## ABSTRACT

## PERFORMANCE EVALUATION AND SEQUENCE CONTROL OF AN AUTOMATED MANUFACTURING SYSTEM

**by**
**Vijaykumar D. Desai**

In an automated sequential manufacturing system Programmable Logic Controllers (PLC) are widely used. As the control specification varies, the control software needs to be rewritten to accommodate the new specification. Since PLC has high flexibility, one can update the current system while it is running thereby making easier implementation. In order to design flexible, reusable and maintainable control software, a good modeling tool is required. Petri nets are such a tool. Which facilitates analysis of behavioral properties, performance evaluation, and systematic construction of discrete event simulators and controllers. In this thesis a system with one robot and five sequential work stations is used as an example of an automated system. To illustrate the Petri net method, performance and other properties of this system are evaluated. The PLC program is also developed for sequence control of the system.

# PERFORMANCE EVALUATION AND SEQUENCE CONTROL OF AN AUTOMATED MANUFACTURING SYSTEM

by
Vijaykumar D. Desai

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

January 1995

APPROVAL PAGE

# PERFORMANCE EVALUATION AND SEQUENCE CONTROL OF AN AUTOMATED MANUFACTURING SYSTEM

Vijaykumar D. Desai

Dr. MengChu Zhou, Thesis Advisor — Date
Assistant Professor
Department of Electrical and Computer Engineering, NJIT

Dr. Anthony Robbi, Committee Member — Date
Associate Professor
Department of Electrical and Computer Engineering, NJIT

Dr. Marshall Kuo, Committee Member — Date
Professor
Department of Electrical and Computer Engineering, NJIT

## BIOGRAPHICAL SKETCH

**Author:** Vijaykumar D. Desai

**Degree:** Master of Science in Electrical Engineering

**Date:** January 1995

**Undergraduate and Graduate Education:**

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1995

- Bachelor of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1984

**Major:** Electrical Engineering

This thesis is dedicated to
my family

## ACKNOWLEDGMENT

I wish to express my sincere thanks to Dr. Zhou for his guidance and encouragement throughout the progress of this thesis. Special thanks to Dr. Robbi and Dr. Kuo for serving as members of the committee.

# TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

Development of a software is dependent on the model it is based on. Petri nets have been extensively used in software development [20]. One of the major advantage of using Petri nets is that the same model is used for the analysis of behavioral properties and performance evaluation.

Petri nets were successfully applied for sequence control. Petri net based sequence controllers are easy to design, implement, and maintain. In the early 80's, Hitachi Ltd. developed a Petri net based sequence controller [16] which was successfully used in real manufacturing applications to control parts assembly systems, and automatic warehouse load/unload system [17]. The use of Petri nets substantially reduced the development time compared with traditional approach. Numerous approaches to synthesis and implementation of Petri net based sequence controllers have been reported in the past few years [4].

Programmable Logic Controllers are commonly used for the sequence control in automated systems. They can be designed using ladder logic diagrams, which are known to be very difficult to debug. Since ladder diagrams are widely used in the industries for hardwired logic circuits, they became a standard way of providing control information from the designers to the users of equipment. When programmable controllers were introduced, this type of circuits was desirable because they were easy to use and interpret for practicing engineers.

In this thesis, a system consisting of five work stations and a shared resource (robot) is used to show how a Petri net model can be developed for performance evaluation and a PLC Program is developed for sequence control. Stations 1 and 2 are loading machines for two different parts via a same robot, Station 3 is a dimpling machine, Station 4 is a marking machine and Station 5 is unloading machine which unloads and separates finished and unfinished goods. Unfinished goods may result due to failure of any work stations. This process can be extended to any number of work stations. Performance evaluation and behavior properties are also studied. Ladder Logic Programs are finally developed to implement a PLC for the system.

# CHAPTER 2

# THEORY of PETRI NETS

## 2.1 Introduction to Petri Nets

In 1962 Carl A. Petri created a net like mathematical tool for the study of communication with automata. The resulting tool was named as Petri nets. Petri net methodology utilizes both mathematical and graphical representations.

Changing control specifications can be easily implemented using such a graphical tool. As mathematical models petri nets reflect behavior of the system. The performance evaluation can be conducted by using either analytical techniques, based on the underlying (semi)-markov processes or discrete event simulation. By incorporating time functions that follow probabilistic distributions to transitions, one can obtain production rates for manufacturing system models, throughput, critical resource utilization and reliability measures.

Some of the most fundamental properties of Petri nets are reachability, boundedness, liveness, reversibility, consistency and persistence as given later in Section 2.4.

## 2.2 Application of Petri Nets

Petri nets (PN's) were successfully used in modeling and analysis of communication protocols [3]. PN's are also applied to the modeling of Just-in-Time manufacturing systems [22]. They are also widely used in flexible manufacturing systems due to their ability to model asynchronous operations, concurrence, deadlock, conflicting events, and event-driven systems [11]. PN's can be used from design through evaluation to control. It is possible to compile the net into control code data for implementation and execution on the shop floor [24]. Petri nets also inspired GRAFCET for the application to finite state automation processes, which became standard for specifying sequence control in France in 1977 [22], and later became an international standard called Sequential Function Charts. Since then more research and developments are done using Petri nets in various applications for automated manufacturing systems. A bibliography of Petri nets [19], published in 1991, contains 4099 entries dealing with Petri net theory and applications.

## 2.3 Description of Petri Nets

A Petri net can be identified by its graph in which there are three types of objects. They are places, transitions, and tokens. Directed arcs connect transitions to places and places to transitions. A place (P) can be shown as a circle, a transition (T) as bar or box and a token as a dot inside the place. Input function (I) defines the directed arcs from places to transitions. Output function (O) defines the directed arcs from transitions to places. In a real system, a place represents status of a resource or an operation. A transition represents either start or completion of an event or process. A place with a token represents resource available or operation being executed.

A marked Petri net is defined as Z = (P, T, I, O, $m_0$). P is a finite set of places $\{p_1, p_2, ...., p_n\}$. T is a finite set of transitions $\{t_1, t_2, ...., t_s\}$ with the following condition,

$P \cup T \neq \varnothing$ and $P \cap T = \varnothing$. Input function (I) defines $(P\times T) \mapsto N$ and Output function (O) define as $(P\times T) \mapsto N$, where N is a set of non-negative integers. If there exist k directed arcs connecting place p to transition t, I(p,t)=k or O(p,t)=k, which graphically represented by a single arc with multiplicity, or weight k. $m_0$: $(P \mapsto N)$ is an initial marking of places. A marking defines current status of the model system. A transition is enabled when each input place p of t contains at least the number of tokens equal to weight k of the directed arc connecting p to t. When a transition, t, is enabled, it may be fired. If fired it removes from each input place p the number of tokens equal to the weight of directed arc connecting p to t. Also, it deposits in each output place p the number of tokens equal to the weight of the directed arc connecting t to p.

The mathematical representation of a PN consists of two matrices. The input matrix, I, is derived from the Petri net graphical representation. The input matrix is an $(n \times s)$ matrix where n is the number of places and s is the number of transitions. The entries in the input matrix correspond to the weights of the input arcs from places (row) to transitions (column). Similarly, the output matrix, O, is an $(n \times s)$ matrix. The entries in the output matrix correspond to the weights of the output arcs from transitions (column) to places (row). Incidence matrix, C, describes the dynamic characteristics of the system and is equal to the difference between the output and input matrices, O - I. This matrix represents the change in tokens of places when transitions fire. Negative (positive) numbers in the incidence matrix signify the consumption (creation) of tokens. The state or marking of a Petri net denotes the number of tokens occupying each place and is captured in one dimension matrix of size n.

To illustrate how Petri nets can be used to model, we consider the following example of Robot #1, Machine #1 and a buffer with limited capacity. Machine #1 acquires Robot #1 and loads parts using Robot #1, then processes it. Finished parts are unloaded into buffer automatically when it is available. The buffer is of limited capacity. When the robot is available, it unloads finished parts into storage from the buffer and the cycle continues.

The net is shown in Figure 2.1 and its places are described in Table 1.

**Table 1** Place Designations of Illustrative Example

| | | | |
|---|---|---|---|
| $p_1$ | Machine #1 (M1) | $p_5$ | M1 process using R1 |
| $p_2$ | Robot #1 (R1) | $p_6$ | Buffer |
| $p_3$ | M1 acquire R1 | $p_7$ | Buffer capacity |
| $p_4$ | Raw Material /Part | $p_8$ | Unload buffer using R1 |

In this example the robot is shared by M1 for loading and by the buffer for unloading. Since the buffer has limited capacity, M1 does not continuously produce finished parts until the buffer is unloaded by R1. Assume that initial marking is $m_0 = (3,1,0,1,0,0,2,0)$, where $p_1$ has more tokens than buffer capacity $p_7$. Then deadlock

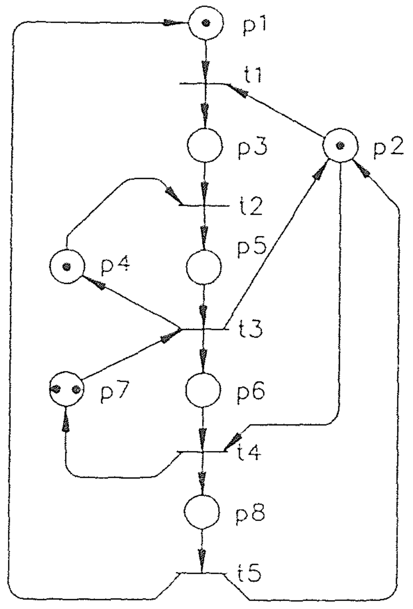


FIGURE 2.1- DEADLOCK FREE PETRI NET MODEL OF ILLUSTRATIVE EXAMPLE

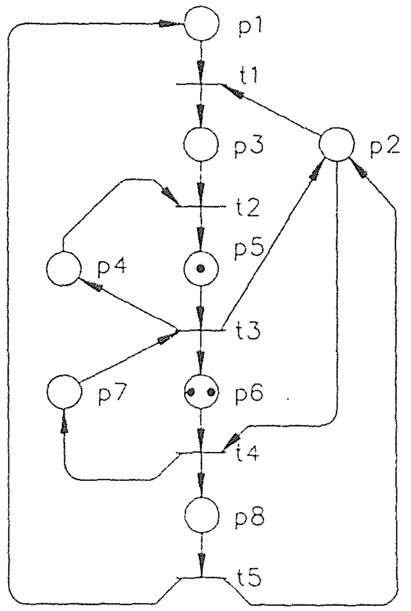


FIGURE 2.2- DEADLOCK OCCURS WHEN NUMBER OF TOKEN IN INITIAL MARKING IN p1 IS GREATER THAN p7 IN ILLUSTRATIVE EXAMPLE

will be reached if firing sequence $t_1,t_2,t_3,t_1,t_2,t_3,t_1,t_2$ occurs. At this time $(0,0,0,0,1,2,0,0)^T$ marking is reached, as shown in Figure 2.2. Since $p_7$ does not have a token, $t_3$ is unable to fire. And also, since $p_2$ does not have a token, and $t_4$ is unable to fire. To avoid this $p_1$ should have less than buffer capacity tokens or a mechanism by which when the buffer is full the robot must unload the buffer before the M1 uses the robot to produce next part. This is a Sequential Mutual Exclusion in which there is a shared resource [22]. This shows that one should verify deadlock condition, especially when there are shared resources present in the system. One of the advantages of Petri net models is that one can analyze some of the properties related to a manufacturing process. They can be behavioral, structural, qualitative properties and logical correctness.

The input matrix I, output matrix O and incidence matrix C for the illustrative example considered are as follows:

$$I=\begin{bmatrix}1&0&0&0&0\\1&0&0&1&0\\0&1&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&1&0&0\\0&0&0&0&1\end{bmatrix}\quad O=\begin{bmatrix}0&0&0&0&1\\0&0&1&0&1\\1&0&0&0&0\\0&0&1&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&1&0\end{bmatrix}\quad C=\begin{bmatrix}-1&0&0&0&1\\-1&0&1&-1&1\\1&-1&0&0&0\\0&-1&1&0&0\\0&1&-1&0&0\\0&0&1&-1&0\\0&0&-1&1&0\\0&0&0&1&-1\end{bmatrix}$$

The behavioral properties depend on initial marking and its significance in manufacturing process is defined as below.

## 2.4 Behavioral Properties of Petri Net

### 2.4.1 Reachability

The reachability set $R(m_0)$ of a Petri net, is defined as the set of all markings which are obtainable from the initial marking $m_0$, through some firing sequence $L(m_0)$ [18]. This means that a system can reach a specific state, or particular functional behavior. This can be verified by finding a sequence of transition firings which would result in transforming a marking $m_0$ to $m_i$. $m_i$ represents a specific state, and the sequence of firings represents the required functioned behavior. Thus the problem of identifying the existence of a specific state $m_i$, of the system can be redefined as the problem of finding if $m_i \in R(m_o)$.

### 2.4.2 Boundedness

A PN is said to be k-bounded for any reachable marking, m, none of the places contains more than k (k is a nonnegative integer number) tokens. A 1-bounded Petri nets is called safe [18]. In manufacturing, boundedness implies to the absence of capacity overflows. Safeness of an operation place, guarantees that there is no attempt to request execution of an ongoing process. If a resource place is safe, it implies only one available resource.

### 2.4.3 Liveness

There are five degrees of liveness, L0-L4. They are defined with respect to a single transition as follows [18]:

L0 "dead" if t can never be fired in any sequence of $L(m_0)$.

L1 "potentially firable" if t can be fired at least once in some firing sequence $L(m_0)$.

L2 given any positive integer k, t can be fired at least k times in some firing sequence $L(m_0)$.

L3 if t appears infinitely often in some firing sequence $L(m_0)$.

L4 "live" if t is L1-live for every marking, m, in $R(m_0)$.

A PN is said to be "live" if all transitions in the net are L4-live. A live PN is one which cannot be deadlocked. This property guarantees that a system can successfully produce and all modeled processes can occur. Coffman at. al. [10] showed that four conditions must hold for a deadlock to occur. These four conditions are:

1. Mutual exclusion: a resource is either available or allocated to a process which has an exclusive access to this resource [21].
2. Hold and Wait: a process is allowed to hold a resource(s) while requesting more resources.
3. No preemption: a resource(s) allocated to a process cannot be removed from the process, until it is released by the process itself.
4. Circular wait: two or more processes are arranged in a chain in which each process waits for resources held by the process next in the chain.

In the illustrative example shown in Figure 2.2, deadlock can be reached when $m_0(p_1)$ is greater than $m_0(p_7)$. It can be shown that all four conditions of deadlock hold.

### 2.4.4 Reversibility

A reversible PN is one that can always return to a home state via some firing sequence in $L(m_n)$. The home state is usually, but not necessarily, the initial state [18]. This property in manufacturing means the system can be initialized from any reachable state.

### 2.4.5 Consistency

A Petri net is said to be (partially) consistent if there exists a marking $m_0$ and a firing sequence $L(m_0)$ returning to $m_0$ such that every (some) transition appears at least once in $L(m_0)$ [18]. Reversibility implies consistency, but consistency does not necessarily imply reversibility. In manufacturing, this means that cyclic activities exist.

### 2.4.6 Persistence

A PN is considered to be persistent if firing of any enabled transition does not disable previously enabled transition [18]. In shared resource allocation a persistent net model implies that there is no conflict among processes.

## 2.5 Analysis Methods

Once the Petri net model is formulated, it must be analyzed to show that it does meet customer requirements and all inputs and outputs are considered. The fundamental methods of analysis are based on the reachability tree and matrix equation representation of a net. Also, a systematic transformation of a Petri net can be done, by reducing the number of places and transitions in a net, and at the same time preserving the properties such as boundedness, conservativeness, liveness, etc. Some of these techniques were discussed in [18].

From the definition of the reachability tree the following observation can be made for the illustrative example when an initial marking is $m_0 = (2,1,0,1,0,0,2,0)^T$ [26]. In this marking transition $t_1$ is enabled. When $t_1$ fires a new marking is obtained; $m_1 = (1,0,1,1,0,0,2,0)^T$. This is a "new" marking in which transition $t_2$ is enabled. Firing $t_2$ in $m_1$ results in $m_2 = (1,0,0,0,1,0,2,0)^T$. This is a "new" marking in which transition $t_3$ is enabled and $m_1$ becomes an "old" marking. Firing $t_3$ in $m_2$ results in $m_3 = (1,1,0,1,0,1,1,0)^T$. This is a "new" marking in which transitions $t_4$ and $t_1$ are enabled and $m_2$ becomes an "old" marking. Firing $t_4$ in $m_3$ results in $m_5 = (1,0,0,1,0,0,2,1)^T$. This is a "new" marking in which $t_5$ is enabled and $m_3$ becomes an "old" marking. Firing $t_5$ in $m_5$ results in $m_0$ which is an "old" marking. At $m_3$ firing $t_1$ results in $m_4 = (0,0,1,1,0,1,1,0)^T$. This is a "new" marking in which $t_2$ is enabled. Firing $t_2$ in $m_4$ results in $m_6 = (0,0,0,0,1,1,1,0)^T$. This is a "new" marking in which transition $t_3$ is enabled and $m_4$ becomes an "old" marking. Firing $t_3$ in $m_6$ results in $m_7 = (0,1,0,1,0,2,0,0)^T$. This is a "new" marking in which $t_4$ is enabled and $m_6$ becomes an "old" marking. Firing $t_4$ in $m_7$ results in $m_8 = (0,0,0,1,0,1,1,1)^T$. This is a "new" marking in which transition $t_5$ is enabled and $m_7$ becomes an "old" marking. Firing $t_5$ in $m_8$ results in $m_3$ which is an "old" marking. Figure 2.3 shows reachability trees for deadlock free and deadlock of the illustrative example under different marking when $m_0(p_1)$ is greater than $m_0(p_7)$ such as $m_0 = (2,1,0,1,0,0,2,0)^T$ and $m_0 = (3,1,0,1,0,0,2,0)^T$ respectively.

A number of properties can be studied by using reachability tree. If each branch of tree contains only zeros and ones, then the net is safe. A transition is dead if it does not appear as an arc label in the tree. In this system all transitions are L2-Live for the above two initial marking, since it is possible to fire any transition at least k times in some firing sequence $L(m_0)$. Finally, reversibility property can also be observed by inspection that $m_0$ is reachable from any marking $m \in R(m_0)$ for $m_0 = (2,1,0,1,0,0,2,0)^T$. This also implies consistency. This net is not persistent since at $m_3$ in Figure 2.3 firing $t_1$ disable $t_4$.

The ordinary Petri nets do not include any concept of time. With this class of nets, it is possible only to describe the logical structure of the modeled system, but not its time evolution. Responding to the need for the temporal performance analysis of discrete-event systems, time has been introduced into Petri nets in a variety of ways. There are two fundamental types of timed Petri nets in the context of performance evaluation. These are Deterministic timed Petri nets, and Stochastic timed Petri nets.

When time delays are modeled as random variables, or probabilistic distributions are added to the transitions of deterministic timed Petri nets models for the conflict resolution, stochastic timed Petri nets models are yielded. In such models, it has become a convention to associate time delays with the transition only. When the random variables are of general distribution or both deterministic and random variables are involved, the resulting net models cannot be solved analytically for general cases. Thus simulation or approximation methods are required. The stochastic timed Petri nets in which the time delay for each transition is assumed to be stochastic and exponentially

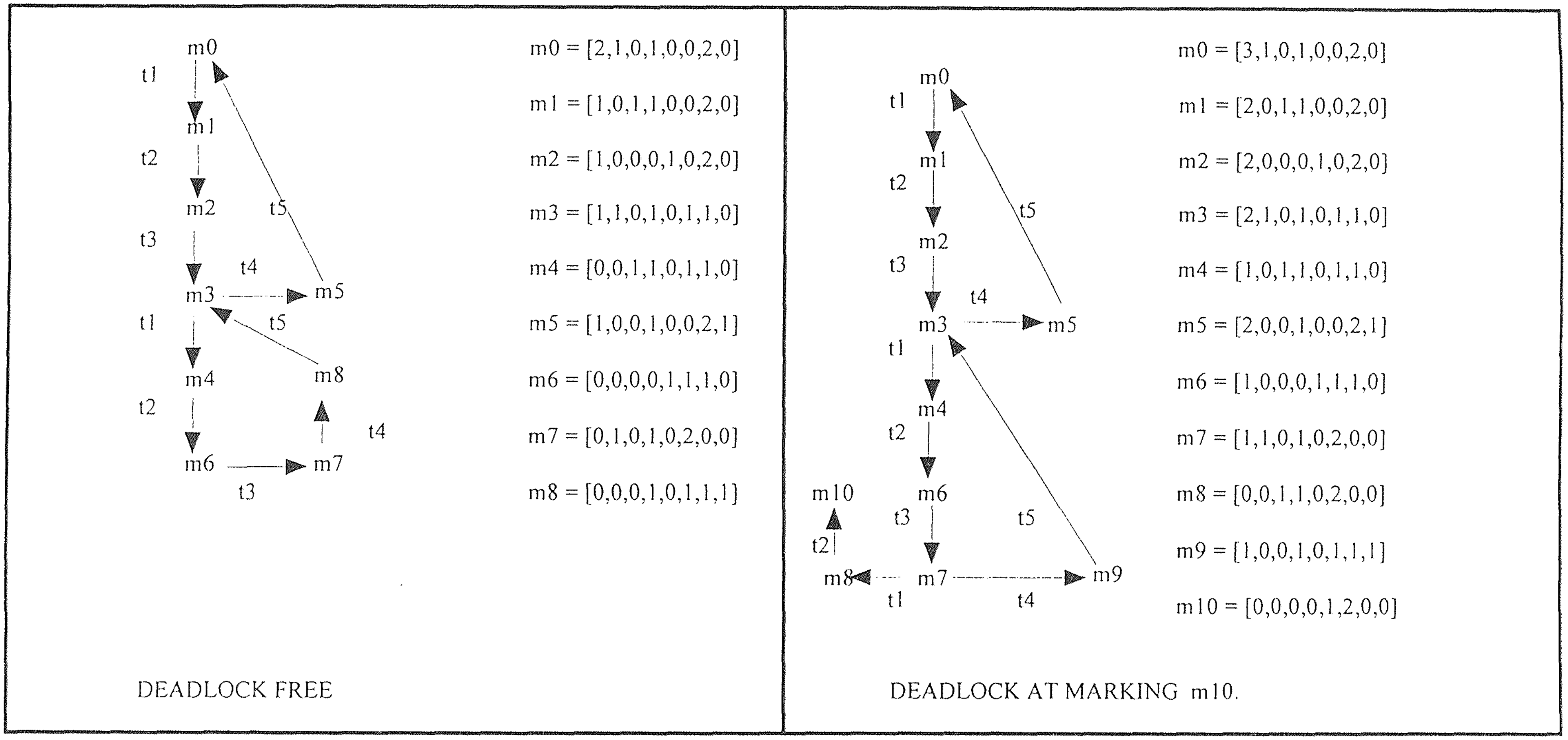


**Figure 2.3** Reachability Graph for Illustrate Example

distributed are called stochastic Petri nets (SPN) [15]. The SPN models which allow for immediate transitions, i.e., with zero time delay, are called generalized SPN (GSPN) [14]. Both models include extensions such as priority transitions, inhibitor arcs, and probabilistic arcs. These models can be converted into their equivalent Markov process representations, and analyzed analytically.

A conflict results when a place has two or more output transition and only one transition can be fire, others become disabled. When a conflict results, it has to be resolved by a mechanism, such as associating priority to transitions. Since this mechanism leads to stochastic nets the use of the deterministic timed Petri nets for performance evaluation has been restricted to the choice-free nets, which can be modeled as marked graphs [6,26].

The continuous time Markov chain is a state machine whose nodes are states or markings and links are labeled to the corresponding transitions. In a continuous time Markov chain transition rate $a_{ij}$ is the rate of a process from state i to state j, $i \neq j$. $a_{ii}$ is determined by

$$\sum_{j=1}^{m} a_{ij} = 0$$

Given the present state $m_i$, the conditional probability that $m_i$ goes to $m_j$ during time interval dt is $a_{ij}$ dt. Also, $1/a_{ij}$ is the expected time between i and j. From the above definition transition rate matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix}$$

Let $\pi$ be a row vector of steady state probabilities. Then, it satisfies:

$$\pi * A = 0 \tag{1}$$

$$\sum_{i=1} \pi_i = 1 \tag{2}$$

For example, the transition rate matrix for the illustrative example shown in Figure 2.1 with the initial marking $m_0 = (2,1,0,1,0,0,2,0)$ is

$$A = \begin{bmatrix} -\lambda_1 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_2 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\lambda_3 & \lambda_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (-\lambda_4-\lambda_5) & \lambda_4 & \lambda_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\lambda_6 & 0 & \lambda_6 & 0 & 0 \\ \lambda_{10} & 0 & 0 & 0 & 0 & -\lambda_{10} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_7 & \lambda_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_8 & \lambda_8 \\ 0 & 0 & 0 & \lambda_9 & 0 & 0 & 0 & 0 & -\lambda_9 \end{bmatrix}$$

By solving equations (1) and (2) steady state probabilities of the process can be solved.

To study performance using SPN the following steps are involved.

1. Develop a stochastic Petri net model of a system. Define the initial state, and obtain the corresponding marking;

2. Generate the reachability graph of the net, and obtain the set of Markov process;

3. Perform the analysis of the Markov model, and;

4. Convert the analysis results into the required performance measures.

In this thesis SPN is used for performance evaluation as given above using SPNP software.

# CHAPTER 3

# DEVELOPMENT OF A MODEL

To show how Petri net models can be used in performance analysis, a five sequenced work station model is used. This process can be extended into a larger system.

## 3.1 System Description

Assume that previously sub-assembled parts are automatically loaded on to pallets and then pallets enter the system one by one. When station #1 (S1) is available the pallet is loaded on to S1. The specification of the process is as follows:

Step 1: The pallet is marked good at S1 within time given by timer T1.

Step 2: S1 acquires robot (R) and then the timer T2 starts. Using R, part #1 is loaded on sub-assembled part.

Step 3: If within given amount of time given by timer T2 S1 failed to load part #1 using R the pallet is marked bad within time given by timer T3 and robot (R) is released. The pallet is released when S2 is available.

Step 4: S2 also verifies that the pallet is marked good or bad, before it acquires R to load part #2.

Step 5: If the pallet is marked bad S2 releases the pallet, when S3 is available.

Step 6: If the pallet is marked good, S2 acquires the robot (R) when available to load part #2, then the timer T4 starts.

Step 7: If S2 fails to load part #2 using R in a given amount of time by timer T4 then the pallet is marked bad at S2 within time given by timer T5. The R and pallet is released when S3 is available.

Step 8: Before S3 starts it process, it also, verifies either the pallet is marked good or bad. If the pallet is marked bad it releases the pallet when S4 is available.

Step 9: At S3 there is s dimpling machine, If the pallet is marked good, timer T6 starts. At S3, using dimpling machine loaded part is dimpled and locked in its place.

Step 10: If Within give amount of time, S3 does not complete its job using dimpling machine, the pallet is marked bad within time given by timer T7 and dimpling machine and the pallet are released when S4 is available.

Step 11: As before S4 verifies if received pallet is good or bad, i.e., previous stations have completed their job or not. So, again if the pallet is marked bad it releases the pallet when S5 is available.

Step 12: At S4 there is a marker machine which for example marks part number and/or date code etc. If the pallet is good S4 using the marker marks the finished product within given amount of time by timer T8. When the job is completed, the pallet and marker are released when S5 is available.

Step 13: If S4 fails, the pallet is marked bad within time given by timer T9.

Step 14: Finally S5 unloads the pallet with a part marked good or bad. At S5 the pallet is verified. If it is marked good, it means that all stations have finished their jobs and finished goods enters the storage or into next stage of production. The verification and unloading of the pallet are done automatically at S5.

Step 15: Finally the pallet is released into the cycle back to S1 when it is available and cycle continues. See Figure 3.1 of a five sequenced work station system.

In this example stations are in sequence and the pallet is not taken out of cycle if job is failed at one station. But each station receives pallet and each of them verifies if pallet is marked good or bad before it continues with its required job. At the unloading station S5, there can be a counter to count how many pallets are good or bad for the efficiency of stations. Also, at each station one can have counter to count consecutive job failed. When the counter accumulative value equals to the preset value it shut down the station and operator is warned. The station is reset with an operator key. The unfinished product can be returned into the system for reprocessing, depending on which station has failed. The Petri net model is shown in Figure 3.4 and place designations in Table 2.

**Table 2** Place Designations for SPNP program.

| | | | | | |
|---|---|---|---|---|---|
| $p_1$ | Start | $p_{18}$ | Part #2 | $p_{35}$ | Process Completed |
| $p_2$ | Pallet Present S1 | $p_{19}$ | Part#2 Load Using R1 | $p_{36}$ | Pallet Present S5 |
| $p_3$ | Pallet Tag Good | $p_{20}$ | Pallet Present S3 | $p_{37}$ | Pallet Check S5 |
| $p_4$ | Part #1 | $p_{21}$ | Pallet Check S3 | $p_{38}$ | Bad Pallet Unload |
| $p_5$ | Ready to Load part #1 | $p_{22}$ | Good Pallet Present S3 | $p_{39}$ | Good Pallet Unload |
| $p_6$ | Timer S1 | $p_{23}$ | Bad Pallet S3 | $p_A$ | Good Pallet from S1 |
| $p_7$ | Wait state | $p_{24}$ | Dimpling Done at S3 | $p_B$ | Good Pallet from S2 |
| $p_8$ | Process completed at S1 | $p_{25}$ | Timer S3 | $p_C$ | Good Pallet from S3 |
| $p_9$ | Process Fail at S1 | $p_{26}$ | Wait State | $p_D$ | Good Pallet from S4 |
| $p_{10}$ | Part #1 Load Using R1 | $p_{27}$ | Process Completed at S3 | $p_{s1}$ | Station #1 (S1) |
| $p_{11}$ | Pallet Present S2 | $p_{28}$ | Pallet Present S4 | $p_{s2}$ | Station #2 (S2) |
| $p_{12}$ | Pallet Check S2 | $p_{29}$ | Pallet Check S4 | $p_{s3}$ | Station #3 (S3) |
| $p_{13}$ | Bad Pallet at S2 | $p_{30}$ | Bad Pallet at S4 | $p_{s4}$ | Station #4 (S4) |
| $p_{14}$ | Good Pallet Present S2 | $p_{31}$ | Good Pallet Present S4 | $p_{s5}$ | Station #5 (S5) |
| $p_{15}$ | Timer S2 | $p_{32}$ | Timer S4 | $p_m$ | Marker(ex. P/N#) |
| $p_{16}$ | Wait State | $p_{33}$ | Marking Done At S4 | $p_d$ | Dimpling Machine |
| $p_{17}$ | Process Completed | $p_{34}$ | Wait State | $p_{R1}$ | Robot #1 (R1) |

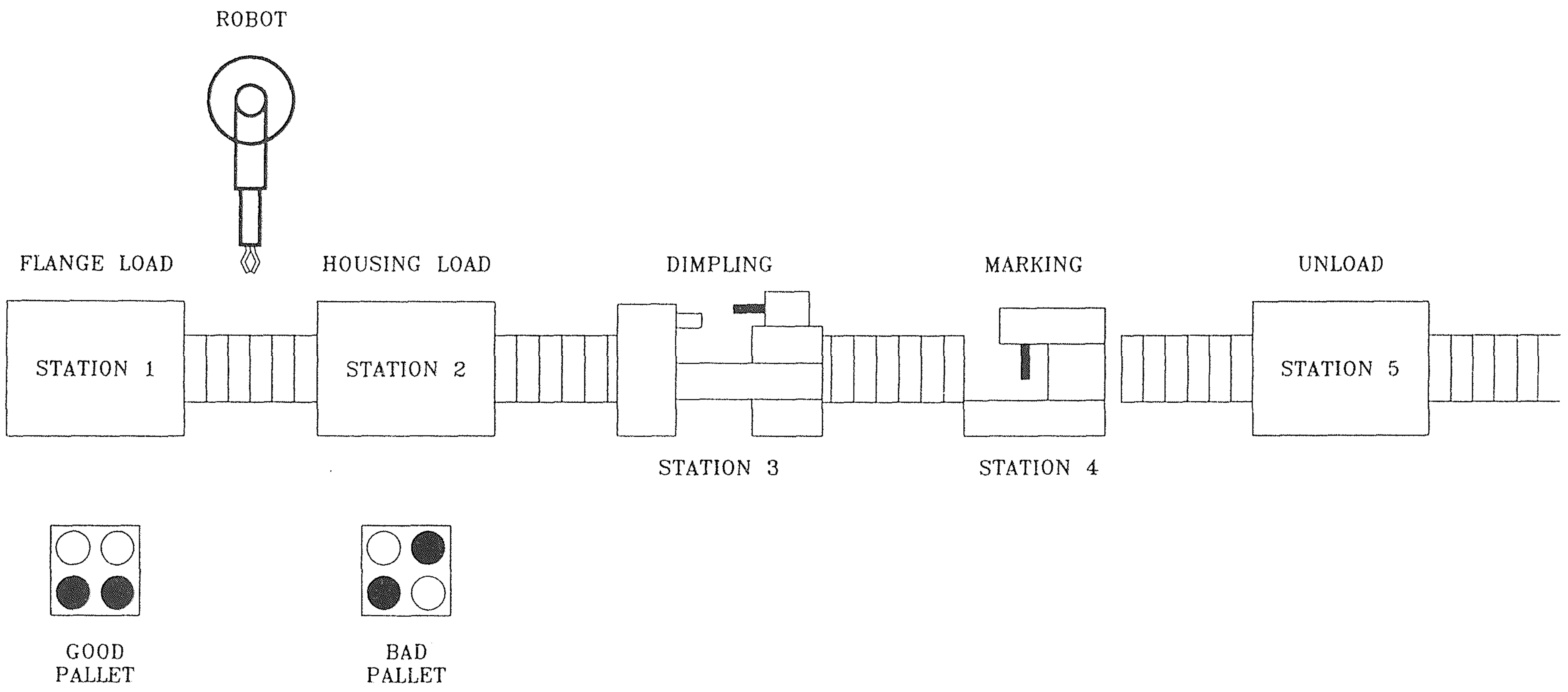


FIGURE 3.1- FIVE SEQUENCE WORK STATION SYSTEM

## 3.2 Building a Graphical Model

A five work station PN model is developed using a bottom-up modular approach. First, S1's process is developed then S2 - S5's. Next, the failure check module is developed for each work station. Later on five work station processes are linked in sequence. Finally, all the modules are linked to complete the system modeling. There are several different methods available to develop a graphical model given in [25].

The following method is used in the model development of the first station. Identify the processes being done at the first station. They are: a pallet being loaded into S1 and marked good, using the robot a part is loaded. For each process in order, create a place to represent its status and add a transition to show the start of a process and a transition to show the end of the process. They are connected with an output arc(s) and with an input arc(s), respectively.

For each kind of resource(s), create and label a place. If an activity place is a starting activity to require the resource(s), add input arc(s) from that resource place to the starting transition of that activity. If an activity is the ending activity to use the resource(s), add output arc(s) from the ending transition to the resource place(s). In the given system for station #1 resources are S1, robot, and part. For those resources places are $p_{s1}$, $p_{R1}$, and $p_4$ respectively. In Table 3 process places and resource places of S1 are given. Starting activities are a pallet at station #1, the pallet is marked good and using robot a part is loaded. The first activity requires S1, therefore, from $p_{s1}$ an input arc is added to transition $t_1$.

**Table 3** Place Designations for S1 Module.

| Places | Processes | places | Resources |
|---|---|---|---|
| $p_1$ | Pallet ready | $p_{s1}$ | Station #1 |
| $p_2$ | Pallet loaded to station 1 | $p_{R1}$ | Robot |
| $p_3$ | Pallet marked good | $p_4$ | Part |
| $p_5$ | Robot is ready with part to be loaded | | |
| $p_{10}$ | Part is being loaded | | |

For the loading of a part using the robot an input arc is added to transition $t_3$ from both resources places $p_{R1}$ and $p_4$, respectively. Finally, the initial marking is specified. $p_1$, $p_{s1}$, $p_{R1}$ and $p_4$ are marked with one token. For example since initially a robot is available, $m_0(p_{R1}) = 1$. The Petri net module for the first station is shown in Figure 3.2. One difference between S1 and the other stations is that, a pallet is marked good at S1 but at the other stations a pallet is verified if it is good or bad.

Similarly, a failure check module is developed. The processes are: start timer, if timer is done, mark pallet bad else the process is completed. In this module resource(s) is not used. Therefore, only arc(s) are to show the flow of the process. Table 4 lists process places and Figure 3.3 shows the Petri net module for failure check.

Similarly, work stations S2 -S5 and their corresponding failure blocks are developed. Finally, all of them are linked to complete the system as shown in Figure 3.4 and place designation in Table 2. It is noted that links are based on the system description as given in Section 3.1.

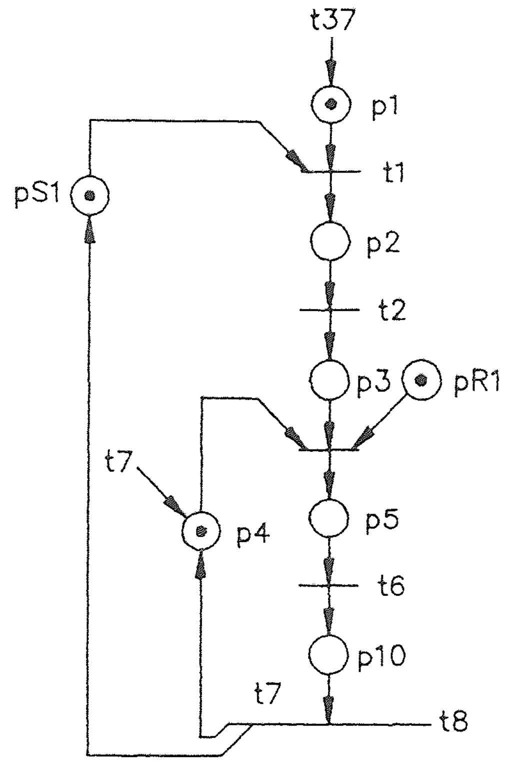


FIGURE 3.2- STATION 1 MODULE OF PETRI NET MODEL

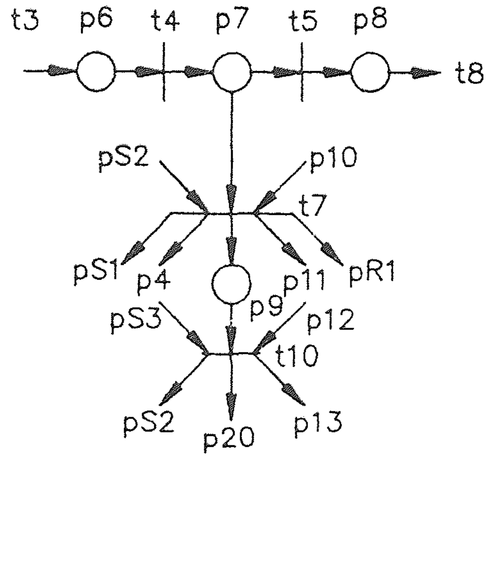


FIGURE 3.3- STATION 1 FAILURE MODULE OF PETRI NET MODEL

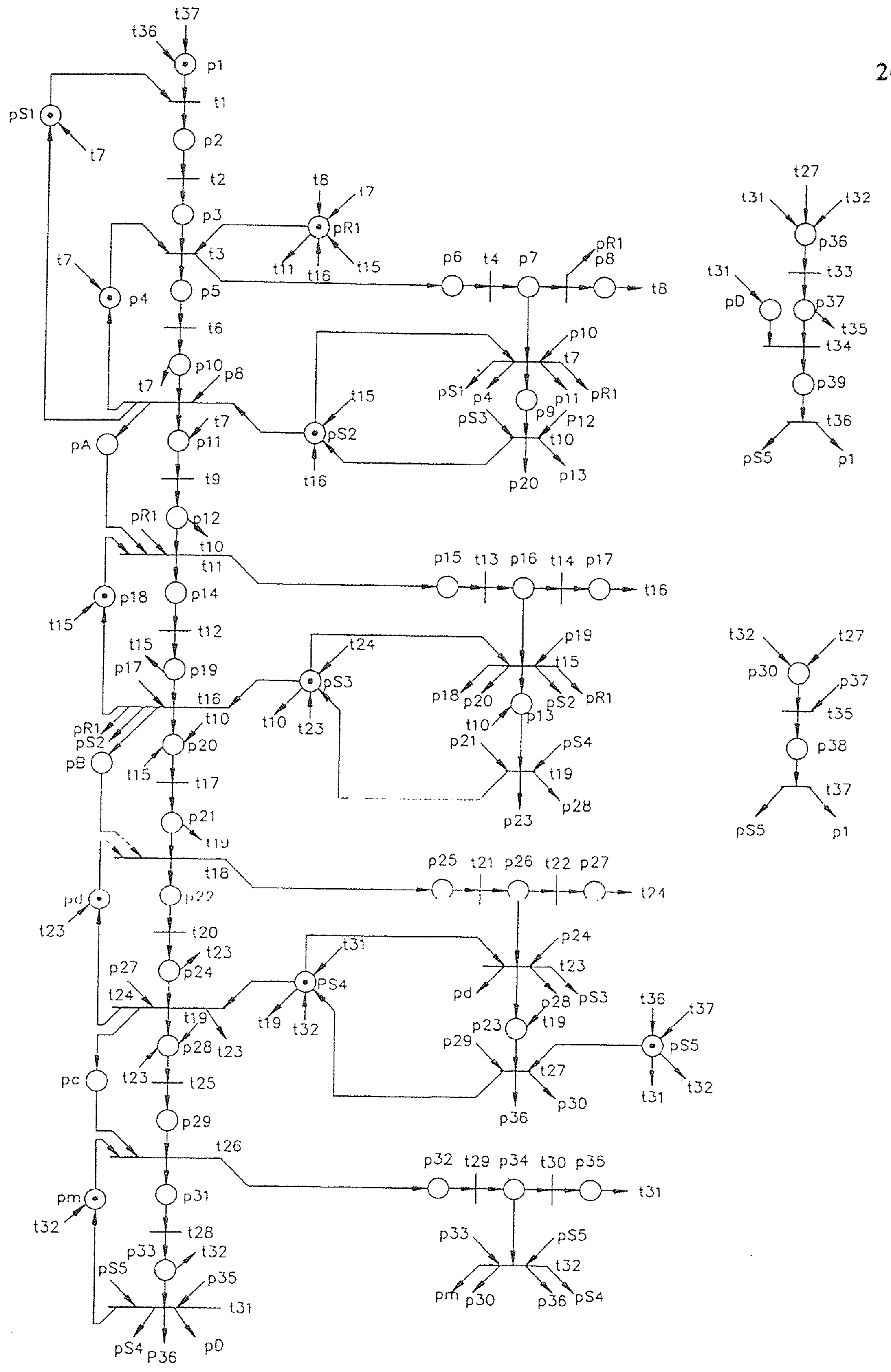


FIGURE 3.4- PETRI NET MODEL OF FIVE SEQUENCE WORK STATION

**Table 4** Place Designations for S1 Failure module.

| Places | Processes |
|---|---|
| $p_6$ | Timer start |
| $p_7$ | Wait state |
| $p_8$ | Process completed |
| $p_9$ | Pallet marked bad |

# CHAPTER 4

# PERFORMANCE EVALUATION OF THE SYSTEM

In order to evaluate the performance of the system, its PN model has to be analyzed. In this study SPNP software package is used for performance evaluation.

In order to run the model with the SPNP software a source code must be written. This code provides the necessary input and also generates certain output. The input consists of places along with their initial markings, transitions along with their firing rates, placement of input and outputs arcs, and any variables that the user intends to change from run to run; e.g., firing rates and failure rates.

## 4.1 Software Packages

There are numerous software packages available for modeling and evaluating given systems. Some of the more popular packages used are SPNP, GRAMAN, SIMAN, SLAM, and SIMSCRIPT. Of these SIMAN, SLAM, and SIMSCRIPT are better suited for simulation. SPNP and GRAMAN were developed around and intended to be used for Petri nets. Beck and Krogh [13] have successfully utilized SIMAN to simulate their modified Petri nets models for manufacturing systems. Also, GreatSPN [8] software is powerful in the sense that it can accept various time variables, inhibitors, and random switches and has simulation capacity. SPNP (Stochastic Petri nets Package) developed by Ciardo [9] has been used by Al-Jaar and Desrochers [2] to investigate the performance of transfer lines and production networks, also by Zhou and Leu [23] to evaluate the performance of two robotic manipulator assembly station for printed circuit boards. A UNIX based version of SPNP has been found to be appropriate tool to evaluate this five work station example. SPNP is written in C and is also available for VMS systems (VAX). The syntax and semantics are based on the C language.

## 4.2 Requirements and Terminology of SPNP

The SPNP software was used to execute the Petri nets model. Due to the complexity of the PN, the input matrix is large and occupies much memory during system execution. Also, when there are more than one pallet in the system, numerous markovian states result and hence the execution of the PN consumes much time. Hence, the model was run on SPNP with an initial marking with one token for pallet and part so that throughput and utilization of resources can be analyzed with SPNP. Example of input terminology for SPNP is described in Table 5. Details of the illustrative example and a five work stations system given in Appendix A and Appendix B respectively. Also, the output of SPNP program is given in Tables 6 - 10 and in Appendix C for the illustrative example and S4 in a five work station system, respectively.

**Table 5** Terminologies for SPNP Software

| |
|---|
| place("p1"); ** Establishes a place p1. ** |
| init("p1",1); ** Defines initial marking of p1 as 1. ** |
| trans("t1"); ** Establishes a transition t1. ** |
| iarc("p1","t1") ** Defines an input arc to t1 from p1. ** |
| oarc("p1","t2") ** Defines an output arc from t2 to p1. ** |
| rateval("t1",1.0) ** Defines firing rate of t1 is 1.0. ** |
| rateval("t2",x) ** Defines firing rate of t2 is variable x,which is entered at the beginning of the program run. ** |
| OUTPUT COMMANDS:<br>reward_type ef0() {return(rate("t1"));} /* throughput */ **<br>pr_expected("throughput,'t1' ",ef0);<br>Defines Expected throughput of transition t1. **<br>reward_type ef1() {return(1-mark("pR1"));}<br>pr_expected("utilization, 'pR1' ",ef1);<br>Defines Expected utilization of resources namely Robot #1. |
| OTHER OUTPUT FILES GENERATED BY SPNP SOFTWARE:<br>iopt(IOP_PR_RSET,VAL_YES);<br>If VAL_YES is specified, the program will print reachability set under ".rg" files. Which helps in analyzing deadlock marking(s) and other properties. |

**Results of Evaluation**

For each given command for output the SPNP gives corresponding output at a designated output files. For the illustrative example some of the output commands and there outputs given by the SPNP are explained below.

● (IOP_PR_RSET, VAL_YES) gives the following results under ".rg" files.

Assign ascending order starting from zero to each places and transitions as given in the SPNP program. This output can be read as marking $m_0$ being a tangible marking and having two tokens in place 0, one token in place 1, zero in place 2, one in place 3, zero in place 4, in place 5, two in place 6 and zero in place 7. If

(IOP_PR_FULL_MARK, VAL_NO) gives short form of _reachset in which only places with tokens are given with corresponding number of tokens in each place. VAL_YES gives long form as given above and is easy to read if SPN has a small number of places.

- (IOP_PR_RGRAPH, VAL_YES) give the following results under ".rg" files.

**Table 6** ".rg" File of the Illustrative Example

```
_nplace = 8;
_ntrans = 5;

_places =                       _transitions =
        0: p1;                          0: t1;
        1: p2;                          1: t2;
        2: p3;                          2: t3;
        3: p4;                          3: t4;
        4: p5;                          4: t5;
        5: p6;
        6: p7;
        7: p8;
_ntanmark = 9;

_nabsmark = 0;

_nvanmark = 0;

_nvanloop = 0;

_nentries = 10;

_reachset =

#      p1     p2     p3     p4     p5     p6     p7     p8

0_t    2      1      :      1      :      :      2      :
1_t    1      :      1      1      :      :      2      :
2_t    1      :      :      :      1      :      2      :
3_t    1      1      :      1      :      1      1      :
4_t    :      :      1      1      :      1      1      :
5_t    1      :      :      1      :      :      2      1
6_t    :      :      :      :      1      1      1      :
```

**Table 6** (Cont.)

```
7_t    :      1      :      1      :      2      :      :
8_t    :      :      :      1      :      1      1      1

_reachgraph =
0      1:     0:1
1      2:     1:1
2      3:     2:1
3      5:     3:1    4:     0:1
4      6:     1:1
5      0:     4:1
6      7:     2:1
7      8:     3:1
8      3:     4:1
```

This output can be read as marking 0 entering marking 1 by firing transition 0 with probability 1. Also, marking 3 can go to marking 5 by firing transition 3 with probability 1 and to marking 4 by firing transition 0 with probability 1.

With IOP_PR_RSET and/or IOP_PR_RGRAPH VAL_TANGIBLE specifies that only tangible markings must be printed in ".rg" file.

- (IOP_PR_MC, VAL_YES) gives ".mc" file. It has default VAL_CTMC, this describes the Continuous Timed Markov Chain (CTMC) derived from the SPN; the vanishing markings are absent and only numerical rates appear. The transition rate matrix is described by rows, since default is set for (IOP_PR_MC_ORDER,VAL_FROMTO) else if VAL_TOFROM its transpose is printed in ".mc" file. The ".mc" file of the illustrative example is given in table 7.

**Table 7** ".mc" File of the Illustrative Example

```
_firstindex = 0;
_nstates = 9;
_nentries = 10;
_order = _FROMTO;
_matrix =
0       1:1
1       2:1
2       3:1
3       4:1     5:1
4       6:1
5       0:1
6       7:1
7       8:1
8       3:1
```

This can be read as for _matrix = 3 4:1 5:1; means that transition rate from state 3 to state 4 is 1 or to state 5 is 1. The _TOFROM is in the following format: 4 ... 3:1 ....;

5 ... 3:1 ....;

Also, these files have IOP_METHOD that allows to set the numerical solution method for the CTMC, the default setting is steady state SOR (_SSSOR), and other options are given in [9]. The precision can be set with FOP_PRECISION with non-negative double value or default is 1e-06, and maximum number of iterations can be set with IOP_ITERATIONS with non-negative int or has default value is 2000.

● (IOP_PR_PROB, VAL_YES) gives ".prb" file. This file gives the steady-state probabilities for tangible markings; corresponds to the result of the CTMC solution. In this file the _method may be changed automatically and values in it reflect the actual choice and may be different from the one declared in ".mc" file. When the maximum number of iterations has reached, it automatically switches to Steady State Gauss-Seidel. The ".prb" file of the illustrative example is given in Table 8.

**Table 8** ".prb" File of the Illustrative Example

```
_firstindex = 0;
_nstates = 9;
_method = _SSSOR;
_precision = 0.000
_iterations = 1;
_probabilities =
0:0.111      1:0.111      2:0.111
3:0.111      4:0.111      5:0.111
6:0.111      7:0.111      8:0.111
_end;
```

Also, output generates ".log" file which contains the number of tangible markings, number of vanishing markings, number of arcs, number of remaining arcs after elimination of redundant arcs, number of remaining arcs after elimination of vanishing markings and solving method. This file is also, shown during the execution of SPNP program. The ".log" file of the illustrative example is given in Table 9.

**Table 9** ".log" File of the Illustrative Example

```
The Reachability graph contains:

        9 tangible markings

        0 vanishing markings

        10 arcs
```

**Table 9** (Cont.)

```
After elimination of redundant arcs:

        # of remaining arcs:      10

After elimination of vanishing markings:

        # of remaining arcs:      10

Solving the Markov chain....

...Markov chain solved

Reading the reachability graph info ...

End of execution.
```

".out" file in Table 10 gives output request in ac_final, Probability of each places having nonempty token, Average token in each places, Probability of enabled transitions and Average throughput of each transitions.

**Table 10** ".out" File of the Illustrative Example

```
EXPECTED: throughput,'t3' = = 0.222
EXPECTED: throughput,'t5' = = 0.222
EXPECTED: utilization, 'Robot' = = 0.667
EXPECTED: utilization, 'Buffer' = = -0.333

AVERAGE:

PLACE        Pr[nonempty]  Av[tokens]
0:p1         0.555              0.667
1:p2         0.333              0.333
2:p3         0.222              0.222
3:p4         0.777              0.777
4:p5         0.222              0.222
5:p6         0.555              0.667
6:p7         0.889              1.333
```

**Table 10** (Cont.)

| | | |
|---|---|---|
| 7:p8 | 0.222 | 0.222 |
| TRANSITION | Pr[enabled] | Av[throughput] |
| 0:t1 | 0.222 | 0.222 |
| 1:t2 | 0.222 | 0.222 |
| 2:t3 | 0.222 | 0.222 |
| 3:t4 | 0.222 | 0.222 |
| 4:t5 | 0.222 | 0.222 |

The Production rate (P) in the five station assembly system is taken to be small so that a result can be obtained from SPNP software. The failure rate (F) is defined as

$$F = \frac{R}{1-R} \times P$$

where R is the rate of failure

The production rate of all the other station was constant while production rate of station in question varies. Similarly, failure rate was constant for all the stations while production rate varies for the station in question. The data has been taken for four stations and at three failure rates namely 1% ,10%, 30%, also data has been taken with one pallet in the system. The data for three failure rates are given in Table 11, Table 12 and Table 13 respectively. The firing rate of transition is exponentially distributed. The time delay is random and exponentially distributed, and the resulting firing rate, λ, is equal to inverse of expected time delay

$$\lambda = \frac{1}{E(\tau)}$$

where E($\tau$) is the average time delay. See Appendix C for the output files of S4 in a five sequence work station system.

## 4.3 Performance Analysis of the System

The outputs requested from SPNP include station and robot utilizations, throughput of each station, as well as system production rate. The throughput of the stations S1, S2, S3, S4 and S5 are the throughput of transitions $t_6$, $t_{12}$, $t_{20}$, $t_{28}$ and $t_{37}$ respectively. The system production rate is the throughput of the last station, i.e., S5. The utilization of a resource is defined as

1 - (token availability of the resource place).

From the Table 11 and Figure 4.2 utilization of S4 is 0.214790 when production speed of the stations S1-S3 and S5 is 5; the failure rate of each stations S1-S3 and S5 is 0.1; the production speed of S4 is 0.1 and failure rate of S4 is 0.1. The token availability in place $p_{s4}$ is 0.785210.

The following observation is made. As the average machine rate increases, the system production rate increases and the utilization of the work station decreases, until the system production rate is saturated. The average firing rate of $t_{36}$ gives throughput of finished goods unloaded at S5 and $t_{37}$ gives throughput of unfinished goods unloaded at S5.

The system production rate =

$$\frac{\textit{Average firing rate of } t_{36}}{\textit{Average firing rate of } t_{36} + \textit{Average firing rate of } t_{37}} \times 100$$

The system production rate and utilization of S4 are given in Figures 4.1 - 4.3 for each failure rate of 1%, 10% and 30% respectively. As shown in Figures 4.1 - 4.3 increase in failure rate of the system reduces production rate. Thus the system efficiency decreases. Due to the production rate (P) in the five station assembly system is taken to be small significant difference in the outputs for three failure rates is not achieved.

**Table 11** System Production rate and Utilization of S4 when Failure rate is 1%

| Average Machine Rate S4 | Average Failure Rate S4 | System Production Rate | Utilization S4 |
|---|---|---|---|
| .020 | .01 | 6.437935E-03 | 4.140652E-01 |
| .040 | .01 | 7.663150E-03 | 3.025704E-01 |
| .060 | .01 | 8.475539E-03 | 2.559349E-01 |
| .080 | .01 | 8.453690E-03 | 2.306162E-01 |
| .100 | .01 | 8.626459E-03 | 2.148873E-01 |
| .120 | .01 | 8.743051E-03 | 2.042707E-01 |
| .140 | .01 | 8.826289E-03 | 1.966889E-01 |

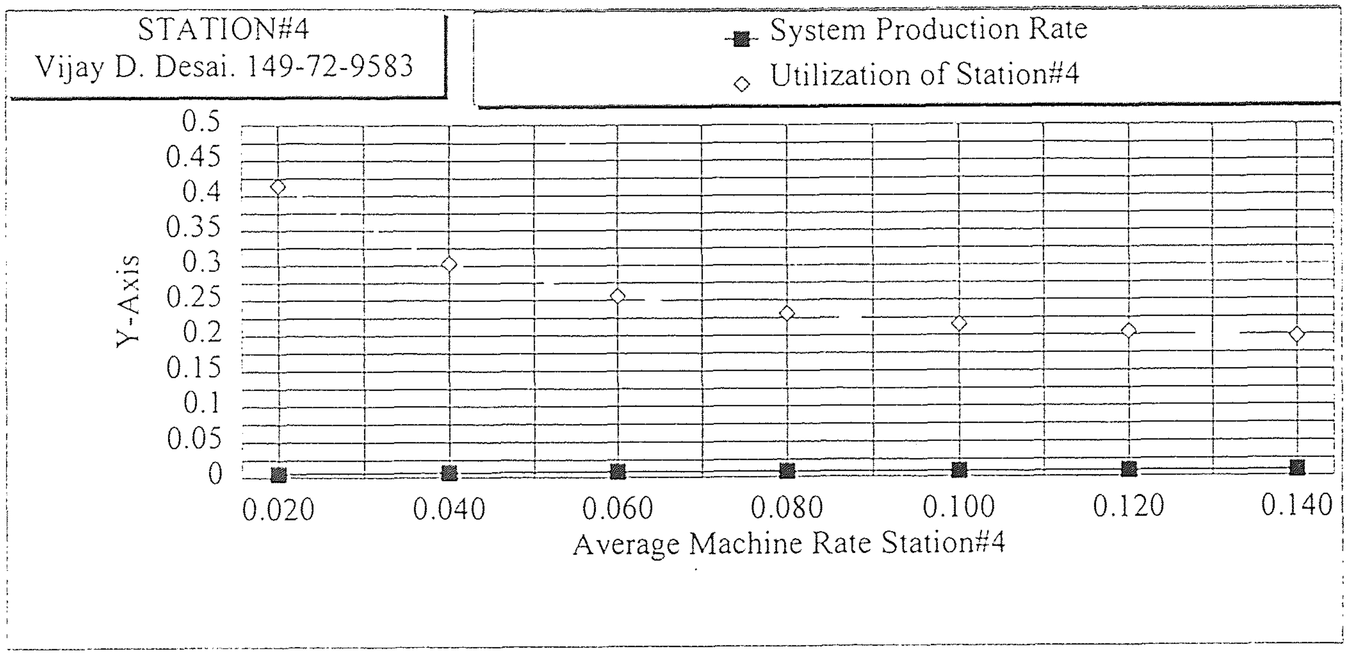


Figure 4.1
Average Failure Rate for S4 is 1%.
Average Machine Rate is 5 and Average Failure Rate is 10% for S1, S2, S3, and S5.

**Table 12** System Production rate and Utilization of S4 when Failure rate is 10%

| Average Machine Rate S4 | Average Failure Rate S4 | System Production Rate | Utilization S4 |
|---|---|---|---|
| .020 | .1 | 6.438082E-03 | 4.140612E-01 |
| .040 | .1 | 7.663084E-03 | 3.025535E-01 |
| .060 | .1 | 8.175379E-03 | 2.558972E-01 |
| .080 | .1 | 8.453401E-03 | 2.305513E-01 |
| .100 | .1 | 8.626012E-03 | 2.147897E-01 |
| .120 | .1 | 8.742420E-03 | 2.041356E-01 |
| .140 | .1 | 8.825454E-03 | 1.965124E-01 |

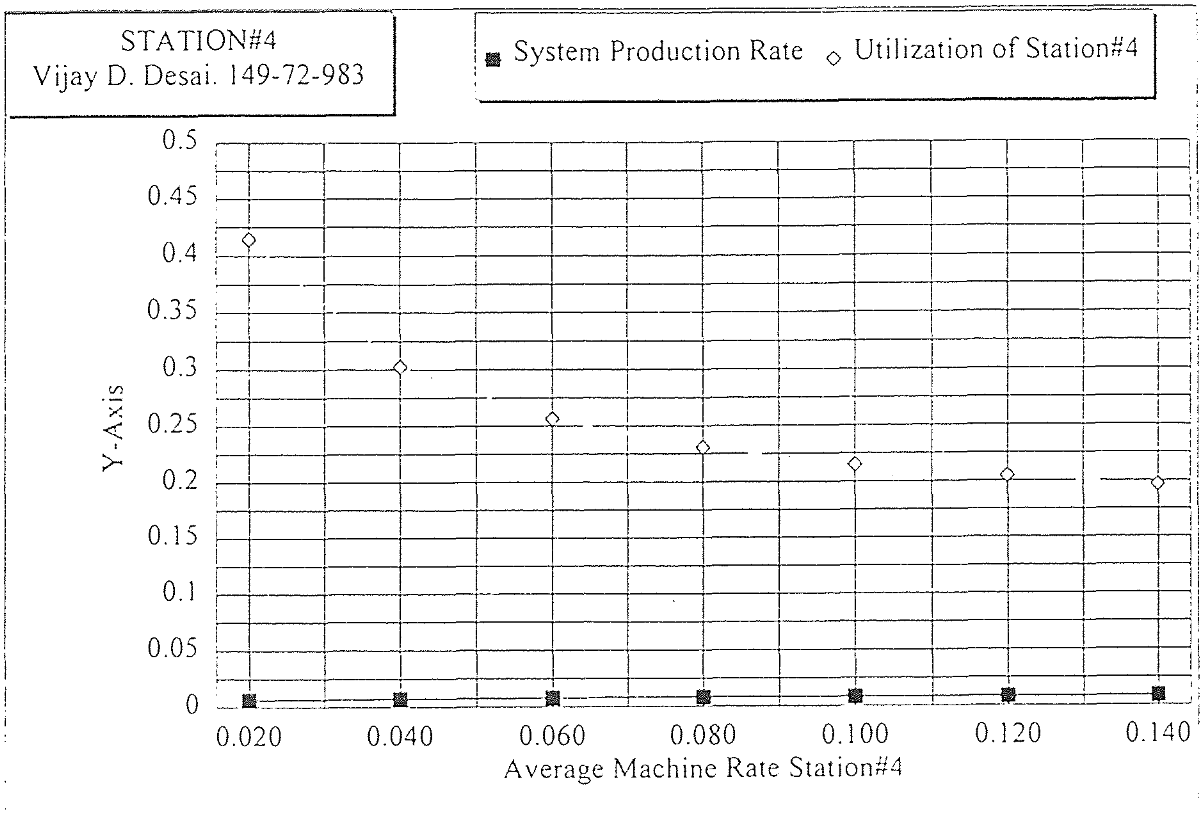


Figure 4.2
Average Failure Rate of S4 is 10%
Average Machine Rate is 5 and Average Failure is 10% for S1, S2, S3, and S5

**Table 13** System Production rate and Utilization of S4 when Failure rate is 30%

| Average Machine Rate S4 | Average Failure Rate S4 | System Production Rate | Utilization S4 |
|---|---|---|---|
| .020 | .3 | 6.438044E-03 | 4.140489E-01 |
| .040 | .3 | 7.662883E-03 | 3.025018E-01 |
| .060 | .3 | 8.174898E-03 | 2.557839E-01 |
| .080 | .3 | 8.452544E-03 | 2.303590E-01 |
| .100 | .3 | 8.624704E-03 | 2.145043E-01 |
| .120 | .3 | 8.740601E-03 | 2.037463E-01 |
| .140 | .3 | 8.823076E-03 | 1.960108E-01 |

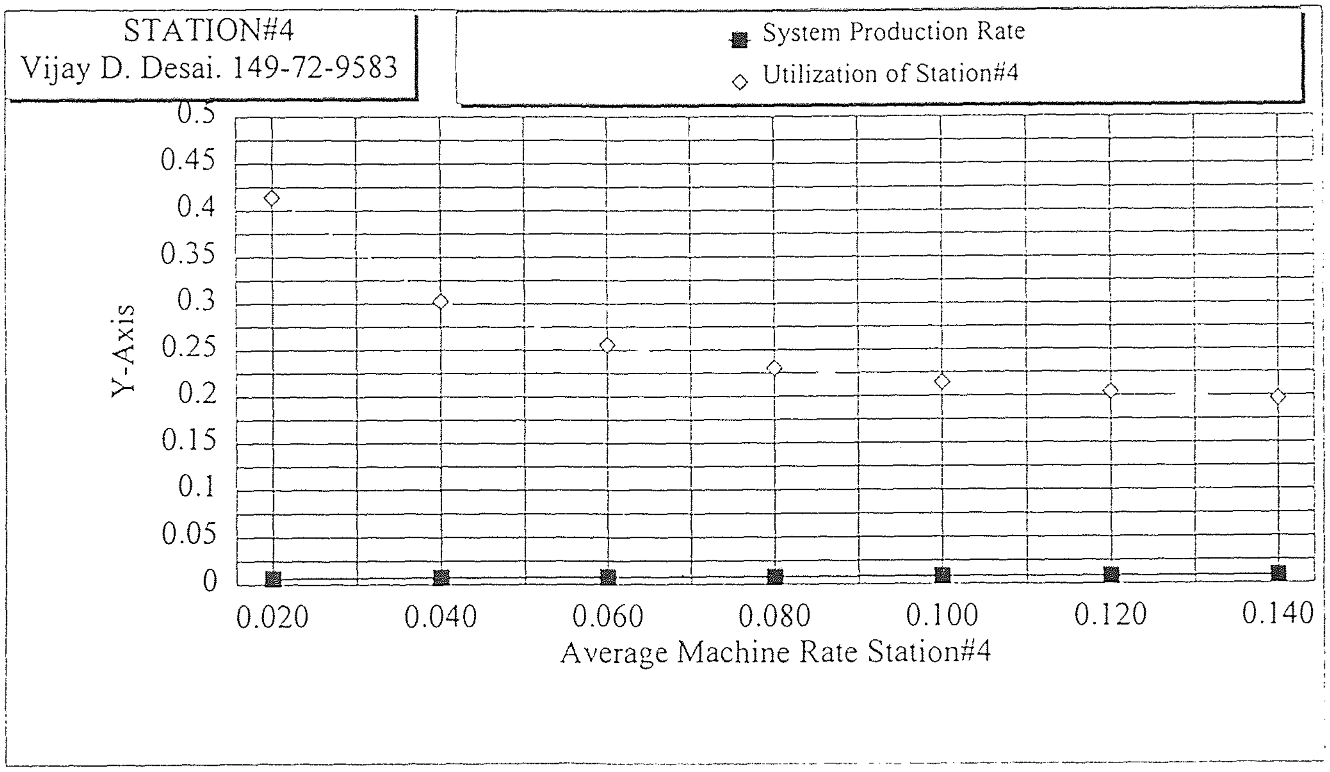


**Figure 4.3**
Average Failure Rate of S4 is 30%.
Average Machine Rate is 5 and Average Failure Rate is 10% for S1, S2, S3 and S5.

From the other output files of SPNP program (Appendix D) the following Petri net properties can be observed. ".log" file states that, the reachability graph contains 45 tangible markings, 0 vanishing markings and 57 arcs. Since this file does not contain any absorbing marking, the system is deadlock free. From ".rg" file, the system is found to be bounded and safe. The system is live, since for any marking $m_0$, it is possible to fire ultimately any transition by executing some firing sequence. The system is reversible, since marking $m_0$ can be reached from any marking $m \in R(m_0)$, which also implies consistency.

# CHAPTER 5

# CONTROL OF THE SYSTEM USING PLC

In this thesis PLC is used to control the system. The application of Programmable Logic Controllers (PLCs) for system control is given in this chapter.

PLC can be defined as a specialized computer processor in a computer family. It is capable of storing instructions to implement control functions such as sequencing, timing, counting, arithmetic, data manipulation and communication.

## 5.1 History of PLC

In 1968 in order to eliminate high costs, Hydramatic Division of the General Motors Corporation replaced inflexible relay-controlled systems with PLCs which are flexible, easily maintainable, and programmable. PLCs are able to sustain in a harsh environment and can be reused in the system. Also, it provides expendability for the future. To achieve these criteria a PLC was developed. In the beginning the PLC was designed to replace relay-controlled systems which were widely used in automotive industry. In 1970's with improvement of microprocessor technology PLCs were capable of data manipulation, arithmetic and communication with other PCs and PLCs. PLCs have diagnostic indicators that aid trouble shooting and PLCs are designed in a modular form for replacement or repair of sub-assemblies. By 1971 PLCs were also being used in food and beverage, metals manufacturing and pulp and paper. CRT technology made PLC programming more flexible and aided a trouble shooting process. Between 1975 and 1979 Hardware and Software enhancements gave PLCs more flexibility in memory capability, remote Input/Output capability, communications and fault detection. Also, a PLC reduces cost of installation and implementation of a control system. With today's technology, PLCs have developed programs that are flexible, shorter, faster and higher memory capacity. Compared with a PC-based controller, they are very expensive. However, the harsh environments still justify their extensive usage.

## 5.2 Description of a PLC

A PLC is composed of two basic sections, the CPU and Input/Output interface to the system. The CPU is composed of three components: the processor, the memory system and power supply of the system. In the operation of PLCs, Inputs/Outputs are connected to field devices which are used in the control of a process. CPU receives and processes information from field devices via control program and then updates the output devices.

The ladder diagrams are used in providing control information from the designers to the users of equipment. Since it is easy to use and interpret, it was widely accepted in the industry. Proper implementation of a control system depends on the knowledge of PLC operation, scanning and instruction programming. In PLC logic functions are programmed and can be easily changed. Relay logic implemented in PLCs is based on three basic functions (AND, OR, NOT). These functions are used either singly or in combinations to form instructions to control devices. These instructions are implemented using ladder diagram language, which is a one step translation from relay logic.

The complete ladder diagram can be thought of as being formed by individual circuits, each circuit having one output. Each of these circuits is known as a rung. A rung is the contact symbology required to control an output in the PLC. Some controllers allow a rung to have multiple outputs, but one output per rung is convention. Each rung is a combination of input conditions (symbols) connected from left to right between two vertical lines, with the symbols that represent the output at the far right. The input symbols are connected in series, parallel or some combination to obtain the desired logic.

Each symbol on the rung has a reference number, which is the address in memory where I/O status is stored. The address for given I/O can be used throughout the program as many times as required by the control logic. In relay logic, additional contacts mean additional hardware, compared to ladder logic. The address reference is dependent on the controller, but most of them are octal (base 8) or decimal (base 10) numbering.

The main function of the ladder diagram program is to control outputs and perform functional operations based on input conditions. The continuity is achieved whenever a path contains contact elements in a closed condition so that power flows from left to right. These contact elements will either close or remain closed according to the status of its reference inputs. The maximum number of ladder contact elements that can be used to program a rung is restricted by the ladder rung matrix, and its size differs among different PLC manufactures and programming device used. One of the most important rules in all PLCs is that of reverse power flow. Reverse power is not allowed in PLC logic to prevent possible sneak paths that could occur in hardwired electromechanical relay systems.

## 5.3 Description of Basics of Ladder Diagrams

The following symbols are used in translating relay control logic to contact symbolic logic.

—] [— **Normally-opened contact:** Input to control logic, either external input or internal output. When evaluated by the program it is examined for its status "1". If so, the contact is closed, else it is open.

—]/[— **Normally-closed contact:** Input to control logic, either external input or internal output. When evaluated by the program it is examined for its status "0" if so then contact is closed else contact will open.

-( )- **Output:** Represents any output that is driven by some combination of input logic. It can be connected to device or can be internal output. If any left-to-right path of a rung has all contact closed then output of the rung is energized.

**Timer:** There are different types of timers supplied by manufacturers. They are 1) Timer ON Delay Energize 2) Timer ON Delay De-energize 3) Timer OFF Delay Energize and 4) Timer OFF Delay De-energize.

Also, there are two formats. Block format timer may have one or two inputs in which case one pin is control and the other is for enable/reset. If both inputs are true, then the timer block starts timing. Ladder format timer has one pin input. If it is high then timer starts. There are two registers used in the timer. One is to store the preset value and the other is to store the accumulated value. The time base is dependent on type of PLC used (e.g. 0.01 sec, 0.1 sec, 1.0 sec, etc.)

-(L)- **Latch output:** A latch output remains energized, even though the status of the contacts which cause the output to energize may change. It remains latched ON until it is unlatched by an unlatch output instruction of the same reference address.

-(U)- **Unlatch output:** The unlatch output instruction is programmed to reset a latched output having the same reference address. If any rung path has logic continuity to the unlatch output, the reference address is unlatched to an OFF condition.

There are other types of contact symbolic logic used in PLC program such as counter, jump to, go to subroutine, return coil, arithmetic instructions, etc. These applications and their instructions can be found in [5].

The programmable controller reads all field input devices, executes the control program and updates all field output devices. This process is called scan.

The last process of updating output is in two steps. First step is to update internal output and continue to execute the program until it has finished evaluating the control program; second step is to update output interface modules thereby the field devices. The scan time, the time it takes to implement scan, consists of the program scan time and the I/O update time. PLC manufactures generally specify the scan time based only on the amount of application memory used (e.g. 10 msec/1K of programmed memory) [5]. The remote location of I/O subsystems increases scan time. Some PLCs provide software instructions that allow the interruption of the continuous program scan in order to receive an input or update an output immediately. These immediate instructions are very useful when the PLC must react instantaneously to a critical input or output.

The power supply requirement to run PLC is usually 120VAC or 220VAC, but some controller provides PLC's to run on 24 VDC. Input supply must be connected to Isolation Transformer and through EMI filter for any fluctuation of line voltage and noise due to on/off of devices connected to PLC causing noise and power loading. If summation of the current requirements for a particular I/O configuration is greater than the total current supplied by the power supply, then the second power supply is required.

The amount of application memory is specified in terms of unit K where each K unit represents 1024 word locations. The total number of storage locations available is indicated by the memory capacity of a particular controller in the units of K. After determining the minimum memory requirements for the application, one should add 25% to 50% more memory for future changes and modifications.

The memory organization and interaction of the data table's I/O mapping and storage area helps to comprehend the functional operation of a programmable controller. Also, it helps in understanding of how the control software program is organized and developed.

In PLC the word length is two bytes, 16 bits. The starting address of control program is not very important but register address references are.

There are different types of input and output interfaces connected to a PLC. Selector switches, pushbuttons, limit switches, proximity switches, and Thumbwheel switches are example of digital or discrete type. The standard ratings for these inputs and/or outputs are 24 Volts AC/DC, 48 Volts AC/DC, 120 or 230 Volts AC/DC, TTL level etc. Detail of these inputs and outputs can be found in [5]. Also, there are Analog I/O that can be connected to PLC [5].

## 5.4 Development of the PLC Program

In the development of PLC program, a modular approach similar to the Petri net model is used. First, all the common rungs are written for all the stations and then these rungs are copied into files for each station. Next, individual rungs and their addresses in each station are added as per each stations job. These process are valid in a PLC, since it does not update any output until it has scanned the program completely. If input conditions are satisfied then the output contacts are made. Thus, the execution of the process is done in the actual sequence, even if the program steps are not in sequence. The PLC program is written considering many of the practical conditions such as motion of robot arms, the use of an operator key for the reset of a work station due to the machine failure.

### 5.4.1 Ladder Logic Program for S1 Module

The portion of flow chart for the sequence of process is shown in Figure 5.1 and the PN model in Figure 3.2, the token in $p_{s1}$ implies that S1 is ready; in ladder logic program (LLP), output B:3/1 is energized in rung 2:0. In rung 2:1 output O:3/0 is energized, when $p_2$ has a token in the PN model. When timer T4:0 is done the pallet is marked good and output O:3/3 is energized in rung 2:6, i.e., $p_3$ has a token in the PN model. When a part for S1 is available in the feeder input I:1/6, I:1/6 becomes a closed contact, as in the PN model $p_4$ has a token. Similarly, I:1/5 becomes a closed contact when robot is available

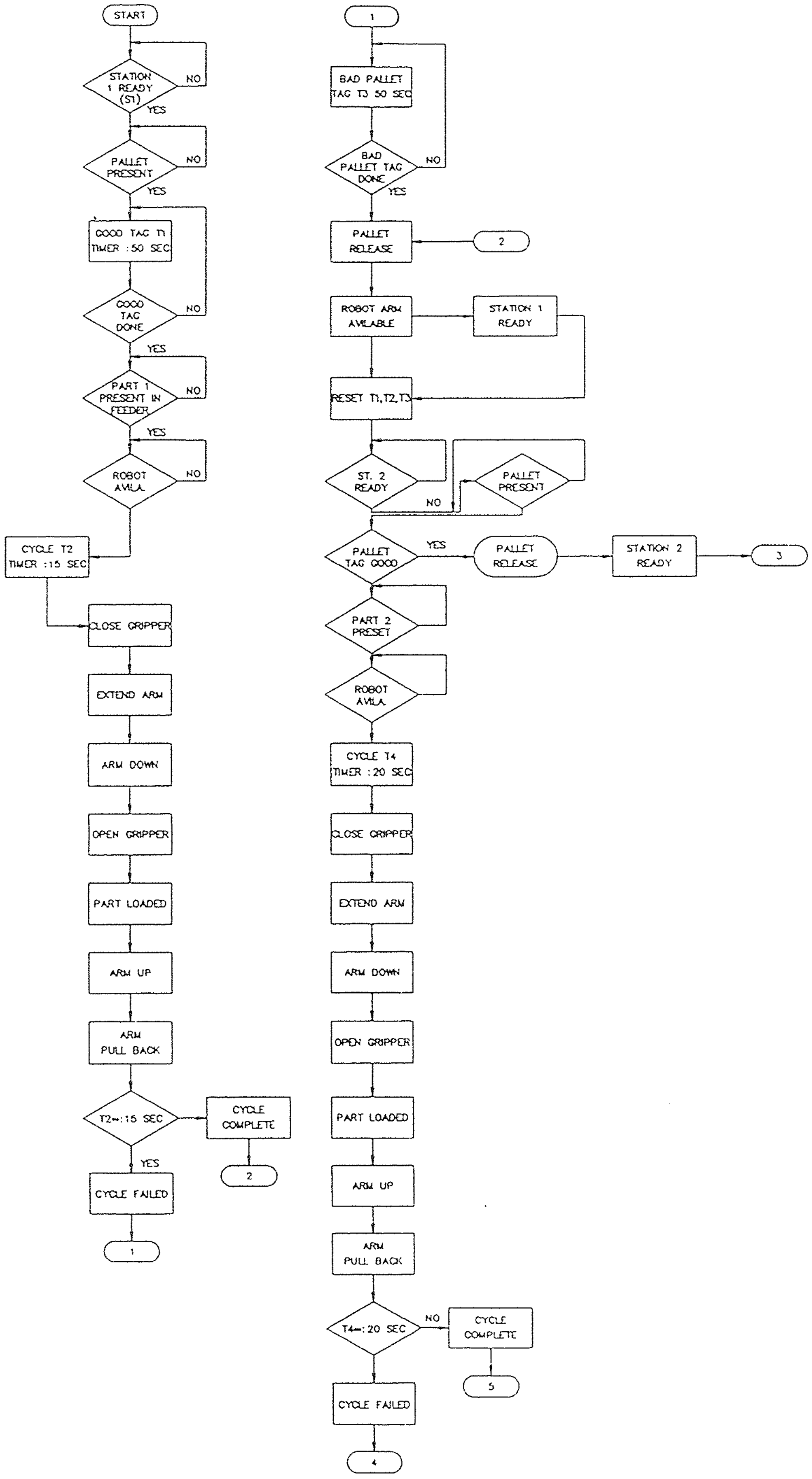


**Figure 5.1** Flow Chart of Five Sequence Work Station For Ladder Logic Program

and ready or $p_{R1}$ has a token. Place $p_5$ having a token implies that robot is ready with the part to be loaded, and output O:3/8 is energized and latched at rung 2:20. The operation at S1 is completed when B:3/5 is energized at rung 2:25. The enabling of a transition in the PN model can be regarded as that all the input conditions are satisfied to have an output to be energized in the LLP's.

### 5.4.2 Ladder Logic Program for S1 Failure Module

Similar to the S1 module, all places and transitions in the PN model of the failure module as shown in Figure 3.3, can be represented in a LLP. In the LLP when an operation fails, a counter C5:0 starts counting the number of consecutive failures of an operation at a given station. If a next operation is successfully completed or, if operator resets the station then the counter is reset. When the counter reaches its preset value it shuts down that particular station and a red light goes on, which is done in rungs 2:3 and 2:4. In the PN model an operation fails when $t_7$ fires; and in the LLP output O:3/9 is energized and latched. A token in place $p_9$ states that a pallet is being marked bad. This process in LLP is done in three rungs with four inputs, two outputs and a timer. Timer T4:1 starts when either the operation at the S1 fails or when an operator resets using a key. When T4:1 starts timing, output O:3/4 is energized and the pallet begins to tag bad. The output, O:3/5, a latched output is energized when timer T4:1 is done and the pallet is tagged bad. The pallet is released when $t_8$ or $t_{10}$ fires in the PN and in LLP input B:3/4 or O:3/5 is closed and output O:3/6 is energized. Once the pallet is released all the latched output are unlatched. Also, in the LLP an operator key unlatches all the latch outputs of the station. The rest of the flow chart is given in Appendix E and S1's LLP is given in Appendix D.

## 5.5 Ladder Logic Program for the System

The ladder logic program for the S1 of the system considered is shown in Appendix D. This has been developed using Allen-Bradly's Advance Programming Software [1]. In the system, 10 slot rack has been used. Each slot has 16 I/O locations. There are different racks, each of which can have different type of slots, different I/O, and CPU given in the software configuration files. Slot 0 is for CPU with 12K user memory, slots 1,2,7,8 are used for inputs with 24VDC and slots 3,4,5,6 are used for discrete outputs. Internal bits for S1 are from B3/0 to B3/10. Similarly, internal bits for S2, S3, S4, and S5 are from B3/11 to B3/20, B3/21 to B3/30, B3/31 to B3/40 and B3/41 to B3/50, respectively. The internal bit in the software can go up to B3/999. There are thirteen timers and five counters used in the system LLD program. Following steps are used to develop the LLP in Allen-Bradly's Software using a PC.

The software is loaded under IPDS/ATTACH/SLC500 directory. At the prompt entering APS allows programming, editing and executing of a PLC program. All the menu are given at the bottom of the screen and controlled by function key from F1 through F10 and ESC key is used for *cancel*. The main menu in the software package is given in Table 14.

**Table 14** Main Menu

| Fl | F2 | F3 | F4 | F5 | F6 | F7 | F7 | F9 | F10 |
|---|---|---|---|---|---|---|---|---|---|
| ONLI NE | ONLI NE CON FI | OFFL INE PRO G/DO C | OFFL INE CON FIG | WH O | SYST EM CONF IGUR ATIO N | FILE OPTI ONS | PRIN T REP ORT S | SYST EM UTIL ITIES | EXI T SY STE M |

The program edited in *offline configuration mode* since no PLC is connected to a communication port. In the file, option file is created and I/O configuration is setup. In the next menu F8, *monitor file*, is entered and at this level file is edited. In the file edit mode, other options are given, such as *insert rung*, *modify rung*, *append rung*, *delete rung*, *edit branch*. In this mode the last rung is always "*end*" which cannot be deleted. At this point all the rungs are added and edited. Once the rung is inserted, the insert instruction is entered. At this point another set of menu is given at the bottom of the screen, where one can enter different types of instructions, for example, timer/counter, comparator, I/O message, and bit (normally open input, normally close input, output, latch output, unlatch output). After an input type, input bit address is entered. The bit address is at which your particular input is connected in the particular slot or it is an internal input bit. For example input I:1/0 is an external input connected in slot one location zero and is a cycle start input. Similarly all the instruction can be edited and at the exit of the file monitor it prompts you to save the file and to overwrite the old one. One can edit any instruction during the running process. At the next scan, the program responds to the new instruction and output is updated accordingly.

The program printout also gives for each contact used in the rung their locations throughout the program.

# CHAPTER 6

# CONCLUSIONS

Petri nets (PN's) are an emerging tool for performance modeling and discrete event control of advanced manufacturing systems. In this thesis, in order to show the application of PN's for modeling and performance analysis, an example system is considered. The system consists of one robot and five work stations. The detailed operation of this system is clearly modeled using the PN model. From the model, the system performance is investigated to study the effects of machine failure rates and machine rates on system throughput and resource utilization. Furthermore, a PLC program to control this system is developed using ladder logic diagram (LLDs) in Allen-Bradley PLC.

The reason for using PNs for system performance is that they are more suitable for the detailed investigation of the effect of system parameters on system performance. However, LLDs are used for programming Programmable Logic Controllers (PLCs) since PLCs are widely used in industry and suitable for the harsh environment. This work aims to convince industrial practitioner to highlight the applicability of PNs for their modeling application.

Further research should focus on using the PN model to generate automatically the LLD for discrete control.

## Appendix A

## SPNP Program of the Illustrative Example

```
# include        "user.h"


/* Illustrative Example */

/* The System is Deadlocked When number of token in p1 is greater than p7 */



parameters() {
        iopt(IOP_PR_FULL_MARK, VAL_YES);
        iopt(IOP_PR_RSET,VAL_YES);
        iopt(IOP_PR_MC,VAL_YES);
        iopt(IOP_PR_RGRAPH,VAL_YES);
        iopt(IOP_PR_PROB,VAL_YES);

}

net() {
      place("p1");
      place("p2");
      place("p3");
      place("p4");
      place("p5");
        place("p6");
        place("p7");
        place("p8");
```

```
    init("p1",2);

    init("p2",1);

    init("p4",1);

    init("p7",2);


    trans("t1");       rateval("t1",1.0);

    trans("t2");       rateval("t2",1.0);

    trans("t3");       rateval("t3",1.0);

    trans("t4");       rateval("t4",1.0);

    trans("t5");       rateval("t5",1.0);


        iarc("t1","p1"); iarc("t1","p2");

        oarc("t1","p3");

        iarc("t2","p3"); iarc("t2","p4");

        oarc("t2","p5");

        iarc("t3","p5"); iarc("t3","p7");

        oarc("t3","p2"); oarc("t3","p4"); oarc("t3","p6");

        iarc("t4","p6"); iarc("t4","p2");

        oarc("t4","p7");

        oarc("t4","p8");

        iarc("t5","p8");

    oarc("t5","p2");

        oarc("t5","p1");

}

/* the following 3 lines should appear in all programs */

    assert() {return(RES_NOERR);}

ac_init() {}
```

```
ac_reach() {fprintf(stderr,"\nThe reachibility graph has been generated\n\n");}


/* user defined out functions */

reward_type ef0() {return(rate("t3"));} /* throughput */

reward_type ef1() {return(rate("t5"));} /* throughput */

reward_type ef6() {return(1-mark("p2"));}

reward_type ef7() {return(1-mark("p7"));}




ac_final() {pr_expected("throughput, 't3' = ",ef0);

          pr_expected("throughput, 't5' = ",ef1);

          pr_expected("utilization, 'Robot' = ",ef6);

          pr_expected("utilization, 'Buffer' = ",ef7);


          pr_std_average();}
```

## Appendix B

## SPNP Program of Five Sequence Work Station System

```
# include       "user.h"

/ SPNP program for five work stations */

float x,y,z,f,a,b;

parameters() {

        iopt(IOP_PR_FULL_MARK, VAL_NO);

        iopt(IOP_PR_RSET,VAL_YES);

        iopt(IOP_PR_MC,VAL_YES);

        iopt(IOP_PR_RGRAPH,VAL_YES);

        iopt(IOP_PR_PROB,VAL_YES);


x=input("production rate of station1,station2,station3 and station5");

b=input("failure rate of station1,station2,station3 and station5");

y=(b/(1-b))*x;

z=input("production rate of station4");

a=input("failure rate of station4");

f=(a/(1-a))*z;

}


net() {

        place("p1");

        place("p2");

        place("p3");

        place("p4");

        place("p5");
place("p6");
place("p7");
place("p8");
place("p9");
place("p10");
place("p11");
place("p12");
place("p13");
place("p14");
place("p15");
place("p16");
place("p17");
place("p18");
place("p19");
place("p20");
place("p21");
place("p22");
place("p23");
place("p24");
place("p25");
place("p26");
place("p27");
place("p28");
place("p29");
place("p30");
place("p31");
place("p32");
place("p33");
place("p34");
place("p35");
place("p36");
place("p37");
place("p38");
place("p39");
place("ps1");
place("ps2");
place("ps3");
place("ps4");
place("ps5");
place("pm");
place("pd");
place("pR1");
place("pA");
place("pB");
place("pC");
place("pD");
init("p1",1);
init("p4",1);
init("p18",1);
init("ps1",1);
init("ps2",1);
init("ps3",1);
init("ps4",1);
init("ps5",1);
init("pd",1);

init("pm",1);

init("pR1",1);



trans("t1");       rateval("t1",x);

trans("t2");       rateval("t2",x);

trans("t3");       rateval("t3",x);

trans("t4");       rateval("t4",y);

trans("t5");       rateval("t5",y);

trans("t6");       rateval("t6",x);

trans("t7");       rateval("t7",y);

trans("t8");       rateval("t8",x);

trans("t9");       rateval("t9",x);

trans("t10");      rateval("t10",y);

trans("t11");      rateval("t11",x);

trans("t12");      rateval("t12",x);

trans("t13");      rateval("t13",y);

trans("t14");      rateval("t14",y);

trans("t15");      rateval("t15",y);

trans("t16");      rateval("t16",x);

trans("t17");      rateval("t17",x);

trans("t18");      rateval("t18",x);

trans("t19");      rateval("t19",y);

trans("t20");      rateval("t20",x);

trans("t21");      rateval("t21",y);

trans("t22");      rateval("t22",y);
trans("t23");      rateval("t23",y);
trans("t24");      rateval("t24",x);
trans("t25");      rateval("t25",x);
trans("t26");      rateval("t26",x);
trans("t27");      rateval("t27",y);
trans("t28");      rateval("t28",z);
trans("t29");      rateval("t29",y);
trans("t30");      rateval("t30",y);
trans("t31");      rateval("t31",x);
trans("t32");      rateval("t32",f);
trans("t33");      rateval("t33",x);
trans("t34");      rateval("t34",x);
trans("t35");      rateval("t35",y);
trans("t36");      rateval("t36",x);
trans("t37");      rateval("t37",y);

iarc("t1","p1"); iarc("t1","ps1");
oarc("t1","p2");
iarc("t2","p2");
oarc("t2","p3");
iarc("t3","p3");
iarc("t3","p4"); iarc("t3","pR1");
oarc("t3","p6"); oarc("t3","p5");
iarc("t4","p6"); oarc("t4","p7");
oarc("t5","pR1");
iarc("t5","p7"); oarc("t5","p8");
iarc("t6","p5"); oarc("t6","p10");
iarc("t7","p7"); iarc("t7","ps2");iarc("t7","p10");
oarc("t7","p9");
oarc("t7","p11"); oarc("t7","ps1"); oarc("t7","pR1");
oarc("t7","p4");
iarc("t8","p10"); iarc("t8","ps2");
iarc("t8","p8");
oarc("t8","p11"); oarc("t8","ps1");
oarc("t8","p4"); oarc("t8","pA");
iarc("t9","p11");
oarc("t9","p12");
iarc("t10","p9"); iarc("t10","p12"); iarc("t10","ps3");
oarc("t10","ps2");oarc("t10","p13"); oarc("t10","p20");
iarc("t11","p12"); iarc("t11","pA");
iarc("t11","p18"); iarc("t11","pR1");
oarc("t11","p14"); oarc("t11","p15");
iarc("t12","p14");
oarc("t12","p19");
iarc("t13","p15");
oarc("t13","p16");
iarc("t14","p16"); oarc("t14","p17");
iarc("t15","p19"); iarc("t15","ps3");
iarc("t15","p16");
oarc("t15","p20"); oarc("t15","p18");
oarc("t15","p13");
oarc("t15","ps2"); oarc("t15","pR1");
iarc("t16","p19"); iarc("t16","ps3");
iarc("t16","p17");
```

```
oarc("t16","p20"); oarc("t16","p18"); oarc("t16","pB");
oarc("t16","ps2"); oarc("t16","pR1");
iarc("t17","p20");
oarc("t17","p21");
iarc("t18","p21"); iarc("t18","pB");
iarc("t18","pd");
oarc("t18","p22"); oarc("t18","p25");
iarc("t19","p13"); iarc("t19","p21");
iarc("t19","ps4");
oarc("t19","ps3"); oarc("t19","p23");
oarc("t19","p28");
iarc("t20","p22");
oarc("t20","p24");
iarc("t21","p25"); oarc("t21","p26");
iarc("t22","p26"); oarc("t22","p27");
iarc("t23","p24"); iarc("t23","p26");
iarc("t23","ps4");
oarc("t23","p28"); oarc("t23","ps3");
oarc("t23","pd"); oarc("t23","p23");
iarc("t24","p24"); iarc("t24","p27");
iarc("t24","ps4");
oarc("t24","p28"); oarc("t24","ps3");
oarc("t24","pd"); oarc("t24","pC");
iarc("t25","p28");
oarc("t25","p29");
iarc("t26","p29");iarc("t26","pC");
iarc("t26","pm");
```

```
oarc("t26","p31"); oarc("t26","p32");
iarc("t27","p29"); iarc("t27","p23");
iarc("t27","ps5");
oarc("t27","ps4");oarc("t27","p30"); oarc("t27","p36");
iarc("t28","p31");
oarc("t28","p33");
iarc("t29","p32");
oarc("t29","p34");
iarc("t30","p34");
oarc("t30","p35");
iarc("t31","p33"); iarc("t31","ps5");
iarc("t31","p35");
oarc("t31","pD");
oarc("t31","ps4"); oarc("t31","p36");
oarc("t31","pm");
iarc("t32","p33"); iarc("t32","ps5");iarc("t32","p34");
oarc("t32","p30");
oarc("t32","ps4"); oarc("t32","p36");
oarc("t32","pm");
iarc("t33","p36");
oarc("t33","p37");
iarc("t34","p37");iarc("t34","pD");
oarc("t34","p39");
iarc("t35","p30"); iarc("t35","p37");
oarc("t35","p38");
iarc("t36","p39");
oarc("t36","p1"); oarc("t36","ps5");
```

```
          iarc("t37","p38");

          oarc("t37","p1"); oarc("t37","ps5");


     /* net is defined , analysis part */

}

/* the following 3 lines should ap1ear in all programs */

          assert() {return(RES_NOERR);}

ac_init() {}

ac_reach() {fprintf(stderr,"\nThe reachibility graph has been generated\n\n");}


/* user defined out functions */

reward_type ef0() {return(rate("t6"));} /* throughput */

reward_type ef1() {return(rate("t12"));} /* throughput */

reward_type ef2() {return(rate("t20"));} /* throughput */

reward_type ef3() {return(rate("t28"));} /* throughput */

reward_type ef15() {return(rate("t36"));} /* throughput */

reward_type ef16() {return(rate("t37"));} /* throughput */


reward_type ef5() {return(rate("t36"));} /* system production rate */

reward_type ef6() {return(1-mark("pR1"));}

reward_type ef7() {return(1-mark("ps1"));}

reward_type ef8() {return(1-mark("ps2"));}

reward_type ef9() {return(1-mark("ps3"));}

reward_type ef10() {return(1-mark("ps4"));}

reward_type ef11() {return(1-mark("ps5"));}

reward_type ef17() {return(1-mark("p38"));}

reward_type ef13() {return(1-mark("p39"));}
reward_type ef14()  {return(1.0-
(mark("ps1")+mark("ps2")+mark("ps3")+mark("ps4")+mark("ps5"))/5.0);}


ac_final() {pr_expected("throughput,'t6' ",ef0);
            pr_expected("throughput,'t12' ",ef1);
            pr_expected("throughput,'t20' ",ef2);
            pr_expected("throughput,'t28' ",ef3);
            pr_expected("throughput,'t36' ",ef15);
            pr_expected("throughput,'t37' ",ef16);
            pr_expected("System Production Rate = ",ef5);
            pr_expected("utilization 'pR1' ",ef6);
            pr_expected("utilization 'ps1' ",ef7);
            pr_expected("utilization 'ps2' ",ef8);
            pr_expected("utilization,'ps3' ",ef9);
            pr_expected("utilization,'ps4'  ",ef10);
            pr_expected("utilization ,'ps5' ",ef11);
            pr_expected("utilization,'p38' ",ef17);
            pr_expected("utilization,'p39'  ",ef13);
            pr_expected("Average utilization  ",ef14);
            pr_std_average();}
```

# APPENDIX C

## Output files of S4 in a Five Sequence Work Station System as Shown in Figure 3.1 and 3.4

".mc" file of the five sequence work station system

```
_firstindex = 0;

_nstates = 45;

_nentries = 57;

_order = _FROMTO;

_matrix =

  0     1:5.000000000000e+00;

  1     2:5.000000000000e+00;

  2     3:5.000000000000e+00;

  3     4:5.555555820465e-01     5:5.000000000000e+00;

  4     6:5.555555820465e-01     7:5.000000000000e+00;

  5     7:5.555555820465e-01;

  6     8:5.000000000000e+00;

  7     8:5.555555820465e-01     9:5.555555820465e-01;

  8    10:5.000000000000e+00;

  9    11:5.000000000000e+00;

 10    12:5.000000000000e+00;

 11    13:5.555555820465e-01;

 12    14:5.000000000000e+00;

 13    15:5.000000000000e+00;

 14    16:5.000000000000e+00    17:5.555555820465e-01;

 15    18:5.555555820465e-01;

 16    19:5.555555820465e-01;

 17    19:5.000000000000e+00    20:5.555555820465e-01;

 18    21:5.000000000000e+00;

 19    13:5.555555820465e-01    22:5.555555820465e-01;

 20    22:5.000000000000e+00;

 21    23:5.555555820465e-01;
```

```
22      24:5.000000000000e+00;
23      25:5.000000000000e+00;
24      26:5.000000000000e+00;
25      27:5.555555820465e-01;
26      28:5.000000000000e+00;
27       0:5.555555820465e-01;
28      29:5.000000000000e+00      30:5.555555820465e-01;
29      31:5.555555820465e-01;
30      31:5.000000000000e+00      32:5.555555820465e-01;
31      18:5.555555820465e-01      33:5.555555820465e-01;
32      33:5.000000000000e+00;
33      34:5.000000000000e+00;
34      35:5.000000000000e+00;
35      36:5.000000000000e+00;
36      37:1.000000014901e-01      38:5.555555820465e-01;
37      39:5.555555820465e-01;
38      39:1.000000014901e-01      40:5.555555820465e-01;
39      23:1.111111138016e-02      41:5.555555820465e-01;
40      41:1.000000014901e-01;
41      42:5.000000000000e+00;
42      43:5.000000000000e+00;
43      44:5.000000000000e+00;
44       0:5.000000000000e+00;
_method = _SSSOR;
_precision = 1e-06;
_iterations = 2000;
_solve = _ALL;
```

".log" file of the five sequence work station system example

```
The reachability graph contains:

     45 tangible markings

      0 vanishing markings

     57 arcs

After elimination of redundant arcs:

          # of remaining arcs:              57

After the elimination of vanishing markings:

          # of remaining arcs:              57

Solving the Markov chain...

ERROR/WARNING: switching from SOR to Gauss-Seidel

...Markov chain solved

Reading the reachability graph info ...

End of execution.
```

".prb" file of the five sequence work station system example

```
_firstindex = 0;

_nstates = 45;

_method = _SSSOR;

_precision = 5.971728161432e-07;

_iterations = 10;

_probabilities =

0:1.339571526201e-02      1:1.339571526201e-02      2:1.339571526201e-02

3:1.205614367832e-02      4:1.205614419571e-03      5:1.085052879309e-01

6:1.339571641177e-04      7:5.967790859484e-02      8:6.764836213062e-03

9:6.630879048944e-03      10:6.764836213062e-03     11:5.967790859484e-02

12:6.764836213062e-03     13:9.979470971818e-03     14:6.088352562724e-03

15:8.981523446364e-02     16:5.479517045168e-02     17:6.088352824008e-04

18:1.167050989431e-02     19:3.013734386600e-02     20:6.764836793693e-05

21:1.050345840403e-01     22:3.416242290500e-03     23:1.168004300357e-02

24:3.416242290500e-03     25:1.051203820196e-01     26:3.416242290500e-03

27:1.051203820196e-01     28:3.074618046789e-03     29:2.767156110161e-02

30:3.074618178737e-04     31:1.521935866526e-02     32:3.416242583718e-05

33:1.725202358168e-03     34:1.725202358168e-03     35:1.725202358168e-03

36:1.315832250914e-02     37:2.368497974000e-03     38:1.115112082622e-02

39:4.289901626404e-03     40:6.195067328768e-02     41:1.715669243212e-03

42:1.715669243212e-03     43:1.715669243212e-03     44:1.715669243212e-03;

_end = ;
```

".out" file of S4 in the five sequence work station system

```
INPUT: production rate of station1,station2,station3 and station5 = 5

INPUT: failure rate of station1,station2,station3 and station5 = 0.1

INPUT: production rate of station4 = 0.1

INPUT: failure rate of station4 = 0.1

EXPECTED: throughput,'t6'  = 0.06697857631

EXPECTED: throughput,'t12'  = 0.0338241810653

EXPECTED: throughput,'t20'  = 0.0170812114525

EXPECTED: throughput,'t28'  = 0.00862601179084

EXPECTED: throughput,'t36'  = 0.00857834621606

EXPECTED: throughput,'t37'  = 0.0584002150178

EXPECTED: System Production Rate =  = 0.00857834621606

EXPECTED: utilization 'pR1'  = 0.276558547445

EXPECTED: utilization 'ps1'  = 0.215135178525

EXPECTED: utilization 'ps2'  = 0.174952052891

EXPECTED: utilization,'ps3'  = 0.154659554432

EXPECTED: utilization,'ps4'   = 0.214789684118

EXPECTED: utilization ,'ps5'  = 0.227067814772

EXPECTED: utilization,'p38'  = 0.89487961798

EXPECTED: utilization,'p39'   = 0.998284330757

EXPECTED: Average utilization   = 0.197320856948

AVERAGE:

=============================================================

  PLACE       Pr[nonempty]        Av[tokens]

 0: p1        1.339571526201e-02  1.339571526201e-02

 1: p2        1.339571526201e-02  1.339571526201e-02

 2: p3        1.339571526201e-02  1.339571526201e-02
```

```
3: p4      8.116562519992e-01   8.116562519992e-01
4: p5      1.339571526201e-02   1.339571526201e-02
5: p6      1.205614316092e-01   1.205614316092e-01
6: p7      6.088352301441e-02   6.088352301441e-02
7: p8      6.898793377180e-03   6.898793377180e-03
8: p9      6.630878764378e-02   6.630878764378e-02
9: p10     1.749480327388e-01   1.749480327388e-01
10: p11     1.339571526201e-02   1.339571526201e-02
11: p12     6.644274480790e-02   6.644274480790e-02
12: p13     9.979470543545e-02   9.979470543545e-02
13: p14     6.764836213062e-03   6.764836213062e-03
14: p15     6.088352301441e-02   6.088352301441e-02
15: p16     3.074617914840e-02   3.074617914840e-02
16: p17     3.483890658436e-03   3.483890658436e-03
17: p18     9.048864071788e-01   9.048864071788e-01
18: p19     8.834875660818e-02   8.834875660818e-02
19: p20     1.339571326232e-02   1.339571326232e-02
20: p21     9.323147675414e-02   9.323147675414e-02
21: p22     3.416242290500e-03   3.416242290500e-03
22: p23     1.167050939346e-01   1.167050939346e-01
23: p24     4.461612212505e-02   4.461612212505e-02
24: p25     3.074617914840e-02   3.074617914840e-02
25: p26     1.552682048314e-02   1.552682048314e-02
26: p27     1.759364784006e-03   1.759364784006e-03
27: p28     1.339571225247e-02   1.339571225247e-02
28: p29     1.067597863985e-01   1.067597863985e-01
29: p30     1.168004250232e-01   1.168004250232e-01
```

```
30: p31        8.626011662304e-02   8.626011662304e-02
31: p32        1.552682048314e-02   1.552682048314e-02
32: p33        8.374068843615e-03   8.374068843615e-03
33: p34        1.544102245262e-02   1.544102245262e-02
34: p35        6.366634253089e-02   6.366634253089e-02
35: p36        1.339571224678e-02   1.339571224678e-02
36: p37        1.068360512628e-01   1.068360512628e-01
37: p38        1.051203820196e-01   1.051203820196e-01
38: p39        1.715669243212e-03   1.715669243212e-03
39: ps1        7.848648214752e-01   7.848648214752e-01
40: ps2        8.250479471088e-01   8.250479471088e-01
41: ps3        8.453404455680e-01   8.453404455680e-01
42: ps4        7.852103158824e-01   7.852103158824e-01
43: ps5        7.729321852276e-01   7.729321852276e-01
44: pm         9.053658145333e-01   9.053658145333e-01
45: pd         9.519676355845e-01   9.519676355845e-01
46: pR1        7.234414525551e-01   7.234414525551e-01
47: pA         1.352967242612e-02   1.352967242612e-02
48: pB         6.832484580999e-03   6.832484580999e-03
49: pC         3.450404716337e-03   3.450404716337e-03
50: pD         3.431338486424e-03   3.431338486424e-03
  TRANSITION     Pr[enabled]         Av[throughput]
 0: t1        1.339571526201e-02   6.697857631003e-02
 1: t2        1.339571526201e-02   6.697857631003e-02
 2: t3        1.339571526201e-02   6.697857631003e-02
 3: t4        1.205614316092e-01   6.697857631003e-02
 4: t5        6.088352301441e-02   3.382418106531e-02
```

| | | |
|---|---|---|
| 5: t6 | 1.339571526201e-02 | 6.697857631003e-02 |
| 6: t7 | 5.967790859484e-02 | 3.315439524472e-02 |
| 7: t8 | 6.764836213062e-03 | 3.382418106531e-02 |
| 8: t9 | 1.339571526201e-02 | 6.697857631003e-02 |
| 9: t10 | 5.967790859484e-02 | 3.315439524472e-02 |
| 10: t11 | 6.764836213062e-03 | 3.382418106531e-02 |
| 11: t12 | 6.764836213062e-03 | 3.382418106531e-02 |
| 12: t13 | 6.088352301441e-02 | 3.382418106531e-02 |
| 13: t14 | 3.074617914840e-02 | 1.708121145250e-02 |
| 14: t15 | 3.013734386600e-02 | 1.674296961281e-02 |
| 15: t16 | 3.416242290500e-03 | 1.708121145250e-02 |
| 16: t17 | 1.339571326232e-02 | 6.697856631159e-02 |
| 17: t18 | 3.416242290500e-03 | 1.708121145250e-02 |
| 18: t19 | 8.981523446364e-02 | 4.989735485909e-02 |
| 19: t20 | 3.416242290500e-03 | 1.708121145250e-02 |
| 20: t21 | 3.074617914840e-02 | 1.708121145250e-02 |
| 21: t22 | 1.552682048314e-02 | 8.626011790842e-03 |
| 22: t23 | 1.521935866526e-02 | 8.455199661656e-03 |
| 23: t24 | 1.725202358168e-03 | 8.626011790842e-03 |
| 24: t25 | 1.339571225247e-02 | 6.697856126237e-02 |
| 25: t26 | 1.725202358168e-03 | 8.626011790842e-03 |
| 26: t27 | 1.050345840403e-01 | 5.835254947153e-02 |
| 27: t28 | 8.626011662304e-02 | 8.626011790840e-03 |
| 28: t29 | 1.552682048314e-02 | 8.626011790842e-03 |
| 29: t30 | 1.544102245262e-02 | 8.578346216061e-03 |
| 30: t31 | 1.715669243212e-03 | 8.578346216061e-03 |
| 31: t32 | 4.289901626404e-03 | 4.766557478090e-05 |

```
32: t33         1.339571224678e-02   6.697856123391e-02

33: t34         1.715669243212e-03   8.578346216061e-03

34: t35         1.051203820196e-01   5.840021501784e-02

35: t36         1.715669243212e-03   8.578346216061e-03

36: t37         1.051203820196e-01   5.840021501784e-02

=============================================================
```

## APPENDIX D

## PLC Ladder Logic Program of S1 in the five Sequence Work Station System

```
Rung 2:0
| CYCLE       |CYCLE STOP|LCR ON S1 |                                    S1 READY   |
| START       |          |          |                                               |
|     I:1         I:1         I:1                                         B3        |
|----] [--------]/[--------] [-------------------------------------------( )-----|
|       0           1           2                                         1         |

B3/1
               -] [-   2:1 2:2
               -( )-   2:0

I:1/0
               -] [-   2:0

I:1/1
               -]/[-   2:0

I:1/2
               -] [-   2:0
               -]/[-   2:4


Rung 2:1
| PALLLET     |S1 READY  |                                           PALLET AT   |
| PRESENT     |          |                                           S1          |
| S1          |          |                                                       |
|     I:1          B3                                                 O:3        |
|----] [--------] [-------------------------------------------------( )-----|
|       3           1                                                 0          |

B3/1
               -] [-   2:1 2:2
               -( )-   2:0

I:1/3
               -] [-   2:1 2:19

O:3/0
               -] [-   2:5
               -( )-   2:1


Rung 2:2
| S1 READY    |                                                      GREEN       |
|             |                                                      LIGHT ON    |
|             |                                                      AT S1       |
|       B3                                                             O:3       |
|----] [-----------------------------------------------------------( )-----|
|       1                                                              1         |

B3/1
               -] [-   2:1 2:2
               -( )-   2:0

O:3/1

               -( )-   2:2
```

```
Rung 2:3
| OPERATION                                                     CYCLE FAIL          |
| FAILED AT                                                     ED COUNTER          |
| S1                                                            AT S1               |
|    O:3                                                      +CTU---------------+  |
|----] [------------------------------------------------------+COUNT UP          +-(CU)-
|       9                                                     |Counter       C5:0+-(DN)
|                                                             |Preset           3|  |
|                                                             |Accum            0|  |
|                                                             +------------------+  |

C5:0
             -CTU-   2:3 2:10
             -RES-   2:11 5:10

O:3/9
             -] [-   2:3 2:7 2:10
             -(L)-   2:14
             -(U)-   2:18 2:26


Rung 2:4
|   CYCLE FAIL                                                          RED LIGHT   |
|   ED 3 TIMES                                                          ON AT S1    |
|                                                                                   |
|       C5:0                                                              O:3       |
|-+----] [-----+---------------------------------------------------------( )-----|
| |      DN    |                                                            2       |
| | LCR ON S1  |                                                                    |
| |            |                                                                    |
| |     I:1    |                                                                    |
| +----]/[-----+                                                                    |
|         2                                                                         |

C5:0/DN
             -] [-   2:4

I:1/2
             -] [-   2:0
             -]/[-   2:4

O:3/2
             -( )-   2:4


Rung 2:5
| PALLET AT                                                     GOOD TAG            |
| S1                                                            TIMER               |
|                                                                                   |
|    O:3                                                      +TON---------------+  |
|----] [------------------------------------------------------+TIMER ON DELAY    +-(EN)-
|       0                                                     |Timer         T4:0--(DN
|                                                             |Time Base     0.01|  |
|                                                             |Preset          50|  |
|                                                             |Accum            0|  |
O:3/0
                -] [-   2:5
                -( )-   2:1

T4:0
                -TON-   2:5

Rung 2:6
| GOOD TAG   |                                                      GOOD        |
| TIMER DONE |                                                      PALLET TAG  |
|            |                                                                  |
|            |                                                                  |
|    T4:0                                                            O:3        |
|----] [------------------------------------------------------------( )-----|
|      DN                                                               3       |

O:3/3
                -] [-   2:12
                -( )-   2:6

T4:0/DN
                -] [-   2:6


Rung 2:7
|    OPERATION                                          BAD PALLET              |
|    FAILED AT                                          TAG TIMER               |
|    S1                                                                         |
|       O:3                                       +TON----------------           |
|-+----] [-----+---------------------------------+TIMER ON DELAY    +-(EN)-|
| |       9     |                                 |Timer         T4:1--(DN)     |
| |             |                                 |Time Base      1.0|          |
| |             |                                 |Preset          50|          |
| |             |                                 |Accum            0|          |
| |             |                                 +------------------           |
| |  OPERATOR   |                                                               |
| |  KEY        |                                                               |
| |     I:1     |                                                               |
| +----] [-----+                                                               |
|         4                                                                     |

I:1/4
                -] [-   2:7 2:11 2:26 3:12 4:21

O:3/9
                -] [-   2:3 2:7 2:10
                -(L)-   2:14
                -(U)-   2:18 2:26

T4:1
                -TON-   2:7


|                                                 +------------------+           |
Rung 2:8
|                                                                         BAD TAG PA |
|                                                                         LLET       |
| T4:1                                                                       O:3     |
|--] [-----------------------------------------------------------------------( )-----|
|    EN                                                                       4      |

O:3/4
              -] [-   3:27 3:28
              -( )-   2:8

T4:1/EN
              -] [-   2:8


Rung 2:9
|                                                                         BAD PALLET |
|                                                                         TAGGED     |
| T4:1                                                                       O:3     |
|--] [-----------------------------------------------------------------------(L)-----|
|    DN                                                                       5      |

O:3/5
              -] [-   2:15
              -(L)-   2:9
              -(U)-   2:18 2:26

T4:1/DN
              -] [-   2:9


Rung 2:10
| OPERATION                                                 CYCLE FAIL             |
| FAILED AT                                                 ED COUNTER             |
| S1                                                        AT S1                  |
|    O:3                                               +CTU---------------+        |
|----] [-----------------------------------------------+COUNT UP          +-(CU)-|
|       9                                              |Counter       C5:0+-(DN) |
|                                                      |Preset           3|       |
|                                                      |Accum            0|       |
|                                                      +------------------+       |

C5:0
              -CTU-   2:3 2:10
              -RES-   2:11 5:10

O:3/9
              -] [-   2:3 2:7 2:10
              -(L)-   2:14
              -(U)-   2:18 2:26
```

```
Rung 2:11
|    CYCLE DONE                                                   CYCLE FAIL |
|                                                                 ED COUNTER |
|                                                                 AT S1      |
|          B3                                                        C5:0    |
|-+----] [-----+--------------------------------------------------(RES)----|
| |        4     |                                                           |
| |  OPERATOR    |                                                           |
| |  KEY         |                                                           |
| |      I:1     |                                                           |
| +----] [-----+                                                             |
|          4                                                                 |

B3/4
                 -] [-   2:11 2:15
                 -(L)-   2:25
                 -(U)-   2:17

C5:0
                 -CTU-   2:3 2:10
                 -RES-   2:11 5:10

I:1/4
                 -] [-   2:7 2:11 2:26 3:12 4:21


Rung 2:12
| ROBOT       |GOOD                                       CYCLE            |
| READY       |PALLET TAG                                 TIMER            |
|             |                                                            |
|    I:1         O:3                          +TON---------------+         |
|----] [--------] [---------------------------+TIMER ON DELAY    +-(EN)-|
|      5           3                          |Timer        T4:2+-(DN)  |
|                                             |Time Base    0.01|        |
|                                             |Preset         15|        |
|                                             |Accum           0|        |
|                                             +------------------+        |

I:1/5
                 -] [-   2:12 2:19

O:3/3
                 -] [-   2:12
                 -( )-   2:6

T4:2
                 -TON-   2:12


Rung 2:13
|                                                           CYCLE TIME |
|                                                           OVER       |
| T4:2                                                          B3     |
|--] [---------------------------------------------------------(L)-----|
|     DN                                                         3      |


B3/3
                 -] [-   2:14
                 -(L)-   2:13
```

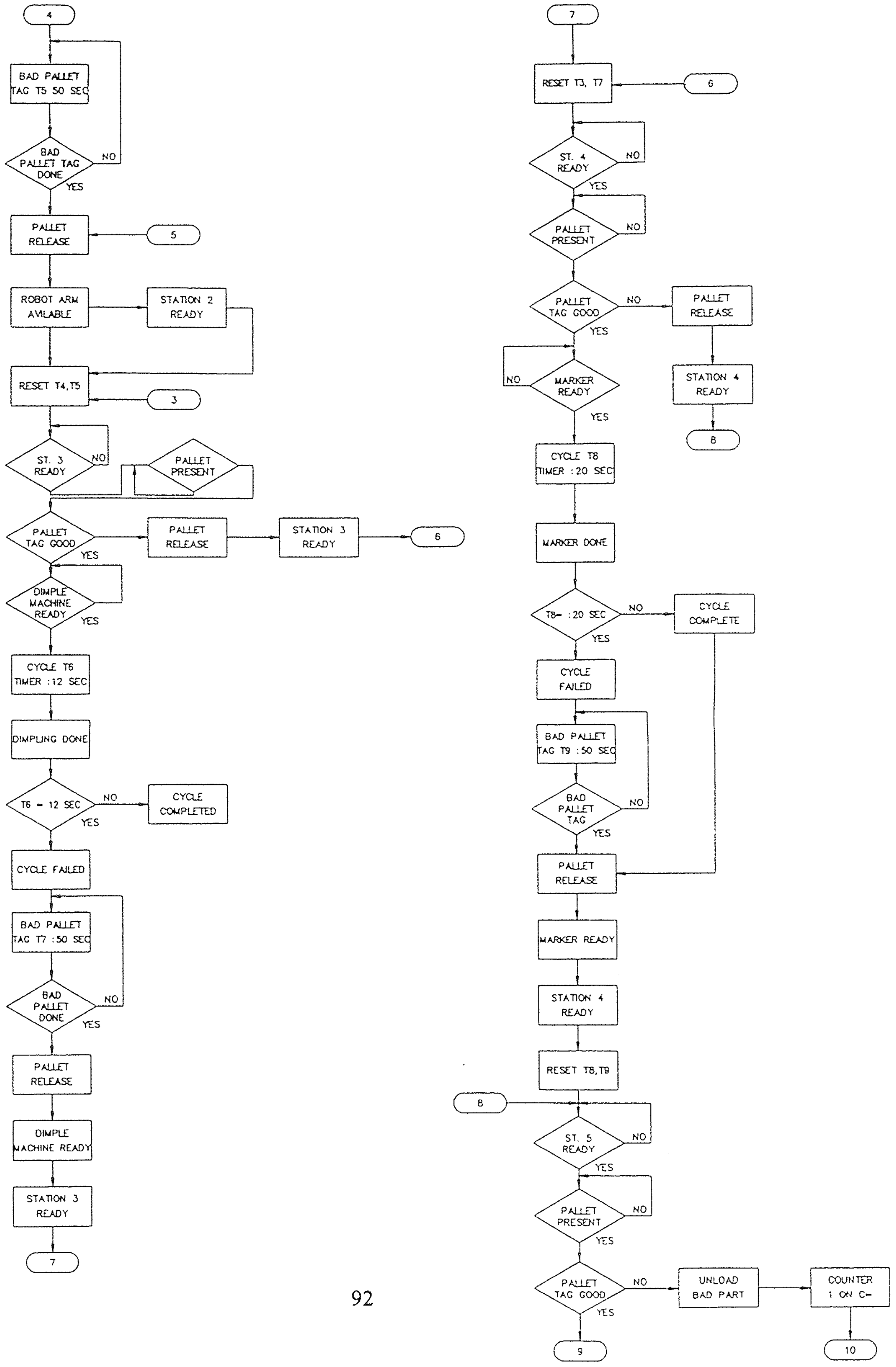
```
T4:2/DN
              -] [-   2:13

Rung 2:14
| CYCLE TIME|                                                        OPERATION   |
| OVER      |                                                        FAILED AT   |
|           |                                                        S1          |
|      B3                                                               O:3      |
|----] [------------------------------------------------------------- (L)-----|
|       3                                                                9     |

B3/3
              -] [-   2:14
              -(L)-   2:13

O:3/9
              -] [-   2:3 2:7 2:10
              -(L)-   2:14
              -(U)-   2:18 2:26


Rung 2:15
|    CYCLE DONE                                                      RELEASE     |
|                                                                    PALLET      |
|         B3                                                           O:3       |
|-+----] [-----+---------------------------------------------------------( )-----|
| |       4    |                                                         6       |
| | BAD PALLET |                                                                 |
| | TAGGED     |                                                                 |
| |     O:3    |                                                                 |
| +----] [-----+                                                                 |
|         5                                                                      |

B3/4
              -] [-   2:11 2:15
              -(L)-   2:25
              -(U)-   2:17

O:3/5
              -] [-   2:15
              -(L)-   2:9
              -(U)-   2:18 2:26

O:3/6
              -] [-   2:16
              -( )-   2:15


Rung 2:16
| RELEASE                                              PALLET REL                |
| PALLET                                               ESED TMR                  |
|     O:3                                            -TON---------------+        |
|----] [---------------------------------------------+TIMER ON DELAY    +-(EN)-|
|       6                                            |Timer        T4:12+-(DN)  |
|                                                    |Time Base     0.01|        |
|                                                                                |
|                                                    |Preset           3|        |
|                                                    |Accum            0|        |
|                                                    +------------------+        |
O:3/6
              -] [-   2:16
              -( )-   2:15

T4:12
              -TON-   2:16 5:11 5:16

Rung 2:17
|                                                              CYCLE DONE |
|                                                                          |
| T4:12                                                              B3   |
|--] [--------------------------------------------------------------(U)-----|
|    DN                                                               4   |

B3/4
              -] [-   2:11 2:15
              -(L)-   2:25
              -(U)-   2:17

T4:12/DN
              -] [-   2:17 5:12 5:18


Rung 2:18
| PALLET                                                  BAD PALLET    |
| RELEASED                                                TAGGED        |
|    I:1                                                    O:3         |
|----] [------------------------------------------------+----(U)------+-|
|      13                                               |     5       | |
|                                                       | OPERATION   | |
|                                                       | FAILED AT   | |
|                                                       | S1          | |
|                                                       |     O:3     | |
|                                                       +----(U)------+ |
|                                                             9          |

I:1/13
              -] [-   2:18

O:3/5
              -] [-   2:15
              -(L)-   2:9
              -(U)-   2:18 2:26

O:3/9
              -] [-   2:3 2:7 2:10
              -(L)-   2:14
              -(U)-   2:18 2:26


Rung 2:19
| PALLLET     |PART AVAIL|ROBOT      |                          EXTEND ARM |
| PRESENT     |ABLE AT FE|READY      |                                     |
| S1          |EDER      |                                                 |
|    I:1          I:1          I:1                               O:3       |

|----] [--------] [--------] [----------------------------------------(L)-----|
|      3           6           5                                      7     |
I:1/3
               -] [-   2:1 2:19

I:1/5
               -] [-   2:12 2:19

I:1/6
               -] [-   2:19 2:20

O:3/7
               -(L)-   2:19 3:18
               -(U)-   2:26 3:20 3:26

Rung 2:20
| ARM EXTEND|PART AVAIL|                                                 CLOSE GRIP |
| ED        |ABLE AT FE|                                                 PER        |
|           |EDER      |                                                            |
|    I:1        I:1                                                        O:3      |
|----] [--------] [------------------------------------------------------(L)-----|
|       7          6                                                          8     |

I:1/6
               -] [-   2:19 2:20

I:1/7
               -] [-   2:20 3:19

O:3/8
               -(L)-   2:20 3:19
               -(U)-   2:24 2:26 3:24 3:26


Rung 2:21
| ARM AT     |GRIPPER                                     ARM DOWN              |
| HOME       |CLOSED                                      TIMER                 |
|    I:1        I:1                                 +TON---------------+         |
|----] [--------] [---------------------------------+TIMER ON DELAY    +-(EN)-|
|       9          8                                |Timer         T4:3+-(DN)  |
|                                                   |Time Base     0.01|        |
|                                                   |Preset           5|        |
|                                                   |Accum            0|        |
|                                                   +------------------+        |

I:1/8
               -] [-   2:21 2:23 3:20 3:21 3:23

I:1/9
               -] [-   2:21 2:23 2:25 3:21 3:23 3:25 3:29

T4:3
               -TON-   2:21
```

```
Rung 2:22
|                                                                     ARM DOWN  |
|                                                                     OUTPUT    |
| T4:3                                                                  O:3     |
|--] [-------------------------------------------------------------------( )-----|
|    DN                                                                  10     |

O:3/10
             -( )-  2:22 3:22

T4:3/DN
             -] [-  2:22


Rung 2:23
| ARM AT    |ARM DOWN  |GRIPPER  |                                   PART LOADE |
| HOME      |          |CLOSED   |                                   D          |
|    I:1         I:1        I:1                                          B3     |
|----] [--------] [--------] [-------------------------------------------(L)-----|
|        9        10          8                                          2      |

B3/2
             -] [-  2:24 2:25
             -(L)-  2:23
             -(U)-  2:25 2:26

I:1/8
             -] [-  2:21 2:23 3:20 3:21 3:23

I:1/9
             -] [-  2:21 2:23 2:25 3:21 3:23 3:25 3:29

I:1/10
             -] [-  2:23 3:23


Rung 2:24
| PART LOADE|                                                         CLOSE GRIP |
| D         |                                                         PER        |
|      B3                                                                 O:3    |
|----] [------------------------------------------------------------------(U)-----|
|       2                                                                  8     |

B3/2
             -] [-  2:24 2:25
             -(L)-  2:23
             -(U)-  2:25 2:26

O:3/8
             -(L)-  2:20 3:19
             -(U)-  2:24 2:26 3:24 3:26
```

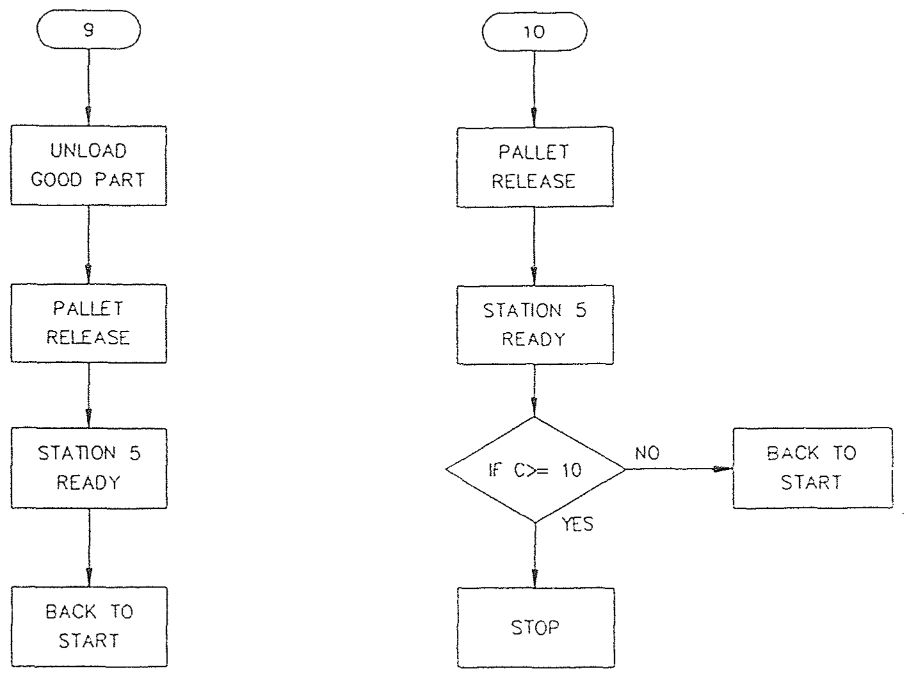
```
Rung 2:25
| ARM AT      |ARM UP      |GRIPPER     |PART LOADE                              CYCLE DONE   |
| HOME        |            |OPEN        |D                                                    |
|    I:1          I:1          I:1          B3                                     B3         |
|----] [--------] [--------] [--------] [------------------------------+----(L)-----+-
|      9           12           11           2                          |      4     |      |
|                                                                       | OPERATION  |      |
|                                                                       | DONE AT S1 |      |
|                                                                       |            |      |
|                                                                       |     B3     |      |
|                                                                       +----(L)-----+      |
|                                                                       |      5     |      |
|                                                                       | PART LOADE |      |
|                                                                       | D          |      |
|                                                                       |     B3     |      |
|                                                                       +----(U)-----+      |
|                                                                              2             |

B3/2
               -] [-   2:24 2:25
               -(L)-   2:23
               -(U)-   2:25 2:26

B3/4
               -] [-   2:11 2:15
               -(L)-   2:25
               -(U)-   2:17

B3/5
               -] [-   3:28
               -]/[-   3:27
               -(L)-   2:25
               -(U)-   3:13

I:1/9
               -] [-   2:21 2:23 2:25 3:21 3:23 3:25 3:29

I:1/11
               -] [-   2:25 2:26 3:25 3:26 3:29

I:1/12
               -] [-   2:25 3:25 3:29


Rung 2:26
| OPERATOR                          CLOSE GRIP                                                   |
| KEY                               PER                                                          |
|    I:1                               O:3                                                       |
|----] [---------+-+----------+-+----(U)-----+----------+----------+----------+-
|      4         | |          | |      8     |          |          |          |   |
|                | |          | | OPERATION  |          |          |          |   |
|                | |          | | FAILED AT  |          |          |          |   |
|                | |          | | S1         |          |          |          |   |
|                | |          | |    O:3     |          |          |          |   |
|                | |          | +----(U)-----+          |          |          |   |


|                | |          |        9                |          |          | |
                 | |          |                         |          |          |
               +++++         +++                       +++        +++        +++
                                                        October 26, 1994   Page 11
Program Listing            Processor File: PROJ.ACH                     Rung 2:26

               +++++        +++                        +++        +++         +++
|               | |          |                          |           |           | |
|               | |          |              BAD PALLET  |           |           | |
|               | |          |              TAGGED      |           |           | |
|               | |          |                 O:3      |           |           | |
|               | |          +----------------(U)------+           |           | |
|               | |                                5               |           | |
|               | |                                     PART LOADE |           | |
|               | |                                     D          |           | |
|               | |                                          B3    |           | |
|               | +----------------------------------------(U)-----+           | |
|               |                                             2                | |
|               | GRIPPER                                          EXTEND ARM  | |
|               | OPEN                                                         | |
|               |    I:1                                              O:3      | |
|               +----] [-----------------------------------------------(U)-----+ |
|                      11                                                 7      |

B3/2
             -] [-  2:24 2:25
             -(L)-  2:23
             -(U)-  2:25 2:26

I:1/4
             -] [-  2:7 2:11 2:26 3:12 4:21

I:1/11
             -] [-  2:25 2:26 3:25 3:26 3:29

O:3/5
             -] [-  2:15
             -(L)-  2:9
             -(U)-  2:18 2:26

O:3/7
             -(L)-  2:19 3:18
             -(U)-  2:26 3:20 3:26

O:3/8
             -(L)-  2:20 3:19
             -(U)-  2:24 2:26 3:24 3:26

O:3/9
             -] [-  2:3 2:7 2:10
             -(L)-  2:14
             -(U)-  2:18 2:26

Rung 2:27
|                                                                                |
|-------------------------------------+END+--------------------------------------|
|                                                                                |
```

## APPENDIX E

## Flow Chart of Five Sequence Work Station System for Ladder Logic Program



4
BAD PALLET TAG T5 50 SEC
BAD PALLET TAG DONE
NO
YES
PALLET RELEASE
5
ROBOT ARM AVILABLE
STATION 2 READY
RESET T4,T5
3
ST. 3 READY
NO
PALLET PRESENT
PALLET TAG GOOD
PALLET RELEASE
STATION 3 READY
6
YES
DIMPLE MACHINE READY
YES
CYCLE T6 TIMER :12 SEC
DIMPLING DONE
T6 = 12 SEC
NO
CYCLE COMPLETED
YES
CYCLE FAILED
BAD PALLET TAG T7 :50 SEC
BAD PALLET DONE
NO
YES
PALLET RELEASE
DIMPLE MACHINE READY
STATION 3 READY
7
7
RESET T3, T7
6
ST. 4 READY
NO
YES
PALLET PRESENT
NO
PALLET TAG GOOD
NO
PALLET RELEASE
YES
STATION 4 READY
8
MARKER READY
NO
YES
CYCLE T8 TIMER :20 SEC
MARKER DONE
T8= :20 SEC
NO
CYCLE COMPLETE
YES
CYCLE FAILED
BAD PALLET TAG T9 :50 SEC
BAD PALLET TAG
NO
YES
PALLET RELEASE
MARKER READY
STATION 4 READY
RESET T8,T9
8
ST. 5 READY
NO
YES
PALLET PRESENT
NO
YES
PALLET TAG GOOD
NO
UNLOAD BAD PART
COUNTER 1 ON C=
YES
9
10



FLOW CHART OF FIVE SEQUENCE WORK STATION
FOR LADDER LOGIC PROGRAM

## REFERENCES

1. Allen-Bradly's Advance Programming Software, Catalog No. 1747-PA2E, Series F. P/N# 40846-915-01, Revision 4.02.

2. R. Y. Al-Jaar and A. A. Desrochers, "Performance Evaluation of Automated Manufacturing System using Generalized stochastic Petri Nets." *IEEE Transactions on Robotics and Automation*, vol. 6, no. 6, pp. 621-639, 1990.

3. J. Billington, G. R. Wheerler, and M. C. Wilbur-Ham, " PROTEAN: A high level Petri net tool for the specification and verification of communicat protocols." *IEEE Transactions on Software Engineering*, vol. 14, no. 3 , pp. 301-31, 1988.

4. K. Brand and J. Kopainsky, "Principles and engineering of process control with Petri nets," *IEEE Transactions on Automatic Control*, vol. 33, no. 2, pp. 138-149, 1988.

5. L. A. Bryan and E. A. Bryan, *Programmable Controllers Theory and Implementatio*n. Industrial Text Co., 1988.

6. J. Campos, J. M. Colom, and M. Silva, "Performance evaluation of repetitive automated manufacturing systems." *Rensselaer's 2nd International Conference on Computer Integrated Manufacturing*, Troy, New york, pp. 74-81, 1990.

7. J. Campos, G. Chiola, J. M. Colom, and M. Silva, "Properties and Performance Bounds for Timed Marked Graphs." *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, vol. 39, no. 5, pp. 386-401, 1992.

8. G. Chiola, " A graphic Petri Net Tool for Performance analysis." *Proc. of Int. Workshop on Modeling Techniques and Performance Evaluation*, France, pp. 323-333, 1987.

9. G. Ciardo, *Manual for the SPNP Package*. Duke University, February 1989.

10. E. G. Coffman, M. J. Elphick, and A. Shoshani, "System Deadlocks." *Computing Surveys*, vol.3, pp. 67-78, 1971.

11. A. Desrochers, "Modeling and Control Using Petri Nets." *Modeling and Control of Manufacturing Systems*, IEEE Computer Society Press, pp. 239-251, 1990.

12. P. Freedman, "Time, Petri Nets, and Robotics." *IEEE Transactions on Robotics and Automation*, vol.7, no.4, pp. 417-433, 1991.

13. B. H. Krogh and C. L. Beck, "Synthesis of place/transition nets for simulation and control of manufacturing Systems," *in Reprints of 4th IFAC/IFORS Symp. Large Scale Systems*, Zurich, pp. 661-666, 1986.

14. A. M. Marsan, G. Balbo, and G. Conte, "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems." *ACM Transactions on Computer Systems*, vol. 2, no. 2, pp. 93-122, 1984.

15. M. K. Molly, "Performance Analysis using Stochastic Petri Nets." *IEEE Transactions on Computer*, vol. 3, no. 9, pp. 913-917, 1982.

16. T. Mutra, N. Komoda, K. Matsumoto, and K. Haruna, "A Petri Net Based Controller for Flexible and Maintainable Sequence Control and its Applications in Factory Automation." *IEEE Transactions on Industrial Electronics*, vol. IE33, no. 1, pp. 1-8, February, 1986.

17. T. Mutra and N. Komoda, "Development of a Petri nets based FA Controller and its applications." *Proc. 10th International Conference on Application and Theory of Petri Nets*, Bonn, Germany, pp. 394-402, 1989.

18. T. Mutra, "Petri Nets: Properties, Analysis and Applications." *IEEE*, vol. 17, no. 4, pp. 541-580, April, 1989.

19. H. Plunnecke and W. reisig, "Bibliography of Petri Nets 1990." *Advances on Petri Nets, G. Rozenberg (Ed.), Lecture Notes in Computer Science*, vol. 524, pp. 317-572, 1991.

20. W. Reising, "Petri nets in software engineering", *in Advances in Petri Nets 1986, part* II, *Lecture Notes in Computer Science*, vol. 255, W. Brauer, W. Reisig, G. Rozenberg (Eds.), Springer Verlag. , pp. 63-98, 1987.

21. M. Silva, "Las redes de Petri en la Automatica y la Informatica." *Editorial AC*, Madrid, 1985.

22. M. Silva and R. Valette, "Petri Nets and Flexible Manufacturing." *E. T. S. Ingenieros Industriales, Technical Report #E50015*, Zaragoza, Spain, pp. 1-43, January 1990.

23. M. C. Zhou and M. C. Leu, " Modeling and Performance Analysis of a Flexible PCB assembly Station using Petri Nets." *Transactions of the ASME, Journal of Electronic Packinging*, vol. 113, no. 4, pp. 410-416, 1991.

24. M. C. Zhou, F. DiCesare, and D. L. Rudolph, "Design and Implementation of a Petri Net Based Supervisory for Flexible Manufacturing System." *Automatica*, vol. 28, no. 6, pp.1199-1208, November, 1992.

25. M. C. Zhou and F. DiCeasare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publishers, Boston, MA, pp. 120, 1993.

26. Zurawski, and M. C. Zhou. "Petri Nets and Industrial Applications: A Survey." *IEEE Transactions Industrial Electronics*, vol. 41, no. 6, pp. 567-583, December 1994.