled. Transition t3 and t2 are marking enabled because there is a token in p4 and p1. Transition t3 does not have an incoming condition signal, but t2 does have one. Transition t2 is condition enabled since there is a token in p5, which makes the condition signal true. The initial marking of Figure 1.18 is μ_o = (1, 0, 0, 1, 1, 0). If t2 fires, a token is removed from p1 and deposited in p2. Since t3 fires simultaneously, a token will be removed from p4 and deposited in p3. The new marking will be μ' = (0, 1, 1, 0, 1, 0). Take into consideration that events arcs can only carry event signals and not tokens. Notice, that p5 and p6 are not affected by the firing of t2 and t3. Furthermore, observe that both transition t2 and t3 fire simultaneously creating a new marking in only one step. If p5 did not have a token, then t2 will no longer be condition enabled and will not be able to fire.

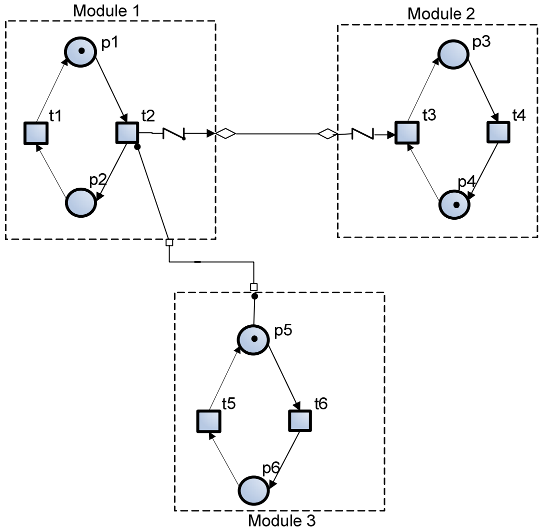


Figure 1.18: Forced Transitions

Another example is shown in Figure 1.19. Transition t3 must be marking, condition and event enabled to fire. The initial marking for the NCES in Figure 1.19 is μ_0 = (1, 0, 0, 1, 1, 0). In this case, transition t2 can fire but t3 cannot. Transition t2 and t3 are marking enabled, but t3 is not condition enabled (no token in p6). If transition t2 fires the new marking will be μ' = (0, 1, 0, 1, 1, 0) and the marking of the places in module 2 will remain unchanged until a token comes back to p1 again.

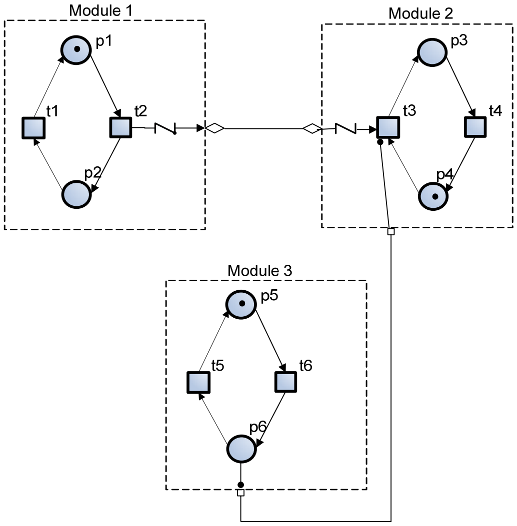


Figure 1.19: Event Signal Firing

1.4.4 Net Condition Event System Example

To illustrate the basic concepts of a NCES, let us consider the process depicted in Figure 1.20. The tank is filled with a mixture via a pump until the level sense high (LSH) sensor goes into alarm. After which, the draining process starts by opening the valve at the bottom of the tank. The mixture is then sent to the next step of the process. The valve will remain open until the level sense low (LSL) sensor goes into alarm. Subsequently, the valve is closed and the refilling process starts again.

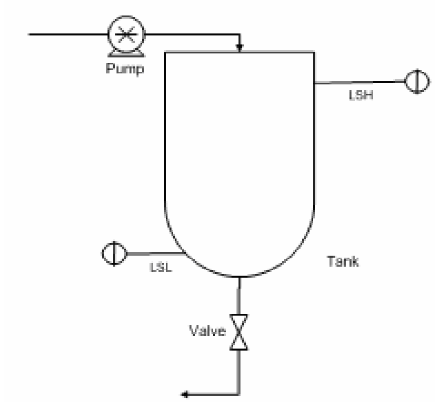


Figure 1.20: Tank Filling and Draining Process

In order to develop a NCES model, it is necessary to separately model each of the devices that are part of the process. In this case, there are 4 devices: a pump, a valve, a LSL, and a LSH. Each of the 4 models must capture the dynamic behavior of the devices. For example, Figure 1.21 shows the NCES model of the valve module. A description of the places and transitions are shown in Table 1.1. Notice that each transition of the module includes a condition signal that is part of the dynamic behavior of the valve. The modeling of the other devices is similar to that of the valve.

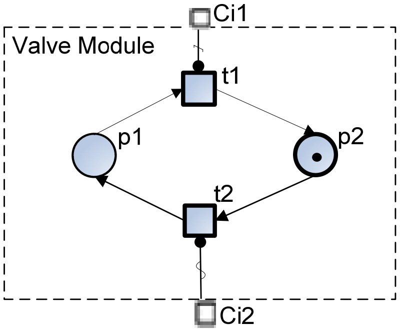


Figure 1.21: NCES Model of the Valve Module

Table 1.1: Places and Transition for the Valve Module

Valve Module			
Transition	Meaning	Place	Meaning
t1	Valve opening	p1	Valve close
t2	Valve closing	p2	Valve open

After the device models are created, they are interconnected by means of their signals to capture the uncontrolled behavior of the system. Let’s first examine the interconnection of three modules: the valve module, the LSH module, and the LSL module as shown in Figure 1.22. Table 1.2 gives a brief description of the places, transitions, and conditions for the interconnected modules. The problem description dictates that the status of the valve is dependent on the status of the LSL and LSH. The valve should remain open (closed) until the tank is completely drained (filled). The signal selected to represent this process is a condition signal. Notice that t4 (valve opening) will not fire until condition C_1^{in} (LSH alarm active) is true. For identification purposes the lines representing the condition signals are drawn differently.

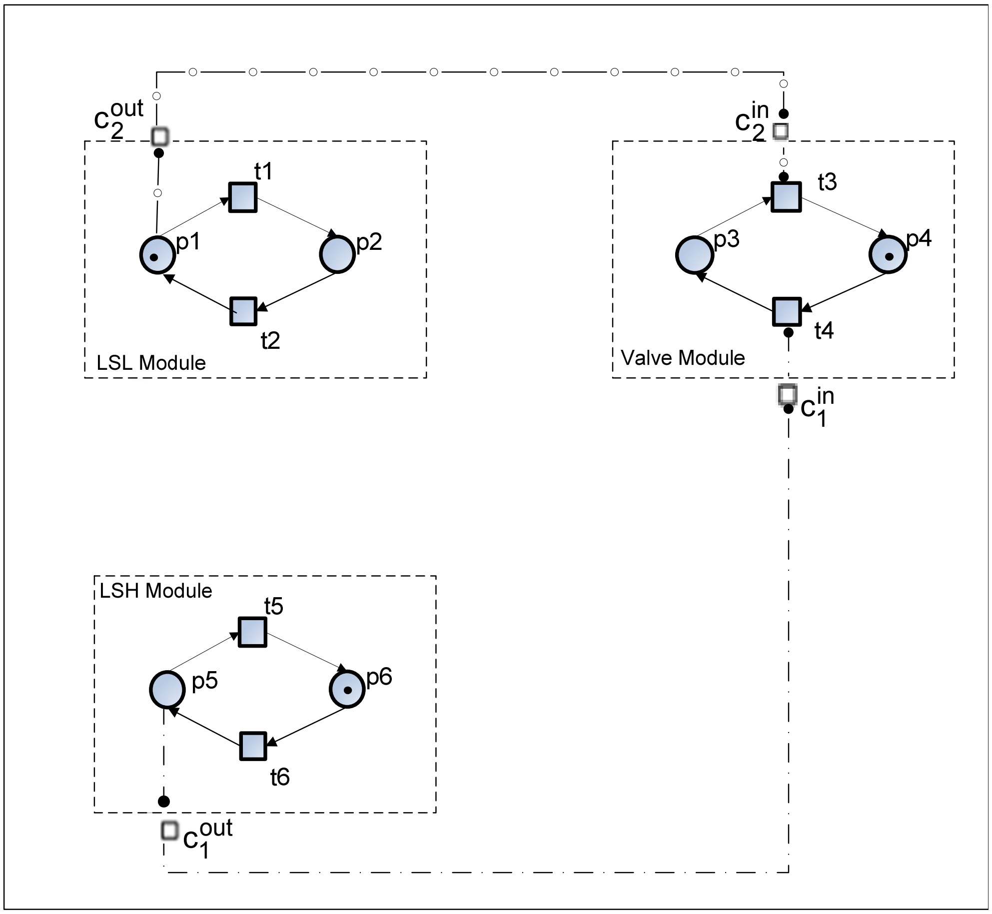


Figure 1.22: Interconnection of Modules

Table 1.2: Places, Transitions, and Conditions for Interconnected Modules

<table>
<tr><th colspan="4">LSL Module</th></tr>
<tr><th>Transition</th><th>Meaning</th><th>Place</th><th>Meaning</th></tr>
<tr><td>t1</td><td>LSL alarm goes passive</td><td>p1</td><td>LSL alarm active</td></tr>
<tr><td>t2</td><td>LSL alarm goes active</td><td>p2</td><td>LSL alarm passive</td></tr>
<tr><th colspan="4">Valve Module</th></tr>
<tr><td>t3</td><td>Valve closing</td><td>p3</td><td>Valve opened</td></tr>
<tr><td>t4</td><td>Valve opening</td><td>p4</td><td>Valve closed</td></tr>
<tr><th colspan="4">LSH Module</th></tr>
<tr><td>t5</td><td>LSH alarm goes passive</td><td>p5</td><td>LSH alarm active</td></tr>
<tr><td>t6</td><td>LSH alarm goes active</td><td>p6</td><td>LSH alarm passive</td></tr>
<tr><th colspan="4">Module Conditions</th></tr>
<tr><th colspan="3">Condition</th><th>Meaning</th></tr>
<tr><td colspan="3">$C_1(C_1^{in}, C_1^{out})$</td><td>LSH alarm is active</td></tr>
<tr><td colspan="3">$C_2(C_2^{in}, C_2^{out})$</td><td>LSL alarm is active</td></tr>
</table>

Figure 1.23 shows the NCES' model for the uncontrolled tank process including the pump. Table 1.3 gives a brief description of the places, transitions, and conditions for the uncontrolled model. From the tank filling and draining process description it is known that the valve and pump do not interact. Just like the valve, the pump is only dependent on the status of LSH and LSL. Condition signals are use to interconnect the four modules.

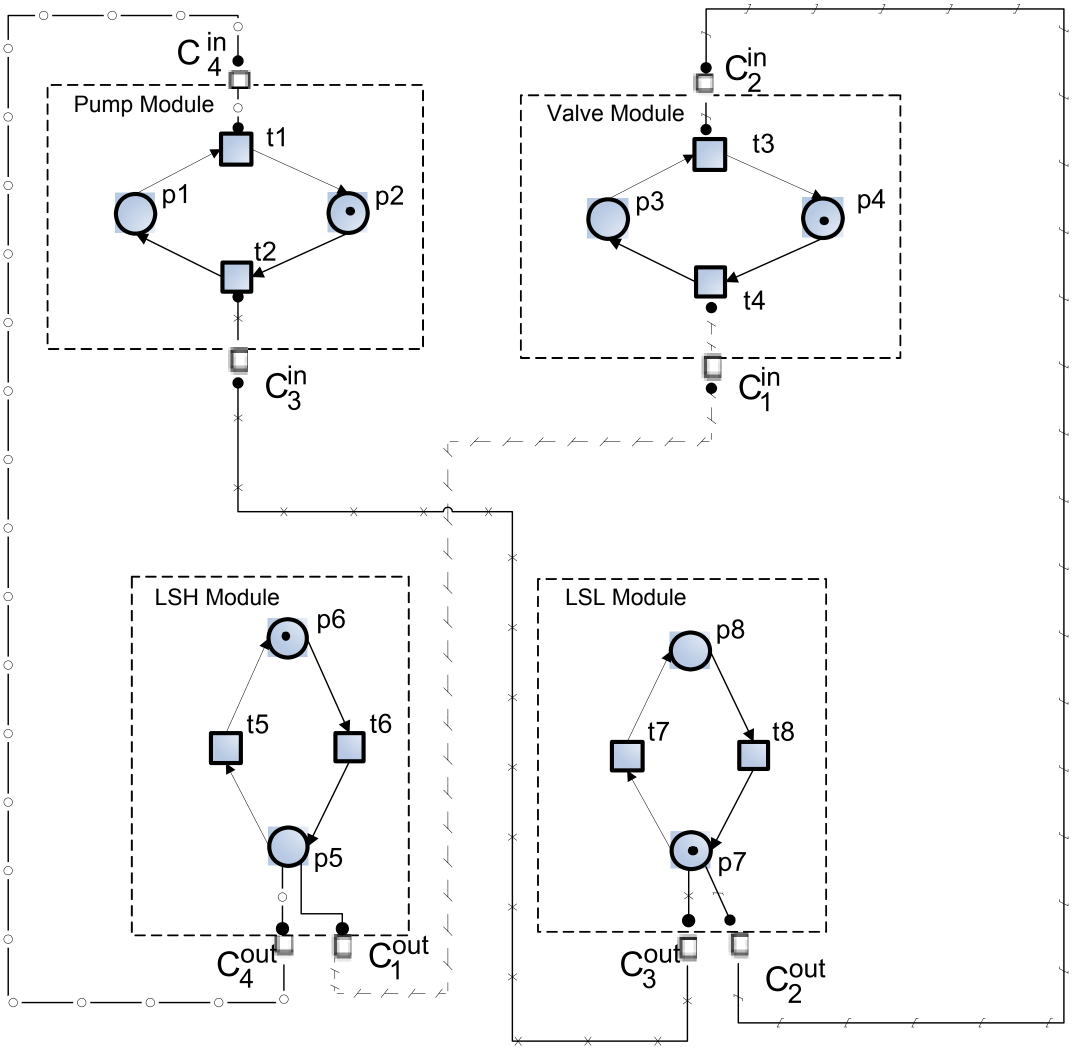


Figure 1.23: NCES Uncontrolled System Model

Table 1.3: Places, Transitions, and Conditions for the Tank Uncontrolled Model

<table>
<tr><th colspan="4">Pump Module</th></tr>
<tr><th>Transition</th><th>Meaning</th><th>Place</th><th>Meaning</th></tr>
<tr><td>t1</td><td>Pump turning off</td><td>p1</td><td>Pump on</td></tr>
<tr><td>t2</td><td>Pump turning on</td><td>p2</td><td>Pump off</td></tr>
<tr><th colspan="4">Valve Module</th></tr>
<tr><th>Transition</th><th>Meaning</th><th>Place</th><th>Meaning</th></tr>
<tr><td>t3</td><td>Valve closing</td><td>p3</td><td>Valve opened</td></tr>
<tr><td>t4</td><td>Valve opening</td><td>p4</td><td>Valve closed</td></tr>
<tr><th colspan="4">LSH Module</th></tr>
<tr><th>Transition</th><th>Meaning</th><th>Place</th><th>Meaning</th></tr>
<tr><td>t5</td><td>LSH alarm goes passive</td><td>p5</td><td>LSH alarm active</td></tr>
<tr><td>t6</td><td>LSH alarm goes active</td><td>p6</td><td>LSH alarm passive</td></tr>
<tr><th colspan="4">LSH Module</th></tr>
<tr><th>Transition</th><th>Meaning</th><th>Place</th><th>Meaning</th></tr>
<tr><td>t7</td><td>LSH alarm goes passive</td><td>p7</td><td>LSL alarm active</td></tr>
<tr><td>t8</td><td>LSL alarm goes active</td><td>p8</td><td>LSL alarm passive</td></tr>
</table>

<table>
<tr><th colspan="2">Module Conditions</th></tr>
<tr><th>Condition</th><th>Meaning</th></tr>
<tr><td>$C_1(C_1^{in}, C_1^{out})$</td><td>LSH alarm is active</td></tr>
<tr><td>$C_2(C_2^{in}, C_2^{out})$</td><td>LSL alarm is passive</td></tr>
<tr><td>$C_3(C_3^{in}, C_3^{out})$</td><td>LSL alarm is passive</td></tr>
<tr><td>$C_4(C_4^{in}, C_4^{out})$</td><td>LSH alarm is active</td></tr>
</table>

1.5 Summary

In this chapter NCES's and PNs, which are the foundation of NCES's, are introduced. The NCES's modeling process is illustrated using a basic tank filling and draining process. NCES's are suitable for modeling complex DEDS due to their input and output structure, which captures the dynamic behavior of complex DEDS more efficiently than other DEDS modeling formalisms.

Chapter 2: Literature Review

Several formalisms are used to model and control discrete event dynamic systems (DEDS). Among them are Finite Automata [9], PNs [12], Temporal Logic [13], and NCES. This thesis focuses on NCES. The following section will review the literature on NCES and their automatic transformation into PLC programming language.

2.1 Evolution of Net Condition Event System

Based on the work of Ramadge and Wonham [9]; R.S. Sreenivas and B.H. Krogh [2] propose a class of discrete event dynamic system (DEDS), which they call Condition Event (C/E) Systems. Condition signals and event signals are the two classes of input and output signals used in C/E systems. Condition signals are piecewise constant signals. Event signals are null except for discrete points of time. Furthermore, event signals are graphically represented by a zigzag symbol (⇝), meanwhile condition signals flow lines use a straight arrow head (→). The authors also define three qualitative properties that characterize C/E systems: causality, time change invariance, and spontaneity. The authors use a conveyor belt example to show the casual interconnection between the physical components of a system. The example proves that condition and event signals offer a more realistic modeling framework than finite automata or formal languages. Furthermore, they are able to verify that the conveyor belt system model has the qualitative properties of a C/E system. C/E systems are also used to model supervisory control applications. Their condition and event signal structure is the same structure used to connect the uncontrolled behavior of the system and its supervisor. The authors are able to develop a C/E language for a C/E system. The C/E language provides a representation for all possible orderings of the conditions and events in a C/E system. Finally, the authors show how to interconnect C/E systems in cascade and feedback configurations to obtain a discrete state model. In a cascade configuration two C/E systems are connected sequentially. The events in the second system are dependent on the events on the first system. In a feedback configuration the condition and events signals form a closed loop with the C/E system.

In [10], Sreenivas and Krogh extend the definition of standard PNs to include auxiliary predicates and an input and output structure to obtain a C/E model (C/E PN's). A PN with auxiliary predicates is a 6 tuple as follows:

$\mathcal{PN} = (\mathcal{P}, \mathcal{T}, \mathcal{A}_i, \mathcal{A}_o, \rho, \mu_o)$ where:

- $\mathcal{P}$ = $\{p_1, p_2, \ldots, p_n\}$ is an ordered set of n places
- $\mathcal{T}$ = $\{t_1, t_2, \ldots, t_m\}$ is an ordered set of m transitions
- $\mathcal{A}_i \in \mathcal{N}^{n \times m}$ is a n x m state input matrix
- $\mathcal{A}_o \in \mathcal{N}^{n x m}$ is a n x m state output matrix
- $\mathcal{P}$ = $\{1, 2, \ldots, m\}$ x $\mathcal{N}^n$ $\{0, 1\}$ is a computable predicate function that defines a predicate on $\mathcal{N}^n$ for each transition $t_i \in \mathcal{T}$
- $\mu_o \in \mathcal{N}^n$ is the initial marking

The input and output structure $\Psi = \{\mathcal{C}_{in}, \mathcal{E}_{in}, \mathcal{C}_{out}, \mathcal{E}_{out}, \mathcal{B}_c, \mathcal{B}_e, \mathcal{C}_s, \mathcal{D}_t\}$ is the same as the one define in Chapter 1. The authors graphically represent condition inputs and outputs by squares and event inputs and outputs by diamonds. Each transition $t_i \epsilon \mathcal{T}$ has two index sets. The first set is a collection of indices of condition inputs denoted as $(\bullet c)$. The second set is a collection of indices of event inputs denoted as $(\bullet e)$. Similarly, each condition output has a set of indices of places and is denoted as $(\bullet p)$. Each event output has a set of indices of transitions and is denoted as $(\bullet t)$. The authors also define five enabling rules: state enabled, condition enabled, event enabled, predicate enabled and maximally forced. Moreover, the authors define an encoding/decoding structure as follows:

$\Theta = (\Theta_u, \Theta_v, \Theta_y, \Theta_z)$ where:

- $\Theta_u: U \rightarrow \{0, 1\}^r$ is a condition input encoding function
- $\Theta_v: V \rightarrow \{0, 1\}^s$ is an event input encoding function
- $\Theta_y: \{0, 1\}^p \rightarrow Y$ is a condition output decoding function
- $\Theta_z: \{0, 1\}^q \rightarrow Z$ is a event output decoding function.

A C/E PN is defined as $\Pi = (\mathcal{N}, \Psi, \Theta)$. The objective of the paper is the construction of a C/E PN's resulting from the interconnection of subsystem models. Specifically, Screenivas and Krogh develop an algorithm to create a C/E PN from two C/E PN subsystems connected in a cascade configuration. The purpose of the algorithm is to construct an equivalent model for the resulting cascade system such that $\mathcal{S}(\Pi) = \mathcal{S}(\mathcal{S}(\Pi_1) \rightarrow \mathcal{S}(\Pi_2))$. They also develop an algorithm to obtain an equivalent C/E PN for feedback configurations. As a result, the C/E

PN's obtained from the algorithms are more compact than the C/E models presented in [2].

M. Rausch and H.-M. Hanisch [1], inspired by the work of Screenivas and Krogh, use a modified C/E PN to model resource allocation problems. The authors propose three modifications for C/E PN; remove the auxiliary predicates, use bounded PN, and introduce two kinds of arcs for the graphical representation of condition and event signals. The authors define $\mathcal{PN} = \{\mathcal{P}, \mathcal{T}, \mathcal{F}, \mu_0)$ where $\mathcal{P}, \mathcal{T}$ and μ_0 are defined in the same manner as in [10]. $\mathcal{F}$ is the arcs including the token weight (incidence matrix). The input and output structure $\Psi = \{\mathcal{C}_{in}, \mathcal{E}_{in}, \mathcal{C}_{out}, \mathcal{E}_{out}, \mathcal{B}_c, \mathcal{B}_e, \mathcal{C}_s, \mathcal{D}_t\}$ is the same as the one define in Chapter 1. The changes transform C/E PN into NCES. The authors define NCES = $\{\mathcal{P}, \mathcal{T}, \mathcal{F}, \Psi, \mu_o, \mathcal{C}_\mathcal{N}, \mathcal{E}_\mathcal{N}\}$. They also define three enabling rules; marking enabled, condition enabled, and event enabled. Furthermore, the authors define spontaneous and forced transitions. They recognize that the model may contain conflicts and not all forced transition will be able to fire. Therefore, they develop an algorithm to determine the maximal step in which all forced transitions must fire. The algorithm is based on another algorithm developed for time PNs that present the same problem [14]. However, their algorithm is extended to analyze models composed of several small modules. The authors use NCES to model a polymer production plant. The plant consists of several reactors that need different quantities of a cooling agent. A controller must ensure that the cooling agent does not overload. The authors start by creating NCES modules for the reactor, for resource allocation, for pressure, and the controller. After which, they connect the four modules using condition and event signals to obtain the controlled model for the polymer plant. The desired behavior of the plant (specifications) is included in terms of forbidden states, which the controlled system avoids. They also developed a reachability graph to verify that the controller works correctly. The resulted example proved that NCES are applicable in the modeling of resource allocation problems.

In [7] Rausch and Hanisch used NCES to synthesis supervisory controller for the forbidden state problems. They define NCES the same way that [1] does. The authors model the uncontrolled behavior of a pusher/conveyor manufacturing system as their example. The modules for the pusher/ conveyor system are safe PN. Before creating the controller, the authors explain several steps that are required in preparation for the algorithm. First, they assign a Boolean function $\mathcal{ET}(t)$ (enabling term) to each transition. The enabling term is determined as follows: $\mathcal{ET}(t) = \bigwedge_{p_i \in F_t} p_i \wedge \bigwedge_{p_j \in CN_t} p_j \wedge \bigwedge_{c_i^{in} \in B_{ct}} c_i^{in} \wedge \bigwedge_{t_i \in EN_t} \mathcal{ET}(t) \wedge \bigwedge_{e_i^{in} \in B_{et}} e_i^{in}$. Secondly, they assign a set of transitions to each place denoted as $\mathcal{T}_{IN}(p_i)$. Thirdly they compute a function for the predecessors as follows: $pred(p) = \bigwedge_{t_i \in t_{IN(p)}} \mathcal{ET}(t)$. Finally, they calculate all place invariants Ip, where $iv = \bigwedge_{p \in Ip} p = 0$. A place invariant is a set of places where the sum of the tokens is constant. Then, the authors present an algorithm that transforms the specification into a controller function. In the algorithm they compute the enabling terms of the forbidden states. After which, they replace each place in the enabling term for its predecessor. From the resulting term, they derive a controller function by replacing the places by the appropriate output signals. The authors realize that the state space can grow exponentially with the size of the model. This algorithm avoids the computation of the whole state space.

L. E. Pinzon, H.-M. Hanisch, M. A. Jafari and T. Boucher in [8] illustrate the advantages and disadvantages for some of the existing synthesis methods for discrete event controllers. This paper discusses the synthesis methods for: formal languages based on the work of Ramadge and Wonham (R&W) [9], [16]; PN [17]; Timed Transitions Models (TTM) [15]; and NCES [1], [7]. The authors use a pusher example to compare the model formalisms implementation methodology for the uncontrolled system model, specification model and controller synthesis. It is concluded that R& W and TTM have difficulty in keeping track of the resources in the uncontrolled system models. On the other hand, PN and NCES use of markings facilitates the tracking of the resources and PN's uncontrolled system model is more complex than NCES uncontrolled system model. NCES are more compact and precise, when modeling the casual behavior of a system. For the specification model, the authors analyze the modeling formalisms ability to model safety and sequential specifications. R&W methodology is able to model both types of specification, but the implementation is not trivial. TTM methodology is also capable of modeling both types of specifications; however, the specification models are not used to synthesize the controller. PN's are not able to model sequential specifications. NCES's are able to model both types of specifications by means of forbidden states. Furthermore, the modeling specification structure for NCES is similar to R&W, but NCES do not consider the whole state space. Finally, the authors analyze the controller synthesis methodologies for each of the modeling formalism. R&W and PN methodology guarantees that the supervisor is maximally permissive. A maximally permissive supervisor is one that only restricts those events which are not legal with respect to the specification. However, R&W algorithms are difficult to implement. NCES and TTM supervisor are not maximally permissive, and provide more efficient solutions because they do not consider the whole state space. Furthermore, NCES can automatically generate control code for PLC's.

L. E. Pinzon, H.-M. Hanisch, and M. A. Jafari along with P. Zhao continue their work [5] to develop a more efficient synthesis method that avoids state space explosions and allows sequential specification. The authors present a NCES uncontrolled system and specification model. The uncontrolled system model is based on a safe PN. However, the example models do not portray any specific manufacturing process. Assuming that all transitions in the uncontrolled system are controllable, the authors obtain the admissible behavior by creating a locking controller. The locking controller uses a condition signal to ensure that a transition t is disabled whenever the controller is in a specific state. In any other controller state, transition t will not be restricted. In other words, the uncontrolled system model sends an event signal to the specification model. The specification model reacts by sending another event signal to the locking controller. Then, the locking controller will send a condition signal to the uncontrolled system that locks the forbidden transition. The main contribution of this paper is an algorithm that obtains the admissible behavior of a system in the presence of uncontrollable events. The algorithm identifies and keeps track of the set of states that enable the uncontrollable transitions. As a result, the authors obtain a locking controller model that prevents the enabling of the uncontrolled transition in a minimally restrictive way and that includes the admissible behavior of the system.

2.2 Automatic Generation of Programmable Logic Controller Language

Programmable Logic Controllers (PLC's) [18 - 19] have been used in automated manufacturing systems since the 1970's. The PLC programming language was design using logic and symbols similar to electrical circuit diagrams. Therefore, electricians and technicians were able to easily use, program, and debug PLC's. The invention of PLC's brought many advantages to the manufacturing floor such as speed, flexibility, and increased performance. The original PLCs were simple devices. Through time, PLC's have increased in complexity, adding several features that improve their programming, debugging, and operation. The standard PLC programming languages [20] include: instruction list (IL), structure text (ST), function block diagram (FBD), sequential function chart (SFC), and Ladder Logic Diagram (LLD). LLD is the most commonly used for programming PLC's.

PLC programming languages are difficult to understand, modify and maintain. Furthermore, the larger the system to control, the more complex the resulting program becomes. Researchers have proposed several methods to address these problems by expressing the control logic using some type of mathematical formalism and then automatically converting the resulting model to a PLC programming language. In addition, these formal models can be used to verify the controller's desirable logical properties and correctness.

Finite Automata (FA), PN, and NCES have been extensively used to model and analyze controllers. The conversion of FA based models is addressed by B. A. Brandin in [21], M. Fabian and A. Hellgreen in [22], and J. Liu and H. Darabi in [23]. Conversion of PN based models is addressed in [24 -27]. There has been research on the transformation of other types of PN models such as T-Timed PN [28, 29]; P-Timed PN [30, 31]; and Coloured PN [32, 33]. While the work on transformation of PN and FA based model is extensive, the work on transformation of NCES models to PLC programming languages is limited. In [11] M. Rausch and B.H. Krogh introduce a methodology that describes the transformation among NCES, Statecharts, and PLC programming languages. The work proposes the transformation of NCES into Instruction List and vice versa and provides a set of rules that transform conditions signals, places and transitions into input and output variables. However, the transformation of events signals is not included. Also, the example discussed does not possess complex control requirements frequently encountered in manufacturing systems.

Chapter 3: DEDS Supervisory Control Synthesis Using NCES

Discrete event dynamic systems (DEDS) require control and coordination in order to satisfy a desired behavior. One important aspect in control theory is the selection of a modeling formalism that can capture the physical characteristics of DEDS and the ability to synthesize a controller. This research utilizes NCES as the model formalism for generating DEDS controllers. Before the controller synthesis is discussed, it is necessary to discuss some basic concepts in Supervisory Control Theory.

3.1 DEDS Supervisory Control Theory

Supervisory controllers ensure the proper operation of a DEDS by enforcing the behavioral requirements. A behavioral requirement is a requirement that a system must follow during its operation. For example, in the tank filling and draining process introduced in Chapter 1 the valve can not be opened when the pump is on. In a supervisory control model, these requirements are defined in terms of safety or sequential specifications. A safety specification refers to forbidden state(s) that the system must avoid. A sequence specification refers to a desired sequence of events the system must follow.

Figure 3.1 shows the interaction of a supervisor controller and an uncontrolled system. The supervisory controller and the uncontrolled system interact as a closed loop system. The supervisory controller observes the events executed by the uncontrolled system and decides which event(s) are allowed next on the uncontrolled system. The event(s) allowed by the supervisor is known as the control pattern. The control pattern ensures that the uncontrolled system operates within the boundaries dictated by the specifications. Furthermore, the supervisory controller has the capability to disable some, not necessarily all, of the possible events on the uncontrolled system. Those events that the supervisory controller can disable are known as controllable events. An example of controllable events is the filling and draining status in the tank filling and draining process. The controller is able to control the exact moment when the filling and draining process starts and ends. The events that the supervisory controller cannot disable are known as uncontrollable events. An example of an uncontrollable event is the level sensors status in the tank filling and draining process. The amount of liquid in the tank triggers the level sensors, but the controller cannot control the exact moment when the level sensors are going to trigger. As a result, the level sensors transitions {t5, t6, t7, and, t8} are uncontrollable transitions.

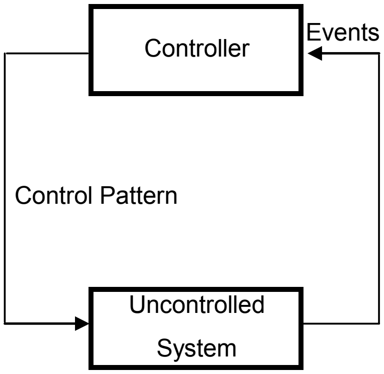


Figure 3.1: Supervisory Control System

The development of a supervisory controller model is divided into three main tasks: the modeling of the uncontrolled system and specifications, controller synthesis, and controller implementation. In the rest of this chapter, these steps will be followed to develop a supervisory control model for the tank filling and draining process introduced in Chapter 1.

3.2 Uncontrolled System and Specification Model

The NCES uncontrolled system model is shown in Figure 1.23 Chapter 1. For convenience, Figure 1.23 and Table 1.3 are reproduced in this chapter in Figure 3.2 and Table 3.1. Notice that the initial conditions are that the tank is empty, the pump is off, the valve is closed, LSH alarm is passive, and LSL alarm is active. From this state (marking μ_0 = (0, 1, 0, 1, 0, 1, 1, 0)) only t2 (pump turning on) can fire and the pump will turn on. Notice that t4 (valve opening) can not fire because the condition signal (C_1^{in}, LSH alarm is active) is not true. During the filling process t7 (LSL alarm goes passive) and t6 (LSH alarm goes active) fire, this means that the tank is full (μ' = (1, 0, 0, 1, 1, 0, 0, 1)). After the tank is completely filled, observe that t1 (pump turning off) and t4 (valve opening) are enabled. If t4 fires, the pump will be on while the valve is opened. This is an undesirable behavior for the tank filling and draining process. Therefore, a specification for the tank filling and draining process is that when the tank is full the pump should turn off first and afterwards the valve can open. This specification will be modeled by means of a sequential specification. The tank filling and draining process will start with the filling process and continue to the draining process. Once the draining process is over the cycle will repeat. As a result, during the filling process the valve will remain closed and the pump will turn on. During the draining process the pump will remain off and the valve will open. The controller must enforce the occurrence of these processes in that given order. During the filling process the controller will disabled the valve and during the draining process the controller will disabled the pump.

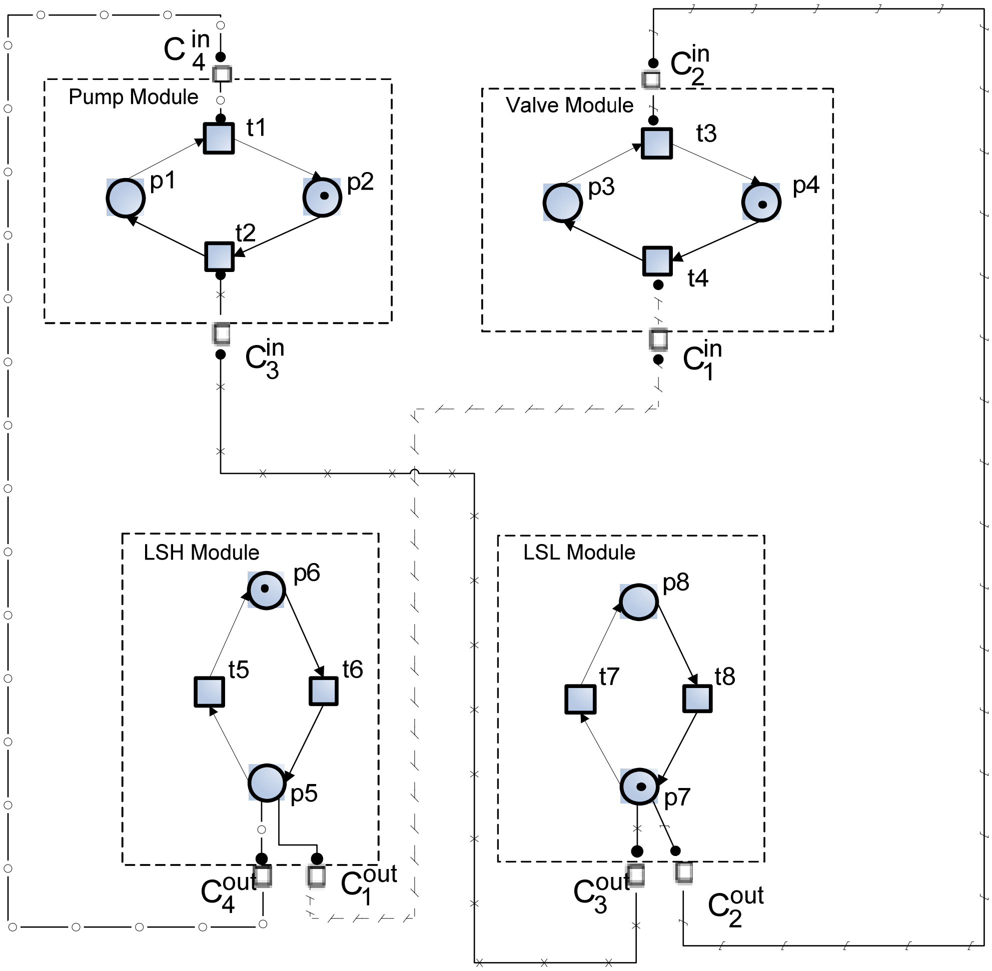


Figure 3.2: Reproduction: Tank NCES Uncontrolled System Model

Table 3.1: Reproduction: Places, Transitions, and Conditions for the Tank Uncontrolled Model

Pump Module			
Transition	Meaning	Place	Meaning
t1	Pump turning off	p1	Pump on
t2	Pump turning on	p2	Pump off
Valve Module			
Transition	Meaning	Place	Meaning
t3	Valve closing	p3	Valve opened
t4	Valve opening	p4	Valve closed
LSH Module			
Transition	Meaning	Place	Meaning
t5	LSH alarm goes passive	p5	LSH alarm active
t6	LSH alarm goes active	p6	LSH alarm passive
LSH Module			
Transition	Meaning	Place	Meaning
t7	LSL alarm goes passive	p7	LSL alarm active
t8	LSL alarm goes active	p8	LSL alarm passive

Module Conditions	
Condition	Meaning
$C_1(C_1^{in}, C_1^{out})$	LSH alarm is active
$C_2(C_2^{in}, C_2^{out})$	LSL alarm is active
$C_3(C_3^{in}, C_3^{out})$	LSL alarm is active
$C_4(C_4^{in}, C_4^{out})$	LSH alarm is active

Figure 3.3 illustrates the specification model. Table 3.2 shows the description of the transitions and places for the specification model. Notice the input and output events shown in Figure 3.3. The input events are the events from the uncontrolled system. The output events are the events sent to the controller from the specification model.

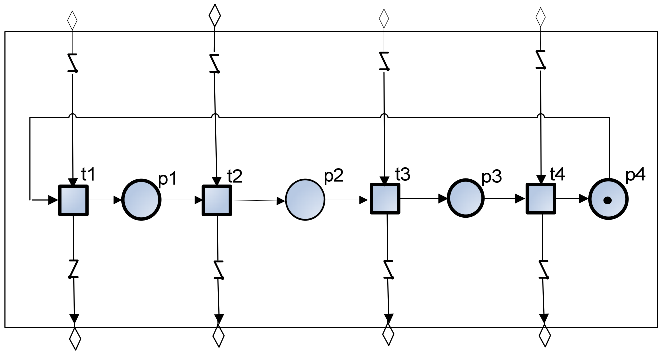


Figure 3.3: Tank Specification Model

Table 3.2: Places and Transitions for the Tank Specification Model

Specification Model			
Transition	Meaning	Place	Meaning
t1	Starting Filling process	p1	Filling process
t2	Stopping Filling process	p2	Filling process stops
t3	Starting draining process	p3	Draining process
t4	Stopping draining process	p4	Draining process stops

3.3 Controller Synthesis

The locking controller (LC) methodology proposed in [5] is used to synthesize the controller. The sequential specification model must include copies of the uncontrolled system place (p_i^c) and its input/output transitions (t_i^c and t_{i+1}^c). The LC will not restrict the transitions that are not part of the sequential specification. If transition t_i in the uncontrolled system model is controllable and forbidden at some place p_i^c in the sequential specification, then transition t_j will be disabled as soon as t_i^c fires and enabled as soon as t_{i+1}^c fires. The ability to enable t_j again, will guarantee that the controller is minimally restrictive. The LC consist of a *co-place* (p_i^{co}) for the specification place p_i^c, and copies of the input/output transitions of p_i^c (t_i^l and t_{i+1}^l) as shown in Figure 3.4. The sequential specifications transitions will be connected to the locking controller transitions via event signals. The LC is connected to the uncontrolled system via a condition signal sent by the co-place. The condition signal will disable transition t_j whenever the specification is in state p_i^c. At any other state, transition t_j will not be restricted by the controller. Figure 3.4 shows the net control structure obtained by adding a locking controller to a generic uncontrolled system and sequential specification model.

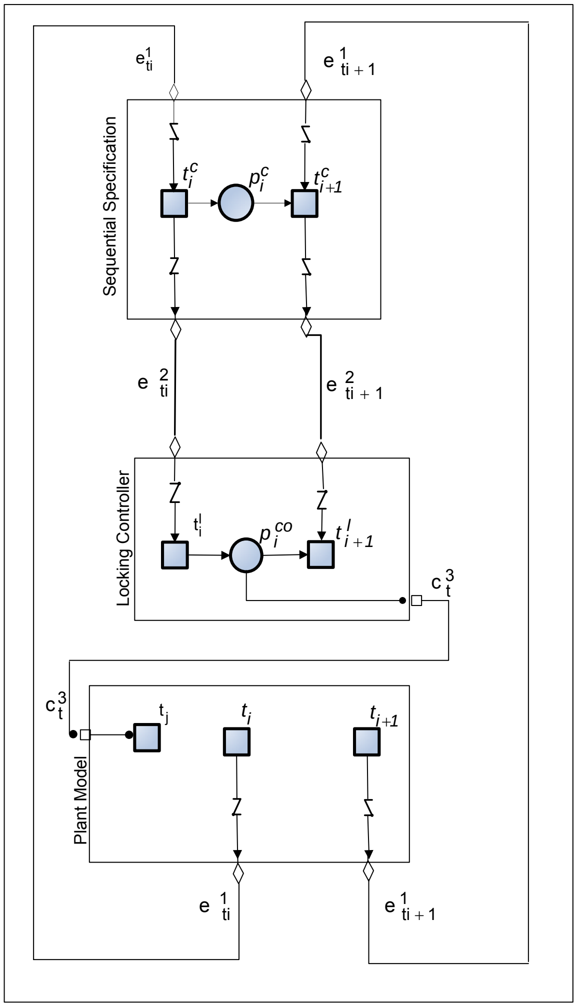


Figure 3.4: Locking Controller

To use the locking controller methodology it is necessary to modify the specification model shown in Figure 3.3. The places and transitions defined in the specifications model are modified to comply with the locking controller. Place p1 (filling process) changes to p_1^c, since it is a copy of place p1 (pump on). In other words, the filling process is defined by turning the pump on. Transition t_2^c (start filling process) is a copy of transition t2 (pump turning on). Transition t_1^c

(stop filling process) is a copy of transition t1 (pump turning off). The rest of the places and transitions are replaced in the same manner and the resulting specification model is shown in Figure 3.5. Table 3.3 describes the transitions and places for the Tank locking controller and specification model.

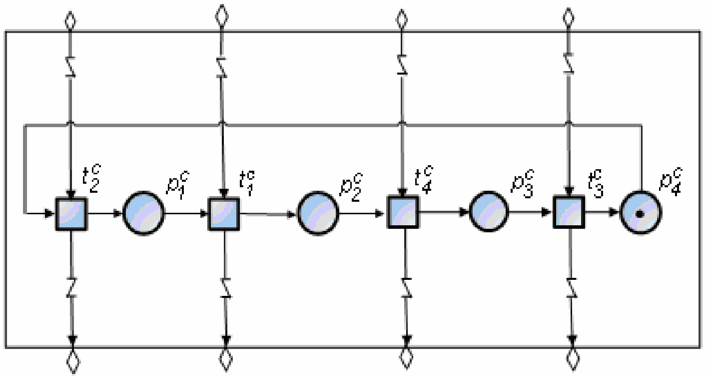


Figure 3.5: Locking Controller Sequential Specification for Tank Model

Table 3.3: Places and Transitions for the Tank LC and Specification Model

Specification Model			
Transition	Meaning	Place	Meaning
t_2^c	Starting Filling process	p_1^c	Filling process
t_1^c	Stopping Filling process	p_2^c	Filling process stops
t_4^c	Starting draining process	p_3^c	Draining process
t_3^c	Stopping draining process	p_4^c	Draining process stops

Figure 3.6 shows the complete controller model for the tank process. Table 3.4 describes the transitions, places, conditions and events of the controller model. The model consists of two locking controller modules. One module controls the valve and the other module controls the pump. The LC module on the left has a co-place p_1^{co} for p1 (pump on). The LC module ensures that the valve is not open during the filling process. The LC disables t4, when the sequential specification is in p_1^c. The LC module on the right has a co-place p_3^{co}

for p3 (valve open). The LC module ensures that the pump will not turn on during the draining process. The locking controller will disable t2, when the sequential specification is in p_3^c. The LC ensures the uncontrolled system meets the behavioral specification that when the tank is full the pump should turn off and afterwards the valve can open.

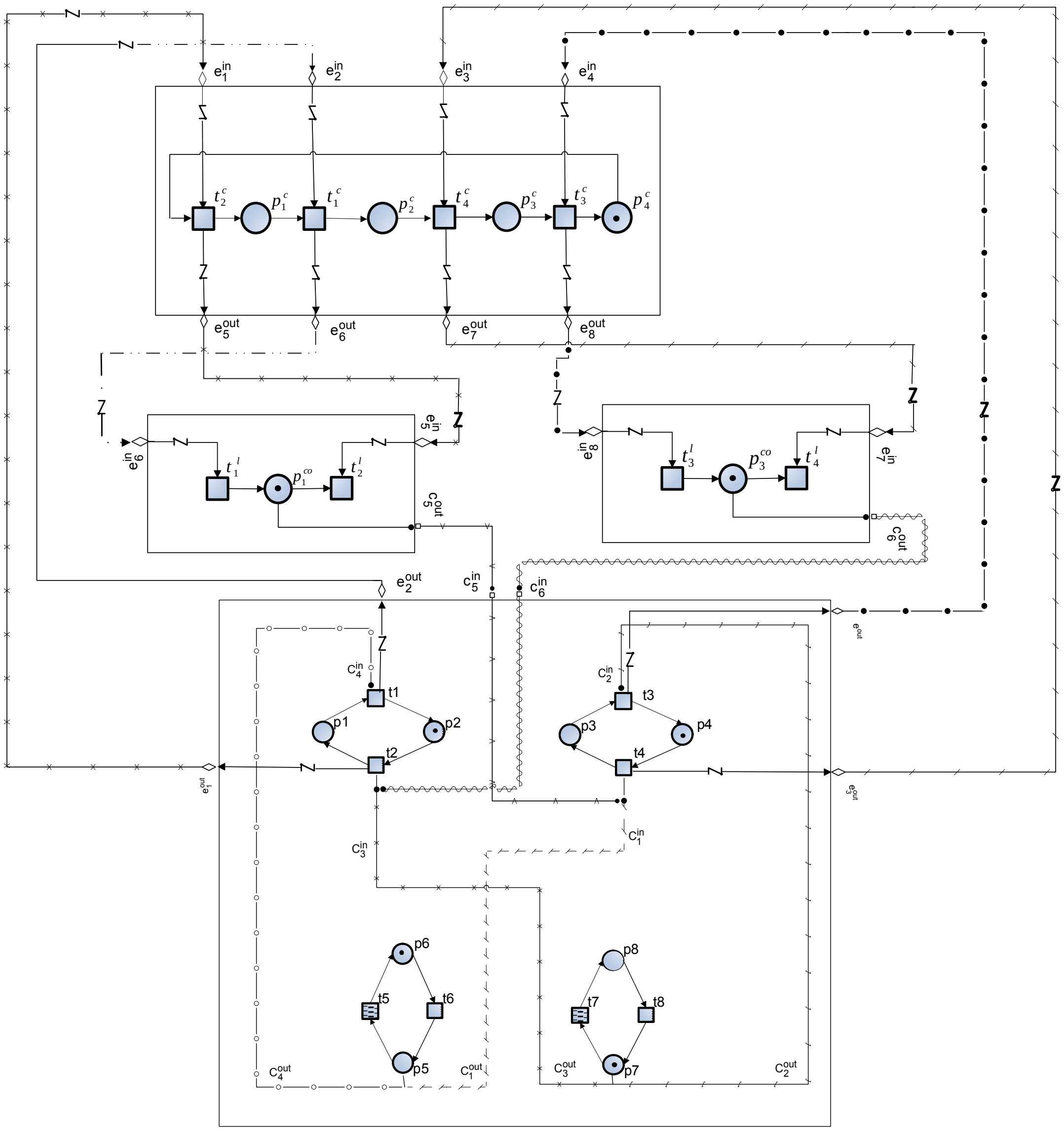


Figure 3.6: Tank Control Model

Table 3.4: Places, Transitions, Conditions, and Events for the Tank Control Model

Uncontrolled System Model			
Transition	Meaning	Place	Meaning
t1	Pump turning off	p1	Pump on
t2	Pump turning on	p2	Pump off
t3	Valve closing	p3	Valve opened
t4	Valve opening	p4	Valve closed
t5	LSH alarm goes passive	p5	LSH alarm active
t6	LSH alarm goes active	p6	LSH alarm passive
t7	LSL alarm goes passive	p7	LSL alarm active
t8	LSL alarm goes active	p8	LSL alarm passive
Specification Model			
Transition	Meaning	Place	Meaning
t_2^c	Starting Filling process	p_1^c	Filling process
t_1^c	Stopping Filling process	p_2^c	Filling process stops
t_4^c	Starting draining process	p_3^c	Draining process
t_3^c	Stopping draining process	p_4^c	Draining process stops
Locking Controller Modules			
Transition	Meaning	Place	Meaning
t_2^l	Copy of t_2^c	p_1^{co}	Co-place of p_1^c
t_1^l	Copy of t_1^c	p_3^{co}	Co-place of p_3^c
t_4^l	Copy of t_4^c		
t_3^l	Copy of t_3^c		

Model Conditions	
Condition	Meaning
$C_1(C_1^{in}, C_1^{out})$	LSH alarm is active
$C_2(C_2^{in}, C_2^{out})$	LSL alarm is active
$C_3(C_3^{in}, C_3^{out})$	LSL alarm is active
$C_4(C_4^{in}, C_4^{out})$	LSH alarm is active
$C_5(C_5^{in}, C_5^{out})$	Filling process active
$C_6(C_6^{in}, C_6^{out})$	Draining Process on
Model Events	
Events	Meaning
$e_1(e_1^{in}, e_1^{out})$	Pump is turning on
$e_2(e_2^{in}, e_2^{out})$	Pump is turning off
$e_3(e_3^{in}, e_3^{out})$	Valve is opening
$e_4(e_4^{in}, e_4^{out})$	Valve is closing
$e_5(e_5^{in}, e_5^{out})$	Filling process is starting
$e_6(e_6^{in}, e_6^{out})$	Filling process is stopping
$e_7(e_7^{in}, e_7^{out})$	Draining process is starting
$e_8(e_8^{in}, e_8^{out})$	Draining process is stopping

Chapter 4: Verification of the Control Model: Reachability Analysis

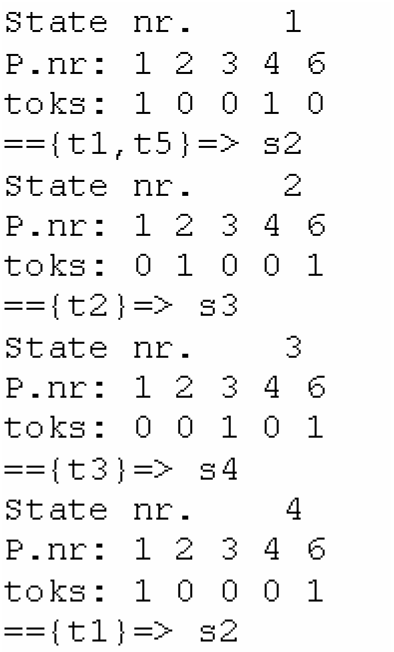
One of the tools for analysis of a NCES is the reachability graph [4]. For some complex systems the size of the reachability tree can become very large, making it difficult to produce manually and analyze visually. In this chapter, a software tool to analyze the properties of NCES is introduced and utilized to verify the correctness of the control model obtained in Chapter 3.

4.1 SESA

A research group from the Humboldt University in Berlin developed a program to support the formal analysis of NCES, which they called SESA [34, 35, and 36]. This software tool allows the user to insert the model information (places, conditions, events, transitions, etc) and the software performs an analysis of the model. The analysis report displays the reachable states and NCES properties pertinent to the model. Figure 4.1 shows an example of a SESA analysis report for the properties of a NCES model. Figure 4.2 shows an example of how SESA displays the reachable states for a NCES model with 6 places. The reachable states are organized by state numbers in chronologically order, where state nr.1 is the initial state. Below each state number there are at least three additional lines. The first line represents the number of places in the model. The second line represents the number of tokens in each place (marking). The final line(s) represent the set of transitions that are enabled followed by the state number the system will move to after the transitions are fired. There could be more than one set of transitions. To properly identify each set there are two equal signs before each set. A more detailed example will be explained in Section 4.2.

```
The net is not conservative.
The net is subconservative.
The net is structurally bounded.
The net is bounded.
The net is not live.
The net is not live and bounded at the same state.
The net is marked.
The net is not marked with exactly one token.
The net has no spontaneous transitions without pre-place and pre-condition.
The net has no spontaneous transitions without post-place and signal arc.
The net has no places without pre-transition but with post-transition.
The net has no places without post-transition but with pre-transition.
Maximal in/out-degree:  2
SCV SCF Ft0 tF0 Fp0 pF0 CPI CTI  B  SB  REV DSt BSt DTr DCF  L   LV L&B
 Y   N   N   N   N   N   ?   ?   Y   Y   ?   ?   ?   ?   ?   N   ?   N
```

Figure 4.1: SESA Analysis Report



```
State nr.    1
P.nr: 1 2 3 4 6
toks: 1 0 0 1 0
=={t1,t5}=> s2
State nr.    2
P.nr: 1 2 3 4 6
toks: 0 1 0 0 1
=={t2}=> s3
State nr.    3
P.nr: 1 2 3 4 6
toks: 0 0 1 0 1
=={t3}=> s4
State nr.    4
P.nr: 1 2 3 4 6
toks: 1 0 0 0 1
=={t1}=> s2
```

Figure 4.2: SESA Reachable States

4.2 SESA Tank Control Model

SESA is used to obtain the reachable states for the tank control model in Figure 3.6 Chapter 3. SESA only allows the use of numbers to label places and transitions. Table 4.1 is provided as a guide to identify the corresponding places and transitions for the SESA results.

Table 4.1: Guide for Places and Transitions of the SESA Tank Control Model

Place	SESA Place	Transition	SESA Transition
p1	p1	t1	t1
p2	p2	t2	t2
p3	p3	t3	t3
p4	p4	t4	t4
p5	p5	t5	t5
p6	p6	t6	t6
p7	p7	t7	t7
p8	p8	t8	t8
p_1^c	p9	t_2^c	t9
p_2^c	P10	t_1^c	t10
p_3^c	P11	t_4^c	t11
p_4^c	P12	t_3^c	t12
p_1^{co}	P13	t_1^I	t13
p_3^{co}	P14	t_2^I	t14
		t_3^I	t15
		t_4^I	t16

The portion of the SESA reachable state for the tank filling and draining control model are displayed in Figure 4.3. The complete results for the SESA reachable states are in Appendix A.

```
State nr.    1
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  0  1  1  0  0  0
0  1  1  1
=={t2,t9,t14}=> s2
=={t2,t6,t9,t14}=> s8
=={t2,t7,t9,t14}=> s47
=={t2,t6,t7,t9,t14}=> s3
=={t6,t7}=> s11
=={t7}=> s35
=={t6}=> s7
State nr.    2
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  0  1  1  0  1  0
0  0  0  1
=={t6,t7}=> s3
=={t7}=> s47
=={t6}=> s8
State nr.    3
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  1  0  0  1  1  0
0  0  0  1
=={t1,t10,t13}=> s4
=={t1,t5,t10,t13}=> s39
=={t1,t8,t10,t13}=> s9
=={t1,t5,t8,t10,t13}=> s40
=={t5,t8}=> s2
=={t8}=> s8
=={t5}=> s47
State nr.    4
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  1  0  0  1  0  1
0  0  1  1
=={t4,t11,t16}=> s5
=={t4,t5,t11,t16}=> s37
=={t4,t8,t11,t16}=> s10
=={t4,t5,t8,t11,t16}=> s6
=={t5,t8}=> s40
=={t8}=> s9
=={t5}=> s39
State nr.    5
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  1  0  0  1  0  0
1  0  1  0
=={t5,t8}=> s6
=={t8}=> s10
=={t5}=> s37
State nr.    6
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  0  1  1  0  0  0
1  0  1  0
=={t3,t12,t15}=> s1
=={t3,t6,t12,t15}=> s7
=={t3,t7,t12,t15}=> s35
=={t3,t6,t7,t12,t15}=> s11
=={t6,t7}=> s5
=={t7}=> s37
=={t6}=> s10
State nr.    7
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  1  0  1  0  0  0
0  1  1  1

=={t2,t9,t14}=> s8
=={t2,t4,t9,t14}=> s25
=={t2,t5,t9,t14}=> s2
=={t2,t4,t5,t9,t14}=> s13
=={t2,t7,t9,t14}=> s3
=={t2,t4,t7,t9,t14}=> s14
=={t2,t5,t7,t9,t14}=> s47
=={t2,t4,t5,t7,t9,t14}=> s48
=={t4,t5,t7}=> s31
=={t5,t7}=> s35
=={t4,t7}=> s32
=={t7}=> s11
=={t4,t5}=> s12
=={t5}=> s1
=={t4}=> s24
State nr.    8
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  1  0  1  0  1  0
0  0  0  1
=={t1,t10,t13}=> s9
=={t1,t5,t10,t13}=> s40
=={t1,t7,t10,t13}=> s4
=={t1,t5,t7,t10,t13}=> s39
=={t5,t7}=> s47
=={t7}=> s3
=={t5}=> s2
State nr.    9
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  1  0  1  0  0  1
0  0  1  1
=={t4,t11,t16}=> s10
=={t2,t4,t11,t16}=> s29
=={t4,t5,t11,t16}=> s6
=={t2,t4,t5,t11,t16}=> s19
=={t4,t7,t11,t16}=> s5
=={t2,t4,t7,t11,t16}=> s18
=={t4,t5,t7,t11,t16}=> s37
=={t2,t4,t5,t7,t11,t16}=> s38
=={t2,t5,t7}=> s27
=={t5,t7}=> s39
=={t2,t7}=> s17
=={t7}=> s4
```

Figure 4.3: Portion of SESA Tank Reachable States

The reachability graph shows that at the initial marking there are 7 sets of transitions enabled for firing. As discussed in Section 3.1, transitions t6 and t7 are uncontrollable (sensors). Within the theory of reachability graph it is possible to think that transitions t6 and t7 are enabled, but in the physical system it is not possible for them to fire. This means that meanwhile the pump is off the LSL will remain active (p7) and the LSH will remain passive (p6). Only the set containing t2, t9, t14 is representative of the physical system. In order to analyze the behavior of the Tank NCES model it is necessary to disregard the states of the reachability graph that are not executable in the physical system. Although the sensors are uncontrollable, their behavior represents the changes in the physical system. To control these transitions, events will be added to the control model. These events will force the uncontrollable transitions to fire in the sequence they logically should. For example, after the pump turns on and the tank starts filling LSL goes passive (t7). A place and transition (p_8^c and t_7^c) are added to the sequence specification to represent this state s shown in Figure 4.4. Furthermore, an event (e_9^c) is added as output to t_7^c and as input to t7 as shown in Figure 4.4. This way the specification can ensure that the LSL (t7) will fire only after the pump has turn on.

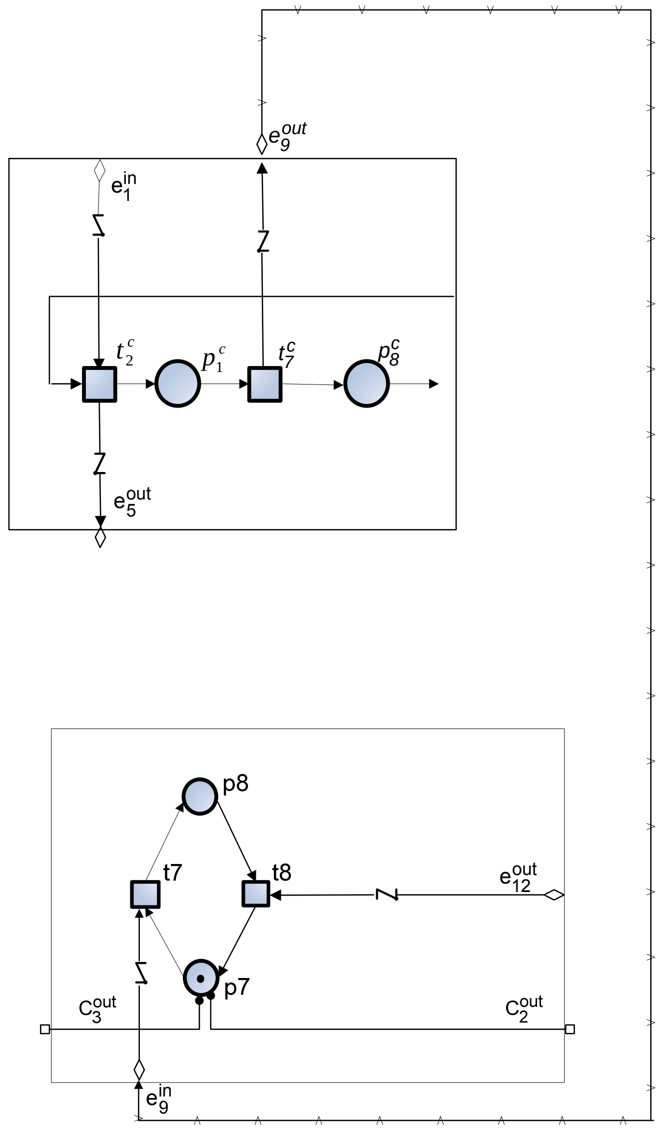


Figure 4.4: Portion of SESA Tank Control Model Pertaining t7

The new tank model is shown in Figure 4.5 and Table 4.2 describes the transitions, places, conditions, and events of the new control model. Note that these modifications are done only for verification purposes, but if the controller is implemented in a physical system the modifications would not be necessary.

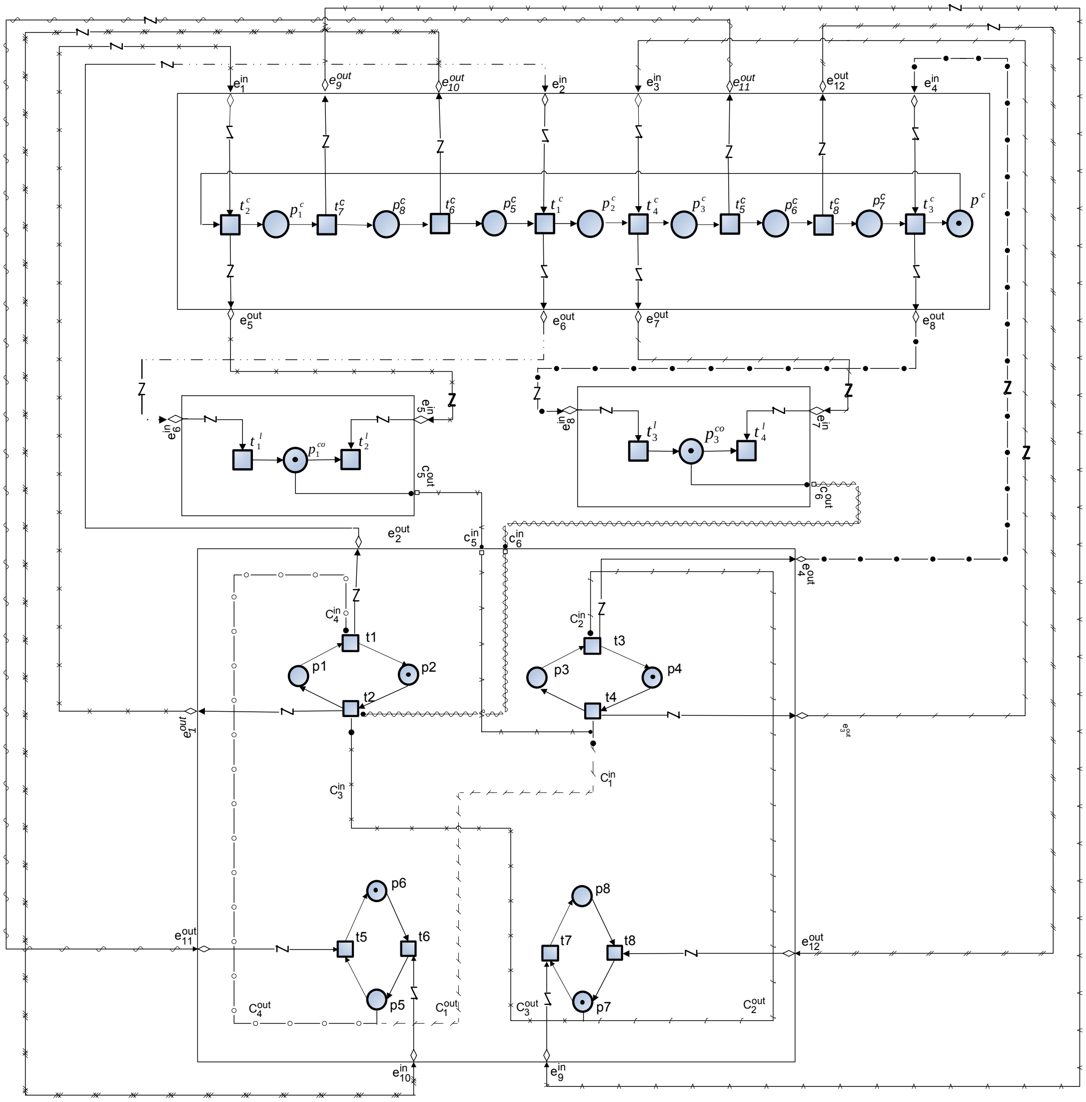


Figure 4.5: SESA Tank Control Model

Table 4.2: Places, Transitions, Conditions, and Events for the SESA Tank Control Model

New Tank Model			
Transition	Meaning	Place	Meaning
T1	Pump turning off	p1	Pump on
T2	Pump turning on	p2	Pump off
T3	Valve closing	p3	Valve opened
T4	Valve opening	p4	Valve closed
T5	LSH alarm goes passive	p5	LSH alarm active
T6	LSH alarm goes active	p6	LSH alarm passive
T7	LSL alarm goes passive	p7	LSL alarm active
T8	LSL alarm goes active	p8	LSL alarm passive
t_2^c	Starting Filling process	p_1^c	Filling process
t_7^c	LSL going passive	p_8^c	LSL passive
t_6^c	LSH going active	p_5^c	LSH active
t_1^c	Stopping Filling process	p_2^c	Filling process stops
t_4^c	Starting draining process	p_3^c	Draining process
t_5^c	LSL going active	p_6^c	LSL active
t_8^c	LSH going passive	p_7^c	LSH passive
t_3^c	Stopping draining process	p_4^c	Draining process stops
t_2^l	Copy of t_2^c	p_1^{co}	Co-place of p_1^c
t_1^l	Copy of t_1^c	p_3^{co}	Co-place of p_3^c
t_4^l	Copy of t_4^c		
t_3^l	Copy of t_3^c		

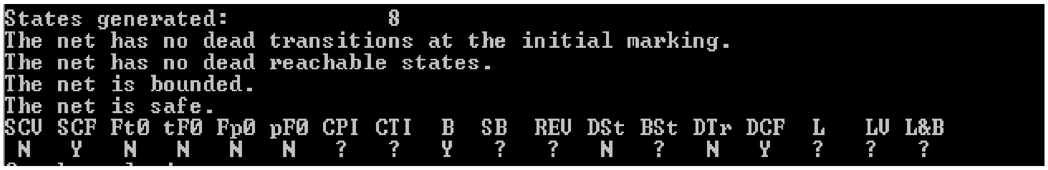
Model Conditions	
Condition	Meaning
$C_1(C_1^{in},C_1^{out})$	LSH alarm is active
$C_2(C_2^{in},C_2^{out})$	LSL alarm is active
$C_3(C_3^{in},C_3^{out})$	LSL alarm is active
$C_4(C_4^{in},C_4^{out})$	LSH alarm is active
$C_5(C_5^{in},C_5^{out})$	Filling process active
$C_6(C_6^{in},C_6^{out})$	Draining Process on

Model Events	
Events	Meaning
$e_1(e_1^{in},e_1^{out})$	Pump is turning on
$e_2(e_2^{in},e_2^{out})$	Pump is turning off
$e_3(e_3^{in},e_3^{out})$	Valve is opening
$e_4(e_4^{in},e_4^{out})$	Valve is closing
$e_5(e_5^{in},e_5^{out})$	Filling process is starting
$e_6(e_6^{in},e_6^{out})$	Filling process is stopping
$e_7(e_7^{in},e_7^{out})$	Draining process is starting
$e_8(e_8^{in},e_8^{out})$	Draining process is stopping
$e_9(e_9^{in},e_9^{out})$	LSL is going passive
$e_{10}(e_{10}^{in},e_{10}^{out})$	LSH is going active
$e_{11}(e_{11}^{in},e_{11}^{out})$	LSH is going passive
$e_{12}(e_{12}^{in},e_{12}^{out})$	LSL is going active

The analysis results from SESA for Figure 4.5 are displayed in Figure 4.6. The SESA reachable states for Figure 4.5 are shown in Figure 4.7. A reachability graph was constructed based on the SESA reachable state and is shown in Figure 4.8. Table 4.3 is provided as a guide to identify the corresponding places and transitions for the SESA results. The reachability states show that at each state there is only one set of transitions enabled for firing. These reachability states results are smaller than Figure 4.3 results and all the reachable states provided are in accordance with the physical system behavior. One can conclude from the analysis report and the reachable states that the NCES possess three behavioral properties:

- Reversible: The initial state; the tank is empty, the pump is off, the valve is closed, LSH alarm is passive, and LSL alarm is active. If a model is reversible, then one can conclude that for each marking there is a transition(s) that will take the system to its initial state. Chapter 1 describes the tank filling and draining process as a continuous cycle. Therefore, the fact that it is reversible proves that the Tank control model follows a cycle and it accurately describes the process behavior.
- Bounded and Safe: The model is 1-bounded, which means that regardless of the firing sequence there will only be one or zero tokens in each place. Since the control model represents the state of the devices (pump: off/on; valve: closed/open; sensors active/passive), it does not make sense in the physical system for a state to have two tokens. The safeness of the model guarantees that the system is modeled correctly, because each of the devices will only be at one of their two possible states in a given period of time.
- Live: The model is live because there are no dead states and all transitions will fire infinitely. This means that regardless of the firing sequence, the model will never be in stuck in a specific state. As a result, the model will never incur a deadlock situation.

The presence of these properties shows that the NCES tank control model developed correctly models the system behavior.



```
States generated:            8
The net has no dead transitions at the initial marking.
The net has no dead reachable states.
The net is bounded.
The net is safe.
SCV SCF Ft0 tF0 Fp0 pF0 CPI CTI  B  SB  REV DSt BSt DTr DCF  L   LV L&B
 N   Y   N   N   N   N   ?   ?   Y   ?   ?   N   ?   N   Y   ?   ?   ?
```

Figure 4.6: SESA Analysis Report for the Tank Control Model

```
State nr.    1
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
toks:  0  1  0  1  0  1  1  0  0  0  0  0  0  0  0  1  1  1
=={t2,t9,t18}=> s2
State nr.    2
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
toks:  1  0  0  1  0  1  1  0  1  0  0  0  0  0  0  0  0  1
=={t7,t10}=> s3
State nr.    3
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
toks:  1  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1
=={t6,t11}=> s4
State nr.    4
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
toks:  1  0  0  1  1  0  0  1  0  0  1  0  0  0  0  0  0  1
=={t1,t12,t17}=> s5
State nr.    5
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
toks:  0  1  0  1  1  0  0  1  0  0  0  1  0  0  0  0  1  1
=={t4,t13,t20}=> s6
State nr.    6
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
toks:  0  1  1  0  1  0  0  1  0  0  0  0  1  0  0  0  1  0
=={t5,t14}=> s7
State nr.    7
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
toks:  0  1  1  0  0  1  0  1  0  0  0  0  0  1  0  0  1  0
=={t8,t15}=> s8
State nr.    8
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
toks:  0  1  1  0  0  1  1  0  0  0  0  0  0  0  1  0  1  0
=={t3,t16,t19}=> s1
```

Figure 4.7: SESA Reachable State for the Tank Control Model

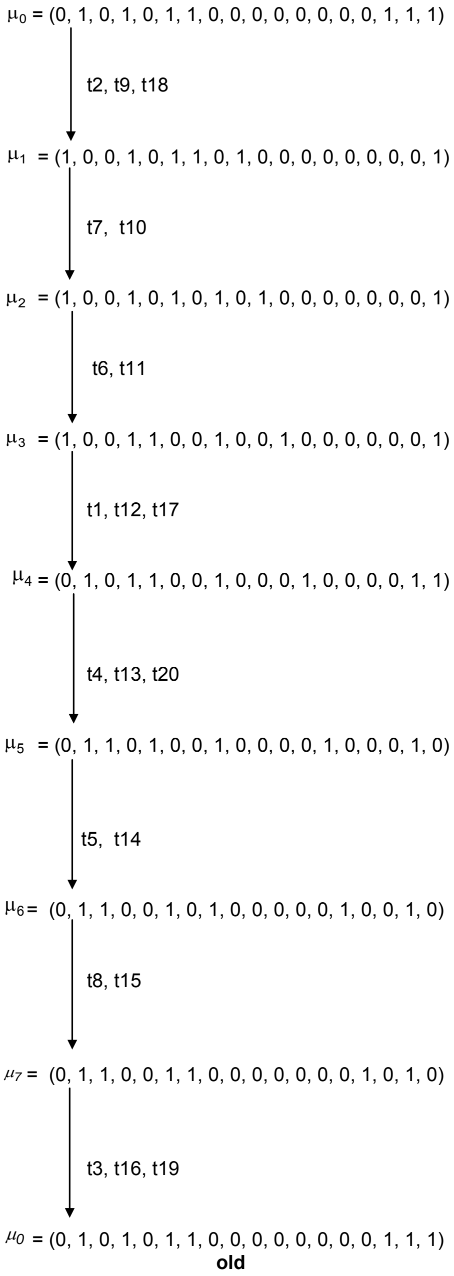


Figure 4.8: Reachability Graph for the Tank Control Model

Table 4.3: Guide for the Places and Transitions for the New SESA Tank Control Model

Place	SESA Place	Transition	SESA Transition
p1	p1	t1	t1
p2	p2	t2	t2
p3	p3	t3	t3
p4	p4	t4	t4
p5	p5	t5	t5
p6	p6	t6	t6
p7	p7	t7	t7
p8	p8	t8	t8
p_1^c	p9	t_2^c	t9
p_8^c	P10	t_7^c	t10
p_5^c	P11	t_6^c	t11
p_2^c	P12	t_1^c	t12
p_3^c	P13	t_4^c	t13
p_6^c	P14	t_5^c	t14
p_7^c	P15	t_8^c	t15
p_4^c	P16	t_3^c	t16
p_1^{co}	P17	t_1^l	t17
p_3^{co}	P18	t_2^l	t18
		t_3^l	t19
		t_4^l	t20

Chapter 5: Controller Implementation as Ladder Logic Diagram

To implement the control model in a PLC, it is necessary to transform the NCES model into some type of PLC programming language. This thesis introduces an algorithm that can automatically transform NCES models into Ladder Logic Diagram.

5.1 Transformation Algorithm

Figure 5.1 shows a flow chart that summarizes the steps of the transformation algorithm. The tank control model is used as an example to illustrate the steps of the transformation algorithm. Note: Terms inside of parenthesis in this algorithm description are specific naming conventions for RSLogix 5000 Version 11.

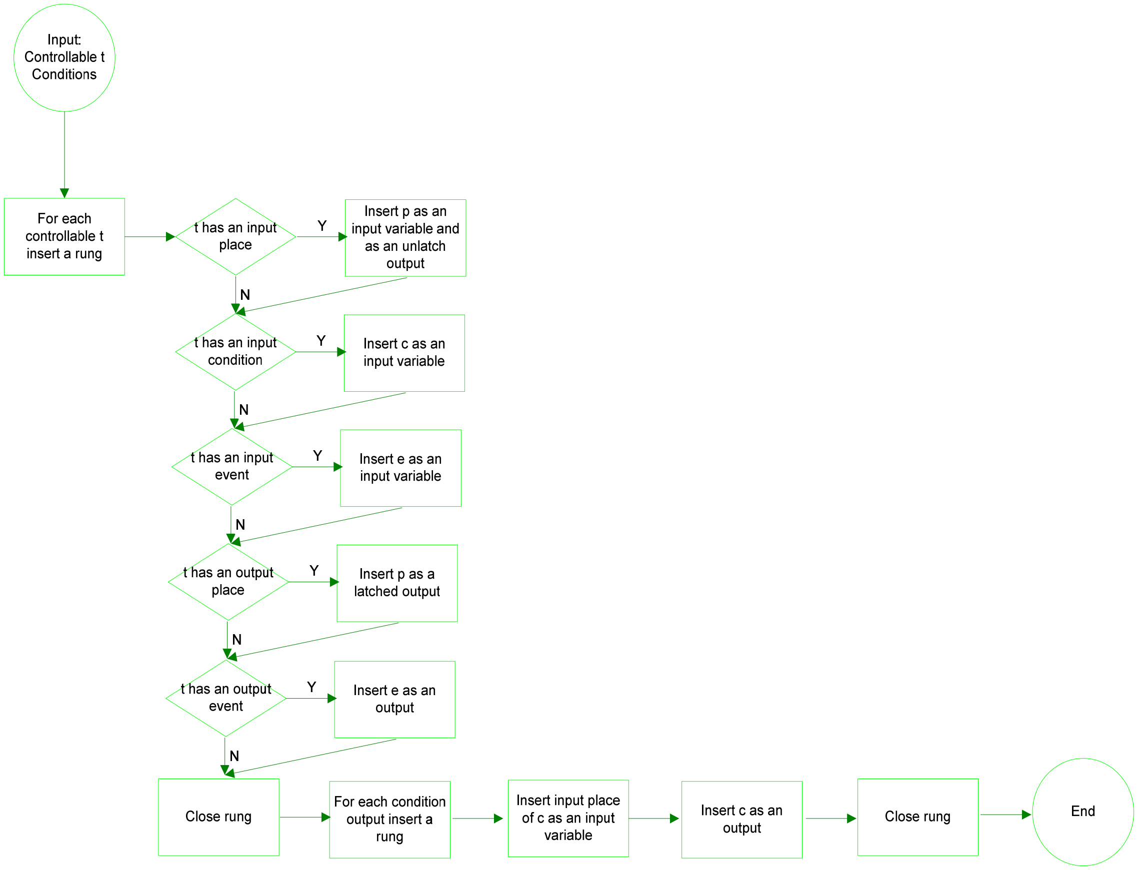


Figure 5.1: Transformation Algorithm Flowchart

5.1.1 Tank Transformation Algorithm

Initialisation :

Given $t \epsilon \mathcal{T}$ and $C \epsilon C_{\mathcal{N}}$ where:

- $\mathcal{T}_c \cap \mathcal{T}_u = \mathcal{T}$
- $\mathcal{T}_c \cap \mathcal{T}_u = \emptyset$
- $C_{in} \cap C_{out} = C_{\mathcal{N}}$
- $C_{in} \cap C_{out} = \emptyset$

$\mathcal{T}_c$ is the set of controllable transitions and $\mathcal{T}_u$ is the set of uncontrollable transitions. Let $\bullet t$ define the set of input places for transition t and $t\bullet$ define the set of output places for transition t. $\mathcal{C}_{in}$ is the set of condition inputs and $\mathcal{C}_{out}$ is the set of condition outputs. Let $\bullet c$ define the set of input places for condition c and $c \bullet$ define the set of output transitions for condition c.

In the tank filling and draining model:

$\mathcal{T}$ = {t1, t2, t3, t4, t5, t6, t7, t8, t_1^c, t_2^c, t_3^c, t_4^c, t_1^l, t_2^l, t_3^l, t_4^l}.

$\mathcal{T}_c$ = {t1, t2, t3, t4, t_1^c, t_2^c, t_3^c, t_4^c, t_1^l, t_2^l, t_3^l, t_4^l}

$\mathcal{T}_u$ = {t5, t6, t7, t8}

$\mathcal{C}_{\mathcal{N}}$ = {C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}}

$\mathcal{C}_{in}$ = {C_1^{in}, C_2^{in}, C_3^{in}, C_4^{in}, C_5^{in}, C_6^{in}, C_7^{in}, C_8^{in}, C_9^{in}, C_{10}^{in}}

$\mathcal{C}_{out}$ = {C_1^{out}, C_2^{out}, C_3^{out}, C_4^{out}, C_5^{out}, C_6^{out}, C_7^{out}, C_8^{out}, C_9^{out}, C_{10}^{out}}

Steps 1.1 through 1.5 transform the places, conditions, and events that interact with transition t1. The tank control model is shown in Figure 3.6 Chapter 3. For convenience purposes, the portion of the tank control model that concerns t1 is shown in Figure 5.2.

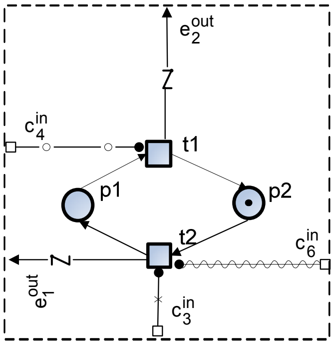


Figure 5.2: Portion of Tank Control Model Pertaining to Transition t1

Step 1: $\forall\ t^{\bullet} \in \mathcal{T}_c$ insert a rung into the Ladder Logic Diagram and:

Step 1.1: $\forall p \in {}^{\bullet}t$ insert p into the rung as an input variable (examine on) and also as an unlatched output (output unlatch). As illustrated in Figure 5.2, p1 is an input place for t1. p1 is inserted into the rung as an input variable and as an unlatched output as shown in Figure 5.3.

Figure 5.3: Tank Control Model Insert Input Place for t1 in LLD

Step 1.2: $\forall\ c \in C_{in}$ insert c into the rung as an input variable. As illustrated in Figure 5.2, C_4 is a condition input for t1. C_4 is inserted into the rung as an input variable as shown in Figure 5.4.

Figure 5.4: Tank Control Model Insert Condition Input for t1 in LLD

Step 1.3: $\forall\ e \in \mathcal{E}_{in}$ insert e into the rung as an input variable. As illustrated in Figure 5.2, t1 has no event inputs. Therefore, no instructions are added to the rung.

Step 1.4: $\forall p \in t^{\bullet}$ insert p into the rung as a latched output (output latch). As illustrated in Figure 5.2, p2 is an output place for t1. p2 is inserted into the rung as a latched output as shown in Figure 5.5.

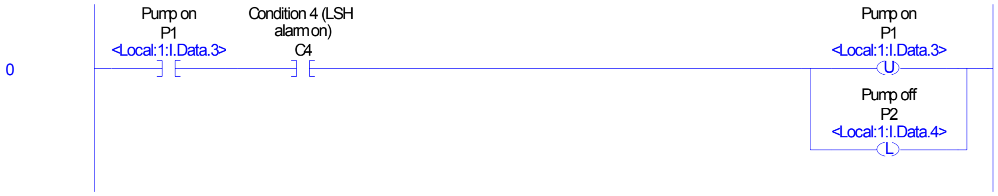


Figure 5.5: Tank Control Model Insert Output Place for t1 in LLD

Step 1.5: $\forall\ e \in \mathcal{E}_{out}$ insert e into the rung as an output (output energize) and end the rung. As illustrated in Figure 5.2, e_2^{out} is an event output for t1. e_2^{out} is inserted into the rung as an output as shown in Figure 5.6.

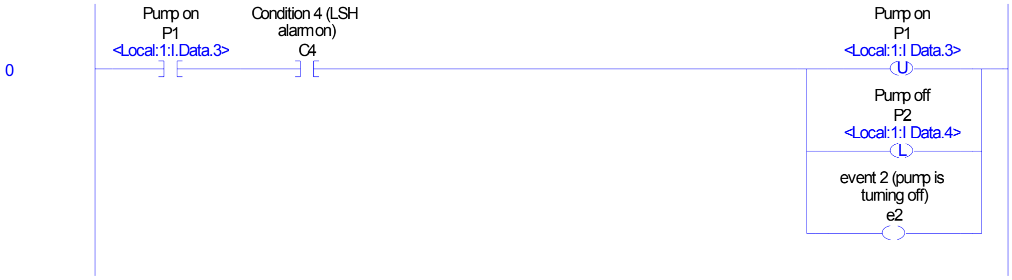


Figure 5.6: Tank Control Model Event Output for t1 in LLD

Steps 2.1 and 2.2 transform the places that interact with condition C_1. For convenience purposes, the portion of the tank control model that concerns C_1 is shown in Figure 5.7.

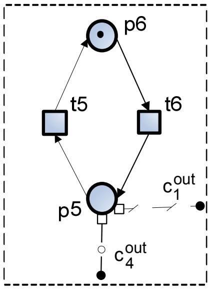


Figure 5.7: Portion of Tank Control Model Pertaining to Condition C_1

Step 2: $\forall\ c \,\epsilon\, C_{out}$, insert a rung into the Ladder Logic Diagram and:

Step 2.1: $\forall p \,\epsilon\, c$ • insert p in the rung as an input variable. As illustrated in Figure 5.7, p5 is an input place for C_1. p5 is inserted into the rung as an input variable as shown in Figure 5.8.

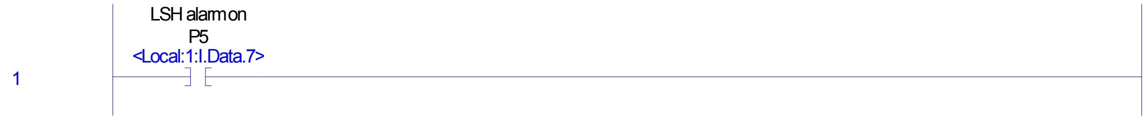


Figure 5.8: Tank Control Model Insert Input Place for C_1 in LLD

Step 2.2: Insert c in the rung as an output variable and end the rung. C_1 is inserted as an output variable into the same rung of step 2.1 as shown in Figure 5.9.

Figure 5.9: Tank Control Model Insert Output C_1 in LLD

Figure 5.7 shows two conditions coming out of p5. Step 2 is repeated to add C4 in the Ladder Logic Diagram.

Step 2: $\forall\ c \,\epsilon\, C_{out}$, insert a rung into the Ladder Logic Diagram and:

Step 2.1: $\forall\ p \,\epsilon\, c$ • insert p in the rung as an input variable. As illustrated in Figure 5.7, p5 is an input place for C_4. p5 is inserted into the rung as an input variable as shown in Figure 5.10.

Figure 5.10: Tank Control Model Insert Input Place for C_4 in LLD

Step 2.2: Insert c in the rung as an output variable and end the rung. C_4 is inserted as an output variable into the same rung of step 2.1 as shown in Figure 5.11.

Figure 5.11: Tank Control Model Insert Output C_4 in LLD

Appendix B includes more examples for the transformation algorithm. The resulting Ladder Logic Diagram is shown in Figure 5.12.

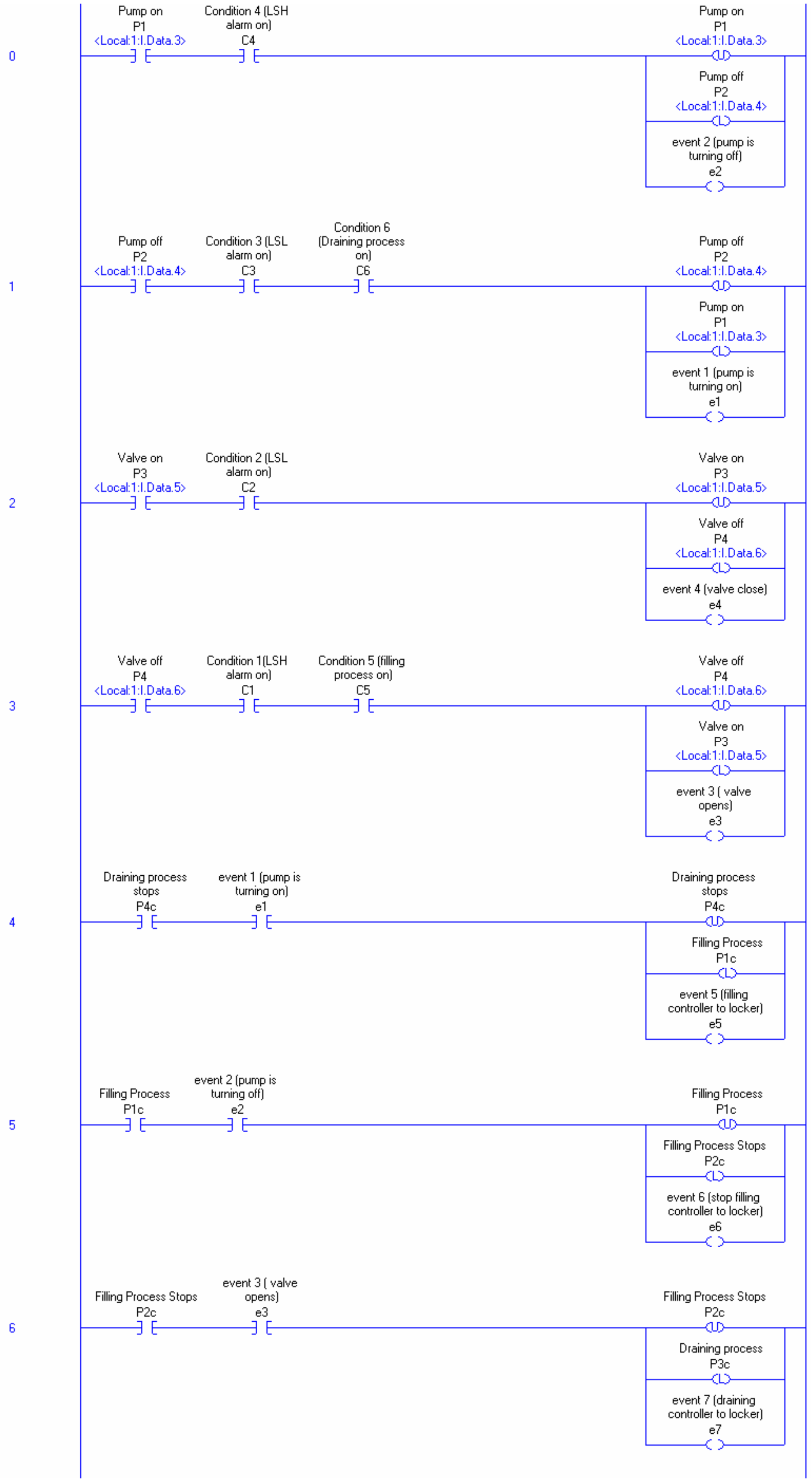


Figure 5.12: Ladder Logic Diagram Tank Control Model

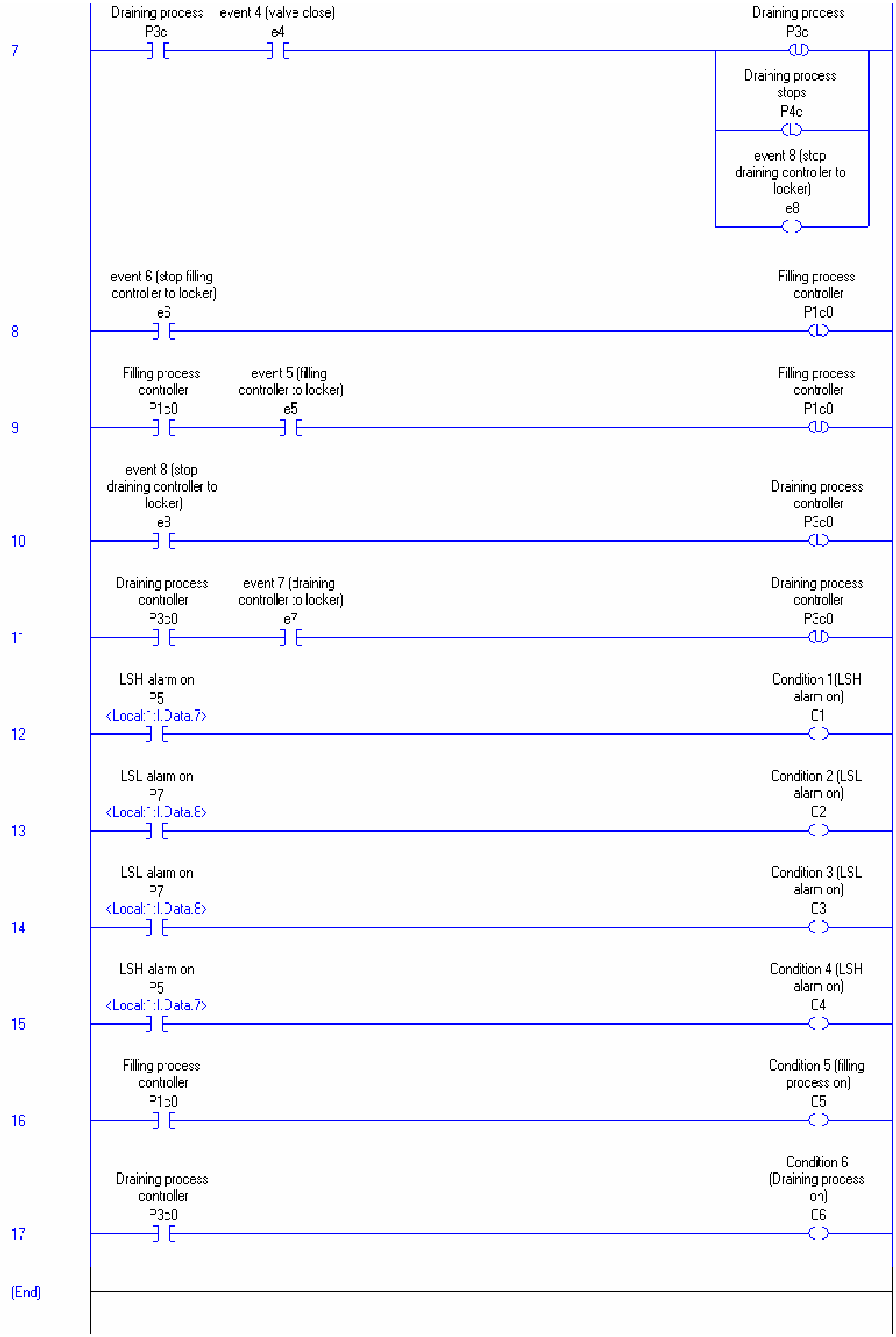
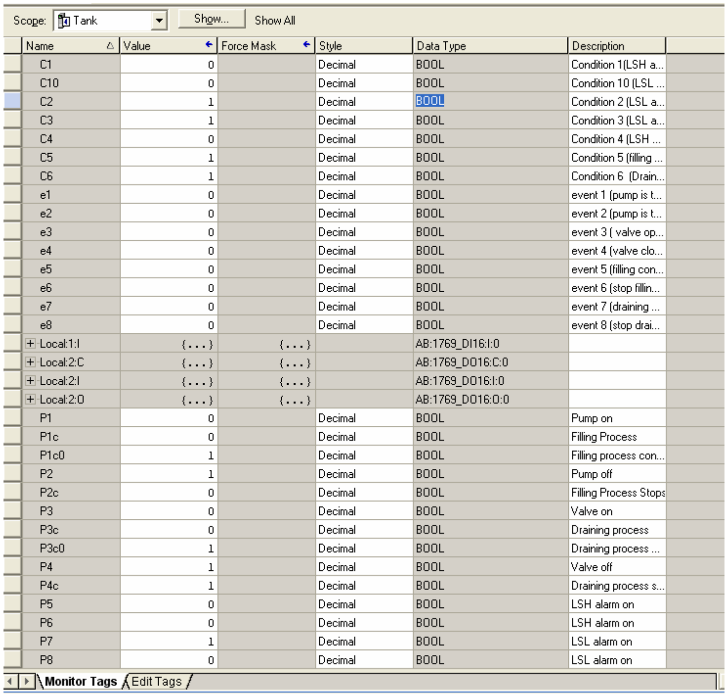


Figure 5.12: (Continued)

5.2 Algorithm Implementation

The Ladder Logic Diagram shown in Figure 3.23a and 3.23b was created using RSLogix 5000 version 11 and downloaded into an Allen Bradley CompactLogix system model 1769 L30 [37]. The Ladder Logic Diagram was verified and no errors were found. The initial conditions of the Tank filling and draining process were simulated by toggling the bits of the input variables. For example, one of the initial conditions is that the pump is off. Therefore, the bit for p2 is toggled to 1. After the bits were toggled the Ladder Logic Diagram was sent online and the tag values were monitor through an RSLogix interface named “monitor tags” as shown in Figure 5.13. The input variables for the level sensors were toggled in the sequence they should logically change states to simulate their response to the physical system. The Ladder Logic Diagram responses to the toggled bits were correct.



Name	Value	Force Mask	Style	Data Type	Description
C1	0		Decimal	BOOL	Condition 1(LSH a...
C10	0		Decimal	BOOL	Condition 10 (LSL ...
C2	1		Decimal	BOOL	Condition 2 (LSL a...
C3	1		Decimal	BOOL	Condition 3 (LSL a...
C4	0		Decimal	BOOL	Condition 4 (LSH ...
C5	1		Decimal	BOOL	Condition 5 (filling ...
C6	1		Decimal	BOOL	Condition 6 (Drain...
e1	0		Decimal	BOOL	event 1 (pump is t...
e2	0		Decimal	BOOL	event 2 (pump is t...
e3	0		Decimal	BOOL	event 3 (valve op...
e4	0		Decimal	BOOL	event 4 (valve clo...
e5	0		Decimal	BOOL	event 5 (filling con...
e6	0		Decimal	BOOL	event 6 (stop fillin...
e7	0		Decimal	BOOL	event 7 (draining ...
e8	0		Decimal	BOOL	event 8 (stop drai...
+ Local:1:I	{...}	{...}		AB:1769_DI16:I:0	
+ Local:2:C	{...}	{...}		AB:1769_DO16:C:0	
+ Local:2:I	{...}	{...}		AB:1769_DO16:I:0	
+ Local:2:O	{...}	{...}		AB:1769_DO16:O:0	
P1	0		Decimal	BOOL	Pump on
P1c	0		Decimal	BOOL	Filling Process
P1c0	1		Decimal	BOOL	Filling process con...
P2	1		Decimal	BOOL	Pump off
P2c	0		Decimal	BOOL	Filling Process Stops
P3	0		Decimal	BOOL	Valve on
P3c	0		Decimal	BOOL	Draining process
P3c0	1		Decimal	BOOL	Draining process ...
P4	1		Decimal	BOOL	Valve off
P4c	1		Decimal	BOOL	Draining process s...
P5	0		Decimal	BOOL	LSH alarm on
P6	0		Decimal	BOOL	LSH alarm on
P7	1		Decimal	BOOL	LSL alarm on
P8	0		Decimal	BOOL	LSL alarm on

Figure 5.13: RSLogix Monitor Tags Interface for the Tank Control Model

Chapter 6: Supervisory Control Synthesis for the HAS-200

The HAS-200(Highly Automated System) is an automated manufacturing system designed for education and training by SMC Corporation, with support from Intel Corporation and in Partnership with the Maricopa Advanced Technology Education Center (MATEC) [38]. The HAS-200 simulates an automated manufacturing process by modeling key work-in-progress stages at 10 hands-on stations. The purpose of the system is to provide a realistic, hands-on experience in automated manufacturing for high school and college students.

6.1 HAS-200 System Overview

The HAS-200 manufactures a range of different products, which are composed of plastic boxes filled with various amounts of colored plastic beads. The final products consist of boxes filled with one color bead or any combination of blue, yellow, and/or red in total amounts of 15, 30 or 45 grams as shown in Figure 6.1. Through various combinations of color and weight, the HAS-200 can manufacture up to 19 different products. Each box also includes a bar code label, which makes it traceable throughout the system and identifies the product color or color combination.

Figure 6.1: HAS-200 Products

In simulating a modern highly automated factory the HAS 200 utilizes the first 4 levels of the Automation Pyramid as shown in Figure 6.2. This means that each station knows its requirements (i.e. color, weight and quantity) and communicates effectively with all other stations, making the entire system constantly aware of its work in progress. The levels of the automation pyramid are explained below.

- Levels I & II*:* These levels represent the equipment used in production (PLC, PC, sensors, etc.) The work on this thesis will be focused on these two levels.
- Levels III & IV: These levels represent the system process operation. Supervisory Control and Data Acquisition (SCADA) provides the infrastructure to control a process and to collect real-time data for automated manufacturing that are running in different physical locations. The primary purpose of a SCADA system is to gather data from controllers, and field devices then display this data on the computer screen. Manufacturing Execution System (MES) is software

that automatically links together the functions of the systems such as resource management, scheduling, maintenance management, statistical quality control, and data collection. The use of MES software provides many benefits. Examples of these benefits include reduced lead time for orders, efficient planning of resources and equipment, improved product quality, reduced manufacturing cycle time, and lower manufacturing costs. All of these benefits contribute to an increased in the manufacturing productivity.

- Level V: This level represents the business part of the system. Enterprise Resource Planning (ERP) automatically ties together all basic functions of a business including those not related to manufacturing. Some of the basic functions are human resources, finance, sales, marketing, accounting, and logistics.

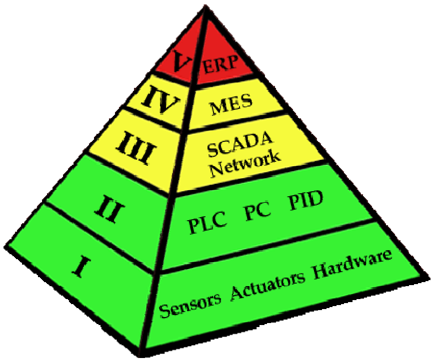


Figure 6.2: Automation Pyramid

The HAS-200 consists of 10 stations. The stations are connected through a conveyor belt that serves as a material handling device for the boxes as shown in Figure 6.3. In addition, each station has a signal tree, a control panel, bar code reader, actuators, sensors and Allen-Bradley CompactLogix PLCs. The stations may be operated locally from a pushbutton control panel at the station, or the entire system can be run in fully automated mode using EdMES (Manufacturing Execution System) software. The EdMES software allows the user to place one or more orders, each order containing different product types and quantities. The EdMES software then coordinates the manufacturing process and directs the stations to manufacture the products. The description of each of the stations operation is as follows:

- Station1 (Multicolor Filling Station): This station supplies empty multicolor-labeled boxes that get filled at production stations 2, 3, and 4.
- Station 2, 3 and 4 (Single-Color Filling Station): These production stations supply single-color boxes to the system, fill boxes with the plastic beads, weigh filled boxes, and place them onto the conveyor belt. They can also fill boxes supplied by the Multicolor Box Feeder station.
- Station 5 and 6 (Metrology Station): These stations measure the height of the material contained in the boxes and compare it with the weight of the order. If the height does not match the specified weight, the box is rejected. The only difference between these two stations is that Station 5 uses a digital encoder to determine the height of the content, while Station 6 takes measurements with an analog linear

potentiometer. Each station has a buffer belt running parallel to the main conveyor belt to decrease the probability of long waiting queues.

- Station 7 (Cap and Label Station): This station inserts plastic caps on the boxes. It also attaches a label on top of the cap.
- Station 8 (Vertical Storage Station) and 9 (Horizontal Storage Station): These stations store both completed boxes ready to be dispatched and “work in progress” boxes waiting to be put back into circulation (such as empty or partially filled boxes). Both stations have a human machine interface (HMI) that can be used to control container movement within in the station. The Vertical Storage can store up to 81 boxes. The Horizontal Storage can store up to 56 boxes.
- Station 10 (Dispatcher Station): This station removes boxes from the conveyor belt and loads them onto a two-part platform. When the platform reaches a maximum of four boxes, the boxes are released onto a ramp and out of the factory.

Figure 6.3: HAS-200 System

6.2 HAS-200 Control Problem Description

The physical layout of the HAS-200 stations is shown in Figure 6.4. From Figure 6.4, one can observe that the typical process for a multicolor product consists of the following:

- A box is provided to the system (Station 1).
- The box is filled (Stations 2, 3, & 4) with the specific product color and quantity required by the “customer” order.
- The box contents are checked and compared to the “customer” order (Station 5 or 6).
- A cap and label are placed on top of the box (Station 7).
- The box is either stored (Station 8 or 9) or dispatched (Station 10).

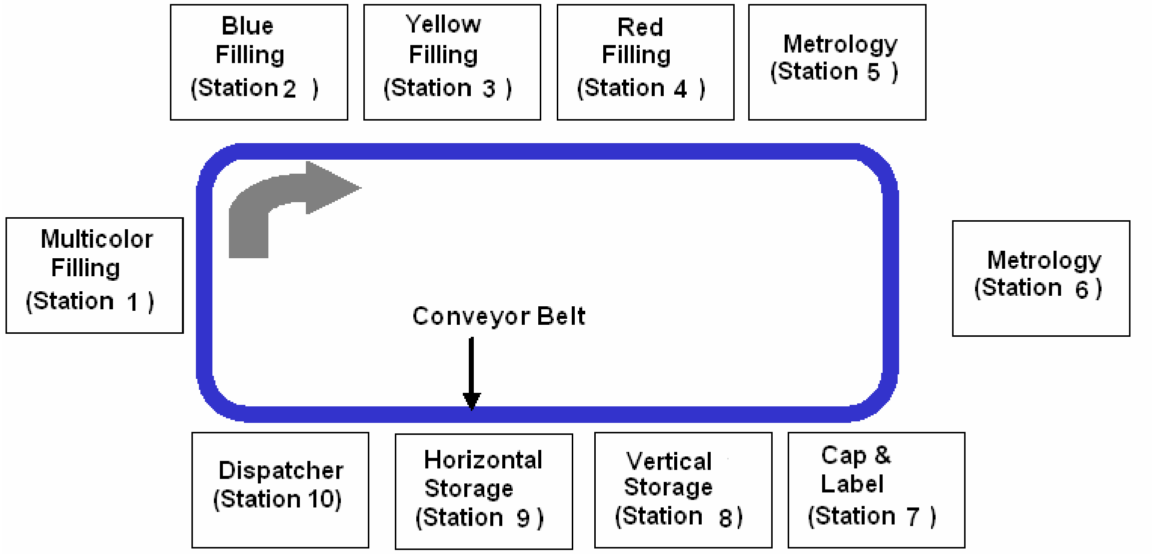


Figure 6.4: HAS-200 Physical Layout

Notice that the conveyor belt moves only in one direction. Therefore, the current multicolor product follows a single filling sequence of blue-yellow-red. The control problem considered in this chapter allows the manufacture the addition of five new set of multicolor products that follow other filling sequences such as blue-red-yellow, yellow-blue-red, yellow-red-blue, red-blue-yellow, and red-yellow-blue. The addition of these filling sequences requires that a box will need to go around the system more than once to be completed as shown in Figure 6.5.

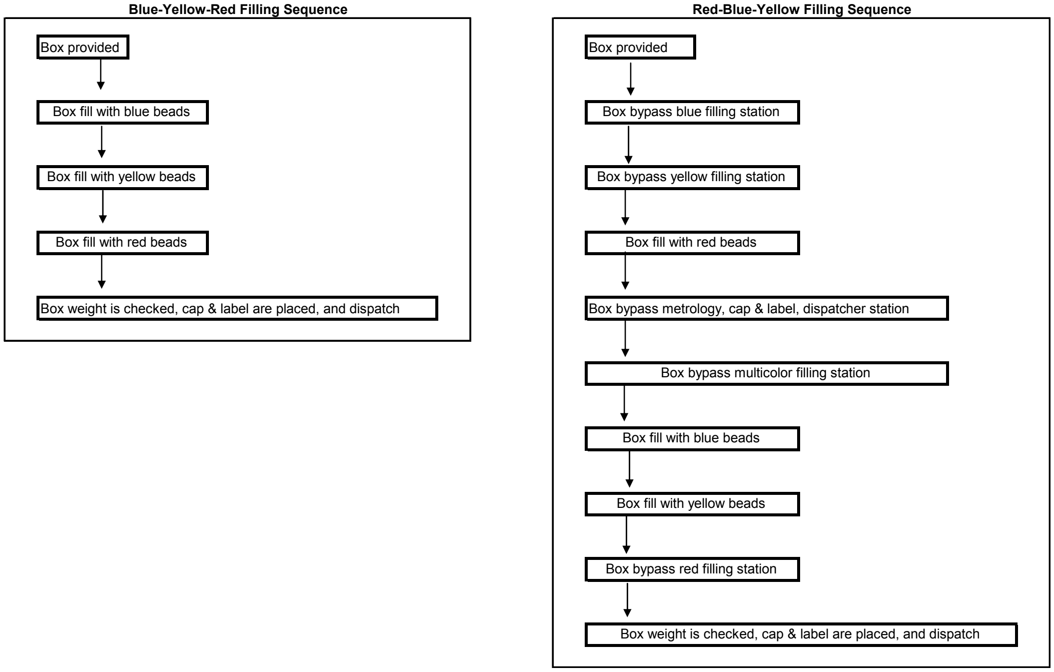


Figure 6.5: HAS-200 Filling Sequence

Development of a supervisory controller model is divided into three main tasks as discussed in Chapter 3: modeling of the uncontrolled system and specifications, controller synthesis and controller implementation. In the rest of this chapter, these tasks will be addressed to develop the new supervisory control model for the HAS-200.

The first task is to develop an uncontrolled model of the system. The uncontrolled model will only include those devices and operations that are relevant to the filling sequences. For example, a box entering a station, a box is processed by a station, a box bypasses a station, or a box leaves a station. The detail steps on how a box gets processed in a station are not pertinent to the filling sequence process. Therefore, these operations are not part of the uncontrolled model. A description of the devices and operations that will be part of the uncontrolled model is given as follows:

- Station 1: Once an order is placed in the system, station 1 will start processing the order. Stopper 1 which is normally retracted will extend to ensure that no box enters into the station. When the process is done, the box is placed on the conveyor belt and the stopper returns to its original position. The conveyor belt will take the box to the next station. If a box arrives to station 1 and there are no other boxes being processed by station 1, the box will continue moving on the conveyor belt until it reaches the next station. If a box arrives to Station 1 and there is a box being processed by Station 1, the box will be stopped by stopper 1 until the process ends. Figure 6.6 shows the physical layout of Station 1. Notice that there are other devices that are part of station1, but are not taken into consideration because these devices are not part of the filling sequence.

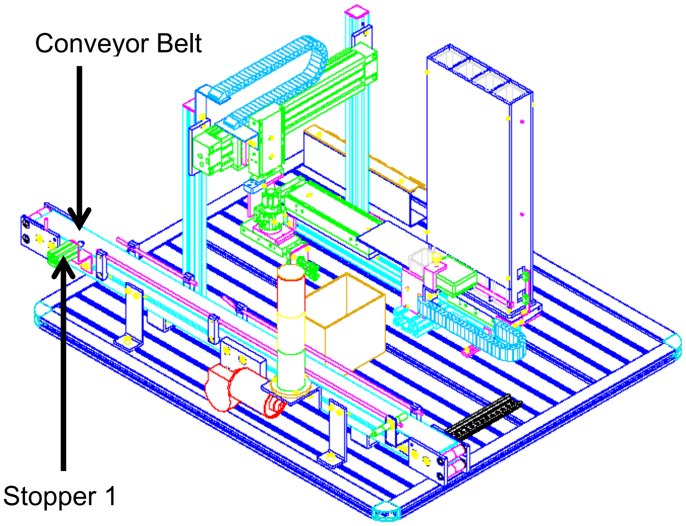


Figure 6.6: HAS-200 Station 1 Layout

– Stations 2 - 4: These three stations perform the same process and have the same devices except that the bead colors are different. Station 2 will be used as the example. Stopper 1 is a dual pin stopper located at the beginning of the station. Stopper 1 consists of 2 pins: Pin 1 is normally retracted and pin 2 is normally extended. If a box is being processed in Station 2, pin 1 is extended and pin 2 is retracted. If a box enters Station 2 and reaches pin 1, this means that a box is being processed in Station 2. The box will have to wait until the process is done and pin 1 retracts. If a box enters Station 2 and reaches pin 2, sensor 1 activates detecting the presence of the box. Then, pin 2 will retract to allow the box inside the station and pin 1 will extend to allow only one box to be processed at a time. The pins will remain in those positions until the box leaves the station. The conveyor belt will move the box until it reaches stopper 2. Stopper 2 is normally down and it holds the box in place so that the bar code reader is able to read the label on the box. If the bar code matches the bar codes in the processing list, the box will be removed from the conveyor belt and processed in Station 2. When the processing is done, the box is placed back on the conveyor belt. Stopper 2 will move up to allow the box to leave Station 2. Stopper 2 and the pins of stopper 1 will return to its original position. The conveyor belt will take the box to the next station. If the bar code does not match the bar codes in the processing list, stopper 2 will move up to allow the box to leave Station 2. Stopper 2 and the pins of stopper 1 will return to its original position. The conveyor belt will take the box to the next station. Sensor 2 will activate if there is a long queue coming from Station 3. If sensor 2 is active, pin 1 will remain extended and no other boxes can be processed or enter Station 2. Figure 6.7 shows the physical layout of stations 2, 3, and 4. For the uncontrolled model, stopper 1 will refer to the actions performed by pin 2. Pin 1 will be omitted from the model, since its actions are opposite of pin 2.

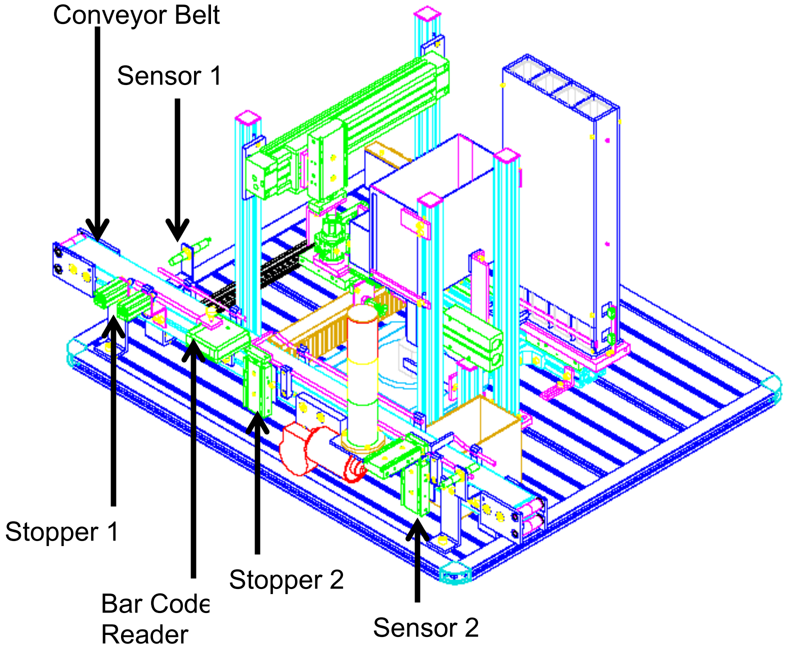


Figure 6.7: HAS-200 Stations 2, 3, and 4 Layout

- Stations 5-10: Although these stations do not perform the same operations, their process will be modeled together. If a box has been filled by stations 2, 3, and 4; then the box will be checked by Stations 5 or 6, processed by Station 7, bypassed Station 8 and 9, and dispatched by Station 10. All these operations will be referred to as processed by Station 5 through 10. If a box has not been filled by station 2, 3, or 4; then it will bypass stations 5 through 10. The physical layout of stations 5 through 10 is shown in Appendix C.

6.3 HAS-200 Uncontrolled Model

Station 1 uncontrolled model is shown in Figure 6.8. Table 6.1 gives a brief description of the places and transitions for Station 1 uncontrolled model.

Table 6.2 gives a brief description of the events and conditions for the model. Notice that if multiple orders of multicolor products are placed at the same time, p7 will have more than one token at a time. This is an undesirable behavior for Station 1. Therefore, a specification for Station 1 is that only one box can be processed at a time in Station 1. For the sake of simplicity and to be able to follow the control model synthesis, only the yellow-red-blue (YRB) sequence uncontrolled, specification and control model will be developed. The rest of this section describes the development of the YRB sequence uncontrolled model for each of the stations of the HAS-200.

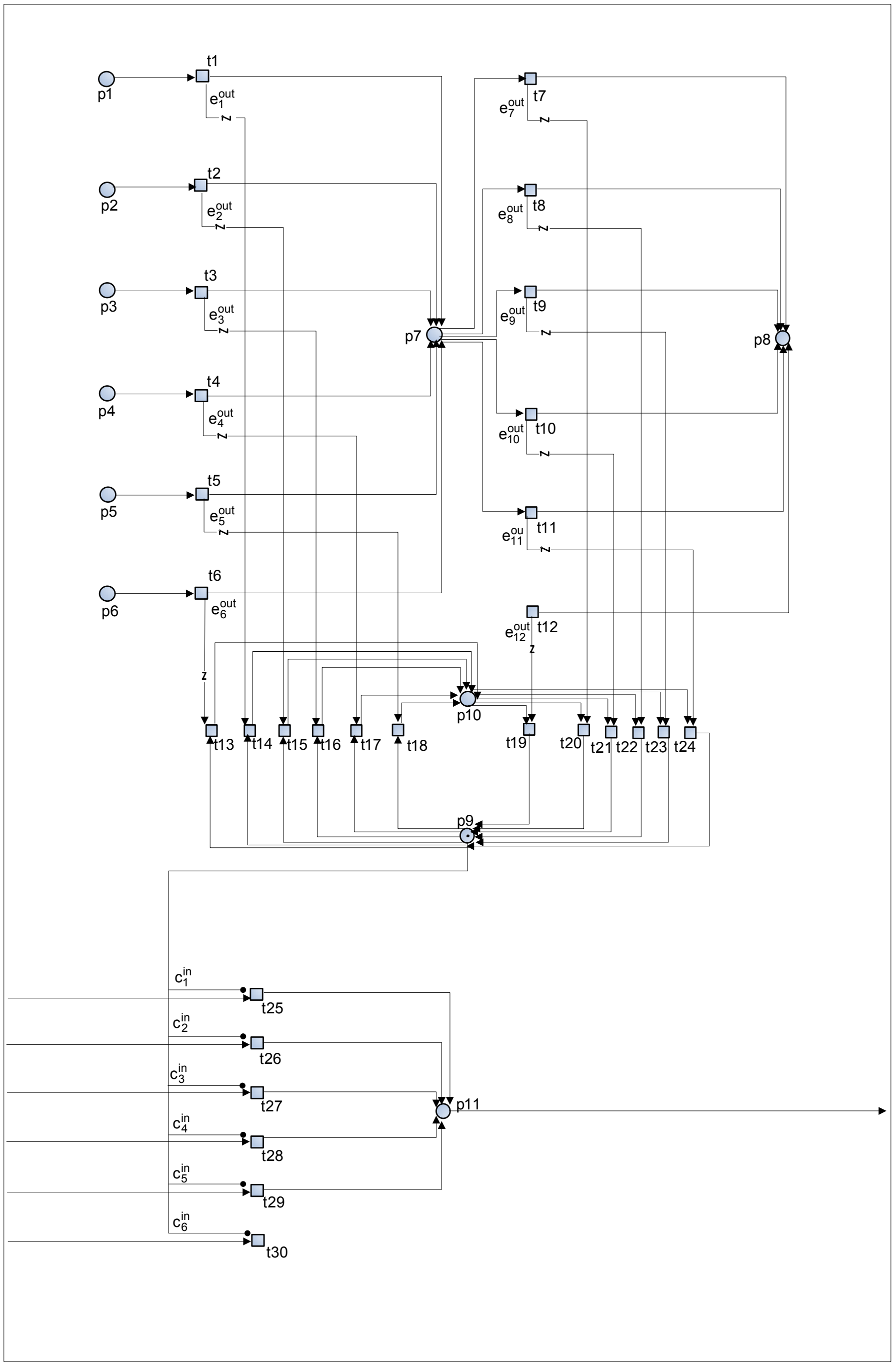


Figure 6.8: Station 1 Uncontrolled Model

Table 6.1: Places and Transitions for Station 1 Uncontrolled Model

Place	Meaning	Transition	Meaning
p1	YRB order	t1	Station 1 start processing order YRB
p2	YBR order	t2	Station 1 start processing order YBR
p3	RBY order	t3	Station 1 start processing order RBY
p4	RYB order	t4	Station 1 start processing order RYB
p5	BRY order	t5	Station 1 start processing order BRY
p6	BYR order	t6	Station 1 start processing order BYR
p7	Box process in station 1	t7	Station 1 done processing box
p8	Box exit station 1	t8	Station 1 done processing box
p9	Station 1 stopper extended	t9	Station 1 done processing box
p10	Station 1 stopper retracted	t10	Station 1 done processing box
p11	Box bypass station 1	t11	Station 1 done processing box
		t12	Station 1 done processing box
		t13	station 1 Stopper extending
		t14	station 1 Stopper extending
		t15	station 1 Stopper extending
		t16	station 1 Stopper extending
		t17	station 1 Stopper extending
		t18	station 1 Stopper extending
		t19	station 1 Stopper retracting
		t20	station 1 Stopper retracting
		t21	station 1 Stopper retracting
		t22	station 1 Stopper retracting
		t23	station 1 Stopper retracting
		t24	station 1 Stopper retracting
		t25	Box moving from station 10 to station 1
		t26	Box moving from station 10 to station 1
		t27	Box moving from station 10 to station 1
		t28	Box moving from station 10 to station 1
		t29	Box moving from station 10 to station 1
		t30	Box moving from station 10 to station 1

Table 6.2: Condition and Events for Station 1 Uncontrolled Model

Condition	Meaning	Events	Meaning
$C_1(C_1^{in}, C_1^{out})$	Station 1 stopper retracted	$e_1(e_1^{in}, e_1^{out})$	Station 1 start processing
$C_2(C_2^{in}, C_2^{out})$	Station 1 stopper retracted	$e_2(e_2^{in}, e_2^{out})$	Station 1 start processing
$C_3(C_3^{in}, C_3^{out})$	Station 1 stopper retracted	$e_3(e_3^{in}, e_3^{out})$	Station 1 start processing
$C_4(C_4^{in}, C_4^{out})$	Station 1 stopper retracted	$e_4(e_4^{in}, e_4^{out})$	Station 1 start processing
$C_5(C_5^{in}, C_5^{out})$	Station 1 stopper retracted	$e_5(e_5^{in}, e_5^{out})$	Station 1 start processing
$C_6(C_6^{in}, C_6^{out})$	Station 1 stopper retracted	$e_6(e_6^{in}, e_6^{out})$	Station 1 start processing
		$e_7(e_7^{in}, e_7^{out})$	Station 1 done processing
		$e_8(e_8^{in}, e_8^{out})$	Station 1 done processing
		$e_9(e_9^{in}, e_9^{out})$	Station 1 done processing
		$e_{10}(e_{10}^{in}, e_{10}^{out})$	Station 1 done processing
		$e_{11}(e_{11}^{in}, e_{11}^{out})$	Station 1 done processing
		$e_{12}(e_{12}^{in}, e_{12}^{out})$	Station 1 done processing

6.3.1 Station 1 Uncontrolled Model

The Station 1 uncontrolled model for the YRB sequence is shown in Figure 6.9. Table 6.3 gives a brief description of the places, events, transitions, and conditions for the uncontrolled model. Notice that if multiple orders of multicolor products are placed at the same time, p2 will have more than one token at a time. This is an undesirable behavior for station 1. Therefore, the specification must force Station 1 to process (provide) only one container at a time. Also, observe that during the processing in Station 1, the stopper will remain extended. As a result, the stopper will not permit any container to bypass while Station 1 is processing.

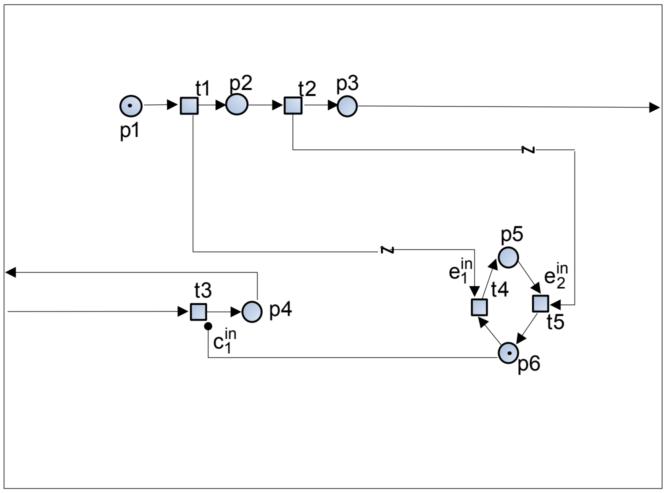


Figure 6.9: Station 1 YRB Uncontrolled Model

Table 6.3: Places, Transitions, Conditions, and Events for Station 1 Uncontrolled Model

Place	Meaning	Transition	Meaning
p1	YRB order	t1	Station 1 start processing order YRB
p2	Box process in station 1	t2	Station 1 done processing box
p3	Box exit station 1	t3	Box moving from station 10 to station 1
p4	Box bypass station 1	t4	station 1 Stopper extending
p5	Station 1 stopper extended	t5	station 1 Stopper retracting
p6	Station 1 stopper retracted		
Condition	Meaning	Events	Meaning
$C_1(C_1^{in}, C_1^{out})$	Station 1 stopper retracted	$e_1(e_1^{in}, e_1^{out})$	Station 1 start processing
		$e_2(e_2^{in}, e_2^{out})$	Station 1 done processing

6.3.2 Station 2 Uncontrolled Model

The Station 2 uncontrolled model for the YRB sequence is shown in Figure 6.10. Table 6.4 gives a brief description of the places, events, transitions, and conditions for the uncontrolled model. The uncontrolled model for stations 3 and 4 are similar to the uncontrolled model of Station 2. Notice that if a box is in front of the bar code reader (p17, Figure 6.10), the box can either be bypassed or processed by station 2. The decision to bypass or process a box will be determined by the sequence specification. Also, observe that once a box enters station 2, stopper1 will extend. Stopper1 will remain extended until the box leaves the station. The Stopper will not allow any box to enter while station 2 is processing or bypassing. The set of events (e_6^{in} and e_{10}^{in}); (e_7^{in} and e_{11}^{in}) will fire when the corresponding transition (box moving from station 2 to station 3) fires. This concept will be shown more clearly in the control model Figure 6.14.

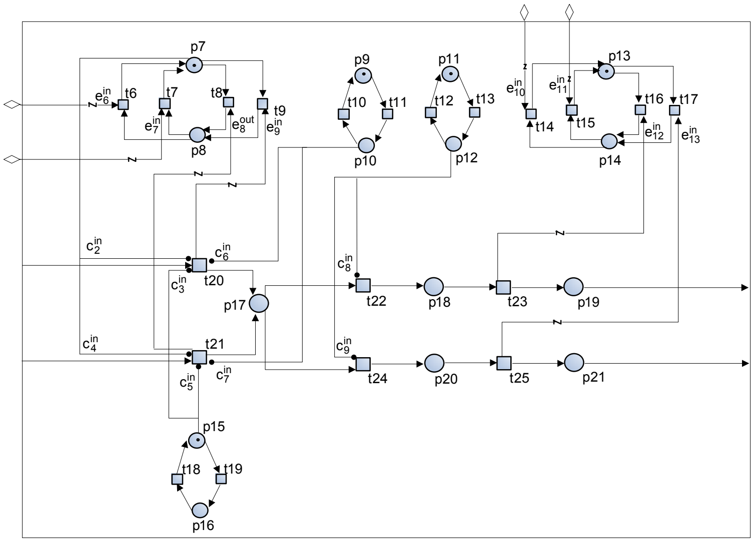


Figure 6.10: Station 2 YRB Uncontrolled Model

Table 6.4: Places, Transitions, Conditions, and Events for Station 2 Uncontrolled Model

Place	Meaning	Transition	Meaning
p7	Station 2 stopper 1 retracted	t6	station 2 Stopper 1 retracting
p8	Station 2 stopper 1 extended	t7	station 2 Stopper 1 retracting
p9	Station 2 sensor 1 passive	t8	station 2 Stopper 1 extending
p10	Station 2 sensor 1 active	t9	Station 2 Stopper 1 extending
p11	Station 2 BCR passive	t10	Station 2 sensor 1 turns passive
p12	Station 2 BCR active	t11	Station 2 sensor 1 turns active
p13	Station 2 Stopper 2 down	t12	Station 2 BCR goes active
p14	Station 2 Stopper 2 up	t13	Station 2 BCR goes passive
p15	Station 2 sensor 2 passive	t14	Station 2 Stopper 2 goes down
p16	Station 2 sensor 2 active	t15	Station 2 Stopper 2 goes down
p17	Box in front of station 2 BCR	t16	Station 2 Stopper 2 goes up
p18	Box process in station 2	t17	Station 2 Stopper 2 goes up
p19	Box exit station 2	t18	Station 2 sensor 2 turns passive
p20	Box bypass station 2	t19	Station 2 sensor 2 turns active
p21	Box exit station 2	t20	Box moving from station 1 to station 2
		t21	Box moving from station 1 to station 2
		t22	Station 2 start processing
		t23	Station 2 done processing box
		t24	Station 2 start bypassing
		t25	Station 2 done bypassing
Condition	Meaning	Events	Meaning
$C_2(C_2^{in}, C_2^{out})$	Station 2 stopper 1 retracted	$e_6(e_6^{in}, e_6^{out})$	Box exit Station 2
$C_3(C_3^{in}, C_3^{out})$	Station 2 sensor 2 passive	$e_7(e_7^{in}, e_7^{out})$	Box exit Station 2
$C_4(C_4^{in}, C_4^{out})$	Station 2 stopper 1 retracted	$e_8(e_8^{in}, e_8^{out})$	Box enters station 2
$C_5(C_5^{in}, C_5^{out})$	Station 2 sensor 2 passive	$e_9(e_9^{in}, e_9^{out})$	Box enters station 2
$C_6(C_6^{in}, C_6^{out})$	Station 2 sensor 1 active	$e_{10}(e_{10}^{in}, e_{10}^{out})$	Box exit Station 2
$C_7(C_7^{in}, C_7^{out})$	Station 2 sensor 1 active	$e_{11}(e_{11}^{in}, e_{11}^{out})$	Box exit Station 2
$C_8(C_8^{in}, C_8^{out})$	Station 2 BCR active	$e_{12}(e_{12}^{in}, e_{12}^{out})$	Station 2 done processing
$C_9(C_9^{in}, C_9^{out})$	Station 2 BCR active	$e_{13}(e_{13}^{in}, e_{13}^{out})$	Station 2 done bypassing

6.3.3 Station 5-10 Uncontrolled Model

The Stations 5-10 uncontrolled model for the YRB sequence is shown in Figure 6.11. Table 6.5 gives a brief description of the places, events, transitions, and conditions for the uncontrolled model. Notice that if a box is in front of the bar code reader (p52, Figure 6.11), the box can either be bypassed or processed by station 5-10. The decision to bypass or process a box will be determined by the sequence specification. Also, observe that once a box is processed the token will remain in p54. The set of events (e_{30}^{in} and e_{36}^{in}); (e_{31}^{in} and e_{37}^{in}) will fire the corresponding transitions for Stopper1 to retract and Stopper 2 to go down in Station 4. This concept will be shown more clearly in the control model Figure 6.14.

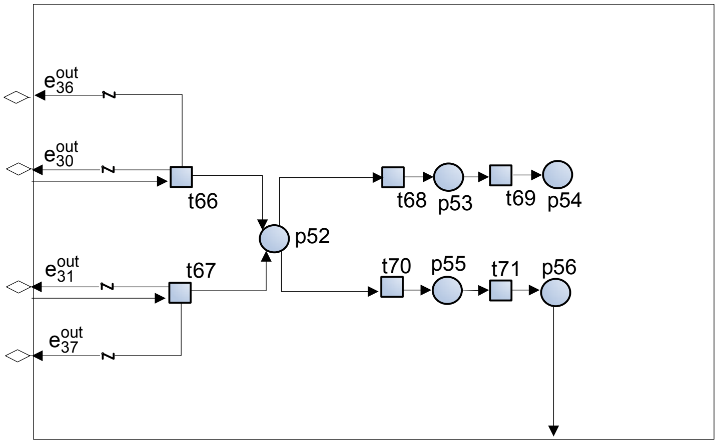


Figure 6.11: Stations 5-10 YRB Uncontrolled Model

Table 6.5: Places, Transitions, Conditions, and Events for Stations 5-10 Uncontrolled Model

Place	Meaning	Transition	Meaning
p52	Box in front of station 2 BCR	t66	Box moving from station 1 to station 2
p53	Box process in station 2	t67	Box moving from station 1 to station 2
p54	Box exit station 2	t68	Station 2 start processing
p55	Box bypass station 2	t69	Station 2 done processing box
p56	Box exit station 2	t70	Station 2 start bypassing
		t71	Station 2 done bypassing
		Events	Meaning
		$e_{30}\ (e_{30}^{in}, e_{30}^{out})$	Box moving from station 4 to station 5
		$e_{31}\ (e_{31}^{in}, e_{31}^{out})$	Box moving from station 4 to station 5
		$e_{36}\ (e_{36}^{in}, e_{36}^{out})$	Box moving from station 4 to station 5
		$e_{37}\ (e_{37}^{in}, e_{37}^{out})$	Box moving from station 4 to station 5

6.4 HAS-200 Controller Synthesis

Figure 6.12 shows the filling sequence logic to complete an YRB order. Figure 6.13 shows the specification and locking controller. The specification developed is based on the filling sequence logic. The first step in Figure 6.12 (box provided) is represented with places p_1^c (order placed), p_2^c (box process) and p_3^c (box leaves) in Figure 6.13. The second step in Figure 6.12 (box bypass blue filling station) is represented with places p_{20}^c (box bypass) p_{21}^c (box leaves) in Figure 6.13. Table 6.6 shows the places and transitions for the specification and the locking controller models. Table 6.7 shows the conditions and events for the specification and locking controller model.

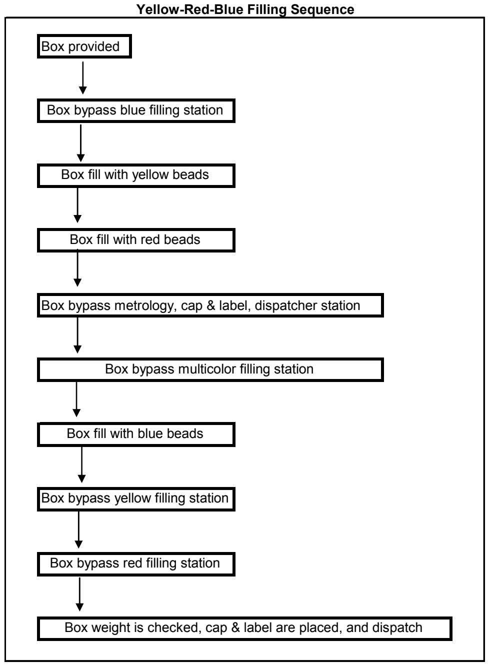


Figure 6.12: YRB Filling Sequence

All stations except for station 1 have two locking controller modules. One module controls the processing and the other module controls the bypassing. For example, If a box needs to be processed in station 2 the specification will remove the token in the bypass locking controller place p_{20}^{co}. No token in place p_{20}^{co} makes the bypassing transition of station 2 disabled. Station 1 has only one locking controller. The locking controller for station 1 ensures that only one box is processed at a time. If a box is being processed in Station 1, the specification will remove the token from the locking controller place p_{2}^{co}. No token in place p_{2}^{co} makes the processing transition of station 2 disabled. Station 1 does not have a bypassing locking controller because; a box does not need to interact with any of the station 1 devices to bypass it. The other stations require that the bar code is read before deciding the action to take (process or bypass). Station 1 will only stop a bypassing box if it is processing another box. As mentioned in section 6.3.1, the uncontrolled model of station 1 does not allow a box to bypass if the station is processing a box. Therefore, this behavior does not need to be modeled by the specification.

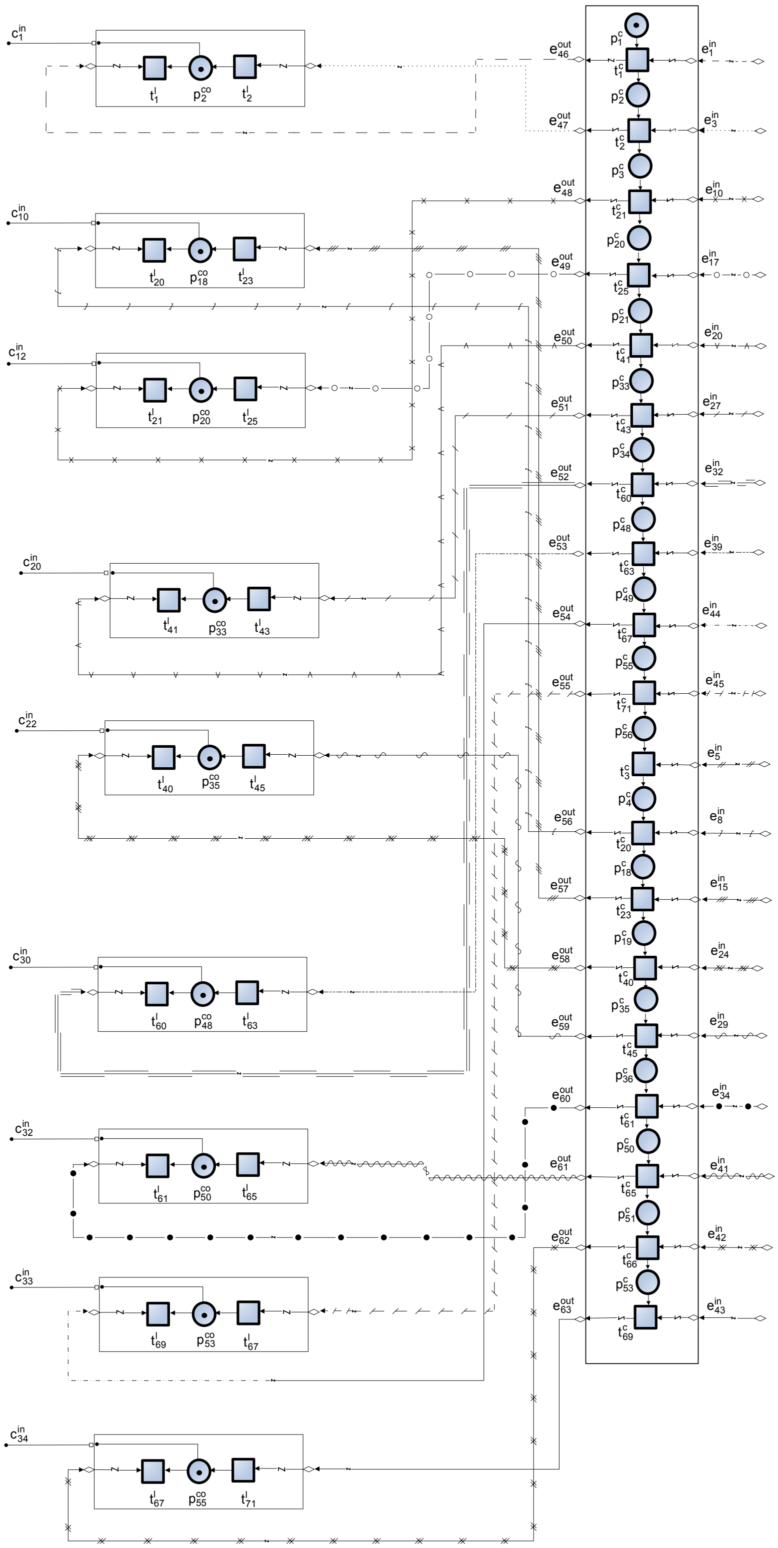


Figure 6.13: YRB Specification and LC Modules

Table 6.6: Places and Transitions for the HAS-200 LC and Specification Model

Place	Meaning	Transition	Meaning
p^{c}_{1}	Order placed	t58	Station 4 sensor 2 turns active
p^{c}_{2}	Box processed Station 1	t59	Station 4 sensor 2 turns passive
p^{c}_{3}	Box Leaves Station 1	t60	Box moving from station 3 to station 4
p^{c}_{20}	Box bypassed Station 2	t61	Box moving from station 3 to station 4
p^{c}_{21}	Box Leaves Station 2	t62	Station 4 start processing
p^{c}_{33}	Box processed Station 3	t63	Station 4 done processing box
p^{c}_{34}	Box Leaves Station 3	t64	Station 4 start bypassing
p^{c}_{48}	Box processed Station 4	t65	Station 4 done bypassing
p^{c}_{49}	Box Leaves Station 4	t66	Box moving from station 4 to station 5
p^{c}_{55}	Box bypassed Station 5-10	t67	Box moving from station 4 to station 5
p^{c}_{56}	Box Leaves Station 10	t68	Station 5-10 start processing
p^{c}_{4}	Box bypassed Station 1	t69	Station 5-10 done processing box
p^{c}_{18}	Box processed Station 2	t70	Station 5-10 start bypassing
p^{c}_{19}	Box Leaves Station 2	t71	Station 5-10 done bypassing
p^{c}_{35}	Box bypassed Station 3	t^{c}_{1}	Station 1 start processing order
p^{c}_{36}	Box Leaves Station 3	t^{c}_{2}	Station 1 done processing box
p^{c}_{50}	Box bypassed Station 4	t^{c}_{21}	Station 2 start bypassing
p^{c}_{51}	Box Leaves Station 4	t^{c}_{25}	Station 2 done bypassing box
p^{c}_{53}	Box processed Station 5-10	t^{c}_{41}	Station 3 start processing
p^{co}_{2}	Co-place of p^{c}_{2}	t^{c}_{43}	Station 3 done processing box
p^{co}_{18}	Co-place of p^{c}_{18}	t^{c}_{60}	Station 4 start processing
p^{co}_{20}	Co-place of p^{c}_{20}	t^{c}_{63}	Station 4 done processing box
p^{co}_{33}	Co-place of p^{c}_{33}	t^{c}_{67}	Station 5-10 start bypassing
p^{co}_{36}	Co-place of p^{c}_{36}	t^{c}_{71}	Station 5-10 done bypassing
p^{co}_{48}	Co-place of p^{c}_{48}	t^{c}_{3}	Station 1 bypassing order
p^{co}_{50}	Co-place of p^{c}_{50}	t^{c}_{20}	Station 2 start processing
p^{co}_{53}	Co-place of p^{c}_{53}	t^{c}_{23}	Station 2 done processing box
p^{co}_{55}	Co-place of p^{c}_{55}	t^{c}_{40}	Station 3 start bypassing
		t^{c}_{45}	Station 3 done bypassing
		t^{c}_{61}	Station 4 start bypassing
		t^{c}_{65}	Station 4 done bypassing
		t^{c}_{66}	Station 5-10 start processing
		t^{c}_{69}	Station 5-10 done processing box
		t^{l}_{1}	Copy of t^{c}_{1}
		t^{l}_{2}	Copy of t^{c}_{2}
		t^{l}_{21}	Copy of t^{c}_{21}
		t^{l}_{25}	Copy of t^{c}_{25}
		t^{l}_{41}	Copy of t^{c}_{41}
		t^{l}_{43}	Copy of t^{c}_{43}
		t^{l}_{60}	Copy of t^{c}_{60}
		t^{l}_{63}	Copy of t^{c}_{63}
		t^{l}_{67}	Copy of t^{c}_{67}
		t^{l}_{71}	Copy of t^{c}_{71}
		t^{l}_{3}	Copy of t^{c}_{3}
		t^{l}_{6}	Copy of t^{c}_{6}
		t^{l}_{20}	Copy of t^{c}_{20}
		t^{l}_{23}	Copy of t^{c}_{23}
		t^{l}_{40}	Copy of t^{c}_{40}
		t^{l}_{45}	Copy of t^{c}_{45}
		t^{l}_{61}	Copy of t^{c}_{51}
		t^{l}_{65}	Copy of t^{c}_{65}
		t^{l}_{66}	Copy of t^{c}_{66}
		t^{l}_{69}	Copy of t^{c}_{69}

Table 6.7: Conditions and Events for the HAS-200 LC and Specification Model

Condition	Meaning	Events	Meaning
$c_1\ (c_1^{in}, c_1^{out})$	Station 1 process	$e_1\ (e_1^{in}, e_1^{out})$	Station 1 start processing
$c_{10}\ (c_{10}^{in}, c_{10}^{out})$	Station 2 process	$e_3\ (e_3^{in}, e_3^{out})$	Station 1 done processing
$c_{12}\ (c_{12}^{in}, c_{12}^{out})$	Station 2 bypass	$e_{10}\ (e_{10}^{in}, e_{10}^{out})$	Station 2 start bypassing
$c_{20}\ (c_{20}^{in}, c_{20}^{out})$	Station 3 process	$e_{17}\ (e_{17}^{in}, e_{17}^{out})$	Station 2 done bypassing
$c_{22}\ (c_{22}^{in}, c_{22}^{out})$	Station 3 bypass	$e_{20}\ (e_{20}^{in}, e_{20}^{out})$	Station 3 start processing
$c_{30}\ (c_{30}^{in}, c_{30}^{out})$	Station 4 process	$e_{27}\ (e_{27}^{in}, e_{27}^{out})$	Station 3 done processing
$c_{32}\ (c_{32}^{in}\ c_{32}^{out})$	Station 4 bypass	$e_{32}\ (e_{32}^{in}, e_{32}^{out})$	Station 4 start processing
$c_{33}\ (c_{33}^{in}, c_{33}^{out})$	Station 5-10 process	$e_{39}\ (e_{39}^{in}, e_{39}^{out})$	Station 4 done processing
$c_{34}\ (c_{34}^{in}, c_{34}^{out})$	Station 5-10 bypass	$e_{44}\ (e_{44}^{in}, e_{44}^{out})$	Station 5-10 start bypassing
		$e_{45}\ (e_{45}^{in}, e_{45}^{out})$	Station 5-10 done bypassing
		$e_5\ (e_5^{in}, e_5^{out})$	Station 1 start bypassing
		$e_8\ (e_8^{in}, e_8^{out})$	Station 2 start processing
		$e_{15}\ (e_{15}^{in}, e_{15}^{out})$	Station 2 done processing box
		$e_{24}\ (e_{24}^{in}, e_{24}^{out})$	Station 2 done bypassing box
		$e_{29}\ (e_{29}^{in}, e_{29}^{out})$	Station 3 start bypassing
		$e_{34}\ (e_{34}^{in}, e_{34}^{out})$	Station 3 done bypassing
		$e_{41}\ (e_{41}^{in}, e_{41}^{out})$	Station 4 start bypassing
		$e_{42}\ (e_{42}^{in}, e_{42}^{out})$	Station 4 done bypassing
		$e_{43}\ (e_{43}^{in}, e_{43}^{out})$	Station 5-10 start processing
		$e_{46}\ (e_{46}^{in}, e_{46}^{out})$	Station 1 start processing
		$e_{47}\ (e_{47}^{in}, e_{47}^{out})$	Station 1 done processing
		$e_{48}\ (e_{48}^{in}, e_{48}^{out})$	Station 2 start bypassing
		$e_{49}\ (e_{49}^{in}, e_{49}^{out})$	Station 2 done bypassing
		$e_{50}\ (e_{50}^{in}, e_{50}^{out})$	Station 3 start processing
		$e_{51}\ (e_{51}^{in}, e_{51}^{out})$	Station 3 done processing
		$e_{52}\ (e_{52}^{in}, e_{52}^{out})$	Station 4 start processing
		$e_{53}\ (e_{53}^{in}, e_{53}^{out})$	Station 4 done processing
		$e_{54}\ (e_{54}^{in}, e_{54}^{out})$	Station 5-10 start bypassing
		$e_{55}\ (e_{55}^{in}, e_{55}^{out})$	Station 5-10 done bypassing
		$e_{56}\ (e_{56}^{in}, e_{56}^{out})$	Station 2 start processing
		$e_{57}\ (e_{57}^{in}, e_{57}^{out})$	Station 2 done processing box
		$e_{58}\ (e_{58}^{in}, e_{58}^{out})$	Station 2 done bypassing box
		$e_{59}\ (e_{59}^{in}, e_{59}^{out})$	Station 3 start bypassing
		$e_{60}\ (e_{60}^{in}, e_{60}^{out})$	Station 3 done bypassing
		$e_{61}\ (e_{61}^{in}, e_{61}^{out})$	Station 4 start bypassing
		$e_{62}\ (e_{62}^{in}, e_{62}^{out})$	Station 4 done bypassing
		$e_{63}\ (e_{63}^{in}, e_{63}^{out})$	Station 5-10 start processing

Figure 6.14 shows the complete controller model for the HAS-200 YRB sequence. Table 6.8 describes the transitions and places for the HAS-200 control model. Table 6.9 describes the events and conditions for the HAS-200 control model. For identification purposes the lines representing the condition signals and event signal between the uncontrolled model, specification, and locking controller are drawn differently.

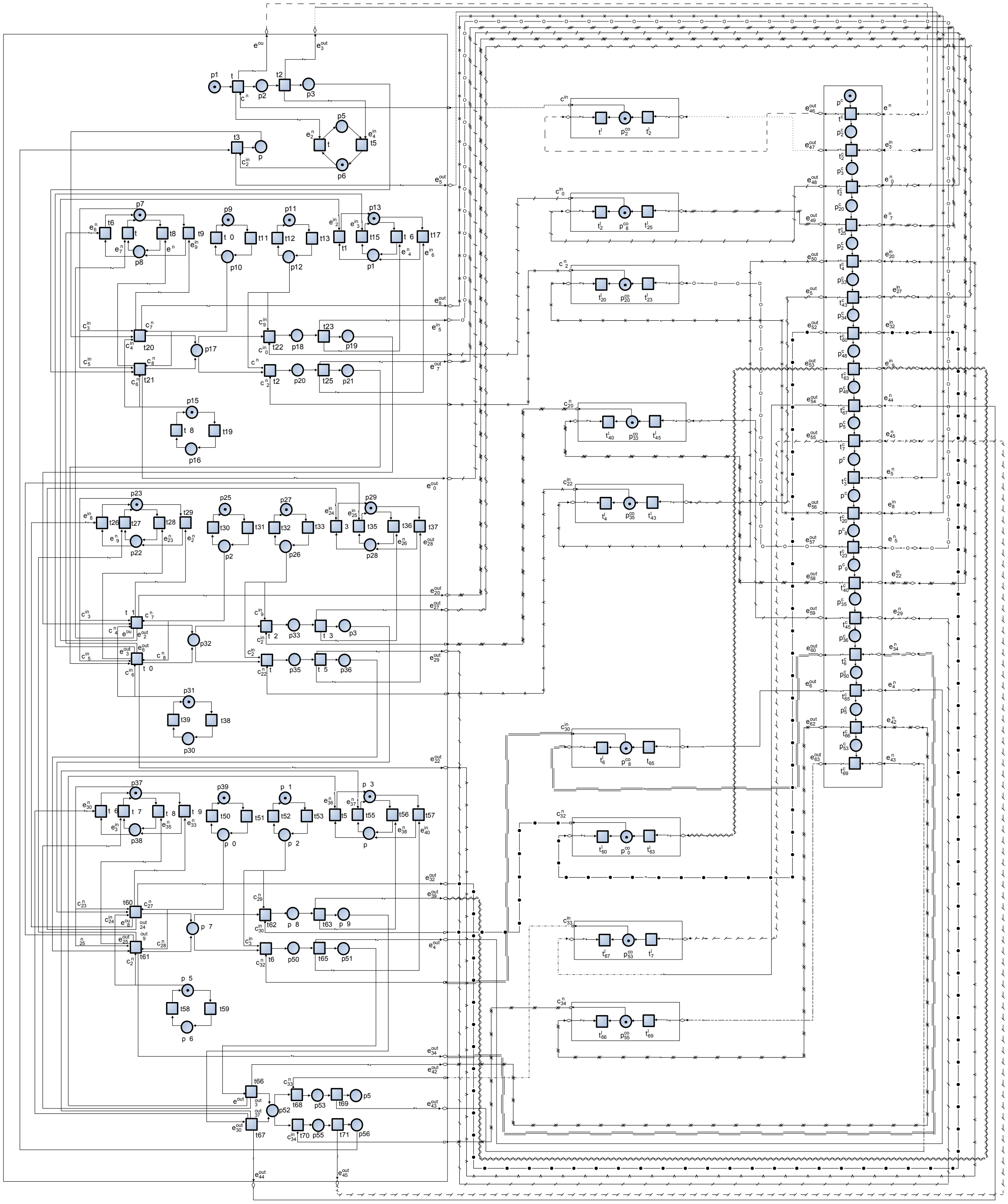


Figure 6.14: HAS-200 YRB Control Model

Table 6.8: Places and Transitions for the HAS-200 Control Model

Place	Meaning	Transition	Meaning	Place	Meaning	Transition	Meaning
p1	YRB order placed	t1	Station 1 start processing order	p^{c}_{1}	Order placed	t57	Station 4 Stopper 2 goes up
p2	Box process in station 1	t2	Station 1 done processing box	p^{c}_{2}	Box processed Station 1	t58	Station 4 sensor 2 turns active
p3	Box exit station 1	t3	Box moving from station 10 to station 1	p^{c}_{3}	Box Leaves Station 1	t59	Station 4 sensor 2 turns passive
p4	Box bypass station 1	t4	station 1 Stopper extending	p^{c}_{20}	Box bypassed Station 2	t60	Box moving from station 3 to station 4
p5	Station 1 stopper extended	t5	station 1 Stopper retracting	p^{c}_{21}	Box Leaves Station 2	t61	Box moving from station 3 to station 4
p6	Station 1 stopper retracted	t6	station 2 Stopper 1 retracting	p^{c}_{33}	Box processed Station 3	t62	Station 4 start processing
p7	Station 2 stopper 1 retracted	t7	station 2 Stopper 1 retracting	p^{c}_{3}	Box Leaves Station 3	t63	Station 4 done processing box
p8	Station 2 stopper 1 extended	t8	station 2 Stopper 1 extending	p^{c}_{8}	Box processed Station 4	t64	Station 4 start bypassing
p9	Station 2 sensor 1 passive	t9	Station 2 Stopper 1 extending	p^{c}_{9}	Box Leaves Station 4	t65	Station 4 done bypassing
p10	Station 2 sensor 1 active	t10	Station 2 sensor 1 turns passive	p^{c}_{55}	Box bypassed Station 5-10	t66	Box moving from station 4 to station 5
p11	Station 2 BCR passive	t11	Station 2 sensor 1 turns active	p^{c}_{56}	Box Leaves Station 10	t67	Box moving from station 4 to station 5
p12	Station 2 BCR active	t12	Station 2 BCR goes active	p^{c}_{4}	Box bypassed Station 1	t68	Station 5-10 start processing
p13	Station 2 Stopper 2 down	t13	Station 2 BCR goes passive	p^{c}_{18}	Box processed Station 2	t69	Station 5-10 done processing box
p14	Station 2 Stopper 2 up	t14	Station 2 Stopper 2 goes down	p^{c}_{19}	Box Leaves Station 2	t70	Station 5-10 start bypassing
p15	Station 2 sensor 2 passive	t15	Station 2 Stopper 2 goes down	p^{c}_{35}	Box bypassed Station 3	t71	Station 5-10 done bypassing
p16	Station 2 sensor 2 active	t16	Station 2 Stopper 2 goes up	p^{c}_{36}	Box Leaves Station 3	t^{c}_{1}	Station 1 start processing order
p17	Box in front of station 2 BCR	t17	Station 2 Stopper 2 goes up	p^{c}_{50}	Box bypassed Station 4	t^{c}_{2}	Station 1 done processing box
p18	Box process in station 2	t18	Station 2 sensor 2 turns passive	p^{c}_{51}	Box Leaves Station 4	t^{c}_{21}	Station 2 start bypassing
p19	Box exit station 2	t19	Station 2 sensor 2 turns active	p^{c}_{53}	Box processed Station 5-10	t^{c}_{25}	Station 2 done bypassing box
p20	Box bypass station 2	t20	Box moving from station 1 to station 2	p^{co}_{2}	Co-place of p^{c}_{2}	t^{c}_{41}	Station 3 start processing
p21	Box exit station 2	t21	Box moving from station 1 to station 2	p^{co}_{18}	Co-place of p^{c}_{18}	t^{c}_{43}	Station 3 done processing box
p22	Station 3 stopper 1 extended	t22	Station 2 start processing	p^{co}_{20}	Co-place of p^{c}_{20}	t^{c}_{60}	Station 4 start processing
p23	Station 3 stopper 1 retracted	t23	Station 2 done processing box	p^{co}_{33}	Co-place of p^{c}_{33}	t^{c}_{63}	Station 4 done processing box
p24	Station 3 sensor 1 active	t24	Station 2 start bypassing	p^{co}_{36}	Co-place of p^{c}_{36}	t^{c}_{67}	Station 5-10 start bypassing
p25	Station 3 sensor 1 passive	t25	Station 2 done bypassing	p^{co}_{8}	Co-place of p^{c}_{48}	t^{c}_{71}	Station 5-10 done bypassing
p26	Station 3 BCR active	t26	Station 3 Stopper 1 retracting	p^{co}_{50}	Co-place of p^{c}_{50}	t^{c}_{3}	Station 1 bypassing order
p27	Station 3 BCR passive	t27	Station 3 Stopper 1 retracting	p^{co}_{53}	Co-place of p^{c}_{53}	t^{c}_{20}	Station 2 start processing
p28	Station 3 Stopper 2 up	t28	Station 3 Stopper 1 extending	p^{co}_{55}	Co-place of p^{c}_{55}	t^{c}_{23}	Station 2 done processing box
p29	Station 3 Stopper 2 down	t29	Station 3 Stopper 1 extending			t^{c}_{40}	Station 3 start bypassing
p30	Station 3 sensor 2 active	t30	Station 3 sensor 1 turns passive			t^{c}_{45}	Station 3 done bypassing
p31	Station 3 sensor 2 passive	t31	Station 3 sensor 1 turns active			t^{c}_{61}	Station 4 start bypassing
p32	Box in front of station 3 BCR	t32	Station 3 BCR goes passive			t^{c}_{65}	Station 4 done bypassing
p33	Box process in station 3	t33	Station 3 BCR goes active			t^{c}_{66}	Station 5-10 start processing
p34	Box exit station 3	t34	Station 3 Stopper 2 goes down			t^{c}_{69}	Station 5-10 done processing box
p35	Box bypass station 3	t35	Station 3 Stopper 2 goes down			t^{l}	Copy of t^{c}
p36	Box exit station 3	t36	Station 3 Stopper 2 goes up			t^{l}_{2}	Copy of t^{c}_{2}
p37	Station 4 stopper 1 extended	t37	Station 3 Stopper 2 goes up			t^{l}_{2}	Copy of t^{c}_{2}
p38	Station 4 stopper 1 retracted	t38	Station 3 sensor 2 turns active			t^{l}_{25}	Copy of t^{c}_{25}
p39	Station 4 sensor 1 active	t39	Station 3 sensor 2 turns passive			t^{l}_{4}	Copy of t^{c}_{4}
p40	Station 4 sensor 1 passive	t40	Box moving from station 2 to station 3			t^{l}_{43}	Copy of t^{c}_{43}
p41	Station 4 BCR active	t41	Box moving from station 2 to station 3			t^{l}_{60}	Copy of t^{c}_{60}
p42	Station 4 BCR passive	t42	Station 3 start processing			t^{l}_{63}	Copy of t^{c}_{63}
p43	Station 4 Stopper 2 up	t43	Station 3 done processing box			t^{l}_{67}	Copy of t^{c}_{6}
p44	Station 4 Stopper 2 down	t44	Station 3 start bypassing			t^{l}_{7}	Copy of t^{c}_{7}
p45	Station 4 sensor 2 active	t45	Station 3 done bypassing			t^{l}_{3}	Copy of t^{c}_{3}
p46	Station 4 sensor 2 passive	t46	Station 4 Stopper 1 retracting			t^{l}_{6}	Copy of t^{c}_{6}
p47	Box in front of station 4 BCR	t47	Station 4 Stopper 1 retracting			t^{l}_{20}	Copy of t^{c}_{20}
p48	Box process in station 4	t48	Station 4 Stopper 1 extending			t^{l}_{23}	Copy of t^{c}_{23}
p49	Box exit station 4	t49	Station 4 Stopper 1 extending			t^{l}_{40}	Copy of t^{c}_{40}
p50	Box bypass station 4	t50	Station 4 sensor 1 turns passive			t^{l}_{45}	Copy of t^{c}_{45}
p51	Box exit station 4	t51	Station 4 sensor 1 turns active			t^{l}_{6}	Copy of t^{c}_{5}
p52	Box in front of station 5 BCR	t52	Station 4 BCR goes passive			t^{l}_{65}	Copy of t^{c}_{65}
p53	Box process in station 5-10	t53	Station 4 BCR goes active			t^{l}_{66}	Copy of t^{c}_{66}
p54	Box exit the system	t54	Station 4 Stopper 2 goes down			t^{l}_{69}	Copy of t^{c}_{69}
p55	Box bypass station 5-10	t55	Station 4 Stopper 2 goes down				
p56	Box exit station 10	t56	Station 4 Stopper 2 goes up				

Table 6.9: Conditions and Events for the HAS-200 Control Model

Condition	Meaning	Events	Meaning
$c_1(c_1^{in}, c_1^{out})$	Station 1 process	$e_1(e_1^{in}, e_1^{out})$	Station 1 start processing
$c_2(c_2^{in}, c_2^{out})$	Station 1 stopper retracted	$e_2(e_2^{in}, e_2^{out})$	Station 1 start processing
$c_3(c_3^{in}, c_3^{out})$	Station 2 stopper 1 retracted	$e_3(e_3^{in}, e_3^{out})$	Station 1 done processing
$c_4(c_4^{in}, c_4^{out})$	Station 2 sensor 2 passive	$e_4(e_4^{in}, e_4^{out})$	Station 1 done processing
$c_5(c_5^{in}, c_5^{out})$	Station 2 stopper 1 retracted	$e_5(e_5^{in}, e_5^{out})$	Station 1 bypass
$c_6(c_6^{in}, c_6^{out})$	Station 2 sensor 2 passive	$e_6(e_6^{in}, e_6^{out})$	Box exit Station 2
$c_7(c_7^{in}, c_7^{out})$	Station 2 sensor 1 active	$e_7(e_7^{in}, e_7^{out})$	Box exit Station 2
$c_8(c_8^{in}, c_8^{out})$	Station 2 sensor 1 active	$e_8(e_8^{in}, e_8^{out})$	Box enters station 2
$c_9(c_9^{in}, c_9^{out})$	Station 2 BCR active	$e_9(e_9^{in}, e_9^{out})$	Box enters station 2
$c_{10}(c_{10}^{in}, c_{10}^{out})$	Station 2 process	$e_{10}(e_{10}^{in}, e_{10}^{out})$	Box enters station 2
$c_{11}(c_{11}^{in}, c_{11}^{out})$	Station 2 BCR active	$e_{11}(e_{11}^{in}, e_{11}^{out})$	Box enters station 2
$c_{12}(c_{12}^{in}, c_{12}^{out})$	Station 2 bypass	$e_{12}(e_{12}^{in}, e_{12}^{out})$	Box exit Station 2
$c_{13}(c_{13}^{in}, c_{13}^{out})$	Station 3 stopper 1 retracted	$e_{13}(e_{13}^{in}, e_{13}^{out})$	Box exit Station 2
$c_{14}(c_{14}^{in}, c_{14}^{out})$	Station 3 sensor 2 passive	$e_{14}(e_{14}^{in}, e_{14}^{out})$	Station 2 done processing
$c_{15}(c_{15}^{in}, c_{15}^{out})$	Station 3 stopper 1 retracted	$e_{15}(e_{15}^{in}, e_{15}^{out})$	Station 2 done processing
$c_{16}(c_{16}^{in}, c_{16}^{out})$	Station 3 sensor 2 passive	$e_{16}(e_{16}^{in}, e_{16}^{out})$	Station 2 done bypassing
$c_{17}(c_{17}^{in}, c_{17}^{out})$	Station 3 sensor 1 active	$e_{17}(e_{17}^{in}\ e_{17}^{out})$	Station 2 done bypassing
$c_{18}(c_{18}^{in}, c_{18}^{out})$	Station 3 sensor 1 active	$e_{18}(e_{18}^{in}, e_{18}^{out})$	Box exit Station 3
$c_{19}(c_{19}^{in}, c_{19}^{out})$	Station 3 BCR active	$e_{19}(e_{19}^{in}, e_{19}^{out})$	Box exit Station 3
$c_{20}(c_{20}^{in}, c_{20}^{out})$	Station 3 process	$e_{20}(e_{20}^{in}, e_{20}^{out})$	Box enters station 3
$c_{21}(c_{21}^{in}, c_{21}^{out})$	Station 3 BCR active	$e_{21}(e_{21}^{in}, e_{21}^{out})$	Box enters station 3
$c_{22}(c_{22}^{in}, c_{22}^{out})$	Station 3 bypass	$e_{22}(e_{22}^{in}, e_{22}^{out})$	Box enters station 3
$c_{23}(c_{23}^{in}, c_{23}^{out})$	Station 4 stopper 1 retracted	$e_{23}(e_{23}^{in}, e_{23}^{out})$	Box enters station 3
$c_{24}(c_{24}^{in}, c_{24}^{out})$	Station 4 sensor 2 passive	$e_{24}(e_{24}^{in}, e_{24}^{out})$	Box exit Station 3
$c_{25}(c_{25}^{in}, c_{25}^{out})$	Station 4 stopper 1 retracted	$e_{25}(e_{25}^{in}, e_{25}^{out})$	Box exit Station 3
$c_{26}(c_{26}^{in}, c_{26}^{out})$	Station 4 sensor 2 passive	$e_{26}(e_{26}^{in}, e_{26}^{out})$	Station 3 done processing
$c_{27}(c_{27}^{in}, c_{27}^{out})$	Station 4 sensor 1 active	$e_{27}(e_{27}^{in}, e_{27}^{out})$	Station 3 done processing
$c_{28}(c_{28}^{in}, c_{28}^{out})$	Station 4 sensor 1 active	$e_{28}(e_{28}^{in}, e_{28}^{out})$	Station 3 done bypassing
$c_{29}(c_{29}^{in}, c_{29}^{out})$	Station 4 BCR active	$e_{29}(e_{29}^{in}, e_{29}^{out})$	Station 3 done bypassing
$c_{30}(c_{30}^{in}, c_{30}^{out})$	Station 4 process	$e_{30}(e_{30}^{in}, e_{30}^{out})$	Box exit Station 4
$c_{31}(c_{31}^{in}, c_{31}^{out})$	Station 4 BCR active	$e_{31}(e_{31}^{in}, e_{31}^{out})$	Box exit Station 4
$c_{32}(c_{32}^{in}, c_{32}^{out})$	Station 4 bypass	$e_{32}(e_{32}^{in}, e_{32}^{out})$	Box enters station 4
$c_{33}(c_{33}^{in}, c_{33}^{out})$	Station 5-10 process	$e_{33}(e_{33}^{in}, e_{33}^{out})$	Box enters station 4
$c_{34}(c_{34}^{in}, c_{34}^{out})$	Station 5-10 bypass	$e_{34}(e_{34}^{in}, e_{34}^{out})$	Box enters station 4
		$e_{35}(e_{35}^{in}, e_{35}^{out})$	Box enters station 4
		$e_{36}(e_{36}^{in}, e_{36}^{out})$	Box exit Station 4
		$e_{37}(e_{37}^{in}, e_{37}^{out})$	Box exit Station 4
		$e_{38}(e_{38}^{in}, e_{38}^{out})$	Station 4 done processing
		$e_{39}(e_{39}^{in}, e_{39}^{out})$	Station 4 done processing
		$e_{40}(e_{40}^{in}, e_{40}^{out})$	Station 4 done bypassing
		$e_{41}(e_{41}^{in}, e_{41}^{out})$	Station 4 done bypassing
		$e_{42}(e_{42}^{in}, e_{42}^{out})$	Box enters station 5
		$e_{43}(e_{43}^{in}, e_{43}^{out})$	Station 5-10 done processing
		$e_{44}(e_{44}^{in}, e_{44}^{out})$	Box enters station 5
		$e_{45}(e_{45}^{in}, e_{45}^{out})$	Station 5-10 done bypassing
		$e_{46}(e_{46}^{in}, e_{46}^{out})$	Station 1 start processing order
		$e_{47}(e_{47}^{in}, e_{47}^{out})$	Station 1 done processing box
		$e_{48}(e_{48}^{in}, e_{48}^{out})$	Station 2 start bypassing
		$e_{49}(e_{49}^{in}, e_{49}^{out})$	Station 2 done bypassing box
		$e_{50}(e_{50}^{in}, e_{50}^{out})$	Station 3 start processing
		$e_{51}(e_{51}^{in}, e_{51}^{out})$	Station 3 done processing box
		$e_{52}(e_{52}^{in}, e_{52}^{out})$	Station 4 start processing
		$e_{53}(e_{53}^{in}, e_{53}^{out})$	Station 4 done processing box
		$e_{54}(e_{54}^{in}, e_{54}^{out})$	Station 5-10 start bypassing
		$e_{55}(e_{55}^{in}, e_{55}^{out})$	Station 5-10 done bypassing
		$e_{56}(e_{56}^{in}, e_{56}^{out})$	Station 1 bypassing order
		$e_{57}(e_{57}^{in}, e_{57}^{out})$	Station 2 start processing
		$e_{58}(e_{58}^{in}, e_{58}^{out})$	Station 2 done processing box
		$e_{59}(e_{59}^{in}, e_{59}^{out})$	Station 3 start bypassing
		$e_{60}(e_{60}^{in}, e_{60}^{out})$	Station 3 done bypassing
		$e_{61}(e_{61}^{in}, e_{61}^{out})$	Station 4 start bypassing
		$e_{62}(e_{62}^{in}, e_{62}^{out})$	Station 4 done bypassing
		$e_{63}(e_{63}^{in}, e_{63}^{out})$	Station 5-10 start processing
		$e_{64}(e_{64}^{in}, e_{64}^{out})$	Station 5-10 done processing box

6.5 HAS-200 Supervisory Controller Verification

SESA is used to obtain the reachable states for the HAS-200 Control model. In order to analyze the behavior of the HAS-200 control model it is necessary to modify the uncontrollable actions of the sensors and bar code readers in Station 2, 3, and 4. To control the sensors and bar code readers, events will be added to their transitions in the control model. These events will force the uncontrollable transitions to fire in a sequence that is representative of the physical system. For example, after the box leaves Station 1 the conveyor belt will take the box to Station 2. Sensor 1 will go active (p10 in Figure 6.15), since the box is in front of Station 2. A place and a transition (p_{10}^{c} and t_{12}^{c}) are added to the sequence specification to represent this state as shown in Figure 6.15. An event (e_{54}^{c}) is added as output to t_{12}^{c} and as input to t12 as shown in Figure 6.15. This way, the specification can ensure that Sensor 1 (p10 and t12 in Figure 6.15) will fire only after the box has left Station1. Also, after the box has entered Station 2 (p15 in Figure 6.15) Sensor 1 will go passive and the bar code reader will go active. Places and transitions (p_{9}^{c}, p_{12}^{c}, t_{11}^{c} and t_{16}^{c} in Figure 6.15) are added to the sequence specification to represent these states. Two events are added to ensure Sensor 1 (p9 and t12 in Figure 6.15) and the bar code reader (p12 and t16 in Figure 6.15) will fire only after the box has entered Station 2. One event (e_{56}^{c}) will be an output to t_{11}^{c} and input to t11 as shown in Figure 6.15. The second event (e_{57}^{c}) will be output to t_{16}^{c} and input to t16 as shown in Figure 6.15. The same procedure is applied for the rest of the sensors and bar code readers. For testing purposes it is assumed that no queues will form at stations 2, 3, and 4. As a result, Sensor2 is removed from the model. The new HAS-200 model is shown in Figure 6.16. Table 6.10 describes the transitions and places of the new control model. Table 6.11 describes the conditions and events of the new control model.

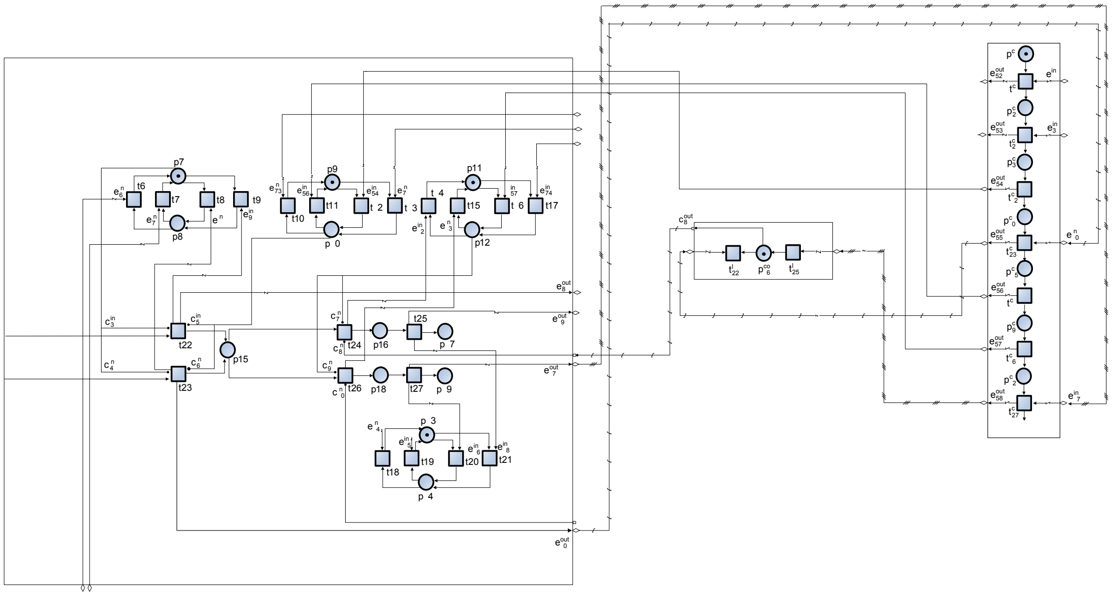


Figure 6.15: Portion of SESA Tank Control Model Pertaining t11, t12, and t16

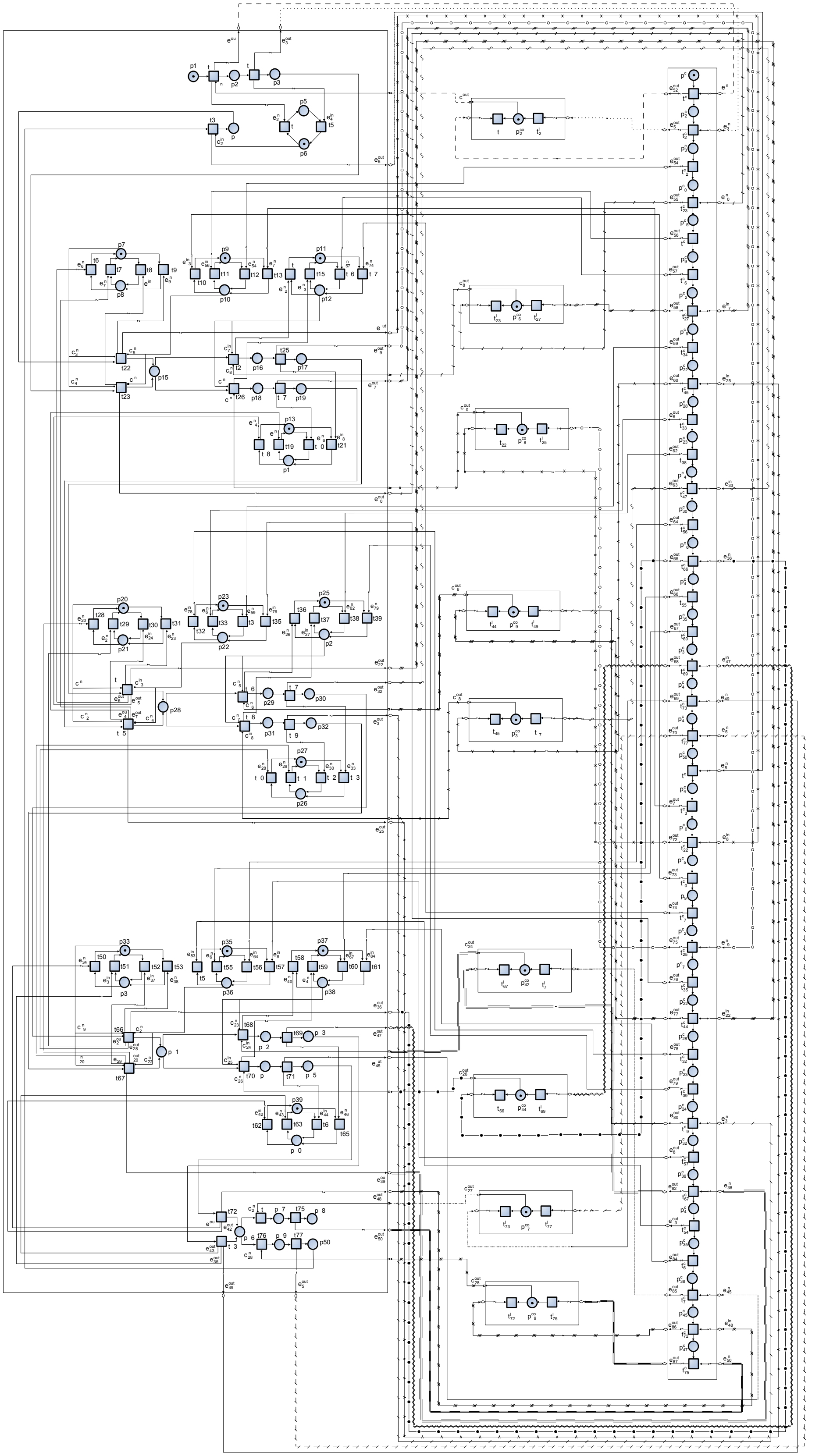


Figure 6.16: SESA HAS-200 Control Model

Table 6.10: Places and Transitions for the SESA HAS-200 Control Model

Place	Meaning	Transition	Meaning	Place	Meaning	**Transition**	**Meaning**
p1	YRB order placed	t1	Station 1 start processing order	p^{c}_{1}	Order placed	t^{c}_{1}	Station 1 start processing order
p2	Box process in station 1	t2	Station 1 done processing box	p^{c}_{2}	Box processed Station 1	t^{c}_{2}	Station 1 done processing box
p3	Box exit station 1	t3	Box moving from station 10 to station 1	p^{c}_{3}	Box Leaves Station 1	t^{c}_{12}	Station 2 sensor 1 goes active
p4	Box bypass station 1	t4	station 1 Stopper extending	p^{c}_{10}	Station 2 sensor 1 goes active	t^{c}_{23}	Station 2 start bypassing
p5	Station 1 stopper extended	t5	station 1 Stopper retracting	p^{c}_{15}	Box bypassed Station 2	t^{c}_{11}	Station 2 sensor 1 goes passive
p6	Station 1 stopper retracted	t6	station 2 Stopper 1 retracting	p^{c}_{9}	Station 2 sensor 1 goes passive	t^{c}_{16}	Station 2 BCR goes active
p7	Station 2 stopper 1 retracted	t7	station 2 Stopper 1 retracting	p^{c}_{12}	Station 2 BCR goes active	t^{c}_{27}	Station 2 done bypassing box
p8	Station 2 stopper 1 extended	t8	station 2 Stopper 1 extending	p^{c}_{19}	Box Leaves Station 2	t^{c}_{34}	Station 3 sensor 1 goes active
p9	Station 2 sensor 1 passive	t9	Station 2 Stopper 1 extending	p^{c}_{22}	Station 3 sensor 1 goes active	t^{c}_{45}	Station 3 start processing
p10	Station 2 sensor 1 active	t10	Station 2 sensor 1 turns passive	p^{c}_{28}	Box processed Station 3	t^{c}_{33}	Station 3 sensor 1 goes passive
p11	Station 2 BCR passive	t11	Station 2 sensor 1 turns passive	p^{c}_{23}	Station 3 sensor 1 goes passive	t^{c}_{38}	Station 3 BCR goes active
p12	Station 2 BCR active	t12	Station 2 sensor 1 turns active	p^{c}_{24}	Station 3 BCR goes active	t^{c}_{47}	Station 3 done processing box
p13	Station 2 Stopper 2 down	t13	Station 2 sensor 1 turns active	p^{c}_{30}	Box Leaves Station 3	t^{c}_{56}	Station 4 sensor 1 goes active
p14	Station 2 Stopper 2 up	t14	Station 2 BCR goes active	p^{c}_{36}	Station 4 sensor 1 goes active	t^{c}_{66}	Station 4 start processing
p15	Box in front of station 2 BCR	t15	Station 2 BCR goes active	p^{c}_{41}	Box processed Station 4	t^{c}_{55}	Station 4 sensor 1 goes passive
p16	Box process in station 2	t16	Station 2 BCR goes passive	p^{c}_{35}	Station 4 sensor 1 goes passive	t^{c}_{60}	Station 4 BCR goes active
p17	Box exit station 2	t17	Station 2 BCR goes passive	p^{c}_{38}	Station 4 BCR goes active	t^{c}_{69}	Station 4 done processing box
p18	Box bypass station 2	t18	Station 2 Stopper 2 goes down	p^{c}_{43}	Box Leaves Station 4	t^{c}_{73}	Station 5-10 start bypassing
p19	Box exit station 2	t19	Station 2 Stopper 2 goes down	p^{c}_{49}	Box bypassed Station 5-10	t^{c}	Station 5-10 done bypassing
p20	Station 3 stopper 1 extended	t20	Station 2 Stopper 2 goes up	p^{c}_{50}	Box Leaves Station 10	t^{c}_{3}	Station 1 bypassing order
p21	Station 3 stopper 1 retracted	t21	Station 2 Stopper 2 goes up	p^{c}_{4}	Box bypassed Station 1	t^{c}_{13}	Station 2 sensor 1 goes active
p22	Station 3 sensor 1 active	t22	Box moving from station 1 to station 2	p^{c}_{10}	Station 2 sensor 1 goes active	t^{c}_{22}	Station 2 start processing
p23	Station 3 sensor 1 passive	t23	Box moving from station 1 to station 2	p^{c}_{15}	Box processed Station 2	t^{c}_{10}	Station 2 sensor 1 goes passive
p24	Station 3 BCR active	t24	Station 2 start processing	p^{c}_{9}	Station 2 sensor 1 goes passive	t^{c}_{17}	Station 2 BCR goes active
p25	Station 3 BCR passive	t25	Station 2 done processing box	p^{c}_{12}	Station 2 BCR goes active	t^{c}_{25}	Station 2 done processing box
p26	Station 3 Stopper 2 up	t26	Station 2 start bypassing	p^{c}_{17}	Box Leaves Station 2	t^{c}_{35}	Station 3 sensor 1 goes active
p27	Station 3 Stopper 2 down	t27	Station 2 done bypassing	p^{c}_{22}	Station 3 sensor 1 goes active	t^{c}	Station 3 start bypassing
p28	Box in front of station 3 BCR	t28	Station 3 Stopper 1 retracting	p^{c}_{28}	Box bypassed Station 3	t^{c}_{32}	Station 3 sensor 1 goes passive
p29	Box process in station 3	t29	Station 3 Stopper 1 retracting	p^{c}_{23}	Station 3 sensor 1 goes passive	t^{c}_{39}	Station 3 BCR goes active
p30	Box exit station 3	t30	Station 3 Stopper 1 extending	p^{c}_{2}	Station 3 BCR goes active	t^{c}_{9}	Station 3 done bypassing
p31	Box bypass station 3	t31	Station 3 Stopper 1 extending	p^{c}_{32}	Box Leaves Station 3	t^{c}_{57}	Station 4 sensor 1 goes active
p32	Box exit station 3	t32	Station 3 sensor 1 turns passive	p^{c}_{36}	Station 4 sensor 1 goes active	t^{c}_{67}	Station 4 start bypassing
p33	Station 4 stopper 1 extended	t33	Station 3 sensor 1 turns passive	p^{c}_{41}	Box bypassed Station 4	t^{c}_{5}	Station 4 sensor 1 goes passive
p34	Station 4 stopper 1 retracted	t34	Station 3 sensor 1 turns active	p^{c}_{35}	Station 4 sensor 1 goes passive	t^{c}_{61}	Station 4 BCR goes active
p35	Station 4 sensor 1 active	t35	Station 3 sensor 1 turns active	p^{c}_{38}	Station 4 BCR goes active	t^{c}_{71}	Station 4 done bypassing
p36	Station 4 sensor 1 passive	t36	Station 3 BCR goes passive	p^{c}_{5}	Box Leaves Station 4	t^{c}_{72}	Station 5-10 start processing
p37	Station 4 BCR active	t37	Station 3 BCR goes passive	p^{c}_{7}	Box processed Station 5-10	t^{c}_{75}	Station 5-10 done processing box
p38	Station 4 BCR passive	t38	Station 3 BCR goes active	p^{co}_{2}	Co-place of p^{c}_{2}	t^{l}_{1}	Copy of t^{c}
p39	Station 4 Stopper 2 up	t39	Station 3 BCR goes active	p^{co}_{16}	Co-place of p^{c}_{16}	t^{l}_{2}	Copy of t^{c}_{2}
p40	Station 4 Stopper 2 down	t40	Station 3 Stopper 2 goes down	p^{co}_{18}	Co-place of p^{c}_{18}	t^{l}_{23}	Copy of t^{c}_{23}
p41	Box in front of station 4 BCR	t41	Station 3 Stopper 2 goes down	p^{co}_{29}	Co-place of p^{c}_{29}	t^{l}_{27}	Copy of t^{c}_{27}
p42	Box process in station 4	t42	Station 3 Stopper 2 goes up	p^{co}_{31}	Co-place of p^{c}_{31}	t^{l}_{45}	Copy of t^{c}_{5}
p43	Box exit station 4	t43	Station 3 Stopper 2 goes up	p^{co}_{2}	Co-place of p^{c}_{2}	t^{l}_{47}	Copy of t^{c}_{4}
p44	Box bypass station 4	t44	Box moving from station 2 to station 3	p^{co}	Co-place of p^{c}	t^{l}_{66}	Copy of t^{c}_{66}
p45	Box exit station 4	t45	Box moving from station 2 to station 3	p^{co}_{8}	Co-place of p^{c}_{47}	t^{l}_{69}	Copy of t^{c}_{69}
p46	Box in front of station 5 BCR	t46	Station 3 start processing	p^{co}_{9}	Co-place of p^{c}_{9}	t^{l}_{73}	Copy of t^{c}_{6}
p47	Box process in station 5-10	t47	Station 3 done processing box			t^{l}	Copy of t^{c}
p48	Box exit the system	t48	Station 3 start bypassing			t^{l}_{22}	Copy of t^{c}_{22}
p49	Box bypass station 5-10	t49	Station 3 done bypassing			t^{l}_{25}	Copy of t^{c}_{25}
p50	Box exit station 10	t50	Station 4 Stopper 1 retracting			t^{l}_{44}	Copy of t^{c}_{44}
		t51	Station 4 Stopper 1 retracting			t^{l}_{9}	Copy of t^{c}_{49}
		t52	Station 4 Stopper 1 extending			t^{l}_{67}	Copy of t^{l}_{67}
		t53	Station 4 Stopper 1 extending			t^{l}_{71}	Copy of t^{c}_{71}
		t54	Station 4 sensor 1 turns passive			t^{l}_{72}	Copy of t^{c}_{72}
		t55	Station 4 sensor 1 turns passive			t^{l}_{5}	Copy of t^{c}_{5}
		t56	Station 4 sensor 1 turns active				
		t57	Station 4 sensor 1 turns active				
		t58	Station 4 BCR goes passive				
		t59	Station 4 BCR goes passive				
		t60	Station 4 BCR goes active				
		t61	Station 4 BCR goes active				
		t62	Station 4 Stopper 2 goes down				
		t63	Station 4 Stopper 2 goes down				
		t64	Station 4 Stopper 2 goes up				
		t65	Station 4 Stopper 2 goes up				
		t66	Box moving from station 3 to station 4				
		t67	Box moving from station 3 to station 4				
		t68	Station 4 start processing				
		t69	Station 4 done processing box				
		t70	Station 4 start bypassing				
		t71	Station 4 done bypassing				
		t72	Box moving from station 4 to station 5				
		t73	Box moving from station 4 to station 5				
		t74	Station 5-10 start processing				
		t75	Station 5-10 done processing box				
		t76	Station 5-10 start bypassing				
		t77	Station 5-10 done bypassing				

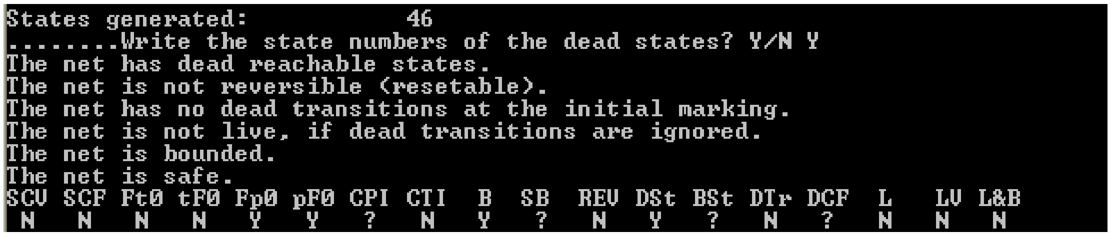
Table 6.11: Conditions and Events for the SESA HAS-200 Control Model

Condition	Meaning	Events	Meaning	Events	Meaning
$c_{1}(c_{1}^{in}, c_{1}^{out})$	Station 1 process	$e_{1}(e_{1}^{in}\ e_{1}^{out})$	Station 1 start processing	$e_{5}(e_{5}^{in}, e_{5}^{out})$	Station 4 done bypassing
$c_{2}(c_{2}^{in}, c_{2}^{out})$	Station 1 stopper retracted	$e_{2}(e_{2}^{in}\ e_{2}^{out})$	Station 1 start processing	$e_{6}(e_{6}^{in}, e_{6}^{out})$	Station 4 done processing
$c_{3}(c_{3}^{in}, c_{3}^{out})$	Station 2 stopper 1 retracted	$e_{3}(e_{3}^{in}\ e_{3}^{out})$	Station 1 done processing	$e_{7}(e_{7}^{in}, e_{7}^{out})$	Station 4 done processing
$c(c^{in}, c^{out})$	Station 2 stopper 1 retracted	$e(e^{in}\ e^{out})$	Station 1 done processing	$e_{8}(e_{8}^{in}, e_{8}^{out})$	Box enters station 5
$c_{5}(c_{5}^{in}, c_{5}^{out})$	Station 2 sensor 1 active	$e_{5}(e_{5}^{in}, e_{5}^{out})$	Station 1 bypass	$e_{9}(e_{9}^{in}, e_{9}^{out})$	Box enters station 5
$c_{6}(c_{6}^{in}, c_{6}^{out})$	Station 2 sensor 1 active	$e_{6}(e_{6}^{in}\ e_{6}^{out})$	Box exit Station 2	$e_{50}(e_{50}^{in}, e_{50}^{out})$	Station 5-10 done processing
$c_{7}(c_{7}^{in}, c_{7}^{out})$	Station 2 BCR active	$e_{7}(e_{7}^{in}\ e_{7}^{out})$	Box exit Station 2	$e_{51}(e_{51}^{in}, e_{51}^{out})$	Station 5-10 done bypassing
$c_{8}(c_{8}^{in}, c_{8}^{out})$	Station 2 process	$e_{8}(e_{8}^{in}\ e_{8}^{out})$	Box enters station 2	$e_{52}(e_{52}^{in}, e_{52}^{out})$	Station 1 start processing order
$c_{9}(c_{9}^{in}\ c_{9}^{out})$	Station 2 BCR active	$e_{9}(e_{9}^{in}\ e_{9}^{out})$	Box enters station 2	$e_{53}(e_{53}^{in}, e_{53}^{out})$	Station 1 done processing box
$c_{10}(c_{10}^{in}, c_{10}^{out})$	Station 2 bypass	$e_{10}(e_{10}^{in}\ e_{10}^{out})$	Box enters station 2	$e_{5}(e_{5}^{in}, e_{5}^{out})$	Station 2 Sensor 1 goes active
$c_{11}(c_{11}^{in}, c_{11}^{out})$	Station 3 stopper 1 retracted	$e_{11}(e_{11}^{in}\ e_{11}^{out})$	Box enters station 2	$e_{55}(e_{55}^{in}\ e_{55}^{out})$	Station 2 start bypassing
$c_{12}(c_{12}^{in}, c_{12}^{out})$	Station 3 stopper 1 retracted	$e_{12}(e_{12}^{in}, e_{12}^{out})$	Station 2 Start Processing	$e_{56}(e_{56}^{in}, e_{56}^{out})$	Station 2 start bypassing
$c_{13}(c_{13}^{in}, c_{13}^{out})$	Station 3 sensor 1 active	$e_{13}(e_{13}^{in}\ e_{13}^{out})$	Station 2 Start Bypassing	$e_{57}(e_{7}^{in}, e_{57}^{out})$	Station 2 start bypassing
$c_{1}(c_{1}^{in}, c_{1}^{out})$	Station 3 sensor 1 active	$e_{1}(e_{1}^{in}\ e_{1}^{out})$	Box exit Station 2	$e_{58}(e_{58}^{in}, e_{58}^{out})$	Station 2 done bypassing box
$c_{15}(c_{15}^{in}, c_{15}^{out})$	Station 3 BCR active	$e_{15}(e_{15}^{in}\ e_{15}^{out})$	Box exit Station 2	$e_{59}(e_{59}^{in}, e_{59}^{out})$	Station 3 Sensor 1 goes active
$c_{16}(c_{16}^{in}, c_{16}^{out})$	Station 3 process	$e_{16}(e_{16}^{in}\ e_{16}^{out})$	Station 2 done bypassing	$e_{60}(e_{60}^{in}, e_{60}^{out})$	Station 3 start processing
$c_{17}(c_{17}^{in}, c_{17}^{out})$	Station 3 BCR active	$e_{17}(e_{17}^{in}\ e_{17}^{out})$	Station 2 done bypassing	$e_{61}(e_{61}^{in}, e_{61}^{out})$	Station 3 start processing
$c_{18}(c_{18}^{in}, c_{18}^{out})$	Station 3 bypass	$e_{18}(e_{18}^{in}\ e_{18}^{out})$	Station 2 done processing	$e_{62}(e_{62}^{in}, e_{62}^{out})$	Station 3 start processing
$c_{19}(c_{19}^{in}\ c_{19}^{out})$	Station 4 stopper 1 retracted	$e_{19}(e_{19}^{in}\ e_{19}^{out})$	Station 2 done processing	$e_{63}(e_{63}^{in}, e_{63}^{out})$	Station 3 done processing box
$c_{20}(c_{20}^{in}, c_{20}^{out})$	Station 4 stopper 1 retracted	$e_{20}(e_{20}^{in}\ e_{20}^{out})$	Box exit Station 3	$e_{64}(e_{64}^{in}\ e_{64}^{out})$	Station 4 Sensor 1 goes active
$c_{21}(c_{21}^{in}, c_{21}^{out})$	Station 4 sensor 1 active	$e_{21}(e_{21}^{in}\ e_{21}^{out})$	Box exit Station 3	$e_{65}(e_{65}^{in}\ e_{65}^{out})$	Station 4 start processing
$c_{22}(c_{22}^{in}, c_{22}^{out})$	Station 4 sensor 1 active	$e_{22}(e_{22}^{in}\ e_{22}^{out})$	Box enters station 3	$e_{66}(e_{66}^{in}\ e_{66}^{out})$	Station 4 start processing
$c_{23}(c_{23}^{in}, c_{23}^{out})$	Station 4 BCR active	$e_{23}(e_{23}^{in}\ e_{23}^{out})$	Box enters station 3	$e_{67}(e_{67}^{in}\ e_{67}^{out})$	Station 4 start processing
$c_{2}(c_{2}^{in}, c_{2}^{out})$	Station 4 process	$e_{2}(e_{2}^{in}, e_{2}^{out})$	Box enters station 3	$e_{68}(e_{68}^{in}\ e_{68}^{out})$	Station 4 done processing box
$c_{25}(c_{25}^{in}, c_{25}^{out})$	Station 4 BCR active	$e_{25}(e_{25}^{in}, e_{25}^{out})$	Box enters station 3	$e_{69}(e_{69}^{in}\ e_{69}^{out})$	Station 5-10 start bypassing
$c_{26}(c_{26}^{in}, c_{26}^{out})$	Station 4 bypass	$e_{26}(e_{26}^{in}\ e_{26}^{out})$	Station 3 Start Processing	$e_{70}(e_{70}^{n}\ e_{70}^{out})$	Station 5-10 done bypassing
$c_{27}(c_{27}^{in}, c_{27}^{out})$	Station 5-10 process	$e_{27}(e_{27}^{in}, e_{27}^{out})$	Station 3 Start Bypassing	$e_{7}(e_{7}^{n}\ e_{7}^{out})$	Station 2 Sensor 1 goes active
$c_{28}(c_{28}^{in}, c_{28}^{out})$	Station 5-10 bypass	$e_{28}(e_{28}^{in}, e_{28}^{out})$	Box exit Station 3	$e_{72}(e_{72}^{n}\ e_{72}^{out})$	Station 2 start processing
		$e_{29}(e_{29}^{in}, e_{29}^{out})$	Box exit Station 3	$e_{73}(e_{73}^{n}\ e_{73}^{out})$	Station 2 start processing
		$e_{30}(e_{30}^{in}, e_{30}^{out})$	Station 3 done bypassing	$e_{74}(e_{74}^{n}\ e_{74}^{out})$	Station 2 start processing
		$e_{31}(e_{31}^{in}\ e_{31}^{out})$	Station 3 done bypassing	$e_{75}(e_{75}^{n}\ e_{75}^{out})$	Station 2 done processing box
		$e_{32}(e_{32}^{in}, e_{32}^{out})$	Station 3 done processing	$e_{76}(e_{76}^{n}\ e_{76}^{out})$	Station 3 Sensor 1 goes active
		$e_{33}(e_{33}^{in}, e_{33}^{out})$	Station 3 done processing	$e_{77}(e_{77}^{n}\ e_{77}^{out})$	Station 3 start bypassing
		$e_{3}(e_{3}^{in}, e_{3}^{out})$	Box exit Station 4	$e_{78}(e_{78}^{n}\ e_{78}^{out})$	Station 3 start bypassing
		$e_{35}(e_{35}^{in}\ e_{35}^{out})$	Box exit Station 4	$e_{79}(e_{79}^{n}\ e_{79}^{out})$	Station 3 start bypassing
		$e_{36}(e_{36}^{in}, e_{36}^{out})$	Box enters station 4	$e_{80}(e_{80}^{n}\ e_{80}^{out})$	Station 3 done bypassing
		$e_{37}(e_{37}^{in}, e_{37}^{out})$	Box enters station 4	$e_{8}(e_{8}^{n}\ e_{8}^{out})$	Station 4 Sensor 1 goes active
		$e_{38}(e_{38}^{in}, e_{38}^{out})$	Box enters station 4	$e_{82}(e_{82}^{n}\ e_{82}^{out})$	Station 4 start bypassing
		$e_{39}(e_{39}^{in}, e_{39}^{out})$	Box enters station 4	$e_{83}(e_{83}^{n}\ e_{83}^{out})$	Station 4 start bypassing
		$e_{0}(e_{0}^{in}, e_{0}^{out})$	Station 4 Start Processing	$e_{84}(e_{84}^{n}\ e_{84}^{out})$	Station 4 start bypassing
		$e_{1}(e_{1}^{in}, e_{1}^{out})$	Station 4 Start Bypassing	$e_{85}(e_{85}^{n}\ e_{85}^{out})$	Station 4 done bypassing
		$e_{2}(e_{2}^{in}\ e_{2}^{out})$	Box exit Station 4	$e_{86}(e_{86}^{n}\ e_{86}^{out})$	Station 5-10 start processing
		$e_{3}(e_{3}^{in}, e_{3}^{out})$	Box exit Station 4	$e_{87}(e_{87}^{n}\ e_{87}^{out})$	Station 5-10 done processing box
		$e(e^{in}, e^{out})$	Station 4 done bypassing		

The SESA software tool is used to analyze the control model for an order of one box. The HAS-200 control model will have only one token in p1 (order placed) and one token in p_1^c (specification order placed). The SESA analysis report is shown in Figure 6.17. The SESA reachable state results are shown in Appendix D. A reachability graph is constructed based on the SESA reachable states results and is shown in Figure 6.18. Table 6.12 is provided as a guide to identify the corresponding places and transitions for the SESA results. The results of the analysis report are explained below.

- Not Reversible: The box leaves the system after it is processed by Station 10. The box does not return to the system to be processed. Therefore, the model does not go back to its initial condition. The model is not reversible, since there is not a firing sequence that will take the system to its initial state.
- Bounded and Safe: The model is bounded, because at all times the number of tokens remains the same in all places. The tokens in the control model represent the state of devices (actuators: up/down; sensors active/passive, etc.) in the system, therefore it does not make sense in the physical system for a device to have two tokens. Furthermore, if there is an order of one box the model can only have one box leaving the system (one token p48, Figure 6.16). The boundedness of the model guarantees that no excess boxes are created during the process. Safeness will only be a requirement in the model for an order of one box. Safeness will guarantee that only one box remains in the system from beginning to end. Note that for orders of two or more the model will no longer be safe, but remains bounded as shown in Figure 6.19, 6.20, and 6.21.
- Not Live: There are no dead transitions at the initial marking. In other words, the model is able to fire at least once and the box is processed by the system. However, there are dead states which make the model not live. All transitions are able to fire at least once, but after t77 (box

exit system p48 in Figure 6.16) fires all transitions become *L0 live*. This is a desired behavior for the HAS-200 system.



```
States generated:           46
........Write the state numbers of the dead states? Y/N Y
The net has dead reachable states.
The net is not reversible (resetable).
The net has no dead transitions at the initial marking.
The net is not live, if dead transitions are ignored.
The net is bounded.
The net is safe.
SCV SCF Ft0 tF0 Fp0 pF0 CPI CTI  B  SB  REV DSt BSt DTr DCF  L   LV L&B
 N   N   N   N   Y   Y   ?   N   Y   ?   N   Y   ?   N   ?   N   N   N
```

Figure 6.17: SESA Analysis Report for HAS-200

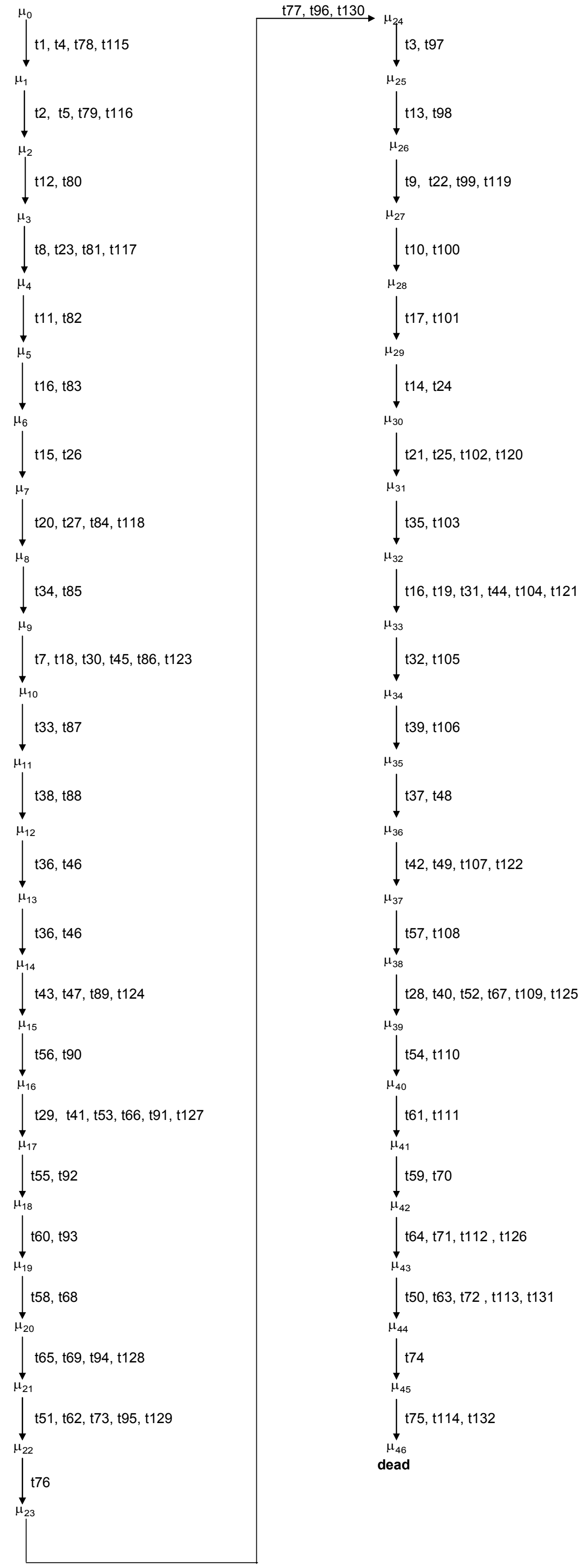
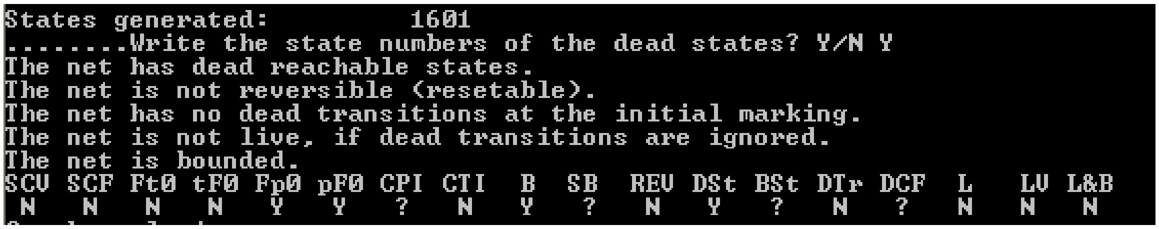


Figure 6.18: HAS-200 Reachability Graph

Table 6.12: Guide for the Places and Transitions for the SESA HAS-200 Control Model

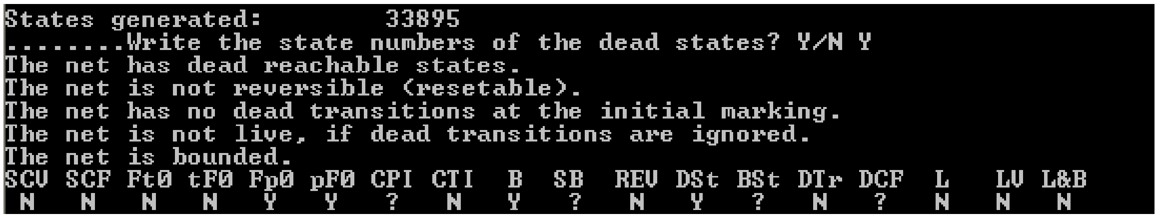
Place	SESA Place	Transition	SESA Transition	Place	SESA Place	Transition	SESA Transition
p1	p1	t1	t1	p_{1}^{c}	p51	t_{1}^{c}	t78
p2	p2	t2	t2	p_{2}^{c}	p52	t_{2}^{c}	t79
p3	p3	t3	t3	p_{3}^{c}	p53	t_{12}^{c}	t80
p4	p4	t4	t4	p_{10}^{c}	p54	t_{23}^{c}	t81
p5	p5	t5	t5	p_{15}^{c}	p55	t_{11}^{c}	t82
p6	p6	t6	t6	p_{9}^{c}	p56	t_{16}^{c}	t83
p7	p7	t7	t7	p_{12}^{c}	p57	t_{27}^{c}	t84
p8	p8	t8	t8	p_{19}^{c}	p58	t_{34}^{c}	t85
p9	p9	t9	t9	p_{22}^{c}	p59	t_{45}^{c}	t86
p10	p10	t10	t10	p_{28}^{c}	p60	t_{33}^{c}	t87
p11	p11	t11	t11	p_{23}^{c}	p61	t_{38}^{c}	t88
p12	p12	t12	t12	p_{24}^{c}	p62	t_{47}^{c}	t89
p13	p13	t13	t13	p_{30}^{c}	p63	t_{56}^{c}	t90
p14	p14	t14	t14	p_{36}^{c}	p64	t_{66}^{c}	t91
p15	p15	t15	t15	p_{41}^{c}	p65	t_{55}^{c}	t92
p16	p16	t16	t16	p_{35}^{c}	p66	t_{60}^{c}	t93
p17	p17	t17	t17	p_{38}^{c}	p67	t_{69}^{c}	t94
p18	p18	t18	t18	p_{43}^{c}	p68	t_{73}^{c}	t95
p19	p19	t19	t19	p_{49}^{c}	p69	t_{77}^{c}	t96
p20	p20	t20	t20	p_{50}^{c}	p70	t_{3}^{c}	t97
p21	p21	t21	t21	p_{4}^{c}	p71	t_{13}^{c}	t98
p22	p22	t22	t22	p_{10}^{c}	p72	t_{22}^{c}	t99
p23	p23	t23	t23	p_{15}^{c}	p73	t_{10}^{c}	t100
p24	p24	t24	t24	p_{9}^{c}	p74	t_{17}^{c}	t101
p25	p25	t25	t25	p_{12}^{c}	p75	t_{25}^{c}	t102
p26	p26	t26	t26	p_{17}^{c}	p76	t_{35}^{c}	t103
p27	p27	t27	t27	p_{22}^{c}	p77	t_{44}^{c}	t104
p28	p28	t28	t28	p_{28}^{c}	p78	t_{32}^{c}	t105
p29	p29	t29	t29	p_{23}^{c}	p79	t_{39}^{c}	t106
p30	p30	t30	t30	p_{24}^{c}	p80	t_{49}^{c}	t107
p31	p31	t31	t31	p_{32}^{c}	p81	t_{57}^{c}	t108
p32	p32	t32	t32	p_{36}^{c}	p82	t_{67}^{c}	t109
p33	p33	t33	t33	p_{41}^{c}	p83	t_{54}^{c}	t110
p34	p34	t34	t34	p_{35}^{c}	p84	t_{61}^{c}	t111
p35	p35	t35	t35	p_{38}^{c}	p85	t_{71}^{c}	t112
p36	p36	t36	t36	p_{45}^{c}	p86	t_{72}^{c}	t113
p37	p37	t37	t37	p_{47}^{c}	p87	t_{75}^{c}	t114
p38	p38	t38	t38	p_{2}^{co}	p88	t_{1}^{l}	t115
p39	p39	t39	t39	p_{16}^{co}	p89	t_{2}^{l}	t116
p40	p40	t40	t40	p_{18}^{co}	p90	t_{23}^{l}	t117
p41	p41	t41	t41	p_{29}^{co}	p91	t_{27}^{l}	t118
p42	p42	t42	t42	p_{31}^{co}	p92	t_{45}^{l}	t119
p43	p43	t43	t43	p_{42}^{co}	p93	t_{47}^{l}	t120
p44	p44	t44	t44	p_{44}^{co}	p94	t_{66}^{l}	t121
p45	p45	t45	t45	p_{48}^{co}	p95	t_{69}^{l}	t122
p46	p46	t46	t46	p_{49}^{co}	p96	t_{73}^{l}	t123
p47	p47	t47	t47			t_{77}^{l}	t124
p48	p48	t48	t48			t_{22}^{l}	t125
p49	p49	t49	t49			t_{25}^{l}	t126
p50	p50	t50	t50			t_{44}^{l}	t127
		t51	t51			t_{49}^{l}	t128
		t52	t52			t_{67}^{l}	t129
		t53	t53			t_{71}^{l}	t130
		t54	t54			t_{72}^{l}	t131
		t55	t55			t_{75}^{l}	t132
		t56	t56				
		t57	t57				
		t58	t58				
		t59	t59				
		t60	t60				
		t61	t61				
		t62	t62				
		t63	t63				
		t64	t64				
		t65	t65				
		t66	t66				
		t67	t67				
		t68	t68				
		t69	t69				
		t70	t70				
		t71	t71				
		t72	t72				
		t73	t73				
		t74	t74				
		t75	t75				
		t76	t76				
		t77	t77				

The HAS-200 control model is inserted in SESA for orders of two and three tokens in p1 and in p_1^c. The SESA results proved that the control model was capable of processing orders of multiple boxes (multiple tokens) while avoiding resource sharing conflicts. However, due to the very large state space of the analysis the results could not be provided in this thesis. Please contact the author for the reachable states output files. Nevertheless, the SESA analysis reports are shown in Figure 6.19 and 6.20. Note that for two and three tokens, the control model has the same properties at the one token control model (not reversible, bounded, and not live).



```
States generated:        1601
........Write the state numbers of the dead states? Y/N Y
The net has dead reachable states.
The net is not reversible (resetable).
The net has no dead transitions at the initial marking.
The net is not live, if dead transitions are ignored.
The net is bounded.
SCV SCF Ft0 tF0 Fp0 pF0 CPI CTI  B   SB  REV DSt BSt DTr DCF  L   LV L&B
 N   N   N   N   Y   Y   ?   N   Y   ?   N   Y   ?   N   ?   N   N   N
```

Figure 6.19: SESA Analysis Report HAS-200 2 Tokens



```
States generated:        33895
........Write the state numbers of the dead states? Y/N Y
The net has dead reachable states.
The net is not reversible (resetable).
The net has no dead transitions at the initial marking.
The net is not live, if dead transitions are ignored.
The net is bounded.
SCV SCF Ft0 tF0 Fp0 pF0 CPI CTI  B   SB  REV DSt BSt DTr DCF  L   LV L&B
 N   N   N   N   Y   Y   ?   N   Y   ?   N   Y   ?   N   ?   N   N   N
```

Figure 6.20: SESA Analysis Report HAS-200 3 Tokens

6.6 HAS-200 Tank Transformation Algorithm

Initialisation:

$\mathcal{T}$ = {t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13, t14, t15, t16, t17, t18, t19, t20, t21, t22, t23, t24, t25, t26, t27, t28, t29, t30, t31, t32, t33, t34, t35, t36, t37, t38, t39, t40, t41, t42, t43, t44, t45, t46, t47, t48, t49, t50, t51, t52, t53, t54, t55, t56, t57, t58, t59, t60, t61, t62, t63, t64, t65, t66, t67, t68, t69, t70, t71, t_1^c, t_2^c, t_3^c, t_{20}^c, t_{21}^c, t_{23}^c, t_{25}^c, t_{40}^c, t_{41}^c, t_{43}^c, t_{45}^c, t_{60}^c, t_{61}^c, t_{63}^c, t_{65}^c, t_{66}^c, t_{67}^c, t_{69}^c, t_{71}^c}

$\mathcal{T}_c$= {t1, t2, t3, t4, t5, t6, t7, t8, t9, t14, t15, t16, t17, t20, t21, t22, t23, t24, t25, t26, t27, t28, t29, t34, t35, t36, t37, t40, t41, t42, t43, t44, t45, t46, t47, t48, t49, t54, t55, t56, t57, t60, t61, t62, t63, t64, t65, t66, t67, t68, t69, t70, t71, t_1^c, t_2^c, t_3^c, t_{20}^c, t_{21}^c, t_{23}^c, t_{25}^c, t_{40}^c, t_{41}^c, t_{43}^c, t_{45}^c, t_{60}^c, t_{61}^c, t_{63}^c, t_{65}^c, t_{66}^c, t_{67}^c, t_{69}^c, t_{71}^c}.

$\mathcal{T}_u$ = {t10, t11, t12, t13, t18, t19, t30, t31, t32, t33, t38, t39, t50, t51, t52, t53, t58, t59}

$\mathcal{C}_\mathcal{N}$= {C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{16}, C_{17}, C_{18}, C_{19}, C_{20}, C_{21}, C_{22}, C_{23}, C_{24}, C_{25}, C_{26}, C_{27}, C_{28}, C_{29}, C_{30}, C_{31}, C_{32}, C_{33}, C_{34}}

$\mathcal{C}_{in}$= {C_1^{in}, C_2^{in}, C_3^{in}, C_4^{in}, C_5^{in}, C_6^{in}, C_7^{in}, C_8^{in}, C_9^{in}, C_{10}^{in}, C_{11}^{in}, C_{12}^{in}, C_{13}^{in}, C_{14}^{in}, C_{15}^{in}, C_{16}^{in}, C_{17}^{in}, C_{18}^{in}, C_{19}^{in}, C_{20}^{in}, C_{21}^{in}, C_{22}^{in}, C_{23}^{in}, C_{24}^{in}, C_{25}^{in}, C_{26}^{in}, C_{27}^{in}, C_{28}^{in}, C_{29}^{in}, C_{30}^{in}, C_{31}^{in}, C_{32}^{in}, C_{33}^{in}, C_{34}^{in} }

$\mathcal{C}_{out}$= {C_1^{out}, C_2^{out}, C_3^{out}, C_4^{out}, C_5^{out}, C_6^{out}, C_7^{out}, C_8^{out}, C_9^{out}, C_{10}^{out}, C_{11}^{out}, C_{12}^{out}, C_{13}^{out}, C_{14}^{out}, C_{15}^{out}, C_{16}^{out}, C_{17}^{out}, C_{18}^{out}, C_{19}^{out}, C_{20}^{out}, C_{21}^{out}, C_{22}^{out}, C_{23}^{out}, C_{24}^{out}, C_{25}^{out}, C_{26}^{out}, C_{27}^{out}, C_{28}^{out}, C_{29}^{out}, C_{30}^{out}, C_{31}^{out}, C_{32}^{out}, C_{33}^{out}, C_{34}^{out} }

Steps 1.1 through 1.5 transform the places, conditions, and events that interact with transition t1. The HAS-200 control model is shown in Figure 6.14. For convenience purposes, the portion of the HAS-200 control model that concerns t1 is shown in Figure 6.21.

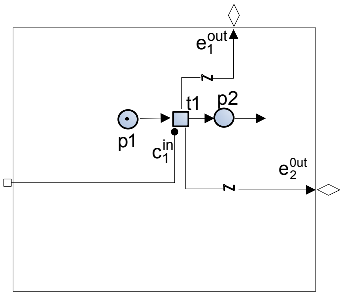


Figure 6.21: Portion of HAS-200 Control Model Pertaining to Transition t1

Step 1: $\forall\ t\bullet\ \epsilon\, \mathcal{T}_c$ insert a rung into the Ladder Logic Diagram and:

Step 1.1: $\forall\ p\ \epsilon\ \bullet t$ insert p into the rung as an input variable (examine on) and also as an unlatched output (output unlatch). As illustrated in Figure 6.21, p1 is an input place for t1. p1 is inserted into the rung as an input variable and as an unlatched output as shown in Figure 6.22.

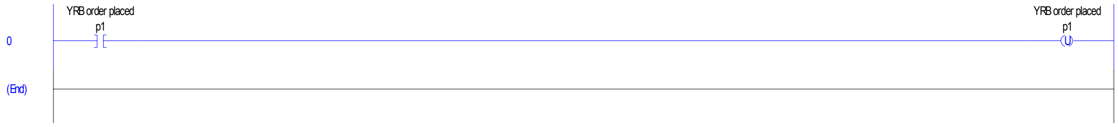


Figure 6.22: HAS-200 Control Model Insert Input Place for t1 in LLD

Step 1.2: $\forall\ c\ \epsilon\ C_{in}$ insert c into the rung as an input variable. As illustrated in Figure 6.21, C_1 is a conditions input for t1. C_1 is inserted into the rung as an input variable as shown in Figure 6.23.

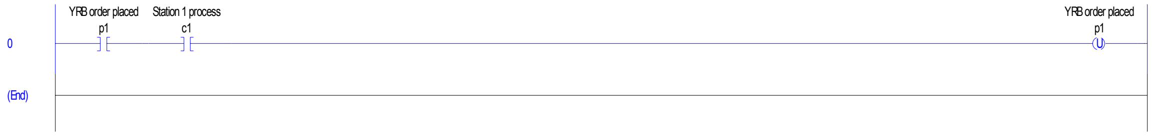


Figure 6.23: HAS-200 Control Model Insert Condition Inputs for t1 in LLD

Step 1.3: $\forall\ e \in \mathcal{E}_{in}$ insert e into the rung as an input variable. As illustrated in Figure 6.21, t1 has no event inputs. Therefore, no instructions are added to the rung.

Step 1.4: $\forall\ p \in t \bullet$ insert p into the rung as a latched output (output latch). As illustrated in Figure 6.21, p2 is an output place for t1. p2 is inserted into the rung as a latched output as shown in Figure 6.24.

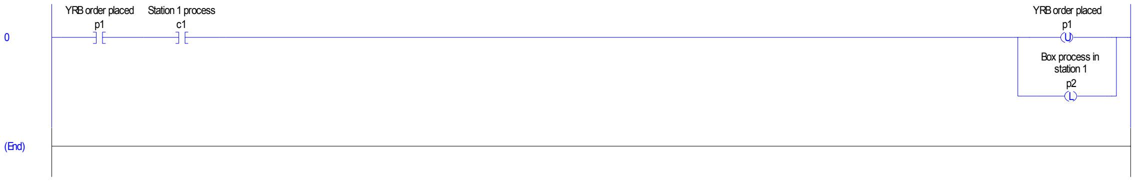


Figure 6.24: HAS-200 Control Model Insert Output Place for t1 in LLD

Step 1.5: $\forall\ e \in \mathcal{E}_{out}$ insert e into the rung as an output (output energize) and the rung ends. As illustrated in Figure 6.21, e_1^{out} and e_2^{out} are event outputs for t1. e_1^{out} and e_2^{out} are inserted into the rung as outputs as shown in Figure 6.25.

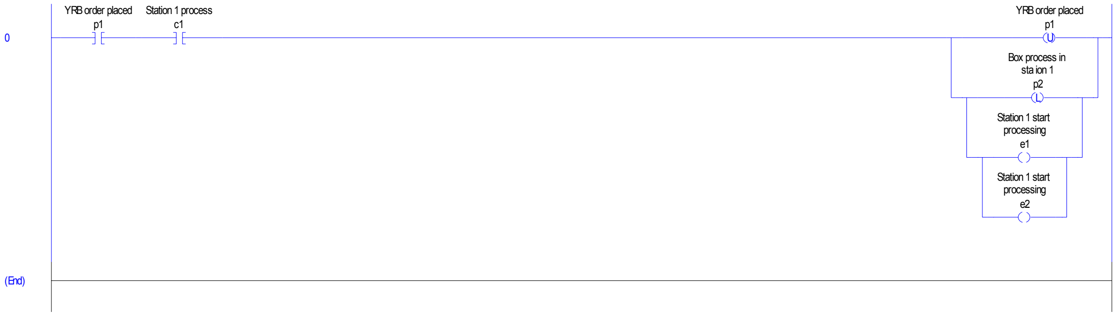


Figure 6.25: HAS-200 Control Model Insert Event Output for t1 in LLD

Steps 2.1 through 2.2 transform the places that interact with condition C_2. For convenience purposes, the portion of the HAS-200 control model that concerns C_2 is shown in Figure 6.26.

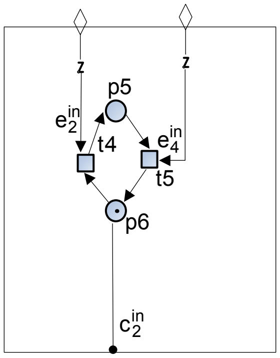


Figure 6.26: Portion of HAS-200 Control Model Pertaining to Condition C_2

Step 2: ∀ $c \epsilon C_{out}$, insert a rung into the Ladder Logic Diagram and:

Step 2.1: ∀ $p \epsilon c \bullet$ insert p in the rung as an input variable. As illustrated in Figure 6.26, p6 is an input place for C_2. p6 is inserted into the rung as an input variable as shown in Figure 6.27.

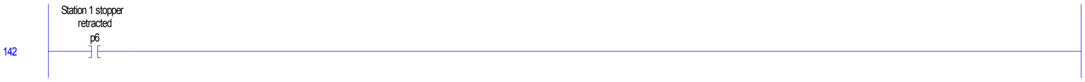


Figure 6.27: HAS-200 Control Model Insert Input Place for C_1 in LLD

Step 2.2: Insert c in the rung as an output variable and end the rung. C_2 is inserted as an output variable into the same rung of step 2.1 as shown in Figure 6.28.

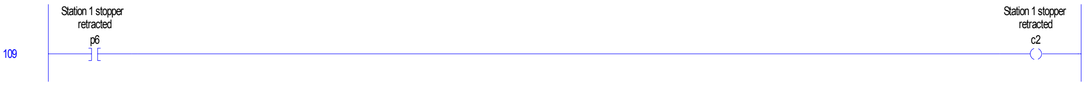


Figure 6.28: HAS-200 Control Model Insert Output C_2 in LLD

The resulting Ladder Logic Diagram is shown in Figure 6.29.

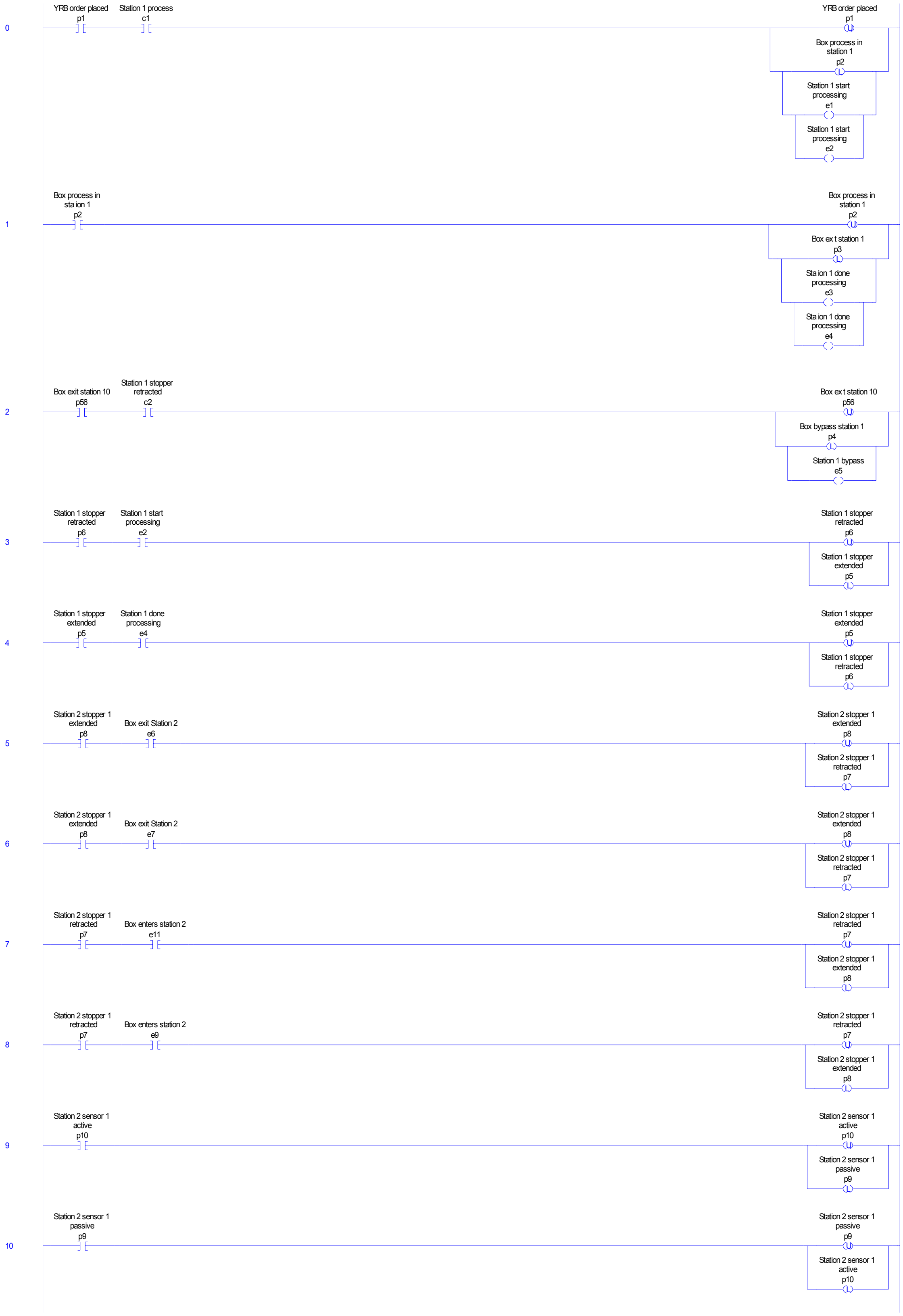


Figure 6.29: Ladder Logic Diagram for the HAS-200 Control Model

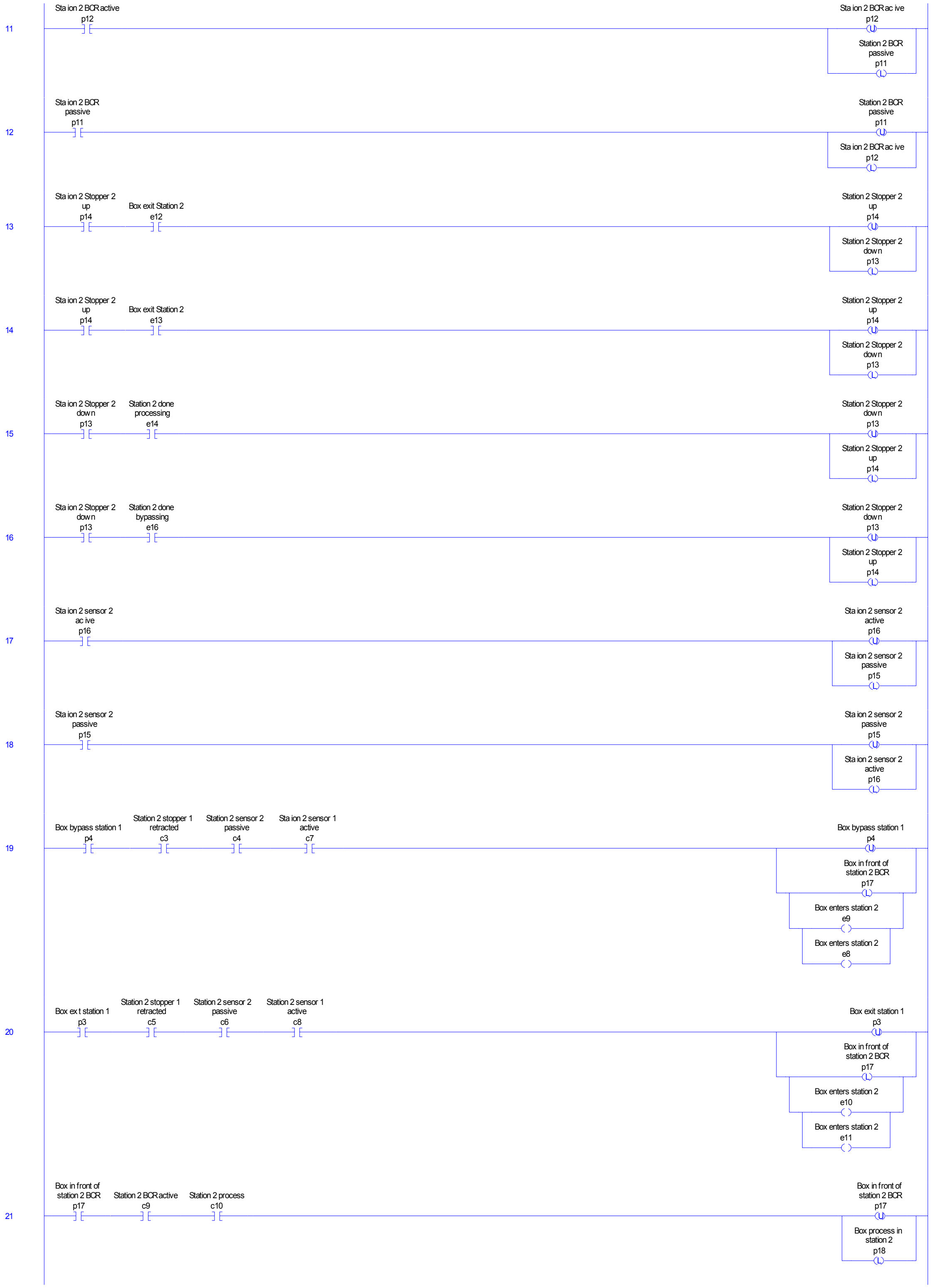


Figure 6.29: (Continued)

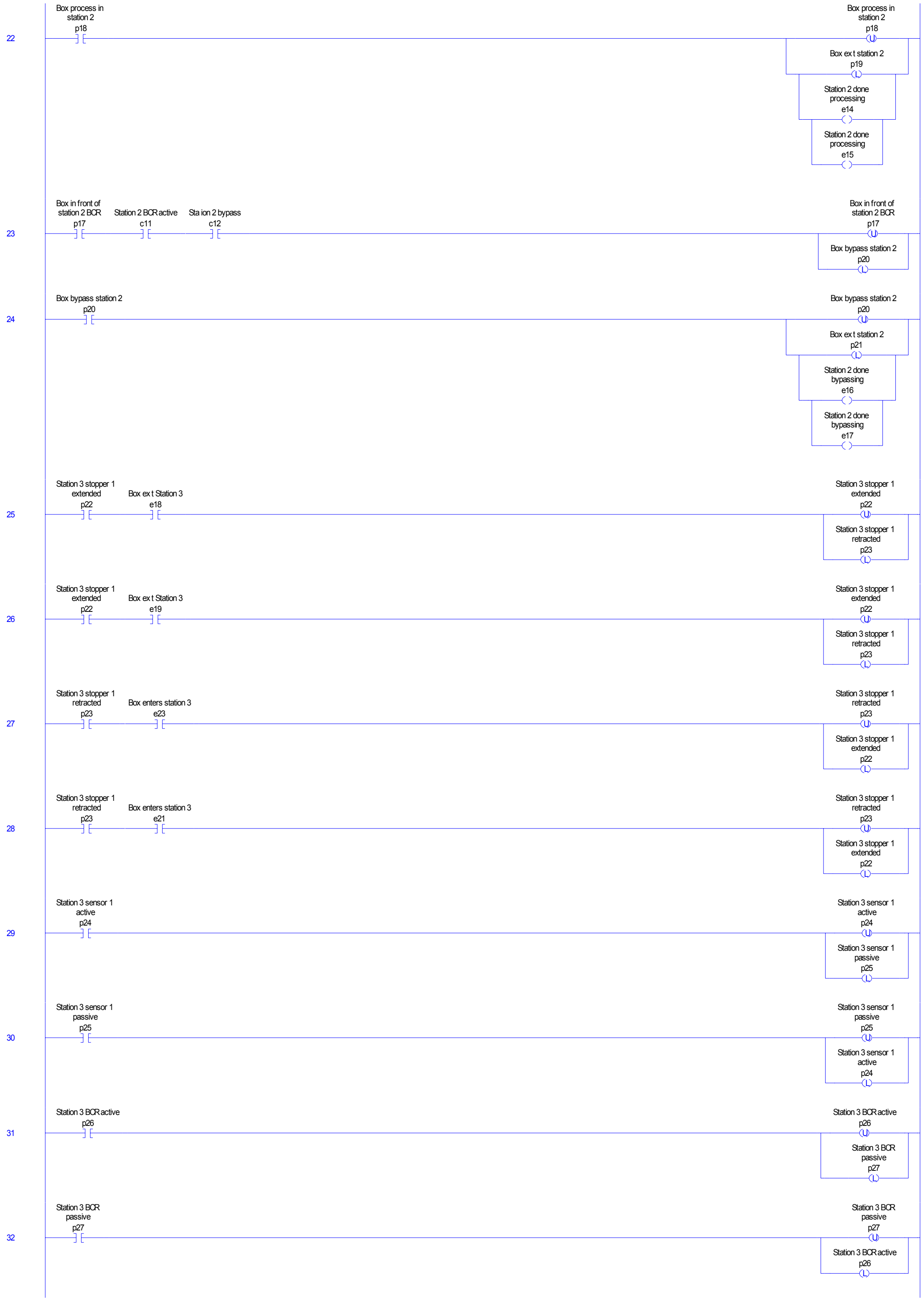


Figure 6.29: (Continued)

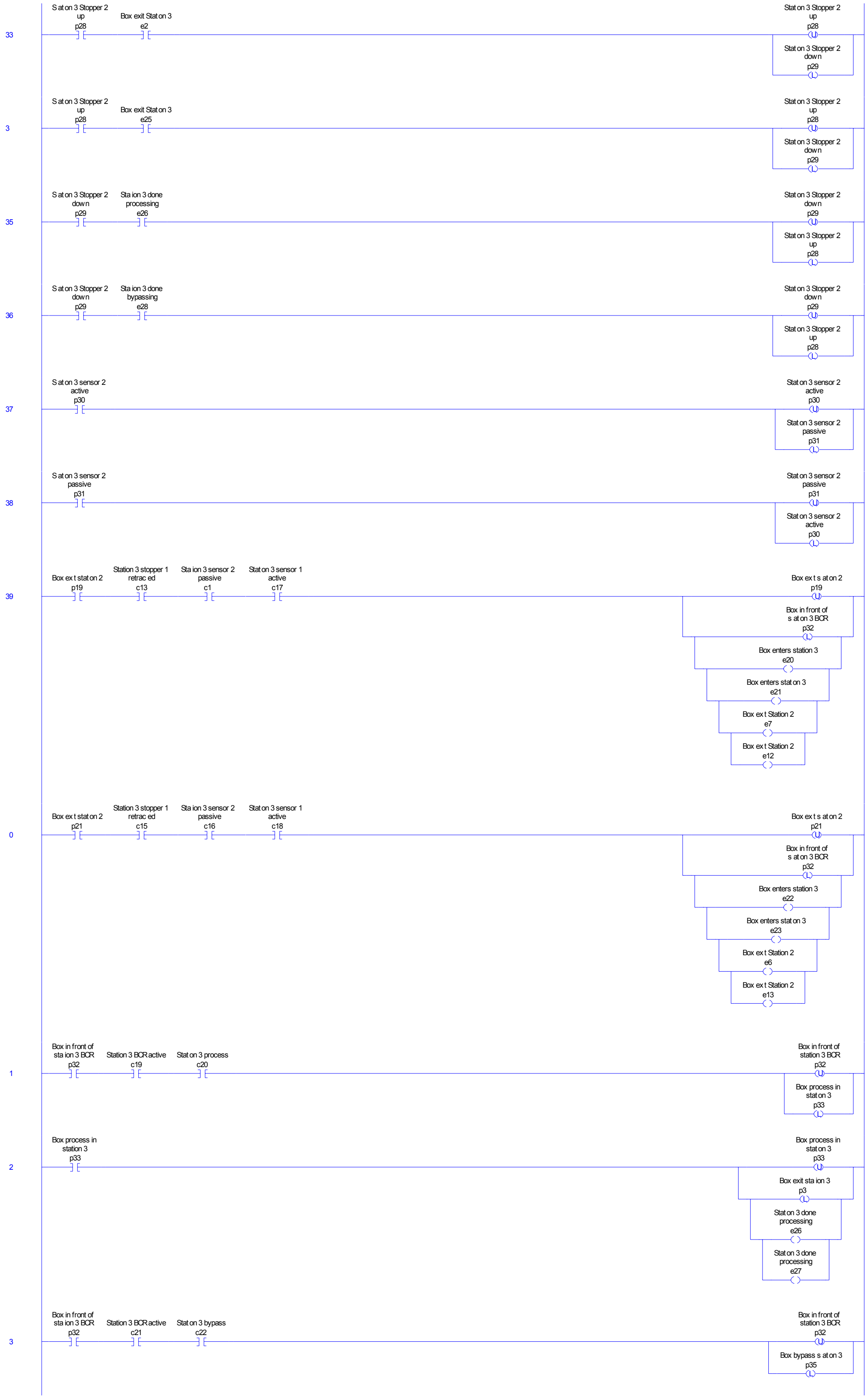


Figure 6.29: (Continued)

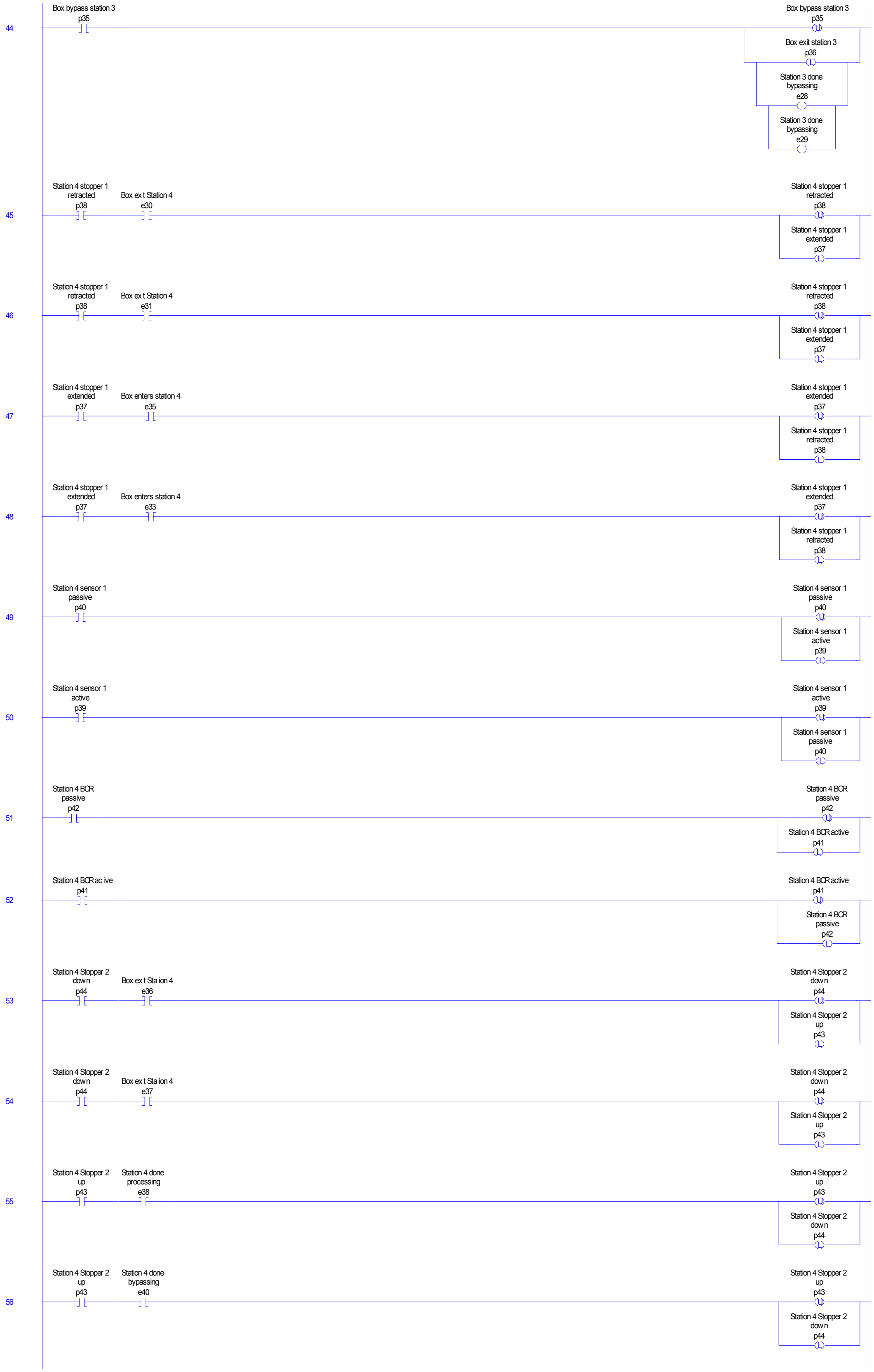


Figure 6.29: (Continued)

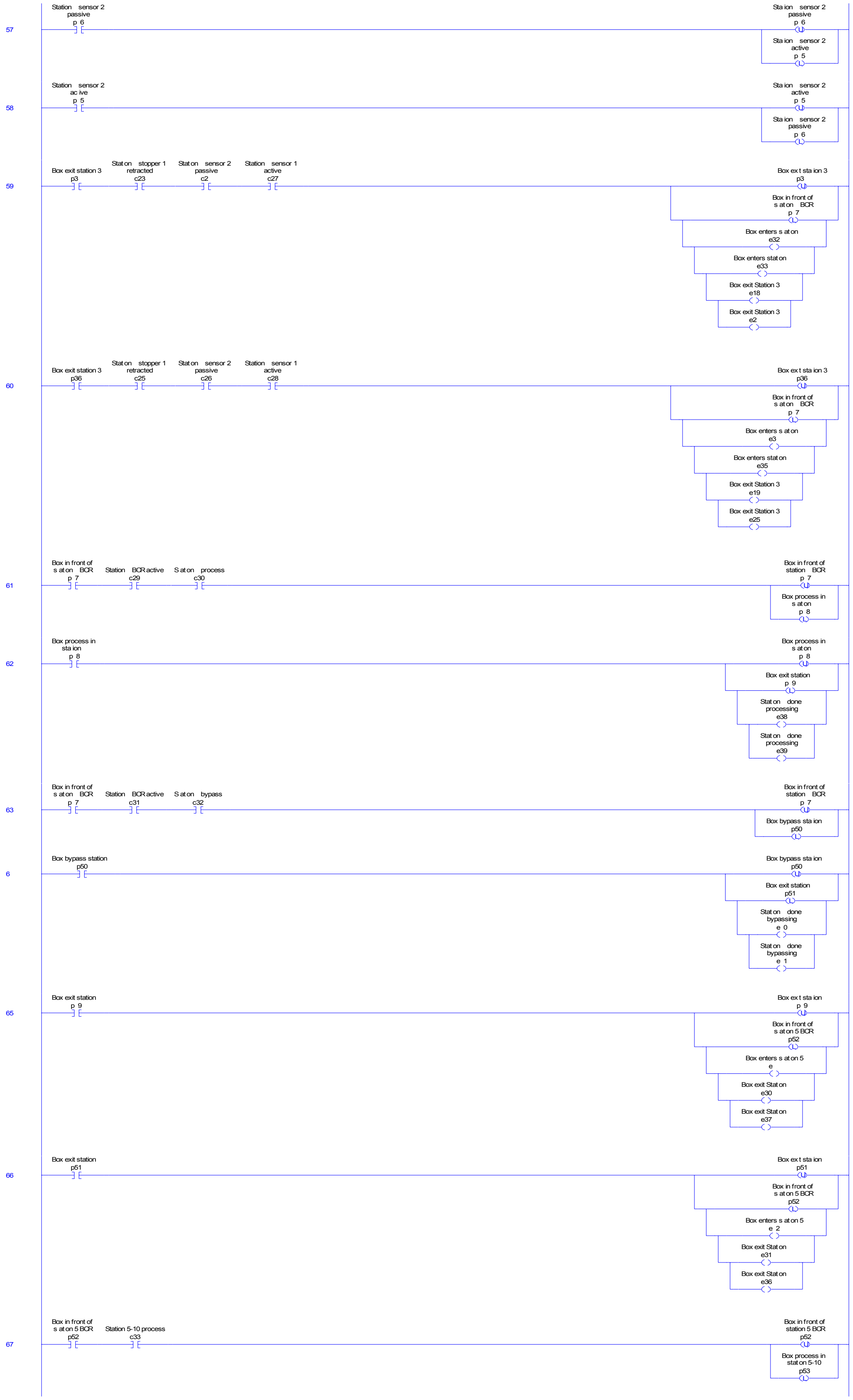


Figure 6.29: (Continued)

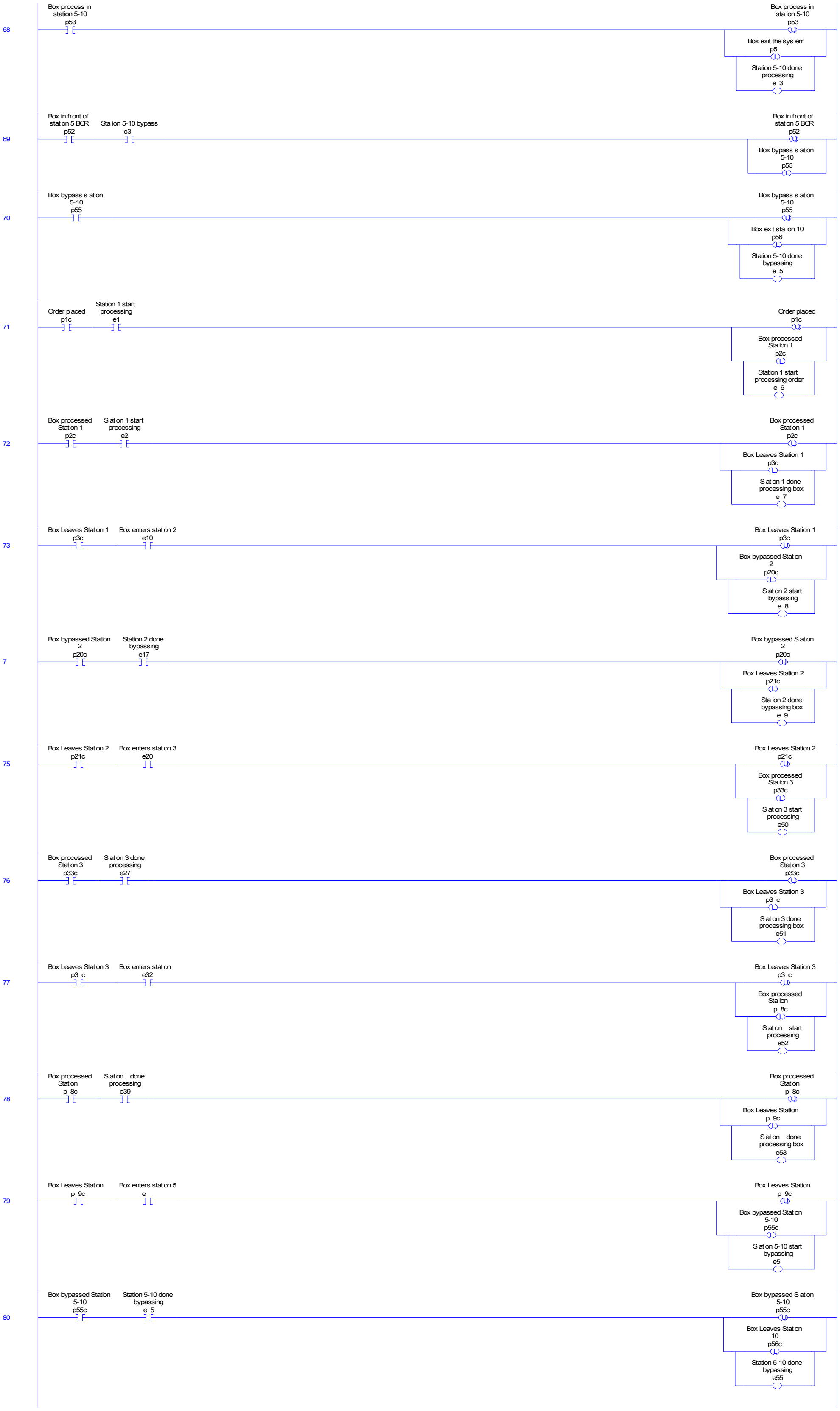


Figure 6.29: (Continued)

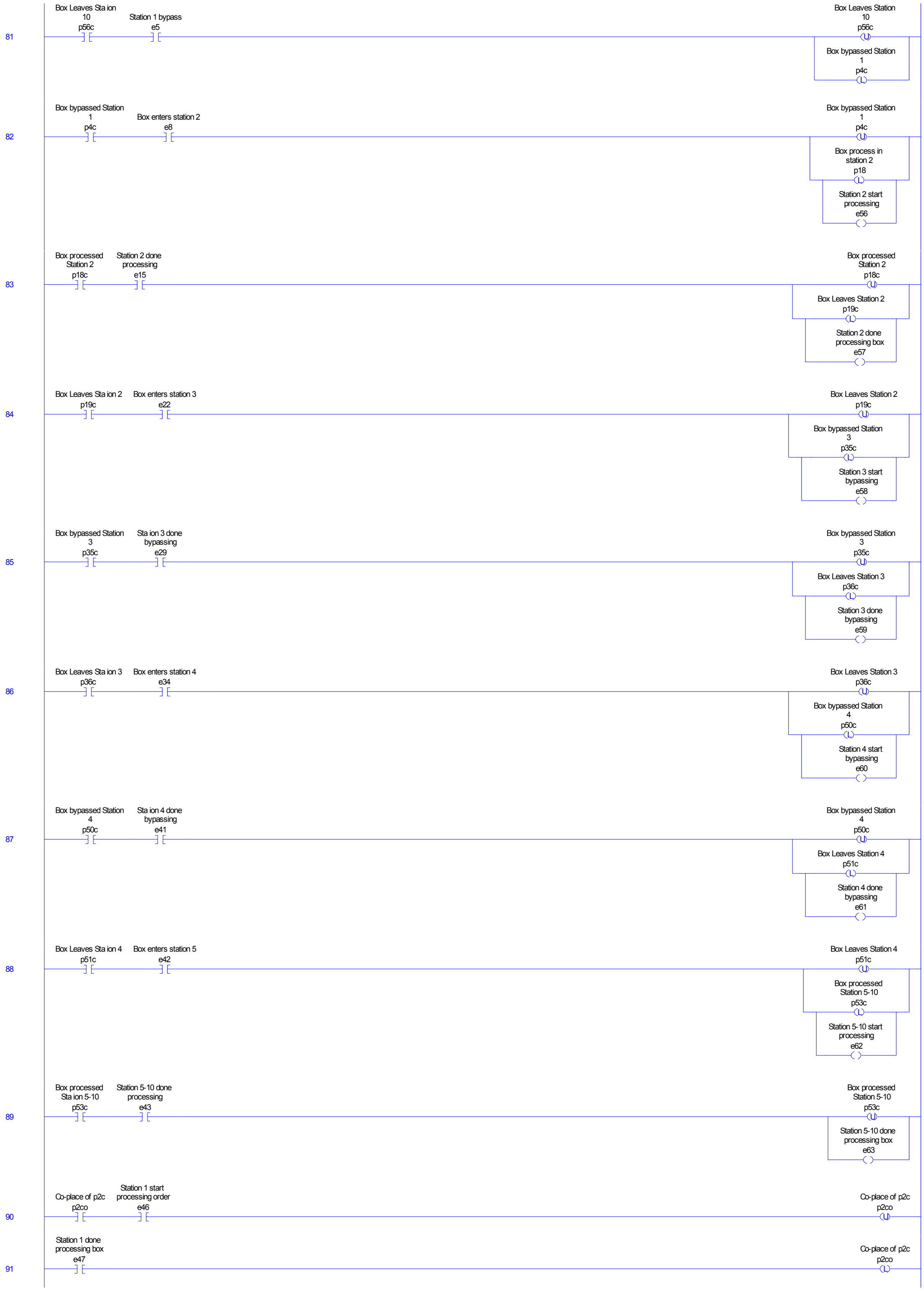


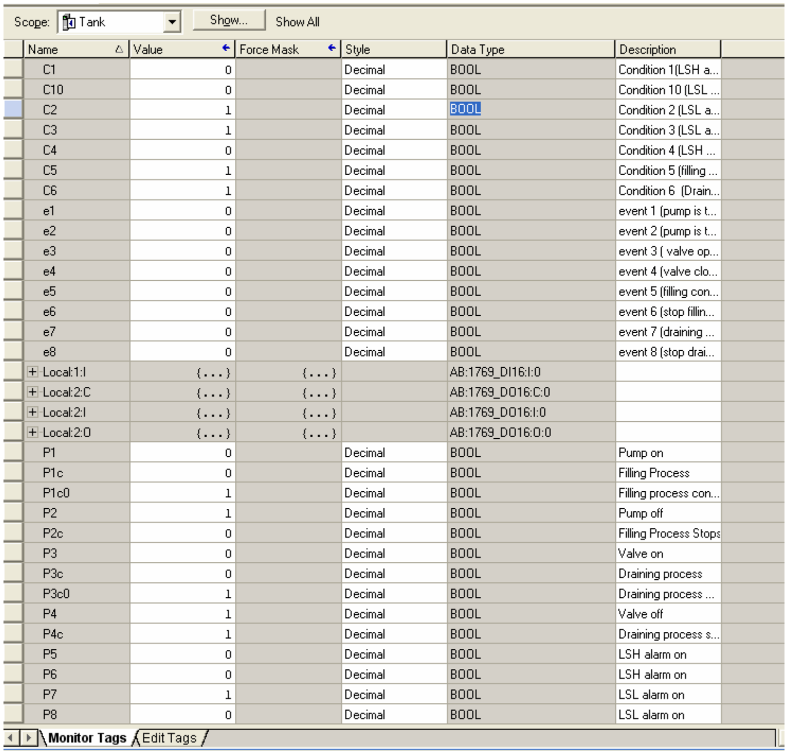
Figure 6.29: (Continued)

Figure 6.29: (Continued)

Figure 6.29: (Continued)

6.7 HAS-200 Algorithm Implementation

The Ladder Logic Diagram shown in Figure 6.29 was created using RSLogix 5000 version 12 and downloaded into an Allen Bradley CompactLogix system model 1769 L30 [37]. The Ladder Logic Diagram was verified and no errors were found. The initial conditions of the HAS-200 were simulated by toggling the bits of the input variables. After the bits were toggled the Ladder Logic Diagram was sent online and the tag values were monitor through an RSLogix interface named “monitor tags” as shown in Figure 6.30. The input variables for the level sensors were toggled in the sequence they should logically change states to simulate their response to the physical system. The Ladder Logic Diagram responses to the toggled bits were correct.



Name	Value	Force Mask	Style	Data Type	Description
C1	0		Decimal	BOOL	Condition 1(LSH a...
C10	0		Decimal	BOOL	Condition 10 (LSL ...
C2	1		Decimal	BOOL	Condition 2 (LSL a...
C3	1		Decimal	BOOL	Condition 3 (LSL a...
C4	0		Decimal	BOOL	Condition 4 (LSH ...
C5	1		Decimal	BOOL	Condition 5 (filling ...
C6	1		Decimal	BOOL	Condition 6 (Drain...
e1	0		Decimal	BOOL	event 1 (pump is t...
e2	0		Decimal	BOOL	event 2 (pump is t...
e3	0		Decimal	BOOL	event 3 (valve op...
e4	0		Decimal	BOOL	event 4 (valve clo...
e5	0		Decimal	BOOL	event 5 (filling con...
e6	0		Decimal	BOOL	event 6 (stop fillin...
e7	0		Decimal	BOOL	event 7 (draining ...
e8	0		Decimal	BOOL	event 8 (stop drai...
+ Local:1:I	{...}	{...}		AB:1769_DI16:I:0	
+ Local:2:C	{...}	{...}		AB:1769_DO16:C:0	
+ Local:2:I	{...}	{...}		AB:1769_DO16:I:0	
+ Local:2:O	{...}	{...}		AB:1769_DO16:O:0	
P1	0		Decimal	BOOL	Pump on
P1c	0		Decimal	BOOL	Filling Process
P1c0	1		Decimal	BOOL	Filling process con...
P2	1		Decimal	BOOL	Pump off
P2c	0		Decimal	BOOL	Filling Process Stops
P3	0		Decimal	BOOL	Valve on
P3c	0		Decimal	BOOL	Draining process
P3c0	1		Decimal	BOOL	Draining process ...
P4	1		Decimal	BOOL	Valve off
P4c	1		Decimal	BOOL	Draining process s...
P5	0		Decimal	BOOL	LSH alarm on
P6	0		Decimal	BOOL	LSH alarm on
P7	1		Decimal	BOOL	LSL alarm on
P8	0		Decimal	BOOL	LSL alarm on

Figure 6.30: RSLogix Monitor Tags Interface for the HAS-200

Chapter 7: Conclusion, Contributions, and Future Research

This chapter provides an overview of the main goal, objectives, and methodology use in this thesis. It summarizes the main findings and discusses their significance. Finally, future areas of research are presented.

7.1 Conclusions

The goal of this thesis was to automatically generate a PLC programming language for manufacturing systems requiring complex control models. This goal was achieved by completing the following three objectives:

- A transformation algorithm to automatically generate a Ladder Logic Diagram from NCES models was developed and presented in Chapter 4. In [10] Rausch and Krogh proposed the transformation of NCES into Instruction List (IL); the algorithm developed in this thesis was based on the ideas proposed in there paper. However, the algorithm in this thesis transforms NCES into Ladder Logic Diagram instead of IL. The structural components of the NCES made it easier to categorize them into Ladder Logic Diagram components.
- This thesis successfully used NCES to model a complex manufacturing system. Previous papers have addressed the modeling of simple DEDS. The significance of this research was to obtain a

NCES control model for a complex manufacturing system such as the HAS-200. The NCES control model of the HAS-200 YRB filling sequence was developed and presented in Section 6.4. The NCES input/output structure proved to capture accurately the behavior of the physical devices that form part of the HAS-200 system. The sequence specification and locking controller models ensured that the model meet the desired behavior. In addition, the HAS-200 control model was analyzed and verified in Section 6.5. The analysis demonstrated the correctness of the NCES control model by identifying the presence or absence of desirable behavioral properties. Furthermore, the Ladder Logic Diagram obtained by transforming the NCES control model will inherit the behavioral properties. Even though, the analysis tools were useful in determining the correctness of the model, the reachable state results were very large and difficult to follow visually.

- The HAS-200 NCES control model was converted into a Ladder Logic Diagram as shown in Section 6.6. The verification of the Ladder Logic Diagram obtained from the transformation is shown in Section 6.7. The resulting Ladder Logic Diagram was very long, but it was easy to follow and understand. The input and output contacts of the Ladder Logic Diagram were toggled to simulate the HAS-200 behavior. The results from the test proved that the Ladder Logic Diagram inherited the behavioral properties desired for the system. The algorithm demonstrates its ability to transform complex manufacturing system.

7.2 Contributions

One of the most important contributions of this thesis is the development of the transformation algorithm. The advantages of using this algorithm are presented below:

- The algorithm converts NCES into Ladder Logic Diagrams, which is one of the most common PLC programming languages.
- The algorithm is able to transform all the components of NCES (places, conditions, events, and transitions) into input and output contacts within the Ladder Logic Diagrams.
- The algorithm has the ability to transformed complex manufacturing control models into Ladder Logic Diagrams.

Another contribution of this research is the successful modeling of a complex manufacturing system using NCES. The lack of documentation indicates that NCES is a modeling formalism that could benefit from further research. This thesis adds valuable documentation about modeling a complex manufacturing system using NCES.

7.3 Future Research

The lack of documentation for modeling complex manufacturing systems using NCES demonstrates that this is an area that needs extensive research. One area of vital interest is determining a better way to verify the correctness of NCES control models. Even though the SESA analysis report was very useful to identify the behavioral properties, the reachable states were very large and difficult to follow visually. For simple manufacturing systems this tool might be appropriate, but for complex systems with 100 places and transitions this tool becomes difficult to use. The use of invariants could be a possible solution to accomplish the analysis of the structural properties [4]. Invariants are based on the incidence matrix and state equation illustrated in Section 1.3.4.2 of this thesis.

Another area of interest for future work would be to investigate the advantages of using NCES against other types of modeling formalism such as coloured petri nets or timed petri nets. Coloured petri nets token structure presents the ability of identifying each token. This might be useful for the modeling of the HAS-200, because there may be cases were the identification of each box process might be needed.

Finally, another area of interest could be the development of a software tool to model and transform NCES. This software tool could have the capability of inserting the NCES components and automatically obtaining a Ladder Logic Diagram. Using NCES to model the physical devices of a system was not difficult but using Microsoft Visio to develop the model was a little tedious. Furthermore, using the algorithm of this thesis as a basis to develop a software program to automatically transform the NCES model to Ladder Logic Diagram will significantly improve the modeling of complex manufacturing systems.

List of References

[1] M. Rausch and H.-M. Hanisch, “Net Condition/Event Systems with Multiple Condition Outputs,” in *proceedings ETFA Conf*., Paris, France, Oct. 1995.

[2] R. S. Sreenivas and B. H. Krogh, “On Condition/Event Systems with Discrete State Realization,” in *Discrete Event Dynamics Systems: Theory and Applications*, 2(1), pp. 209-236, 1991.

[3] R. Zurawski and M. C. Zhou, “Petri Nets and Industrial Applications: A Tutorial,” *IEEE Trans. Ind. Electron.*, vol. 41, no. 6, pp. 567-583, 1994.

[4] T. Murata, “Petri Nets: Properties, Analysis and Applications,” *in proceedings IEEE*, vol. 77, no.4, pp. 541-580, 1989.

[5] L. Pinzon, M. A. Jafari, H.-M. Hanisch and P. Zhao, “Modeling Admissible Behavior Using Event Signals,” *in IEEE Conf. Systems, Man and Cybernetics*, vol. 34, no.3, pp. 1435-1448, 2004.

[6] H. J. Genrich and G. Thieler-Mevissen, “The Calculus of Facts,” in *Mathematical Foundation of Computer Science*, New York: Springer-Verlag, pp. 588-595, 1976.

[7] M. Rausch and H.-M. Hanisch, “Synthesis of Supervisory Controllers Based on a Novel Representation of Condition/Event Systems,” *in proceedings IEEE Conf. Systems, Man and Cybernetics*, Vancouver, BC, Canada, pp 3069-3074, Oct. 22-25, 1995.

[8] L.E. Pinzon, M. A. Jafari, H.-M. Hanisch and T. Boucher, “A Comparative Study of Synthesis Methods for Discrete Event Controllers,” 1997.

[9] P.J Ramadge and W. M. Wonham, “Supervisory Control of a Class of Discrete-Event Processes,” *SIAM Control Optimization*, 25(1), 1987.

[10] R. S. Sreenivas and B. H. Krogh, “Petri Net based models for Condition/Event Systems,” *in proceedings of 1991 American Control Conference*, volume3, pp. 2899-2904, Boston, USA, 1991.

[11] M. Rausch and B. H. Krogh, “Transformation Between Different Model Forms in Discrete Event Systems,” *in IEEE Conf. Systems, Man and Cybernetics*, volume 3, pp. 2841-2846, Orlando, USA, October 1997.

[12] M. A. Jafari, “Supervisory Control Specification and Synthesis,” *in Zhou, M.C.: Petri Nets in Flexible and Agile Automation*, pages 337-368, Dordrecht, NL: Kluwer Academic Publishers, 1995.

[13] J.S. Ostroff, “Temporal Logic for Real-Time Systems,” *Research Studies Press*, Taunton, UK, 1989.

[14] H.-M. Hanisch, “Analysis of Place/Transition Nets with Timed Arcs and Its Application to Batch Process Control,” pages 282-299. Lecture Notes in Computer Science 691. Springer-Verlag, Berlin, 1993.

[15] J.S. Ostroff, “Synthesis of Controllers for Real-Time Discrete Event Systems,” *in IEEE proceedings of the 28th conference on Decision and Control*, Tampa, Florida, December 1989.

[16] P.J Ramadge and W. M. Wonham, W.M. “Modular Feedback Logic for Discrete Event Systems,”' *SIAM Journal of Control and Optimization*, 25(5):1202-1218, September, 1987.

[17] A. Giua, “Petri Nets as Discrete Event Models for Supervisory Control,” *PhD Thesis.* Dept. of Computer and Systems Engineering, Rensselaer Polytechnic Institute. Troy, N.Y. July 1992.

[18] J. Stenerson, “Fundamentals of Programmable Logic Controllers, Sensors and Communications,” 2nd edition, New Jersey: Prentice Hall, 1999.

[19] F. Petruzella, “Programmable Logic Controllers,” 3rd edition, NY: McGraw-Hill Companies, 2005.

[20] IEC Standard 1131-3: Programmable Controllers- Part 3: Programming Languages, International Electrotechnical Commission (IEC), 1993.

[21] B. A. Brandin, “The Real-Time Supervisory Control of an Experimental Manufacturing Cell,” *in IEEE Transactions on Robotics and Automation*, Vol. 12(1), February 1996.

[22] M. Fabian and A. Hellgren, “PLC-based Implementation of Supervisory Control for Discrete Event Systems,” *in proceedings of the 37the IEEE conference on Decision & Control*, Tampa, Florida, USA, December 1998.

[23] J. Liu and H. Darabi, “Ladder Logic Implementation of Ramadge-Wonham Supervisory Controller,” *in proceedings of the 6th International Workshop on Discrete Event System WODES’02,* Zaragoza, Spain, October 2002.

[24] T. Suesut, P. Inban, P. Nilas, P. Rerngreun and S. Gulphanich, “Interpretation Petri Net Model to IEC 1131-3: LD For programmable Logic Controller,” *in proceedings of the IEEE 2004 Conference on Robotics Automation and Mechanics,* Singapore, December 2004.

[25] T. Satoh, H. Oshima, K. Nose, and S. Kumagai, “Automatic Generation System of Ladder List Program By Petri Net,” *in Proceedings of the IEEE International Workshop on Emerging Technologies on Factory Automation - Technology For The Intelligent Factory,* pp. 128-133, 1992.

[26] S. Rattigan, “Using Petri Nets to Develop Programs for PLC Systems,” *Lecture Notes in Computer Science 616: Application and Theory of Petri Nets,* pp. 368 – 372, 1992.

[27] M. A. Jafari, T.O. Boucher, “A Rule-Base System For Generating Ladder Logic Control Program from a High Level System Model,” *in Journal of Intelligent Manufacturing*, 5, pp. 103-120, 1994.

[28] AH Jones, M Uzam, and N. Ajlouni “Design of Discrete Event Control Systems for Programmable Logic Controllers Using T-Timed Petri Nets,” *in IEEE International Symposium on Computer-Aided Control System Design - CACSD’96,*Michigan, USA, September 15-17, 1996.

[29] AH Jones, M Uzam, and N. Ajlouni “Conversion of Petri Net Controllers for Manufacturing Systems into Ladder Logic Diagrams,” in IEEE, 1996.

[30] A. H. Jones, M. Uzam, A. H. Khan, D. Karimzadgan, and S. B. Kenway, “ A General Methodology for Converting Petri Nets into Ladder Logic: The TPLL Methodlogy,” *in proceedings of the CIMAT’96*, France, pp. 357-362.

[31] M. Uzam and A. H. Jones, “Design of a Discrete Event Control System for a Manufacturing System Using Token Passing Ladder Logic,” *in CESA’96 IMACS Conference*, Lille, France July 9-12, 1996.

[32] M. Uzam and AH Jones, **“**Towards a Unified Methodology for Converting Coloured Petri Net Controllers into Ladder Logic Using TPU: Part I – Methodology,” *in Intemational Workshop on Discrete Event Systems, WODES’96,* Edinburgh, UK, August 1996.

[33] A.H. Jones and M Uzam, **“**Towards a Unified Methodology for Converting Coloured Petri Net Controllers into Ladder Logic Using TPLJ: Part II - An Application,” *in Intemational Workshop on Discrete Event Systems, WODES’96,* Edinburgh, UK, August 1996.

[34] V.N. Dubinin, H.-M. Hanisch, and S Karras, **“**Building of Reachability Graph Extractions Using a Graph Rewriting System,” *in proceedings of the 7th International Conference of Science and Technology, NITis 2006,* Penza, Russia, *vol.1, pp160-171,* 2006.

[35] Berthold Paul “SESA Tool Description”, Humboldt University Berlin 2002. http://www.ele.auckland.ac.nz/~vyatkin/nces/imatch/SESA.pdf

[36] Peter H. Starke, and Stephen Roch, “SESA: Signal Net System Analyzer”, Humboldt University Berlin 1999. http://www2.informatik.hu-berlin.de/lehrstuehle/automaten/tools/

[37] Rockwell Automation Web site. Available at: http://www.rockwellautomation.com/. Accessed February 3, 2006.

[38] MATEC Emerging Skills for Highly Automated Environments Web site. Available at: http://www.matec.org/has/. Accessed February 3, 2006.

Appendices

Appendix A: SESA Reachable States for the Tank Control Model

```
State nr.    1
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  0  1  1  0  0  0
0  1  1  1
=={t2,t9,t14}=> s2
=={t2,t6,t9,t14}=> s8
=={t2,t7,t9,t14}=> s47
=={t2,t6,t7,t9,t14}=> s3
=={t6,t7}=> s11
=={t7}=> s35
=={t6}=> s7
State nr.    2
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  0  1  1  0  1  0
0  0  0  1
=={t6,t7}=> s3
=={t7}=> s47
=={t6}=> s8
State nr.    3
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  1  0  0  1  1  0
0  0  0  1
=={t1,t10,t13}=> s4
=={t1,t5,t10,t13}=> s39
=={t1,t8,t10,t13}=> s9
=={t1,t5,t8,t10,t13}=> s40
=={t5,t8}=> s2
=={t8}=> s8
=={t5}=> s47
State nr.    4
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  1  0  0  1  0  1
0  0  1  1
=={t4,t11,t16}=> s5
=={t4,t5,t11,t16}=> s37
=={t4,t8,t11,t16}=> s10
=={t4,t5,t8,t11,t16}=> s6
=={t5,t8}=> s40
=={t8}=> s9
=={t5}=> s39
State nr.    5
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  1  0  0  1  0  0
1  0  1  0
=={t5,t8}=> s6
=={t8}=> s10
=={t5}=> s37
State nr.    6
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  0  1  1  0  0  0
1  0  1  0
=={t3,t12,t15}=> s1
=={t3,t6,t12,t15}=> s7
=={t3,t7,t12,t15}=> s35
=={t3,t6,t7,t12,t15}=> s11
=={t6,t7}=> s5
=={t7}=> s37
=={t6}=> s10
State nr.    7
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  1  0  1  0  0  0
0  1  1  1

=={t2,t9,t14}=> s8
=={t2,t4,t9,t14}=> s25
=={t2,t5,t9,t14}=> s2
=={t2,t4,t5,t9,t14}=> s13
=={t2,t7,t9,t14}=> s3
=={t2,t4,t7,t9,t14}=> s14
=={t2,t5,t7,t9,t14}=> s47
=={t2,t4,t5,t7,t9,t14}=> s48
=={t4,t5,t7}=> s31
=={t5,t7}=> s35
=={t4,t7}=> s32
=={t7}=> s11
=={t4,t5}=> s12
=={t5}=> s1
=={t4}=> s24
State nr.    8
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  1  0  1  0  1  0
0  0  0  1
=={t1,t10,t13}=> s9
=={t1,t5,t10,t13}=> s40
=={t1,t7,t10,t13}=> s4
=={t1,t5,t7,t10,t13}=> s39
=={t5,t7}=> s47
=={t7}=> s3
=={t5}=> s2
State nr.    9
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  1  0  1  0  0  1
0  0  1  1
=={t4,t11,t16}=> s10
=={t2,t4,t11,t16}=> s29
=={t4,t5,t11,t16}=> s6
=={t2,t4,t5,t11,t16}=> s19
=={t4,t7,t11,t16}=> s5
=={t2,t4,t7,t11,t16}=> s18
=={t4,t5,t7,t11,t16}=> s37
=={t2,t4,t5,t7,t11,t16}=> s38
=={t2,t5,t7}=> s27
=={t5,t7}=> s39
=={t2,t7}=> s17
=={t7}=> s4
```

Figure A.1: SESA Tank Reachable States for the Tank Control Model

```
=={t2,t5}=> s41
=={t5}=> s40
=={t2}=> s28
State nr.   10
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  1  0  1  0  0  0
1  0  1  0
=={t3,t12,t15}=> s7
=={t3,t5,t12,t15}=> s1
=={t3,t7,t12,t15}=> s11
=={t3,t5,t7,t12,t15}=> s35
=={t5,t7}=> s37
=={t7}=> s5
=={t5}=> s6
State nr.   11
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  1  0  0  1  0  0
0  1  1  1
=={t4,t5,t8}=> s12
=={t5,t8}=> s1
=={t4,t8}=> s24
=={t8}=> s7
=={t4,t5}=> s31
=={t5}=> s35
=={t4}=> s32
State nr.   12
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  0  1  1  0  0  0
0  1  1  1
=={t2,t9,t14}=> s13
=={t2,t3,t9,t14}=> s2
=={t2,t6,t9,t14}=> s25
=={t2,t3,t6,t9,t14}=> s8
=={t2,t7,t9,t14}=> s48
=={t2,t3,t7,t9,t14}=> s47
=={t2,t6,t7,t9,t14}=> s14
=={t2,t3,t6,t7,t9,t14}=> s3
=={t3,t6,t7}=> s11
=={t6,t7}=> s32
=={t3,t7}=> s35
=={t7}=> s31
=={t3,t6}=> s7
=={t6}=> s24
=={t3}=> s1
State nr.   13
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  0  1  1  0  1  0
0  0  0  1
=={t3,t6,t7}=> s3
=={t6,t7}=> s14
=={t3,t7}=> s47
=={t7}=> s48
=={t3,t6}=> s8
```

```
=={t6}=> s25
=={t3}=> s2
State nr.   14
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  1  0  0  1  1  0
0  0  0  1
=={t1,t10,t13}=> s15
=={t1,t5,t10,t13}=> s44
=={t1,t8,t10,t13}=> s26
=={t1,t5,t8,t10,t13}=> s16
=={t5,t8}=> s13
=={t8}=> s25
=={t5}=> s48
State nr.   15
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  1  0  0  1  0  1
0  0  1  1
=={t5,t8}=> s16
=={t8}=> s26
=={t5}=> s44
State nr.   16
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  0  1  1  0  0  1
0  0  1  1
=={t2,t3,t6,t7}=> s17
=={t3,t6,t7}=> s4
=={t2,t6,t7}=> s45
=={t6,t7}=> s15
=={t2,t3,t7}=> s27
=={t3,t7}=> s39
=={t2,t7}=> s42
=={t7}=> s44
=={t2,t3,t6}=> s28
=={t3,t6}=> s9
=={t2,t6}=> s43
=={t6}=> s26
=={t2,t3}=> s41
=={t3}=> s40
=={t2}=> s46
State nr.   17
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  1  0  0  1  0  1
0  0  1  1
=={t4,t11,t16}=> s18
=={t1,t4,t11,t16}=> s5
=={t4,t5,t11,t16}=> s38
=={t1,t4,t5,t11,t16}=> s37
=={t4,t8,t11,t16}=> s29
=={t1,t4,t8,t11,t16}=> s10
=={t4,t5,t8,t11,t16}=> s19
=={t1,t4,t5,t8,t11,t16}=> s6
=={t1,t5,t8}=> s40
=={t5,t8}=> s41
```

Figure A.1: (Continued)

```
=={t1,t8}=> s9
=={t8}=> s28
=={t1,t5}=> s39
=={t5}=> s27
=={t1}=> s4
State nr.   18
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  1  0  0  1  0  0
1  0  1  0
=={t1,t5,t8}=> s6
=={t5,t8}=> s19
=={t1,t8}=> s10
=={t8}=> s29
=={t1,t5}=> s37
=={t5}=> s38
=={t1}=> s5
State nr.   19
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  0  1  1  0  0  0
1  0  1  0
=={t3,t12,t15}=> s20
=={t3,t6,t12,t15}=> s30
=={t3,t7,t12,t15}=> s36
=={t3,t6,t7,t12,t15}=> s21
=={t6,t7}=> s18
=={t7}=> s38
=={t6}=> s29
State nr.   20
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  0  1  1  0  0  0
0  1  1  1
=={t6,t7}=> s21
=={t7}=> s36
=={t6}=> s30
State nr.   21
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  1  0  0  1  0  0
0  1  1  1
=={t1,t4,t5,t8}=> s12
=={t4,t5,t8}=> s22
=={t1,t5,t8}=> s1
=={t5,t8}=> s20
=={t1,t4,t8}=> s24
=={t4,t8}=> s34
=={t1,t8}=> s7
=={t8}=> s30
=={t1,t4,t5}=> s31
=={t4,t5}=> s33
=={t1,t5}=> s35
=={t5}=> s36
=={t1,t4}=> s32
=={t4}=> s23
=={t1}=> s11
```

```
State nr.   22
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  0  1  1  0  0  0
0  1  1  1
=={t3,t6,t7}=> s21
=={t6,t7}=> s23
=={t3,t7}=> s36
=={t7}=> s33
=={t3,t6}=> s30
=={t6}=> s34
=={t3}=> s20
State nr.   23
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  1  0  0  1  0  0
0  1  1  1
=={t1,t5,t8}=> s12
=={t5,t8}=> s22
=={t1,t8}=> s24
=={t8}=> s34
=={t1,t5}=> s31
=={t5}=> s33
=={t1}=> s32
State nr.   24
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  1  0  1  0  0  0
0  1  1  1
=={t2,t9,t14}=> s25
=={t2,t3,t9,t14}=> s8
=={t2,t5,t9,t14}=> s13
=={t2,t3,t5,t9,t14}=> s2
=={t2,t7,t9,t14}=> s14
=={t2,t3,t7,t9,t14}=> s3
=={t2,t5,t7,t9,t14}=> s48
=={t2,t3,t5,t7,t9,t14}=> s47
=={t3,t5,t7}=> s35
=={t5,t7}=> s31
=={t3,t7}=> s11
=={t7}=> s32
=={t3,t5}=> s1
=={t5}=> s12
=={t3}=> s7
State nr.   25
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  1  0  1  0  1  0
0  0  0  1
=={t1,t10,t13}=> s26
=={t1,t3,t10,t13}=> s9
=={t1,t5,t10,t13}=> s16
=={t1,t3,t5,t10,t13}=> s40
=={t1,t7,t10,t13}=> s15
=={t1,t3,t7,t10,t13}=> s4
=={t1,t5,t7,t10,t13}=> s44
=={t1,t3,t5,t7,t10,t13}=> s39
```

Figure A.1: (Continued)

```
=={t3,t5,t7}=> s47
=={t5,t7}=> s48
=={t3,t7}=> s3
=={t7}=> s14
=={t3,t5}=> s2
=={t5}=> s13
=={t3}=> s8
State nr.   26
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  1  0  1  0  0  1
0  0  1  1
=={t2,t3,t5,t7}=> s27
=={t3,t5,t7}=> s39
=={t2,t5,t7}=> s42
=={t5,t7}=> s44
=={t2,t3,t7}=> s17
=={t3,t7}=> s4
=={t2,t7}=> s45
=={t7}=> s15
=={t2,t3,t5}=> s41
=={t3,t5}=> s40
=={t2,t5}=> s46
=={t5}=> s16
=={t2,t3}=> s28
=={t3}=> s9
=={t2}=> s43
State nr.   27
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  0  1  0  1  0  1
0  0  1  1
=={t6,t8}=> s28
=={t8}=> s41
=={t6}=> s17
State nr.   28
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  1  0  1  0  0  1
0  0  1  1
=={t4,t11,t16}=> s29
=={t1,t4,t11,t16}=> s10
=={t4,t5,t11,t16}=> s19
=={t1,t4,t5,t11,t16}=> s6
=={t4,t7,t11,t16}=> s18
=={t1,t4,t7,t11,t16}=> s5
=={t4,t5,t7,t11,t16}=> s38
=={t1,t4,t5,t7,t11,t16}=> s37
=={t1,t5,t7}=> s39
=={t5,t7}=> s27
=={t1,t7}=> s4
=={t7}=> s17
=={t1,t5}=> s40
=={t5}=> s41
=={t1}=> s9
```

```
State nr.   29
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  1  0  1  0  0  0
1  0  1  0
=={t3,t12,t15}=> s30
=={t1,t3,t12,t15}=> s7
=={t3,t5,t12,t15}=> s20
=={t1,t3,t5,t12,t15}=> s1
=={t3,t7,t12,t15}=> s21
=={t1,t3,t7,t12,t15}=> s11
=={t3,t5,t7,t12,t15}=> s36
=={t1,t3,t5,t7,t12,t15}=> s35
=={t1,t5,t7}=> s37
=={t5,t7}=> s38
=={t1,t7}=> s5
=={t7}=> s18
=={t1,t5}=> s6
=={t5}=> s19
=={t1}=> s10
State nr.   30
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  1  0  1  0  0  0
0  1  1  1
=={t1,t4,t5,t7}=> s31
=={t4,t5,t7}=> s33
=={t1,t5,t7}=> s35
=={t5,t7}=> s36
=={t1,t4,t7}=> s32
=={t4,t7}=> s23
=={t1,t7}=> s11
=={t7}=> s21
=={t1,t4,t5}=> s12
=={t4,t5}=> s22
=={t1,t5}=> s1
=={t5}=> s20
=={t1,t4}=> s24
=={t4}=> s34
=={t1}=> s7
State nr.   31
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  0  1  0  1  0  0
0  1  1  1
=={t6,t8}=> s24
=={t8}=> s12
=={t6}=> s32
State nr.   32
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  1  0  0  1  0  0
0  1  1  1
=={t5,t8}=> s12
=={t8}=> s24
=={t5}=> s31
```

Figure A.1: (Continued)

```
State nr.   33
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  0  1  0  1  0  0
0  1  1  1
=={t6,t8}=> s34
=={t8}=> s22
=={t6}=> s23
State nr.   34
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  1  0  1  0  0  0
0  1  1  1
=={t1,t3,t5,t7}=> s35
=={t3,t5,t7}=> s36
=={t1,t5,t7}=> s31
=={t5,t7}=> s33
=={t1,t3,t7}=> s11
=={t3,t7}=> s21
=={t1,t7}=> s32
=={t7}=> s23
=={t1,t3,t5}=> s1
=={t3,t5}=> s20
=={t1,t5}=> s12
=={t5}=> s22
=={t1,t3}=> s7
=={t3}=> s30
=={t1}=> s24
State nr.   35
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  0  1  0  1  0  0
0  1  1  1
=={t6,t8}=> s7
=={t8}=> s1
=={t6}=> s11
State nr.   36
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  0  1  0  1  0  0
0  1  1  1
=={t6,t8}=> s30
=={t8}=> s20
=={t6}=> s21
State nr.   37
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  0  1  0  1  0  0
1  0  1  0
=={t6,t8}=> s10
=={t8}=> s6
=={t6}=> s5
State nr.   38
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  0  1  0  1  0  0
1  0  1  0
=={t6,t8}=> s29
=={t8}=> s19
=={t6}=> s18
State nr.   39
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  0  1  0  1  0  1
0  0  1  1
=={t6,t8}=> s9
=={t8}=> s40
=={t6}=> s4
State nr.   40
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  0  1  0  1  1  0  0  1
0  0  1  1
=={t2,t6,t7}=> s17
=={t6,t7}=> s4
=={t2,t7}=> s27
=={t7}=> s39
=={t2,t6}=> s28
=={t6}=> s9
=={t2}=> s41
State nr.   41
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  0  1  1  0  0  1
0  0  1  1
=={t6,t7}=> s17
=={t7}=> s27
=={t6}=> s28
State nr.   42
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  0  1  0  1  0  1
0  0  1  1
=={t6,t8}=> s43
=={t8}=> s46
=={t6}=> s45
State nr.   43
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  1  0  1  0  0  1
0  0  1  1
=={t1,t3,t5,t7}=> s39
=={t3,t5,t7}=> s27
=={t1,t5,t7}=> s44
=={t5,t7}=> s42
=={t1,t3,t7}=> s4
=={t3,t7}=> s17
=={t1,t7}=> s15
=={t7}=> s45
=={t1,t3,t5}=> s40
=={t3,t5}=> s41
=={t1,t5}=> s16
=={t5}=> s46
=={t1,t3}=> s9
```

Figure A.1: (Continued)

```
=={t3}=> s28
=={t1}=> s26
State nr.   44
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  0  1  1  0  0  1  0  1  0  1
0  0  1  1
=={t6,t8}=> s26
=={t8}=> s16
=={t6}=> s15
State nr.   45
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  1  0  0  1  0  1
0  0  1  1
=={t1,t5,t8}=> s16
=={t5,t8}=> s46
=={t1,t8}=> s26
=={t8}=> s43
=={t1,t5}=> s44
=={t5}=> s42
=={t1}=> s15
State nr.   46
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  0  1  1  0  0  1
0  0  1  1
=={t3,t6,t7}=> s17
=={t6,t7}=> s45
=={t3,t7}=> s27
=={t7}=> s42
=={t3,t6}=> s28
=={t6}=> s43
=={t3}=> s41
State nr.   47
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  0  1  0  1  0  1  1  0
0  0  0  1
=={t6,t8}=> s8
=={t8}=> s2
=={t6}=> s3
State nr.   48
P.nr:  1  2  3  4  5  6  7  8  9 10
11 12 13 14
toks:  1  0  1  0  0  1  0  1  1  0
0  0  0  1
=={t6,t8}=> s25
=={t8}=> s13
=={t6}=> s14
```

Figure A.1: (Continued)

Appendix B: Tank Algorithm Implementation

T= {t1, t2, t3, t4, t5, t6, t7, t8, t_1^c , t_2^c , t_3^c ,t_4^c , t_1^l , t_2^l , t_3^l , t_4^l }.

T_c = {t1, t2, t3, t4, t_1^c , t_2^c , t_3^c ,t_4^c , t_1^l , t_2^l , t_3^l , t_4^l }

T_u = {t5, t6, t7, t8}

C_N = {C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}}

C_{in} = { C_1^{in} , C_2^{in} , C_3^{in} , C_4^{in} , C_5^{in} , C_6^{in} , C_7^{in} , C_8^{in} , C_9^{in} , C_{10}^{in} }

C_{out} = { C_1^{out} , C_2^{out} , C_3^{out} , C_4^{out} , C_5^{out} , C_6^{out} , C_7^{out} , C_8^{out} , C_9^{out} , C_{10}^{out} }

Below are some of the steps to obtain the ladder diagram for the tank filling and draining example Figure 3.6 in Chapter 3. Step 1.1.1 for transition t1 and Step 2.1.1 for C_1 are shown in Chapter 5 Section 5.1. For convenience purposes, the portion of the tank control model that concerns t2 is shown in Figure B.1.

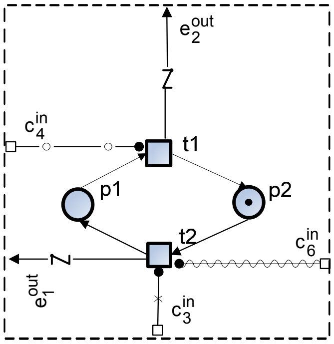


Figure B.1: Portion of Tank Control Model Pertaining to Transition t2

Step 1: $\forall\ t\bullet \epsilon\, \mathcal{T}_c$ insert a rung into the Ladder Logic Diagram and:

Step 1.2.1: $\forall\ p\ \epsilon \bullet t$ insert p into the rung as an input variable (examine on) and also as an unlatched output (output unlatch). p2 is an input place for t2 as shown in Figure B.1. p2 is inserted into the rung as an input variable and as an unlatched output as shown in Figure B.2.

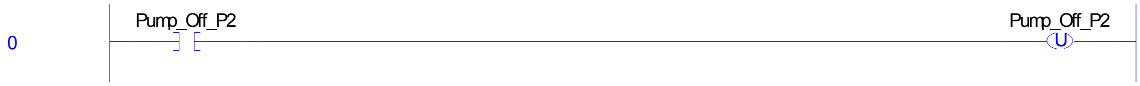


Figure B.2: Tank Control Model Inserted Input Place for t2 in LLD

Step 1.2.2: $\forall\ c\ \epsilon\, C_{in}$ insert c into the rung as an input variable. C_3 and C_6 are conditions inputs for t2 as shown in Figure B.1. C_3 and C_6 are inserted into the rung as input variables as shown in Figure B.3.

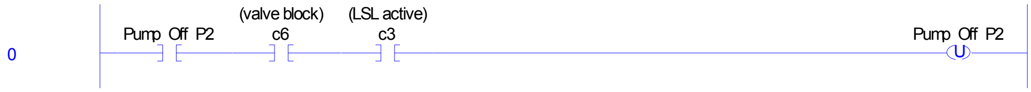


Figure B.3: Tank Control Model Inserted Condition Inputs for t2 in LLD

Step 1.2.3: $\forall\ e\ \epsilon\, \mathcal{E}_{in}$ insert e into the rung as an input variable. t2 has no event inputs. Therefore, no instructions are added to the rung.

Step 1.2.4: $\forall\ p \in t \bullet$ insert p into the rung as a latched output (output latch). P1 is an output place for t2 as shown in Figure B.1. p1 is inserted into the rung as a latched output as shown in Figure B.4.

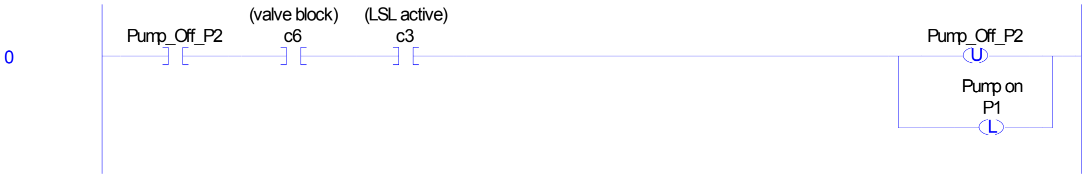


Figure B.4: Tank Control Model Inserted Output Place for t2 in LLD

Step 1.2.5: $\forall\ e \in \mathcal{E}_{out}$ insert e into the rung as an output (output energize) and the rung ends. e_1^{out} is an event output for t1 as shown in B.1. e_1^{out} is inserted into the rung as an output as shown in Figure B.5.

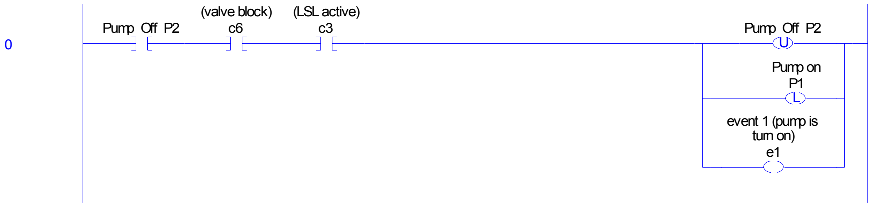


Figure B.5: Tank Control Model Inserted Event Output for t1 in LLD

For convenience purposes, the portion of the tank control model that concerns t_1^c is shown in Figure B.6.

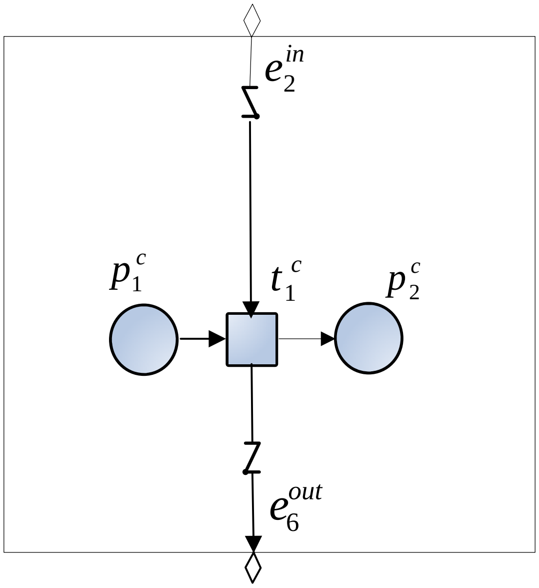


Figure B.6: t_1^c Portion of Tank Control Model

Step 1.3.1: $\forall\, p \,\epsilon \bullet t$ insert p into the rung as an input variable (examine on) and also as an unlatched output (output unlatch). p_1^c is an input place for t_1^c as shown in Figure B.6. p_1^c is inserted into the rung as an input variable and as an unlatched output as shown in Figure B.7.

Figure B.7: Tank Control Model Inserted Input Place for t_1^c in LLD

Step 1.3.2: $\forall\ c \in \mathcal{C}_{in}$ insert c into the rung as an input variable. t_1^c has no condition inputs. Therefore, no instructions are added to the rung.

Step 1.3.3: $\forall\ e \in \mathcal{E}_{in}$ insert e into the rung as an input variable. e_2^{in} is an event input for t_1^c as shown in Figure B.6. e_2^{in} is inserted into the rung as an input variable as shown in Figure B.8.

Figure B.8: Tank Control Model Inserted Event Inputs for t_1^c in LLD

Step 1.3.4: $\forall\ p \in t\bullet$ insert p into the rung as a latched output (output latch). p_2^c is an output place for t_1^c as shown in Figure B.6. p_2^c is inserted into the rung as a latched output as shown in Figure B.9.

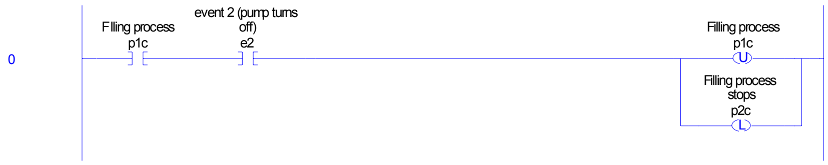


Figure B.9: Tank Control Model Inserted Output Place for t_1^c in LLD

Step 1.3.5: ∀ $e \in \mathcal{E}_{out}$ insert e into the rung as an output (output energize) and the rung ends. e_6^{out} is an event output for t_1^c as shown in Figure B.6. e_6^{out} is inserted into the rung as an output as shown in Figure B.10.

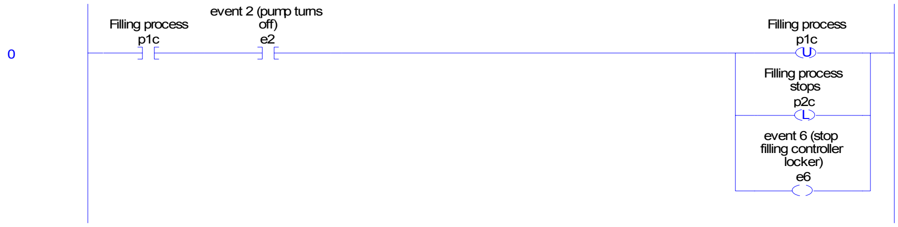


Figure B.10: Tank Control Model Inserted Event Output for t_1^c in LLD

For convenience purposes, the portion of the tank control model that concerns t_1^l is shown in Figure B.11.

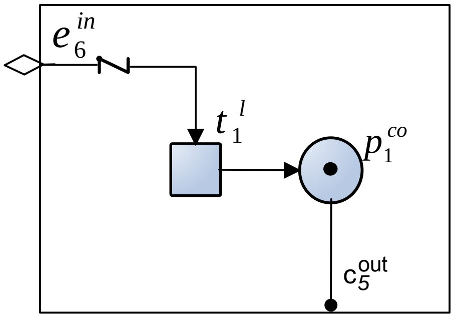


Figure B.11: t_1^l Portion of Tank Control Model

Step 1.4.1: ∀ $p \in \bullet t$ insert p into the rung as an input variable (examine on) and also as an unlatched output (output unlatch). t_1^l has no input place as shown in Figure A.11. Therefore, no instructions are added to the rung.

Step 1.4.2: $\forall\, c \in \mathcal{C}_{in}$ insert c into the rung as an input variable. t_1^l has no condition inputs as shown in Figure B.11. Therefore, no instructions are added to the rung.

Step 1.4.3: $\forall\, e \in \mathcal{E}_{in}$ insert e into the rung as an input variable. e_6^{in} is an event input for t_1^l as shown in Figure B.11. e_6^{in} is inserted into the rung as an input variable as shown in Figure B.12.

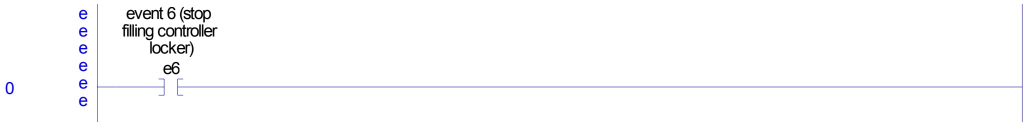


Figure B.12: Tank Control Model Inserted Event Inputs for t_1^l in LLD

Step 1.4.4: $\forall\, p \in t\bullet$ insert p into the rung as a latched output (output latch). p_1^{co} is an output place for t_1^l as shown in Figure B.11. p_1^{co} is inserted into the rung as a latched output as shown in Figure B.13.

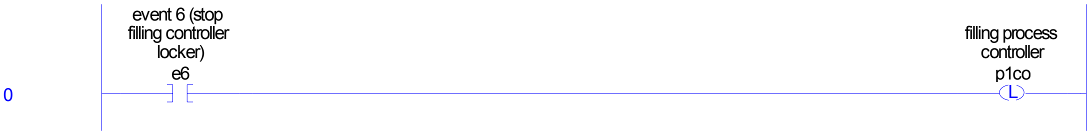


Figure B.13: Tank Control Model Inserted Output Place for t_1^l in LLD

Step 1.4.5: $\forall\ e \in \mathcal{E}_{out}$ insert e into the rung as an output (output energize) and the rung ends. t_1^l has no event outputs as shown in Figure B.11. Therefore, no instructions are added to the rung.

For convenience purposes, the portion of the tank control model that concerns C_2 and C_3 is shown in Figure B.14.

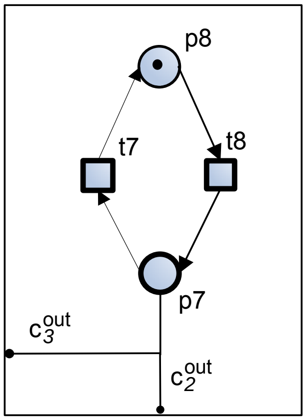


Figure B.14: Portion of Tank Control Model Pertaining Conditions C_2 and C_3

Step 2.2: $c \in C_{out}$, insert a rung into the Ladder Logic Diagram and:

Step 2.2.1: $\forall\ p \in c\bullet$ insert p in the rung as an input variable. P7 is an input place for C_2 as shown in Figure B.14. p7 is inserted into the rung as an input variable as shown in Figure B.15.

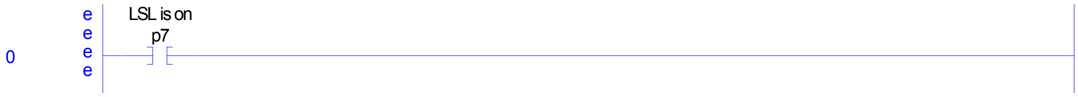


Figure B.15: Tank Control Model Insert Input Place for C_2 in LLD

Step 2.2.2: Insert *c* in the rung as an output variable and end the rung. C_2 is inserted as an output variable into the same rung of step 2.2.1 and end the rung as shown in Figure B.16.

Figure B.16: Tank Control Model Insert Output C_2 in LLD

Step 2.3: $c \epsilon C_{out}$, insert a rung into the Ladder Logic Diagram and:

Step 2.3.1: $\forall\ p \epsilon c \bullet$ insert *p* in the rung as an input variable. P7 is an input place for C_3 as shown in Figure B.14. p7 is inserted into the rung as an input variable as shown in Figure B.17.

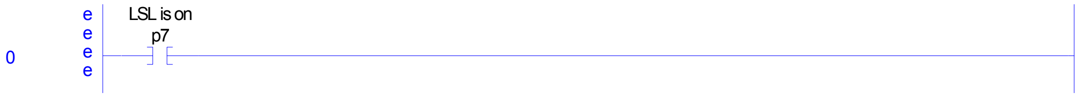


Figure B.17: Tank Control Model Insert Input Place for C_3 in LLD

Step 2.3.2: Insert *c* in the rung as an output variable and end the rung. C_3 is inserted as an output variable into the same rung of step 2.2.1 as shown in Figure B.18.

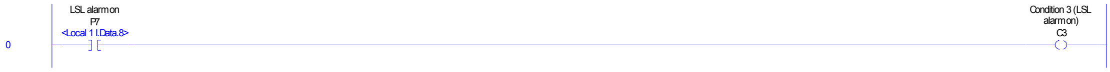


Figure B.18: Tank Control Model Insert Output C_3 in LLD

Appendix C: Station 5-10 Layout

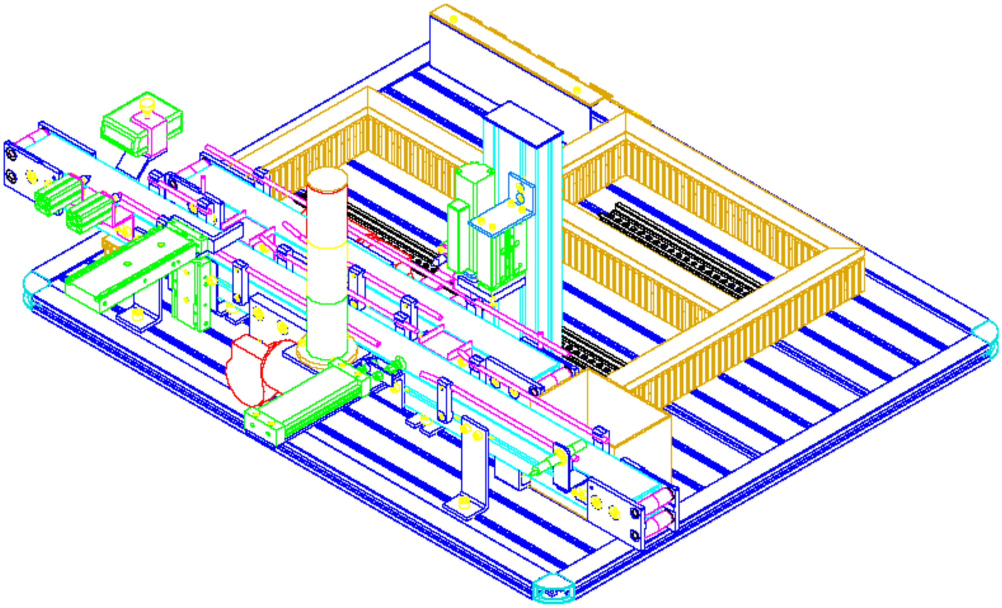


Figure C.1: Stations 5 and 6 Layout

Appendix C: (Continued)

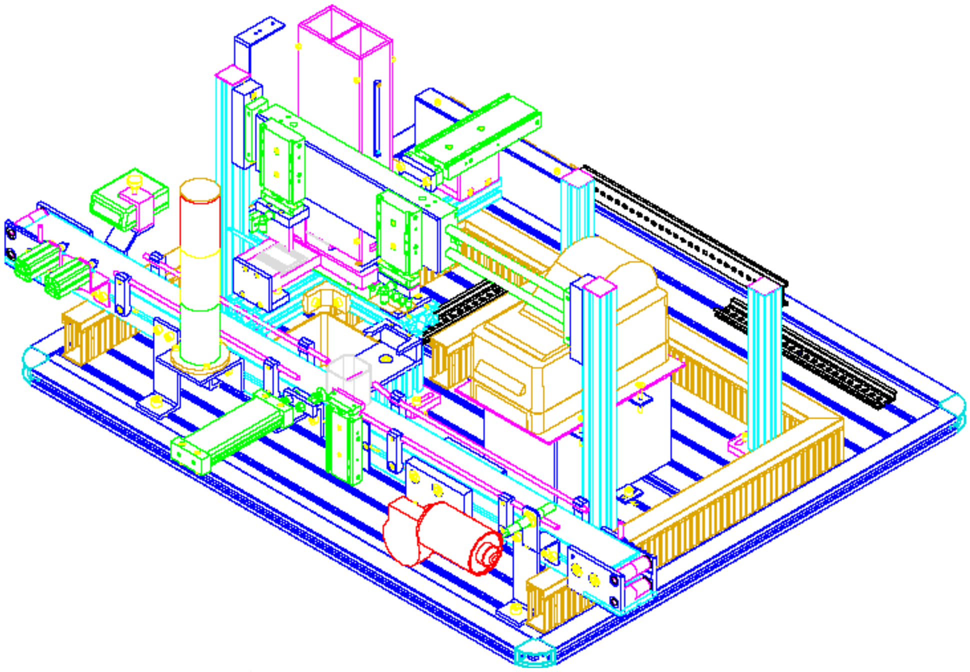


Figure C.2: Station 7 Layout

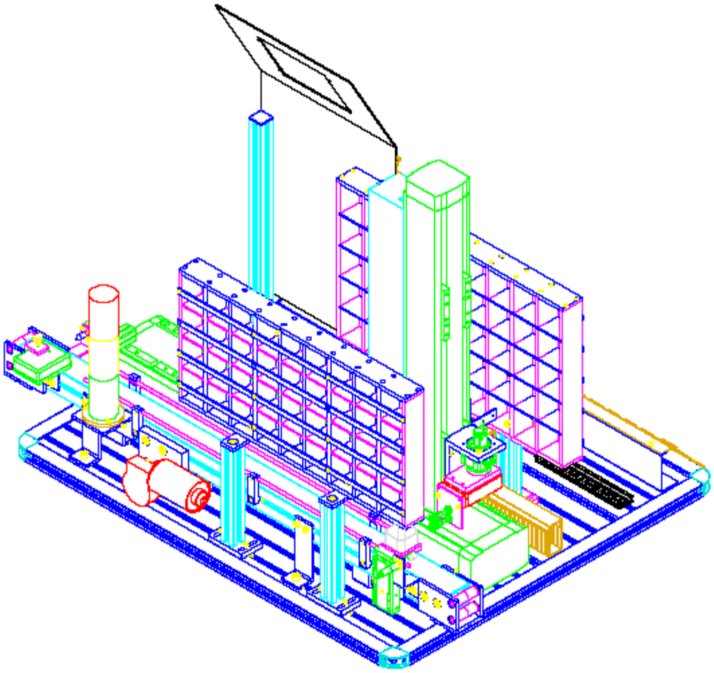


Figure C.3: Station 8 Layout

Appendix C: (Continued)

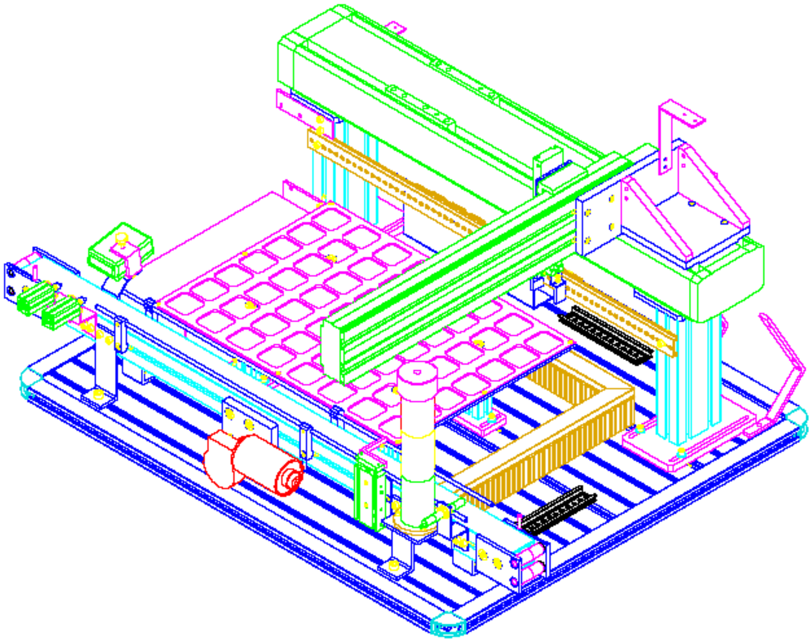


Figure C.4: Station 9 Layout

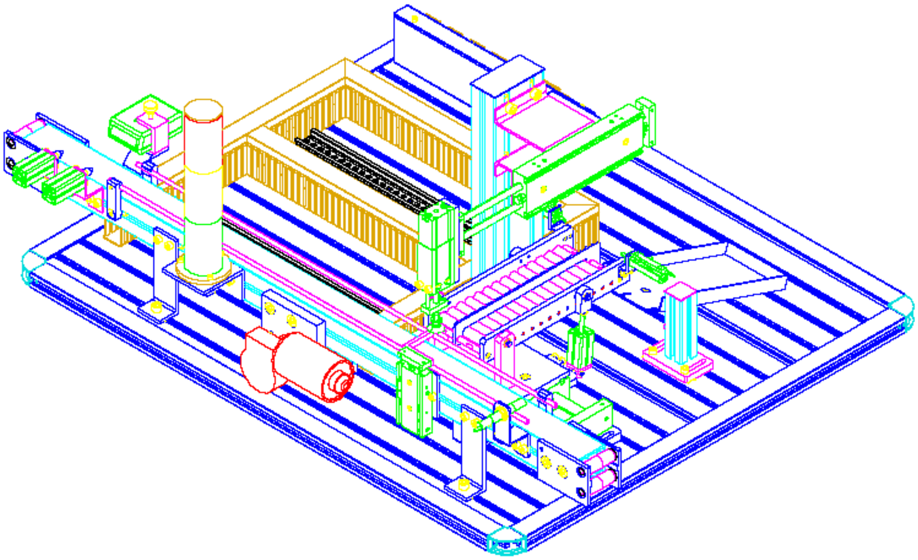


Figure C.5: Station 10 Layout

Appendix D: SESA Reachable States for the HAS-200

```
State nr.    1
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  1  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t1,t4,t78,t115}=> s2
State nr.    2
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  1  0  0  1  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1
    :  1  1  1  1
=={t2,t5,t79,t116}=> s3
State nr.    3
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  1  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t12,t80}=> s4
State nr.    4
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  1  0  0  1  1  0  0  1  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t8,t23,t81,t117}=> s5
State nr.    5
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  0  1  1  0  1  0  1  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  1
    :  1  1  1  1
=={t11,t82}=> s6
```

Figure D.1: SESA HAS-200 Reachable States

```
State nr.    6
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  1  0  1  0  1  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  1
    :  1  1  1  1
=={t16,t83}=> s7
State nr.    7
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  0  1  1  0  1  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  1
    :  1  1  1  1
=={t15,t26}=> s8
State nr.    8
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  1  0  1  0  0  0  0  1  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  1
    :  1  1  1  1
=={t20,t27,t84,t118}=> s9
State nr.    9
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  1  0  0  1  0  0  0  0  1  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t34,t85}=> s10
State nr.   10
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  1  0  0  1  0  0  0  0  1  0  1  1  0
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t7,t18,t30,t45,t86,t123}=> s11
```

Figure D.1: (Continued)

```
State nr.   11
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  1  0
    :  0  1  0  1  1  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  0
    :  1  1  1  1
=={t33,t87}=> s12
State nr.   12
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  0  1  0  1  1  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  0
    :  1  1  1  1
=={t38,t88}=> s13
State nr.   13
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  1  0  0  1  1  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  0
    :  1  1  1  1
=={t36,t46}=> s14
State nr.   14
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  0  1  0  1  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  0
    :  1  1  1  1
=={t43,t47,t89,t124}=> s15
State nr.   15
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  0  1  1  0  0  0  1  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t56,t90}=> s16
```

Figure D.1: (Continued)

```
State nr.   16
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  0  1  1  0  0  0  1  0  0  1  0  0  1  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t29,t41,t53,t66,t91,t127}=> s17
State nr.   17
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  0  1  1  0  1  0  1  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  0  1  1
=={t55,t92}=> s18
State nr.   18
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  0  1  1
=={t60,t93}=> s19
State nr.   19
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  1  0  0  1  1  0  1  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  0  1  1
=={t58,t68}=> s20
State nr.   20
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  1  0  1  0  1  0  0  1  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  0  1  1
=={t65,t69,t94,t128}=> s21
```

Figure D.1: (Continued)

```
State nr.   21
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  1  0  1  0  0  1  0  0  1  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t51,t62,t73,t95,t129}=> s22
State nr.   22
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  1
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  0  1
=={t76}=> s23
State nr.   23
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  0  1
=={t77,t96,t130}=> s24
State nr.   24
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t3,t97}=> s25
State nr.   25
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  1  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t13,t98}=> s26
```

Figure D.1: (Continued)

```
State nr.   26
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  1  0  1  1  0  0  1  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t9,t22,t99,t119}=> s27
State nr.   27
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  0  1  1  0  1  0  1  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  1  1
    :  1  1  1  1
=={t10,t100}=> s28
State nr.   28
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  1  0  1  0  1  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  1  1
    :  1  1  1  1
=={t17,t101}=> s29
State nr.   29
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  0  1  1  0  1  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  1  1
    :  1  1  1  1
=={t14,t24}=> s30
State nr.   30
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  1  0  1  0  0  1  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  1  1
    :  1  1  1  1
=={t21,t25,t102,t120}=> s31
```

Figure D.1: (Continued)

```
State nr.   31
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  1  0  0  1  0  0  1  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t35,t103}=> s32
State nr.   32
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  0  1  1  0  1  0  0  1  0  0  1  0  0  0  1  1  0
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t6,t19,t31,t44,t104,t121}=> s33
State nr.   33
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  1  0
    :  0  1  0  1  1  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  1  1  0  1
    :  1  1  1  1
=={t32,t105}=> s34
State nr.   34
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  0  1  0  1  1  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  1  1  0  1
    :  1  1  1  1
=={t39,t106}=> s35
State nr.   35
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  1  0  0  1  1  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  1  1  1  0  1
    :  1  1  1  1
=={t37,t48}=> s36
```

Figure D.1: (Continued)

```
State nr.   36
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  0  1  0  1  0  0  0  1  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  1  1  1  0  1
    :  1  1  1  1
=={t42,t49,t107,t122}=> s37
State nr.   37
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  0  1  1  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t57,t108}=> s38
State nr.   38
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  1  0  0  1
    :  0  1  1  0  0  0  0  0  1  1  0  0  1  1  0  1  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
=={t28,t40,t52,t67,t109,t125}=> s39
State nr.   39
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  0  1  1  0  1  0  1  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  1  1  1  1
    :  0  1  1  1
=={t54,t110}=> s40
State nr.   40
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  1  1  1  1
    :  0  1  1  1
=={t61,t111}=> s41
```

Figure D.1: (Continued)

```
State nr.   41
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  1  0  0  1  1  0  1  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  1  1  1  1
    :  0  1  1  1
=={t59,t70}=> s42
State nr.   42
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  1  0  1  0  1  0  0  0  0  1  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  1  1  1  1
    :  0  1  1  1
=={t64,t71,t112,t126}=> s43
State nr.   43
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  0  1  1  0  1  0  0  1  0  0  0  0  1  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  1  1  1
    :  1  1  1  1
=={t50,t63,t72,t113,t131}=> s44
State nr.   44
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  1
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1
    :  1  1  1  0
=={t74}=> s45
State nr.   45
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1
    :  1  1  1  0
=={t75,t114,t132}=> s46
```

Figure D.1: (Continued)

```
State nr.   46
P.nr:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
    : 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    : 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
    : 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92
    : 93 94 95 96
toks:  0  0  0  0  0  1  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1
    :  0  1  0  1  0  0  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0
    :  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    :  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
    :  1  1  1  1
dead state
```

Figure D.1: (Continued)

About the Author

Natalia Sandberg was born Natalia Palacio Marino on October 25, 1980. After graduation from Marymount School in Barranquilla, Colombia. Natalia attended Universidad Del Norte and subsequently transferred to The University of South Florida where she attained a Bachelors of Science in Industrial Engineering. She was recognized as the outstanding graduate student from the Department of Industrial Engineering upon graduation in the fall semester 2003. After working for Baxter Healthcare Natalia chose to pursue a Master of Science in Industrial Engineering. She is currently married to Brian Sandberg and works as an Application Engineer at Invensys Process Systems in Houston, TX. For more information on this thesis please contact the Author at natyp25@yahoo.com.