

SIMULATION MODELING AND ANALYSIS OF PRINTED CIRCUIT BOARD
ASSEMBLY LINES

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Pradip Dinkarrao Jadhav

Certificate of Approval

John L. Evans
Associate Professor
Industrial and Systems Engineering

Jeffrey S. Smith, Chair
Professor
Industrial and Systems Engineering

Jorge F. Valenzuela
Associate Professor
Industrial and Systems Engineering

Stephen L. McFarland
Dean
Graduate School

SIMULATION MODELING AND ANALYSIS OF PRINTED CIRCUIT BOARD
ASSEMBLY LINES

Pradip Dinkarrao Jadhav

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
December 16, 2005

SIMULATION MODELING AND ANALYSIS OF PRINTED CIRCUIT BOARD
ASSEMBLY LINES

Pradip Dinkarrao Jadhav

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

THESIS ABSTRACT
SIMULATION MODELING AND ANALYSIS OF PRINTED CIRCUIT BOARD
ASSEMBLY LINES

Pradip Dinkarrao Jadhav

Master of Science, December 16, 2005

(B.S.M.E., D.Y. Patil College of Engineering, Shivaji University, 2000)

125 Typed Pages

Directed by Dr. Jeffrey S. Smith

This thesis focuses on simulation modeling and analysis of printed circuit board assembly lines. The objectives of the study include assessing the feasibility of process flow logic and relative impact of changing line configurations. It is aimed to identify constraints or bottlenecks and development of improvement strategies accordingly. Additionally, complex interactions between various resources are examined to identify effective operator allocation and allocate buffer space. A six-step methodology is developed for this purpose. PCB assembly template is used for simulation modeling in Arena 7.01. We introduce a unique method of integrating the resource state graph into simulation modeling for PCB assembly lines and analyzing it to identify constraints,

bottlenecks, and potential for improvement in terms of throughput and operator allocation.

ACKNOWLEDGEMENTS

The author would like to thank Dr. Jeffrey S. Smith for his valuable and timely guidance on this thesis and Dr. John L. Evans and Dr. Jorge F. Valenzuela for their kind consent to serve on the thesis committee. Thanks are also due to all family members and friends for their support and to colleagues Skylab, McKenzie, and Charles for their valuable help and assistance at various stages of this work.

Style manual or journal used Computers and Industrial Engineering.

Computer software used Microsoft Word 2003.

TABLE OF CONTENTS

LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER 1 INTRODUCTION	1
1.1 Printed Circuit Board Manufacturing	2
1.2 Simulation Modeling and Analysis	5
1.3 Overview of Thesis	6
CHAPTER 2 LITERATURE REVIEW	8
2.1 Models of Manufacturing System	8
2.2 Analytical Models of Serial Production Lines	10
2.3 Simulation in Serial Production Lines	16
2.3.1 General Simulation Studies	16
2.3.2 Machine Interference and Operator Allocation	19
2.3.3 Buffer Allocation	20
2.3.4 Simulation Environment	21
2.3.5 Bottleneck Detection and Throughput Improvement	22
2.4 Conclusions	26
CHAPTER 3 PROBLEM DEFINITION	28
CHAPTER 4 METHODOLOGY	36
4.1 Collect Input Data	36

4.2	Develop Static Model of Process Line	. . .	38
4.3	Develop and Validate Simulation Model	. . .	43
4.4	Run Simulation Model with Current Configuration	. . .	44
4.5	Develop and Analyze Resource State Graph	. . .	45
4.6	Modify Configuration and Repeat	. . .	53
CHAPTER 5	CASE STUDIES	54
5.1	Simulation Modeling and Analysis of Product X Line	. . .	54
5.2	Simulation Modeling and Analysis of Product Y Line	. . .	64
5.3	Comparison of Bottleneck Detection Methods	. . .	66
5.4	Simulation Modeling and Analysis of Product Z Line	. . .	69
CHAPTER 6	CONCLUSIONS AND FUTURE WORK	74
REFERENCES	78
APPENDICES	84
APPENDIX I	PROCESS FLOWCHART AND RESOURCE STATE		
	GRPAHS FOR PRODUCT X LINE	85
APPENDIX II	PROCESS FLOWCHART AND RESOURCE STATE		
	GRPAHS FOR PRODUCT X LINE	97
APPENDIX III	PROCESS FLOWCHART AND RESOURCE STATE		
	GRPAHS FOR PRODUCT X LINE	103

LIST OF TABLES

4.1	Example Input Datasheet	39
4.2	Example Static Model – Details	40
4.3	Example Static Model – Input/Output	41
5.1	Static model Results for Product X Line	55
5.2	Comparison of Static and Simulation Model Results for Product X Line	55
5.3	Relative Impact of Adding Capacity at Final Tester, Install Cover, and Conformal Coat 2 for Product X Line	60
5.4	Percentage Time Spent in State Other Than Idle and Blocked for Product X Line Scenario 5	62
5.5	Comparison of Static and Simulation Model Results for Product Y Line	64
5.6	Average Active Durations for Product Y Line Current Configuration	67
5.7	Effective Average Active Durations for Product Y Line Current Configuration	68
5.8	Effective Average Active Durations for Product Y Line Scenario 1	69
5.9	Comparison of Static and Simulation Model Results for Product Z Line	70

LIST OF FIGURES

1.1	Type 1 SMT Manufacturing	3
1.2	Basic Structure of Serial Production Line	4
3.1	Typical Machine on PCB Assembly Line	29
3.2	Failure Process of a Typical Machine on PCB Assembly Line	30
3.3	Component Part Exhaustion at a Typical Machine on PCB Assembly Line	30
3.4	Input Dialog for Process Plus Module	31
3.5	Input Dialog for Multi-stage Process Module	32
3.6	Resource States - Busy and Idle	33
3.7	Resource States - Blocked	33
3.8	Resource States – Replenish and Failure	34
3.9	Resource States – Repair and Exhaust	35
4.1	Methodology for Simulation Modeling and Analysis for PCB Assembly Lines	36
4.2	Example Resource State Graph 1	47
4.3	Example Resource State Graph 2	49
4.4	Example Resource State Graph 3	52
5.1	Final Testers for Product X Line – Two Boards Available for Processing Simultaneously	59

5.2	Final Testers for Product X Line – Only One Board Available for Processing	59
5.3	Final Testers for Product X Line – Final Tester 1 Busy, Final Tester 2 Idle	59
I.1	Product X Line – Process Flow Chart	86
I.2	Product X Line – Resource State Graph for Current Configuration	87
I.3	Product X Line – Resource State Graph for Alternative 1	88
I.4	Product X Line – Resource State Graph for Alternative 2	89
I.5	Product X Line – Resource State Graph for Current Configuration 1	90
I.6	Product X Line – Resource State Graph for Scenario 1	91
I.7	Product X Line – Resource State Graph for Scenario 2	92
I.8	Product X Line – Resource State Graph for Scenario 3	93
I.9	Product X Line – Resource State Graph for Scenario 4	94
I.10	Product X Line – Resource State Graph for Scenario 5	95
I.11	Product X Line – Resource State Graph for Scenario 6	96
II.1	Product Y Line – Process Flowchart	98
II.2	Product Y Line – Process Flowchart	99
II.3	Product Y Line – Resource State Graph for Current Configuration	100
II.4	Product Y Line – Resource State Graph for Scenario 1	101
II.5	Product Y Line – Resource State Graph for Scenario 2	102
III.1	Product Z Line – Process Flowchart	104
III.2	Product Z Line – Resource State Graph for Current Configuration	105
III.3	Product Z Line – Resource State Graph for Scenario 1	106

III.4	Product Z Line – Resource State Graph for Scenario 2	.	.	.	107
III.5	Product Z Line - Resource State Graph for Scenario 3	.	.	.	108
III.6	Product Z Line - Resource State Graph for Scenario 4	.	.	.	109
III.7	Product Z Line - Resource State Graph for Scenario 5	.	.	.	110
III.8	Product Z Line - Resource State Graph for Scenario 6	.	.	.	111
III.9	Product Z Line - Resource State Graph for Scenario 7	.	.	.	112

CHAPTER 1

INTRODUCTION

Printed circuit board (PCB) assembly lines fall under the general category of serial production lines. These lines are characterized by asynchronous part transfer, capacitated buffers, unreliable workstations, imperfect production, multiple part types, and shared operators. Common problems faced by the managers of these lines include designing configuration of existing or future lines to meet target production rates, assessing the impact of changing line configurations, identifying bottlenecks or constraints of the current system, deciding where to add capacity or allocate buffers, and operator task allocation.

Analytical models for such systems require simplifying assumptions, which might not represent the real line behavior accurately. Simulation modeling can capture the complex behavior and interaction between various components of PCB assembly lines. This thesis aims at modeling the PCB assembly lines using PCB assembly template, capturing the complex interactions between its components. A six-step methodology for simulation modeling and analysis of PCB assembly lines and its implementation in simulation package Arena is discussed in this thesis. The methodology is designed to address and analyze the abovementioned common problems faced by managers and help them make better decisions. This methodology has been implemented for modeling and

analysis of real electronics assembly lines for a local PCB manufacturer. The specifics of this thesis pertain to PCB assembly lines, but, the methodology is equally applicable to the general class of serial production lines with proper adaptation of the template.

1.1 Printed Circuit Board Manufacturing

The following discussion about PCB manufacturing is taken from (Hollomon, 1989) and (Capillo, 1990). Printed circuit board assembly can be classified into two categories based on the type of components placed on the board: a) *Surface Mount Technology* (SMT), and b) *Insertion Mount Technology* (IMT). SMT has certain advantages over IMT, which is why it is more preferred method. SMT offers higher packaging densities, higher performance speeds, more and faster automated manufacturing, and reduced storage as well as manufacturing space. Thus, SMT lowers the overall costs. But, as some components may not be available in surface mount packages, some PCB assemblies may employ a mix of SMT and IMT.

Figure 1.1 shows general steps involved in SMT process for PCB assembly. Six steps involved in SMT manufacturing are explained as follows: a) **Attachment Media Dispensing:** Attachment media is the substance which holds the components in position after placement and prior to soldering. It is applied using screen printing, stenciling or dispensing, b) **Component Placement:** This could be manual, semi-automated or automated, c) **Curing:** Solder-paste curing is done by prolonged bake in a low temperature oven whereas adhesive curing involves baking, ultraviolet-light exposure, or combination of both, d) **Soldering:** This may be either reflow soldering or wave

soldering, e) Cleaning: The boards are cleaned to remove the harmful contaminants, and
 f) Testing: Finally, the boards go through testing for circuitry, assembly integrity, solder joint quality, and functionality. Depending upon the production volume, curing, soldering, and cleaning stages can be carried out using either a batch type or an inline machine, which has a conveyor inside it on which the boards travel.

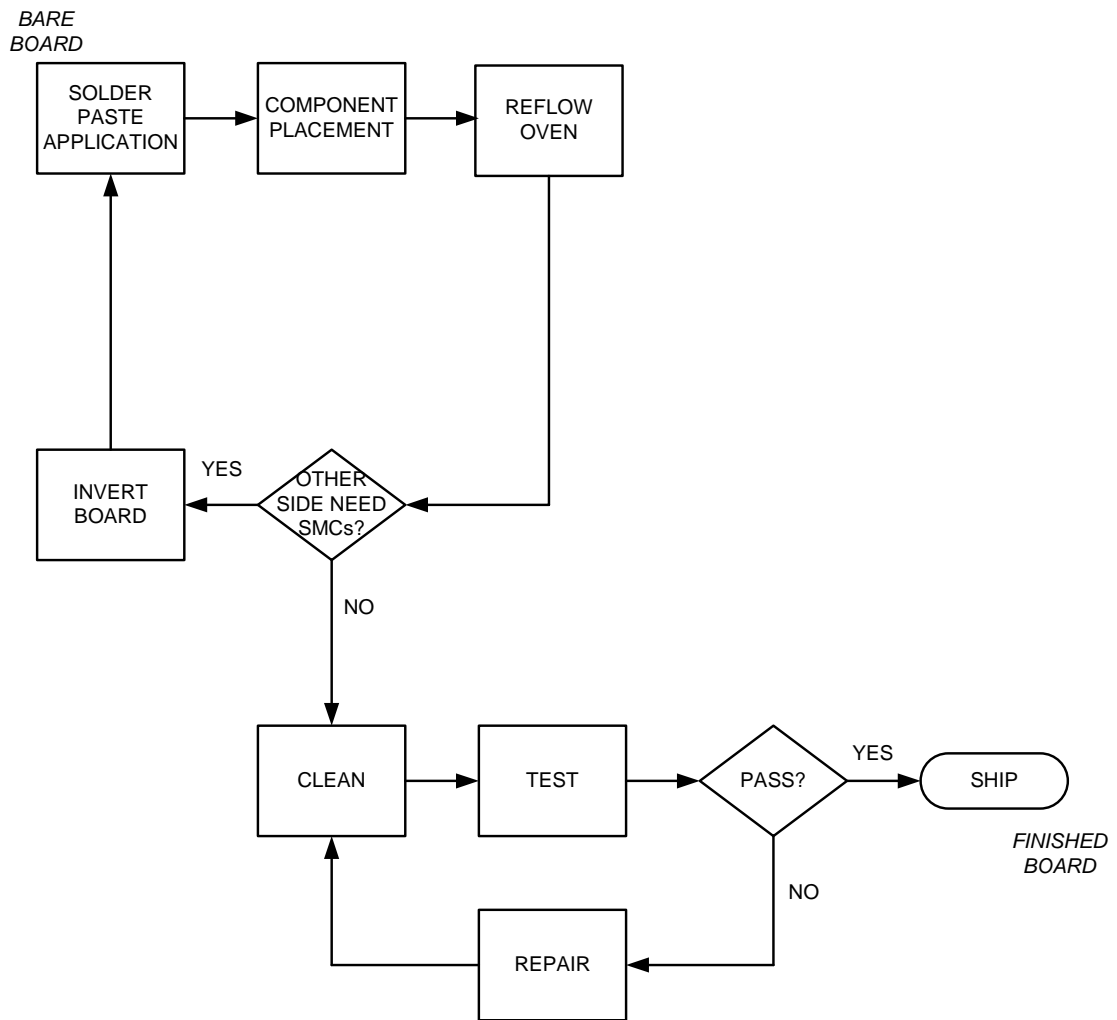


Figure 1.1: Type 1 SMT Manufacturing, (Hollomon, 1989).

Depending upon whether only SMCs or combination of SMCs and IMCs are populated onto the board, the SMT process is categorized into Type 1, Type 2, and Type 3 SMT manufacturing. Type 1 SMT is an exclusive surface-mount method with components mounted either on one or both sides of the board. Type 2 SMT Manufacturing involves a mixture of SMCs and IMCs on at least one side of the board. In Type 3 SMT one side is populated exclusively with IMCs and the other side is exclusively populated with SMCs. For a detailed discussion on SMT manufacturing reader is directed to Hollomon (1989) and Capillo (1990).

Based on volume of production, manufacturing systems are classified as being mass, batch or job shop. Mass manufacturing, also known as product flow layout, is characterized by high volume of production and low product flexibility. The product flow layout can be further classified based on the method of part transfer into: synchronous transfer, asynchronous transfer, and continuous transfer. The systems with synchronous part transfer are known as production lines. The PCB assembly line under consideration fit into the category of serial production lines. The general structure of such serial production lines is shown in Figure 1.2. It consists of K stations arranged in series, each station having buffer preceding it. The buffer before first station may be finite or infinite,

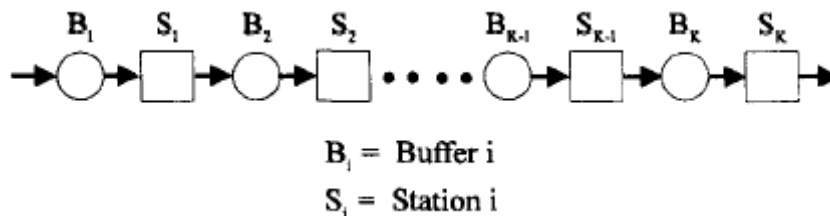


Figure 1.2: Basic Structure of Serial Production Line

but, all the inter-stage buffers are finite. Parts enter this system at station 1, pass through all stations where they undergo some operation, and finally leave the system after k^{th} station. Blocking and starving caused by the capacitated buffers is the main cause of production loss in serial production lines.

1.2. Simulation Modeling and Analysis

Law & Kelton (2000) discuss systems and the ways to model the systems. It is rarely feasible to experiment with the actual system in order to assess the impact of altering it, as in some cases, system under consideration might not even exist. So, one has to experiment with the model of the system to gain understanding of how corresponding system behaves. For a scientific study of a system, we have to make certain assumptions about how the system works. These assumptions take the form of mathematical or logical relationship, and these relationships constitute a model. As mentioned in Kamath (1994), analytical models require minimal data and offer quick results, but, several assumptions have to be made to obtain analytically tractable model. These assumptions usually affect the accuracy of the results. In the real world, most of the times, the relationships that compose the mathematical model are highly complex, precluding the possibility of analytical solution. Such models can be studied by means of simulation. Simulation models can handle such complexity of real-world systems and can be made to mimic the real systems as accurately as possible. The simulation models can be classified as either discrete-event or continuous. In rest of this document, simulation model refers to discrete-event simulation, unless specified otherwise.

The steps involved in a simulation study can be grouped into: a) *Input analysis*: involves collection and analysis of data, and definition as well as validation of conceptual model, b) *Model development*: involves simulation model development followed by verification and validation of the simulation model, and c) *Output analysis*: where performance metrics of the system are determined and analyzed.

Some of the advantages of simulation are: a) for highly complex, real-world systems with stochastic elements that cannot be evaluated analytically, simulation often proves to be the only type of investigation possible, b) simulation enables estimation of performance metrics of an existing or non-existing system under certain set of operating conditions, c) alternative proposed system configurations can be compared using simulation, d) simulation offers better control over experimental conditions as compared to experimenting with the actual system itself, and e) animation provided by simulation enables better understanding of the system and offers an effective way to present results to upper management.

1.3 Overview of Thesis

This thesis focuses on the procedure for simulation modeling and analysis of PCB assembly lines. The PCB assembly template designed by Mukkamala, Smith & Valenzuela (2003) and Mukkamala (2003) was used for the modeling in Arena 7.01. This template was enhanced as a part of this study. It reduces modeling efforts and offers a unique simulation output category. The objectives behind the analysis of PCB assembly lines are to: a) assess the feasibility of the process flow logic and relative impact of

alternative line configurations, b) assess the ability to meet planned production rates/quantities, c) identify bottleneck operation(s) and evaluate improvement strategies, d) examine complex interaction between resources, e) identify optimal operator assignment, and f) allocate buffer space.

A six-step methodology is proposed for this purpose, which offers a unique way to develop and analyze the simulation models. These steps are: a) collect input data, b) develop a static model of process line, c) develop and validate a discrete-event simulation model using PCB assembly template, d) run the simulation model with current configuration, e) analyze the *resource state graph* for current configuration, and f) modify configuration and repeat.

In the next chapter we discuss literature pertaining to analytical and simulation studies for the general class of serial production lines. Chapter 3 discusses the PCB assembly template, the statistics provided by template modules, and the problem dealt with in this thesis. In Chapter 4, the methodology developed for simulation modeling and analysis of PCB assembly lines is presented. Chapter 5 presents three case studies which demonstrate implementation of the proposed methodology. Finally, conclusions of this thesis and directions for future work are presented in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

This chapter discusses models of a manufacturing system and their classification, followed by discussion of analytical as well as simulation models developed by various authors for different issues associated with flow lines in general and serial production lines in specific.

2.1 Models of Manufacturing Systems

Manufacturing systems models can be categorized based on the purpose as being either generative or evaluative. Generative models give optimal solution to satisfy user's objective function. Evaluative models, also known as descriptive models, on the other hand do not provide an optimal solution, but, evaluate a given set of decisions by providing user with performance measures (Papadopoulos & Heavy, 1996). Unlike generative models, evaluative models provide insight into the behavior of the system and involve the decision maker in decision-loop (Kamath, 1994). The evaluative models are associated with performance evaluation where the task is to determine how well a system is performing based on measures such as throughput, utilization, and flow times.

Another categorization of models for production lines is Bernoulli lines and Markovian lines (D'Souza, 2004). In Bernoulli lines, when a workstation is not blocked

or starved, during each cycle, it will produce a part with fixed probability and fail to do so with complementary probability. Here, the duration of downtime of a workstation is generally short and comparable with the cycle time. On the other hand, Markovian lines involve workstations which change their state from up to down or visa-versa, during an infinitesimal time interval with a fixed probability rate. The downtimes of the workstations are generally much longer than the cycle time in this case. This thesis is focused on Markovian Lines.

The models can also be classified based on the type of manufacturing systems. The manufacturing systems can be classified mainly into continuous or discrete part flow systems. We are concerned with the discrete part manufacturing only. The discrete part manufacturing systems can be further classified based on volume of production into mass, batch, and job shop. Mass manufacturing is characterized by high production volumes. It is also called product-flow layout and can be further categorized according to method of part transfer. The systems with synchronous part transfer are known as transfer lines. Systems with asynchronous part transfer are called as production lines or power-and-free systems. These lines are characterized by high volume turnover and low product flexibility. The PCB assembly lines under consideration can be categorized as serial production lines.

Additionally, there are some other properties of the production lines that need to be addressed. Some systems are built with machines in parallel, either to increase production or to achieve greater reliability. In some of the systems two or more buffers can feed a single machine. A machine might take a part from each of the upstream buffer to assemble a part out of them. Finally, some systems require parts to be fitted onto

pallets. As the number of pallets is limited, the parts sometimes have to wait to get processed. The models of serial production lines can further be classified based on whether they can incorporate these kinds of subsystems.

Based on the solution methodology, the models can either be analytical models or simulation models. Analytical models provide formulae or computational procedures to calculate performance measures of the system in case of evaluative models, whereas for generative models they provide optimal values of decision variables. The analytical models usually require minimal data and are quick in providing the results. But, for complex systems several simplifying assumptions need to be made to obtain analytically tractable models, and these simplifying assumptions may not represent the behavior of the real system accurately. Simulation models on the other hand can be used to model considerably complex systems as accurately as desired. Some of the analytical and simulation models published in literature for serial production lines are discussed here.

2.2 Analytical Models of Serial Production Lines

Papadopoulos and Heavy (1996) provide an extensive literature review on queuing network models for production and transfer lines, with 257 references. Queuing networks are categorized into open queuing networks and closed queuing networks. Open queuing networks are used to model production lines. The models in their basic form resemble the configuration as described in section 1.1. The common assumptions made in most of the models are: steady state operation, zero transport times between stations, operation-dependent single-machine failures, no scrapping of parts, single part type, first-

in-first-out (FIFO) queuing discipline, exponential processing times, times to failure, and repair times, and availability of ample repair personnel. The queuing network models are broadly classified based on the assumptions made as follows: a) models may consider either discrete or continuous part flow, b) and the line may be assumed to have either reliable or unreliable machines, c) the service distributions can be exponential, phase-type or general, d) the models may consider the line as either saturated or non-saturated. A line is saturated if the first station is never starved and last station is never blocked; otherwise it is said to be non-saturated, and e) finally, the models differ based on the type of blocking considered. The type of blocking is *blocking after service* (BAS) if a station processes a part and gets blocked if there is no space for the part on downstream conveyor or downstream buffer. The other type of blocking is *blocking before service*, where station would begin processing a part only if there is space for it in the downstream buffer. In rest of the document we consider only BAS blocking unless mentioned otherwise.

We can also categorize these models into exact result models and approximate models. The exact models are based on one of three methods: closed form solution, state model method and holding time method (HTM). The literature reviewed suggests that, exact models are available only for short production lines up to three stations. The approximate models are divided into three categories. The approximate models with exponential service times and reliable machines, which are mainly based on decomposition method. The idea behind decomposition method is to decompose the system into smaller subsystems which are easier to deal with, deriving a set of equations which determines unknown parameters for each subsystem and then developing an

algorithm to solve these equations. A complex characterization of these subsystems leads to better approximation of real behavior of the line, but, also results in increased computational complexity. Hence, there is a trade-off between accuracy and complexity. Second category of approximate models is for lines with general service times and reliable workstations. There are two main approaches for these models: phase-type approach where the general service distributions are approximated by phase-type distribution followed by application of decomposition method; and generalized expansion method which can handle split and merge configurations. Third category of approximate models is for unreliable production lines. Both decomposition and aggregation methods are used for these models. Aggregation method is less accurate as it ignores blocking and starving.

Additionally, some of the continuous part flow models are used as approximation of unreliable asynchronous lines with deterministic service times. For particular references in all of the categories mentioned above, reader is directed to Papadopoulos & Heavy (1996), Govil & Fu (1999), and Dallery & Gershwin (1992). Dallery & Gershwin (1992) also present review of models which relax some of the typical assumptions like models which consider scrapping of parts, parallel machines in a stage, limited repair personnel, and non-zero transfer time between stations.

Next, some of the analytical studies for serial production lines are discussed briefly with the method used and the corresponding assumptions or drawbacks.

Tempelmeier & Burger (2001) presented an approximation procedure for analysis of saturated production lines which accounts for station breakdowns as well as imperfect production with scrapping. Generally distributed random processing times were

considered with operation dependent failures; the time to failure being exponentially distributed. It is assumed that there are ample repair personnel resulting in independent repair times at different stations.

Kavusturucu & Gupta (1999) developed an analytical methodology to calculate throughput of manufacturing flow lines with finite buffers and unreliable stations. They use expansion method along with decomposition method. The arrival stream follows Poisson distribution, processing times and repair times being exponentially distributed. Further drawback of this method is that it works better only with higher values of buffer capacity.

Hillier & So (1991) studied the effect of length of machine uptimes, machine downtimes, and inter-stage storage capacity on throughput of production lines with more than three stations using exact analytical methods. Their results for four to five-station lines show that it is desirable to have frequent short downtimes than infrequent long downtimes. For longer lines up to eight to nine stations, a heuristic method is proposed to estimate required storage space to compensate for machine breakdowns. This method assumes that all workstations are subject to breakdowns; all stations are identical and have same distributions for failure processes.

Papadopoulos & Vidalis (2001) developed an algorithm based on segmentation approach to address the buffer allocation problem for short unbalanced production lines having up to six machines that are subject to breakdowns. Single-machine, operation-dependent failures are assumed with exponentially distributed times to failure. Processing times and repair times are assumed to follow Erlang-K distribution.

Heavy, Papadopoulos & Browne (1993) present a methodology to calculate the throughput rate of unreliable serial production lines with Erlang distributed processing times and repair times, exponentially distributed times to failure, and buffers of non-identical capacities between successive stations. The methodology involves generation of state transition matrix, which limits the size of the model severely due to the curse of dimensionality.

Tempelmeier (2003) considers the problems faced by industrial planners which deal with real-life asynchronous production lines with unreliable machines and random processing times. It is claimed that the performance evaluation and optimization problems for real-life production lines can be handled using analytical algorithms or approaches like “Accelerated Dallery-David-Xie” (ADDX), decomposition method based on stopped arrival G/G/1/N queuing model, and combinations thereof to tackle different aspects like balanced or unbalanced lines, parallel stations, stochastic or deterministic processing times, and failures. But, the methods proposed do not work well for small buffer sizes. Also, complex subsystems other than parallel subsystems and operator interference issues are not dealt with.

One of the main assumptions made in most of the analytical models of the production line is that of availability of ample operators. With highly automated production lines like the PCB assembly lines under consideration, operators are responsible for tending more and more machines. Usually operators are responsible for a group of four to five machines. When the service demands within this group of machines are not synchronized, these machines experience interference, which decreases the productivity of manufacturing systems. This problem has been addressed in isolation as

machine interference problem. In Goss (2000) machine interference is defined as “the time that equipment is available to run but is waiting for an operator to finish serving other equipment.” Accordingly, the equipment must be in one of the following states: processing, down, starved, or machine interference.

Machine interference accounts for the time when both machine and material are simultaneously available but the machine is not processing due to operator unavailability. This is the distinguishing factor between what we call *operator interference* and machine interference. Here, time spent waiting for the operator for repair/replenishment is not considered. This waiting for operator when machine can't process is considered as “Down” state (which also includes the actual repair time), whereas we regard this state as either Failure or Exhaust (depending upon whether machine encountered a failure or component part exhaustion), and these two states combined are regarded as the operator interference. Machine interference as described in Goss (2000) could be more appropriate for situations where significant time is spent in setups. In case of PCB assembly lines, one of the most important factors decreasing productivity is the unpredictable downtime. Hence, we stick with operator interference which helps to concentrate on the waiting for the operator when the resource can't continue processing due to either machine failure or component part exhaustion. In this thesis, for a given operator allocation, operator interference in work area of each operator is observed from the resource state graph. Based on this observation, we come up with strategies either to reduce the number of operators by altering the task allocations or to cross-train operators so as to reduce interference, thereby increasing the throughput.

2.3 Simulation in Serial Production Line Analysis

Smith (2003) presents a survey of use of simulation for manufacturing systems.

The references thereof are categorized into design, operation, and simulation language/package development for manufacturing system applications. Law & McComas (1997) discuss how simulation can be used in design of new manufacturing systems and to improve the performance of existing ones. Kiran, Kaplan & TamerUnal (1993) provide an overview of application of simulation in various manufacturing scenarios in electronics manufacturing including PCB assembly. The problems they identified in PCB assembly that could be addressed with simulation are capacity analysis, operator requirement analysis, material handling system design and analysis, and process improvement and automation analysis. They identified that capacity analysis in the form of machine and operator requirement analysis tops the list of issues.

2.3.1 General Simulation Studies

Huang, Dismukes, Shi, Su, Razzak, Bodhale et al (2003) presented a methodology for productivity measurement and analysis at factory level. In this study, authors have developed an *Overall Throughput Effectiveness* (OTE) metric based on analysis of *Overall Equipment Effectiveness* (OEE). These metrics are then integrated with simulation analysis for manufacturing productivity improvement. Then, simulation is run to investigate the significance of various improvement opportunities. Sangeetha (2004) extended this concept by employing the subsystem recognition algorithm to disintegrate complete system to its basic subsystems and automatically evaluate the performance

metrics to identify the bottleneck. But, there is no provision to capture the operator interference. Also, formulae to calculate OEE for rework and more complex subsystems are not available.

Hegde, Ramamurthy, Tadikamala & Kekre (1988)) investigated alternative line configurations for their impact on throughput and WIP inventory levels. The sensitivity of various designs to variability in processing time and variability due to breakdowns and repairs is studied using experimental analysis. This approach of experimental analysis becomes infeasible as number of factors increases.

Gebus (2004) described how discrete-event simulation can be used in production optimization of electronics assembly lines by comparing different production alternatives. The results from the simulation are used as input for the optimization algorithm. Here, modeling is specifically focused on comparing scheduling policies only.

Kouikoglou (1994) developed a discrete-event, continuous flow model for serial production system with unreliable machines having exponential, gamma or deterministic processing times, finite buffers, and limited repair personnel. This approach results in much faster models than the conventional simulators. Optimal repair allocation to maximize the throughput is also addressed. Phillis, Kouikoglou, Sourlas & Manousiouthakis (1997) extended this concept to consider problem of buffer and repair allocation using branch and bound procedure together with combination of discrete event simulation and perturbation analysis. The limitation of this approach is that if the events (like machine repair, rate change etc) occur frequently, then, the advantage of faster execution is lost.

Jeong, Perry & Zhou (2005) presented a study to quantify the gain due to parallel segments in flow lines. Due to characteristic configuration, in flow lines, a single machine interruption stops the entire line with capacitated buffers. Hence, the throughput for the PCB assembly lines is much lower than the capacity of the bottleneck machine. Changing the serial configuration of a segment to parallel segment can increase the throughput. These parallel segments can be observed for placement machines on many PCB lines. Also, there could be mechanism that allows boards to by-pass a busy or interrupted machine called a cluster. These are also observed on the PCB assembly lines that we studied and referred here as serial-layout, parallel operation processes. This mechanism differs from the parallel configuration in that it has separate buffers, there is a delay in transfer, and the fact that a board once by-passed a machine cannot return back. The authors have proved using analytical and simulation models that parallel line with the by-pass mechanism has significant advantages over a serial line in an automated SMT assembly; though it cannot achieve the throughput gain of a pure parallel system. Similar results are observed in the experimentation done as part of this thesis.

Ingalls & Eckersly (1992) bring forth some of the difficult modeling issues related to electronics manufacturing. They attribute the complexity in modeling to the nature of testing and rework processes and placement processes. Amongst the simulation issues identified, the first is automated assembly, where placement machines can have significantly different failure characteristics (MTTF) and repair distributions (MTTR) for different feeders. These factors can significantly affect the model flow and throughput and generalization of these differences can result in large errors. To avoid this generalization the best strategy is to collect the data from system files, which record the

transaction data, residing on the machine itself. Similar problem was experienced during data collection for the case studies presented in this thesis for the component part exhaustion rates for different reels of the placement machines. But, as later the manufacturer decided to splice the reels; this did not have significant impact. Another simulation issue is that of material handling systems and it is suggested that the material handling and control systems be modeled thoroughly unless modeler knows for sure that the control logic is very straight-forward and that it is certainly known that material handling system is not the bottleneck in the system.

2.3.2 Machine Interference and Operator Allocation

Kotcher (2001) used simulation and predicted that increasing staffing among a group of already lightly loaded machine operators, overstaffing would improve the throughput significantly. This counterintuitive result was attributed to the situations where production equipment has frequent and unpredictable need for operators, but, the operators being unavailable as they are tending other machines. A method of estimating such operator-induced throughput loss is described. Goss (2000) in his thesis used simulation to evaluate operator cross-training program by measuring machine interference. The correlation between workload of an operator in home area and workload in other areas are analyzed to decide upon the cross-training strategies.

The statistics provided by the template modules are helpful in this regard. In this thesis, resource states Failure and Exhaust, which represent the time lost due to operator unavailability, are observed in resource state graph for the task of operator reallocation.

2.3.3 Buffer Allocation Problem

Conway, Maxwell, McClain & Thomas (1988) investigate the behavior of the buffered lines and the distribution and quantity of work-in-process inventory that accumulates. Some of the results presented in this paper are: a) the loss of capacity due to interference in asynchronous serial systems occurs mainly in the first few stations; longer lines are only a little worse than the short ones and the loss depends upon the degree of variability exhibited by processing time, b) the loss in capacity due to interference can be reduced by placing buffers; but, the improvement diminishes rapidly with increasing size of buffers, c) for a symmetric line, central placement of buffers is better, but, near-center placement is almost as good as center-placement, d) buffers provide less improvement in unbalanced lines and buffers towards the bottlenecks are preferred, and e) buffer capacity against random failures should be in multiples of average quantity that one station produces during the repair of another station and the size of the multiple depends upon the degree of variability exhibited by repair time.

Abdul-Kader & Gharbi (2002) address the problem of capacity estimation of a multi-product line with unreliable workstations and intermediate buffers. A simulation-based experimental design methodology is proposed to improve performance expressed in terms of cycle time. The objectives are to determine where to locate/allocate the influential buffers so that variability in the cycle time is small and effects of the uncontrollable variables like the failure and repair are minimized. But, as the number of stations increases, number of factors becomes large and hence number of simulation runs required becomes very large. Hence, application of this method becomes infeasible for longer lines.

Enginarlar, Li & Meerkov (2002) investigated smallest level of buffering needed in unreliable serial lines for exponential, Erlang, and Rayleigh reliability; the smallest buffer being measured in terms of average downtime. Analytical methods like aggregation as well as simulation are used and rules-of-thumb are provided for selecting buffer capacity. But, this method assumes that each machine has the same efficiency, the efficiency being the ratio of the up time to the total time.

Harris & Powell (1999) developed a simplex search algorithm for optimal buffer allocation for reliable, balanced/unbalanced serial lines. The algorithm uses simulation estimate of throughput for every candidate configuration, with simulation run length being adjusted at run time of algorithm to save simulation run time. The drawback of this method is that it does not take machine breakdowns into consideration. Also, the number of candidate allocations to be simulated during each iteration of the procedure increases as the number of stations increases.

2.3.4 Simulation Environment

Farrington, Rogers, Schroer, Swain & Evans (1995) developed an environment to simplify the creation of simulation models. This environment consists of three interconnected elements, namely problem definer, static analyzer, and code generator. These elements are connected through data transfer links and feedback loops and provide increasing level of modeling capability, aiding the simulation model development. The problem definer is used to develop the initial definition of an electronics assembly line through use of a graphical user interface. Static analyzer is then used to do the static

analysis on each component of the manufacturing line model. The code generator generates the code for the simulation packages in required format. Doss & Ülgen (2004) explored concept of building application-specific models based on discrete event simulation software engines for use by non-simulation personnel. The goal was to reduce the model development and modification time and complexity. This is accomplished by building data-driven, template models through user interfaces that are external to the model. In this thesis a framework for automation of the proposed methodology is given which is formed on the basis of the studies mentioned above. This automation will expedite the model development and enable the managers with basic simulation knowledge make changes to the simulation model and try out different scenarios.

2.3.5 Bottleneck Detection and Throughput Improvement

In order to improve the throughput of the system, the throughput of the bottlenecks has to be improved. But, finding the bottleneck itself is not a trivial task. Conventional methods for bottleneck detection include waiting time method and the workload method represented by the percentage of time the machine is active. With the waiting time method, the accuracy is compromised if we have finite buffers. When measuring workload, there could be multiple resources for which the difference between the percentages of time being active could be very small. Hence, the results might not be always accurate. Roser, Nakano & Tanaka (2001) developed a method for bottleneck detection in discrete event system by examining the average duration of a machine being active, for all the machines. In this method the state of the machine is categorized as

either being active or inactive. The *active duration* of a machine is defined as the duration of time the machine is in any of the active states consecutively, without being interrupted by any of the inactive states. Then the machine with the *longest average active duration* is identified as the bottleneck, as it is least likely to be interrupted by other machines, and is most likely to dictate the overall systems throughput. But, this method does not take into account parallel subsystems and batch processing as is shown in Chapter 5.

Kuo, Lim & Meerkov (1996) formulated and analyzed a new definition of production bottlenecks from the total system point of view. They define the bottleneck as the machine for which the sensitivity of the system's performance index to its production rate in isolation is largest. A method is developed which analyzes the manufacturing blockages and starvations of each machine leading to simple rules for bottleneck identification, obviating the need to calculate the production rate sensitivities. Based on this approach, Chiang, Kuo & Meerkov (2000) developed a method for identifying *down-time bottlenecks* in longer serial production lines with unreliable machines and finite buffers. Later Chiang, Kuo & Meerkov (2001) investigated *cycle-time bottlenecks* for production lines with Markovian machines having different cycle times. It is demonstrated based on cycle-time bottleneck and down-time bottleneck procedures that the performance of manufacturing systems with Markovian machines can be improved either by improving machine reliability or by increasing speed of part processing. A hypothesis is advanced that, irrespective of the statistics for machine reliability, the probabilities of manufacturing blockage and starvation, can be used as tool for bottleneck identification in a manner compatible with procedure proposed in these studies. But, it is

not clear how these methods work with parallel, split and merge, and even more complex subsystems.

Throughput has been identified as the most important parameter affected by the presence of a bottleneck in production lines. D'Souza (2004) in his thesis described a throughput-based technique for identifying bottleneck in production system using discrete event simulation. This technique enables capturing both static and dynamic resources using a single parameter, *Drop in Throughput (DIT)*. The general procedure involves identifying the target throughput, followed by sequential addition of resources to the simulation model and collecting information on throughput as well as DIT for each of these reduced systems; then the bottleneck is identified as the station whose addition to the system causes largest drop in throughput. The drawbacks are: the effect of breakdowns is not considered and each step in progression needs modification of the routing of the system.

Given the complexity and random events involved in the manufacturing system, the system may change over time. As a result the bottlenecks might also shift from one machine to another as the system undergoes gradual changes, for example a machine failure might render a non-bottleneck machine to become momentary bottleneck.

Lawrence & Buss (1994) examined the phenomenon of *shifting bottlenecks*. A *bottleneck shiftiness measure* is proposed for investigating several policies for reducing the shiftiness. This measure represents the complexity involved in planning and control of the system. A counterintuitive result is shown where bottleneck shiftiness can be reduced while simultaneously improving flow time by increasing the capacity of the non-bottleneck work-centers, thereby reducing the possibility of principal capacity bottleneck

being starved for work. But, the shiftiness is calculated using the bottleneck probabilities which in turn are calculated based on the long-run proportion of the time a given work center has more jobs in its queue than any other; which is inappropriate when the buffers are capacitated.

Roser, Nakano & Tanaka (2002) provide a method to detect the shifting bottlenecks. The method of active durations proposed in Roser, Nakano & Tanaka (2002) is extended to determine the shifting bottlenecks. The overlap of active period of a bottleneck with the previous or subsequent bottleneck represents the shifting of the bottleneck from one machine to another. It can be determined at any given time if a machine is *non-bottleneck*, a *shifting bottleneck*, or a *sole bottleneck*. To determine the primary and secondary bottlenecks over a period of time, the percentages of time for which a machine is either a sole bottleneck or part of shifting bottlenecks are compared. Based on this approach Roser, Nakano & Tanaka (2002a) present method of calculating the sensitivity of the manufacturing system throughput to the various variables associated with the machines using a single simulation. Here, the events consisted in bottleneck period are analyzed. Knowing the bottleneck periods and events therein, the percentage effect of each machine state onto the overall throughput is calculated. The drawback of this method is that it is true only as long as the changes in the machine variable do not result in drastic changes in bottleneck of the system.

Roser, Nakano & Tanaka (2003) formulated a buffer allocation model based on a single simulation. Every occurrence of blocking and starving for each machine is analyzed in this method, followed by finding the cause of these occurrences. Also, the buffer locations on the path between the idle machine and the cause of that idleness are

tracked and the time a possible buffer location is part of a path is determined for each machine.

2.4 Conclusions

In this chapter a review of literature pertaining to analytical as well as simulation models was presented for serial production lines, which is the category the PCB assembly lines under consideration fit into. Specifically, the review focused on the categorization of the modeling methods, analytical and simulation models for throughput estimation, buffer allocation, machine interference and operator allocation, bottleneck detection and throughput improvement, and simulation modeling environments to reduce modeling efforts and speed-up the analysis process. It is observed that, the analytical methods, though fast, ask for several simplifying assumptions which render the model unrepresentative of real behavior of the systems under consideration. Most of the models are based on the assumption of exponential distribution for processing times, times to failure or times to repair or some combination of these three; which is not a correct representation of a well-engineered workstation. Apart from this, few other aspects which need to be considered are unreliable workstations, finite buffers, scrapping of parts, multiple part types, limited personnel, and parallel, split and merge or ever more complex configurations of subsystems. While few models consider some of these aspects, they ignore the rest. On the other hand, simulation studies found in the literature address these issues, but, most of them are designed to handle, in isolation, the problems like buffer

allocation or operator allocation or bottleneck detection. Also, the simulation studies presented here have certain limitations as discussed.

Thus, a comprehensive tool or method; which captures the complex interactions within manufacturing systems like the PCB lines under consideration, and enables managers tackle the problems in various domains like the ones mentioned above, is missing. This thesis is an attempt to provide this tool or method. A methodology and related tools are proposed, which can be applied to: estimate throughput of an existing or non-existing PCB assembly line, assess the feasibility of the proposed line configurations and the relative impact of changing these configurations. It offers a visual tool to tackle the problems like bottleneck/constraint detection and throughput improvement, operator allocation or reallocation, and buffer allocation. Given a line configuration and allocation of resources, application of this methodology will enable managers make better decisions with regard to addition or allocation/reallocation of resources like machines, operators or buffers.

The next section discusses the PCB assembly template which forms the basis for this work, the statistics provided by the template modules and the problem dealt with in this thesis

CHAPTER 3

PROBLEM DEFINITION

In this section a brief overview of the PCB assembly template developed in Mukkamala, Smith & Valenzuela (2003) and Mukkamala (2003) is given. The information and statistics provided by the template modules and the potential use of this information for analysis is discussed.

Various resources that can be commonly found on a PCB assembly line are: a) processing equipment, b) testing equipment, c) operator, d) conveyor, e) pallet, f) traverser, g) buffer, and h) component parts inventory. For all of these resources many modeling components need to be considered. Additionally, the characteristic configuration of serial production line with capacitated buffers necessitates a tool to capture the blocking and starving due to interdependence of processing steps and operators. With models of real life PCB lines being quite large in size, capturing the complex interactions and extracting the relevant statistics such as resource states using modules from commercially available simulation software makes the modeling phase very time-consuming and tedious. Additionally, developing simulation models for related problems in the same domain is a repetitive process. Templates are useful in this regard. A PCB assembly template for modeling in Arena 7.01 has been developed in Mukkamala, Smith & Valenzuela (2003) and Mukkamala (2003) and as mentioned there,

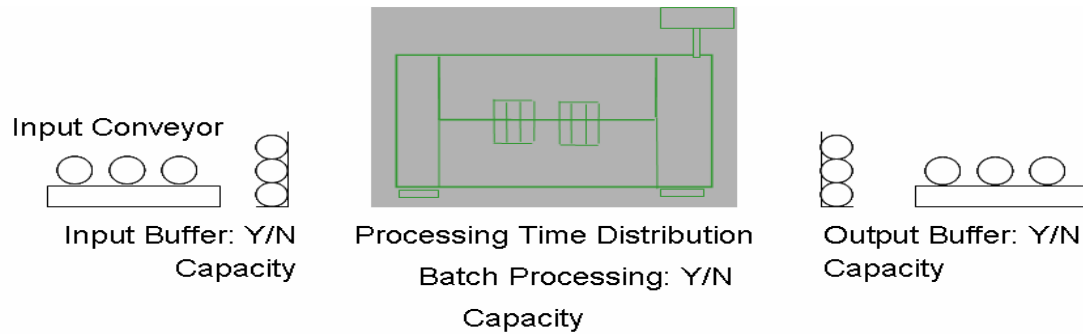


Figure 3.1: Typical Machine on PCB Assembly Line

“Templates are reusable components which encapsulate the domain-specific logic and hide the modeling details.” The template modules reduce the modeling efforts and facilitate custom reporting.

The work in this thesis is primarily based on the PCB assembly template and hence an overview of the PCB assembly template is given here. In order to understand the template, the configuration of a typical machine on a PCB assembly line is explained as follows. As shown in Figure 3.1, the machine has an input conveyor, it can have a finite-capacity input buffer, it has the processing station where boards can be processed individually or in batches, then it can have a finite-capacity output buffer, and finally it has an output conveyor. Also, the machine might require set up or indexing every cycle. The indexing time and the processing time can be variable following certain distribution. As depicted in Figure 3.2, this machine can fail after certain *Time/Cycles to Failure* depending upon whether it is time or cycle dependent failure. This machine will then be repaired by a *Repair Operator* on the line in certain time given by *Repair Delay*. With some probability: *Major Failure percentage*, this operator won't be able to repair the

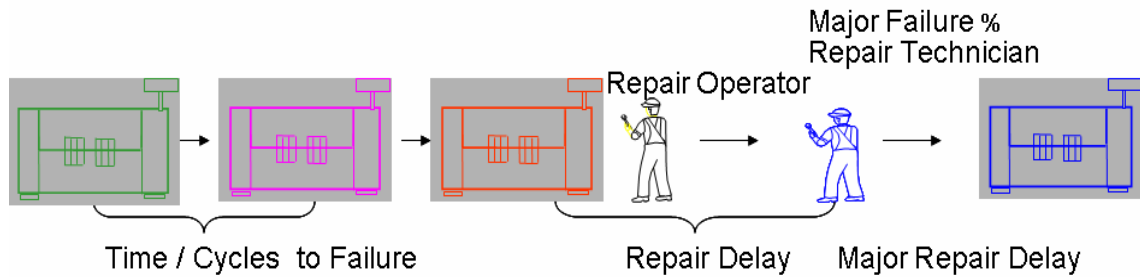


Figure 3.2: Failure Process of a Typical Machine on PCB Assembly Line

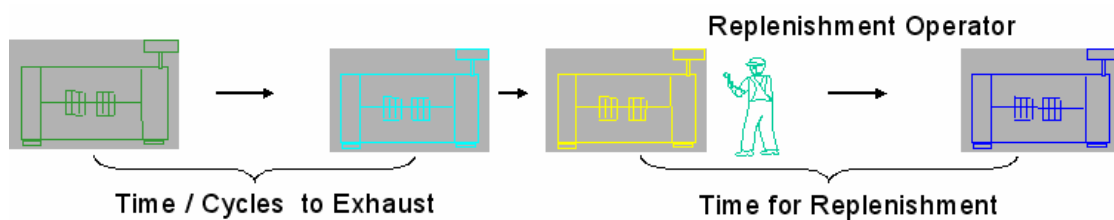


Figure 3.3: Component Part Exhaustion at Typical Machine on PCB Assembly Line

machine and then he/she will call a *Repair technician* who will repair the machine in time *Major Repair Delay*. Similarly, as shown in Figure 3.3, the machine will stop for component part replenishment after some interval given by *Time/Cycle to Exhaust* and the *Replenishment Operator* will replenish the component parts within some interval given by *Time for Replenishment* to get the machine working again.

All this information and corresponding logic is encapsulated in *Process Plus Module* of PCB assembly template (Mukkamala, Smith & Valenzuela, 2003 and Mukkamala 2003). Figure 3.4 shows the input dialog boxes for the Process Plus Module. The modules available in the PCB assembly template are: a) Process Plus Module, b) Inline Process Module, c) Inspection Module, d) Board Destacker Module, e) Board

Stacker Module, f) Conveyor Module. As a part of this thesis, a new module, Multi-stage Process Module, was added to the PCB assembly template. This module is designed for a class of placement machines with failures and exhaust, which simultaneously place

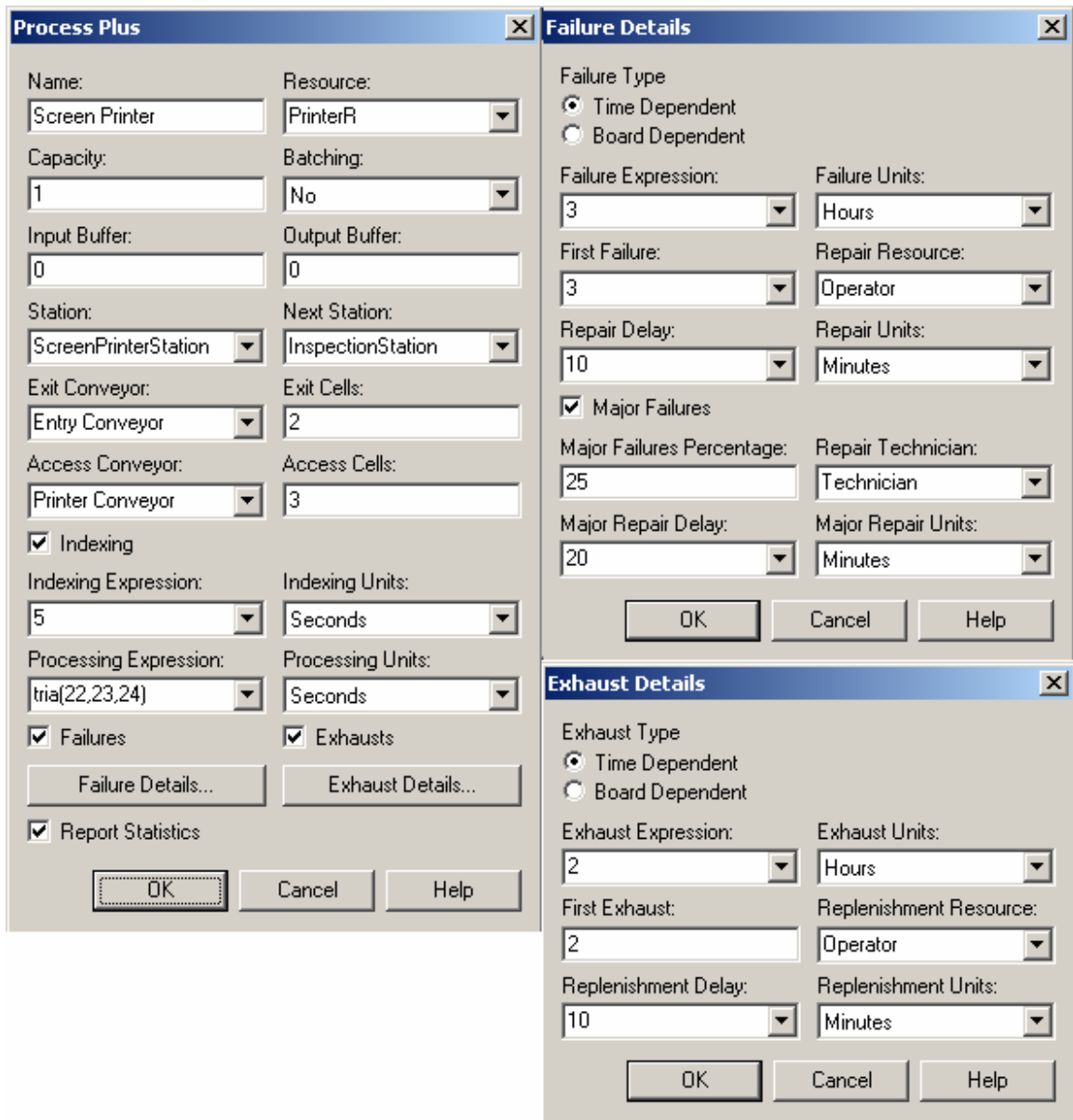


Figure 3.4: Input Dialog for Process Plus Module, Mukkamala (2003)

components with multiple heads onto certain number of boards traveling on a conveyor.

Figure 3.5 shows the input dialog for this module.

The template modules are designed to provide custom information and statistics.

One can track the resource state, number of major and minor failures, and number of

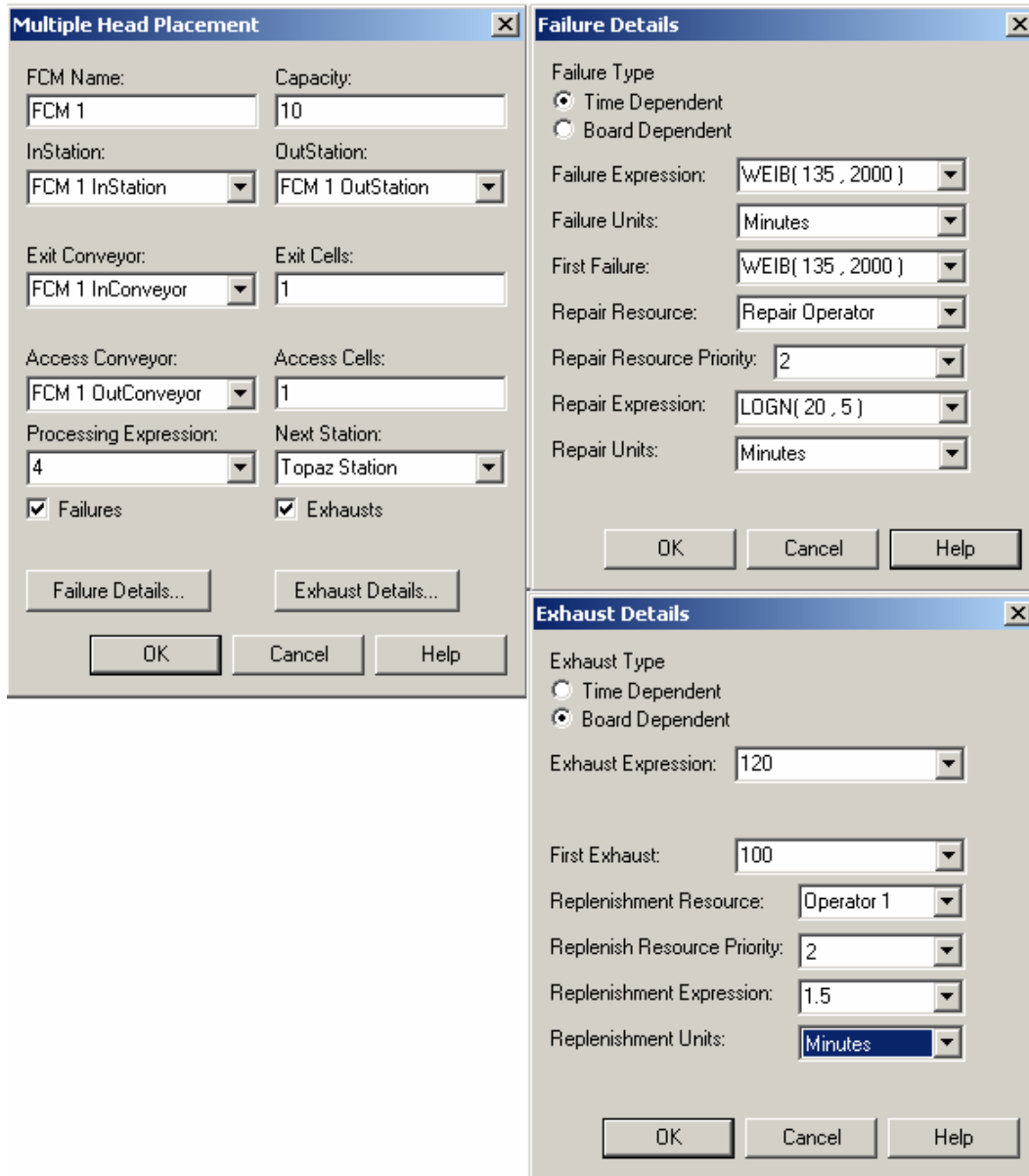


Figure 3.5: Input Dialog for Multi-stage Process Module

exhausts in user view during runtime. This is particularly useful for verifying the flow logic of model during verification stage. The statistics captured by the modules in the template include time spent by an entity in a module, number of entities processed by a module and the most important one is percentage of total time spent by a module in each of following states: *busy*, *idle*, *blocked*, *failure*, *repair*, *exhaust*, and *replenish*.

The distinction between seven states is very important for our analysis purpose. Hence, these seven states are explained with figures as follows. Figure 3.6 explains states *Busy and Idle*. Here, resource A is working on a board and its state is termed as *Busy*; whereas resource B has finished processing on the board and the board has released the resource B, and it has no part to work on. This state of resource B is called *Idle*. Figure 3.7 depicts *Blocked* state. In this case, resources A and B have finished processing on the

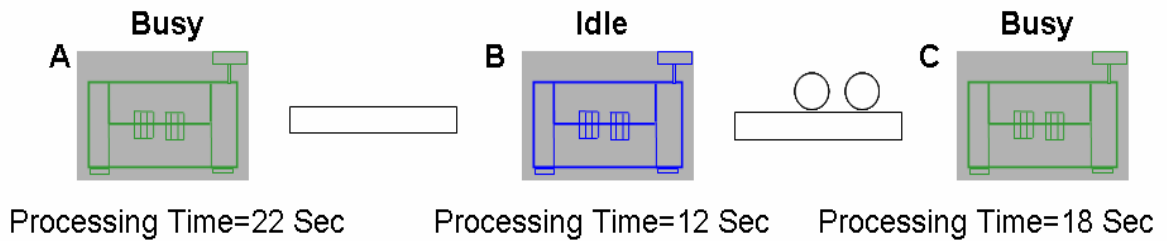


Figure 3.6: Resource States – Busy and Idle

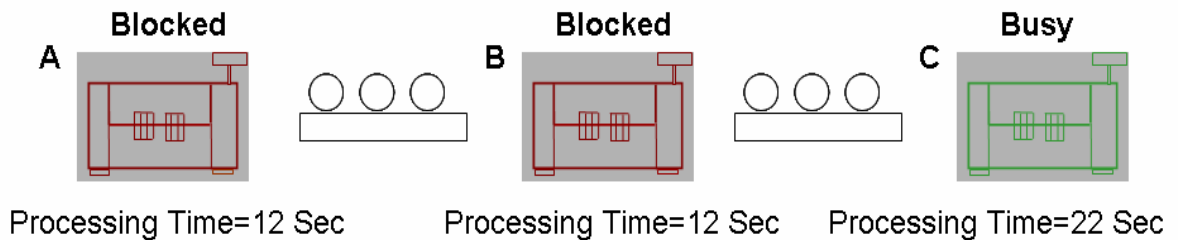


Figure 3.7: Resource States – Blocked

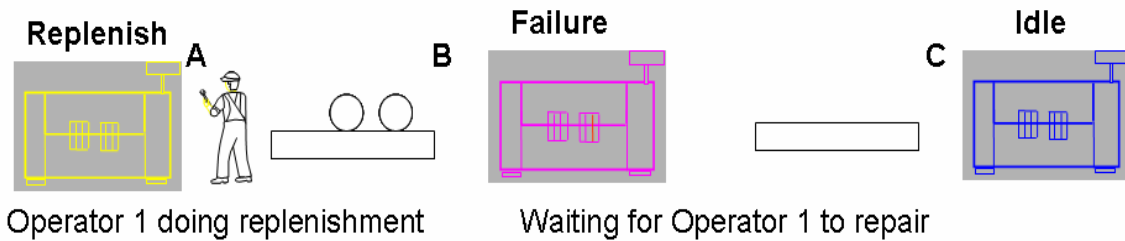


Figure 3.8: Resource States - Replenish and Failure

boards they were working on and they can't release these boards as the conveyors on the output sides of these resources are full. In this case, resources A and B are said to Blocked. In Figure 3.8, resource states *Replenish and Failure* are depicted. In this situation, resource A has encountered a stoppage due to component parts exhaustion and Operator 1 is replenishing the component parts inventory. This state of the resource A is called Replenish. At the same time resource B has encountered a failure and Operator 1 is supposed to repair it. But, as Operator 1 is busy at resource A, resource B is waiting for Operator 1. This waiting state of the resource B is called Failure state. Finally, Figure 3.9 explains the states *Exhaust and Repair*. In this case resource B has encountered a failure and Operator 1 is repairing it. This state of resource B is called Repair. At the same time, resource C has now exhausted component parts inventory, and is waiting on Operator 1 to come and replenish this inventory. This waiting state of resource C is called Exhaust.

This ability to capture blocking, starving, failures, and exhausts provides an excellent tool to analyze interactions between resources within a manufacturing system. The Busy state essentially tells the effective utilization of the resource. Idle and Blocked states signify the starvation and blocking caused by inherent speed mismatch, machine failures, and component part exhaustions. Repair and Replenish states show the

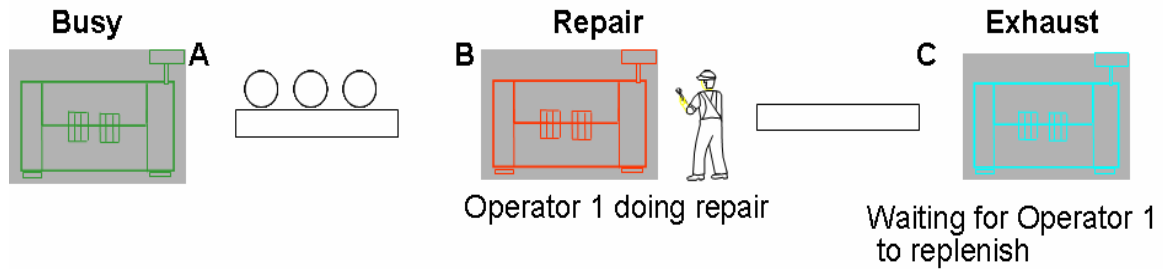


Figure 3.9: Resource States – Repair and Exhaust

productive time lost due to machine failures and component part exhaustions. Failure and Exhaust states capture the complex interactions between resources, namely, machines and operators. These two states combined represent operator interference, which is the time lost due to unavailability of operators when a machine goes down.

The goal of this thesis is to use this information provided by the template modules to analyze the PCB assembly lines. Specifically, the problem dealt with in this thesis is to determine the throughput of the system for certain configuration and then observe the patterns of resource states to identify the constraints or bottlenecks and evaluate the improvement strategies like adding capacity, operator and buffer allocation or reallocation. This analysis is aimed to support the managers in decision making in issues like designing or redesigning line configurations; to answer questions like where to add capacity or buffers, and how to allocate operators efficiently so as to improve the throughput.

CHAPTER 4

METHODOLOGY

This chapter discusses the methodology developed for simulation modeling and analysis of PCB assembly lines. It involves six steps as explained in Figure 4.1, to be followed in order. Following is the detailed discussion of each of the step.

4.1 Collect Input Data

This is one of the most important steps for successfully completing any simulation project. Simulation being a descriptive modeling technique predicts the output for certain set of input conditions. More often than not, simulation modelers lack the domain specific knowledge, and have to depend upon the decision makers of the

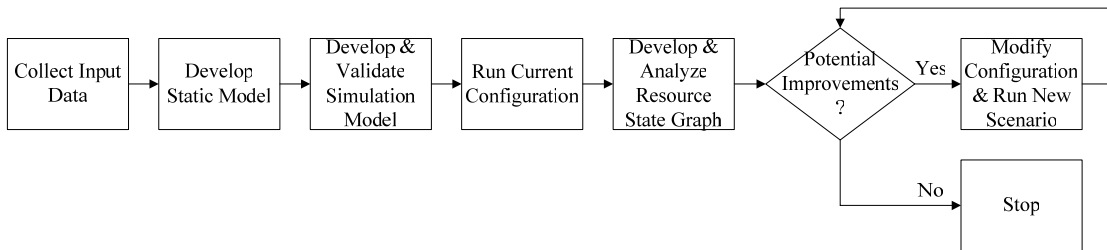


Figure 4.1: Methodology for Simulation Modeling and Analysis for PCB Assembly Line

particular domain (Mukkamala, Smith & Valenzuela, 2003). So, with most real systems being complex and stochastic in nature, it is of prime importance to ensure that the input data being plugged into simulation model represents the actual systems fairly accurately. For this purpose the simulation modelers and the decision makers pertaining to the particular domain should work closely to extract accurate data from the system.

This data may be extracted from historical databases of the line under consideration (if the manufacturing line already exists) or a similar line (if line is non-existent, i.e. proposed). For making the data collection step fast and easy, the set of input parameters has been standardized, which is summarized as follows:

i) Machine Data:

- a. processing type: single-board or batch
- b. indexing time (if required)
- c. cycle time (constant or random variable following certain distribution)
- d. failure data such as distribution for *time to failure* or *cycles to failure*, distribution for *time to repair*, and *repair resource*
- e. exhaust data such as distributions for *time to exhaust* or *cycles to exhaust*, *time for replenishment*, and *replenishment resource*

ii) Operator data:

- a. tasks allocated
- b. task times

iii) Inspection and rework data:

- a. passing percentages
- b. rework resource

- c. time for rework
- iv) Oven data:
 - a. for an inline process: the length and speed of conveyor are required
 - b. for a batch type process: batch size and delay are required
- v) Buffer data:
 - a. capacity of the buffer
 - b. queuing discipline (e.g. LIFO/FIFO, etc.)
- vi) Conveyor data:
 - a. length in terms of number of parts that can be accommodated
 - b. speed of the conveyor
- vii) Traverser/shuttle data:
 - a. loading time
 - b. transfer time
 - c. unloading time
- viii) Layout of the line depicting the detailed process flow logic.

Part of Input datasheet for one of the projects is shown in Table 4.1.

4.2 Develop Static Model of the Process Line

Based on the input data collected in Step 1, a static model is prepared for the validation and benchmarking of the simulation model. Static model is a spreadsheet based model developed in Microsoft Excel. In the static model, expected throughput is calculated for each individual process using the mean values of cycle times, failure and

repair rates, and exhaust and replenishment rates. Then, the throughput which is the least among all the machines or processes is identified as the throughput of the whole line. In doing so, we ignore: a) the inherent interdependence between the processes (induced by the capacitated buffers), b) the variability (induced by the unpredictable breakdowns and component part exhaustions), c) operator interference, and d) the re-entrant flow after detection of failures at various inspection stages and subsequent rework. As these dynamic aspects of line behavior are not considered, we call this model as static model.

For simplicity, the sub-systems like serial-layout, parallel-operation are approximated by parallel-operation processes where multiple, identical resources are

Table 4.1: Example Input Datasheet

Process	Placement	Load Conn.	Oven
Resource	FCM	Operator 6	Thermal Cure
Raw Cycle Time (Sec)-Var A	27.89	19.20	N/A
Raw Cycle Time (Sec)-Var B	30.02	19.20	N/A
Raw Cycle Time (Sec)-Var C	25.23	19.20	N/A
Time to Failure-TTF (Min)	WEIB(2510,.57)	N/A	WEIB(10100,.53)
Time to Repair-TTR (Min)	LOGN(48.6, 87)	N/A	WEIB(45.3,.81)
Repair Resource	Operator 2 & 3	N/A	Operator 9
Cycles to Exhaust 1	20	35	N/A
Time to Replenish 1 (Min)	1	0.5	N/A
Replenishment Resource	Operator 2 & 3	Operator 6	N/A
Cycles to Exhaust 2	N/A	280	N/A
Time to Replenish (Min)	N/A	1	N/A
Replenishment Resource	N/A	Operator 6	N/A
Passing Percentage	N/A	N/A	N/A
Length	N/A	N/A	38
Speed/Indexing Time	N/A	N/A	7.63 Sec

Table 4.2: Example Static Model - Details

Process	FCM 1	Load Connector	Thermal Cure
Raw Cycle Time (Sec.)-Var A	13.84	19.2	21.238
Raw Cycle Time (Sec.)-Var B	14.97	19.2	21.238
Raw Cycle Time (Sec.)-Var C	11.65	19.2	21.238
Max Panels per year-Var A: Scenario 1	508,136	366,282	331,133
Max Panels per year-Var B: Scenario 1	545,774	425,533	384,699
Max Panels per year-Var C: Scenario 1	470,498	285,484	258,089
Time To Failure (Sec.)	246,862	0	1,098,556
Time To Repair (Sec.)	2,916	0	3,235
Expected Failure Time (Sec.)	244,325	0	60,914
Expected Production Time - Var A (Sec.)	6,949,543	7,032,614	7,011,903
Expected Production Time - Var B (Sec.)	8,073,734	8,170,243	8,146,181
Expected Production Time - Var C (Sec.)	5,416,556	5,481,302	5,465,159
Panels per year -Var A: Scenario 2	502,134	366,282	330,158
Panels per year -Var B: Scenario 2	539,327	425,533	383,566
Panels per year -Var C: Scenario 2	464,940	285,484	257,329
Cycles to Exhaust 1	40	12	0
Time to Replenish (Sec.)	60	60	0
Expected Total Prod. Time(Sec.) Var A	6,196,341	5,201,204	7,011,903
Expected Total Prod. Time(Sec.) Var B	7,264,743	6,042,575	8,146,181
Expected Total Prod. Time(Sec.) Var C	4,719,145	4,053,879	5,465,159
Cycles to Exhaust 2	0	180	0
Time to Replenish (Sec.)	0	60	0
Expected Production Time -Var A (Sec.)	6,196,341	5,110,905	7,011,903
Expected Production Time -Var B (Sec.)	7,264,743	5,937,669	8,146,181
Expected Production Time -Var C (Sec.)	4,719,145	3,983,500	5,465,159
Panels per year -Var A: Scenario 3	447,712	266,193	330,158
Panels per year -Var B: Scenario 3	485,286	309,253	383,566
Panels per year -Var C: Scenario 3	405,076	207,473	257,329

Table 4.3: Example Static Model - Input/Output

Shift length	426	Minutes/shift
Number of shifts	17	Shifts/week
Year length	48	Weeks/year
Total production time	347,616	Minutes/year
Changeover time	600	Seconds
Changeover frequency	6	Changeovers/week
Total changeover time	2,880	Minutes/year
Production Mix	A	34.00%
	B	39.50%
	C	26.50%
	Total	100.00%
Scenario 3 Results	Types	Expected Boards Per Year
	A	266,193
	B	309,253
	C	207,473
	Total	713,724
Scenario 2 Results	Types	Expected Boards Per Year
	A	330,158
	B	383,566
	C	257,329
	Total	642,788
Scenario 1 Results	Types	Expected Boards Per Year
	A	331,133
	B	384,699
	C	258,089
	Total	973,921

represented by a single equivalent resource. For example, a serial-layout, parallel operation process, like the in-circuit testers, with each resource having cycle time of 60 seconds will be approximated by a single resource having cycle time of 20 seconds. The input parameters like the available production time per shift, number of shifts per day, number of days per year, the cycle times, failure/repair rates, and exhaust/replenishment rates can be changed easily to adapt to updates in the values of the same. Three scenarios are developed for the static model as follows:

- i) Scenario 1: Excluding both machine failures and part exhaustions,
- ii) Scenario 2: Including machine failures but excluding part exhaustions,
- iii) Scenario 3: Including both machine failures and part exhaustions.

The production quantities thus obtained are the upper bounds on the actual production, because, the production losses due to capacitated buffers, rejections at inspection stages, operator interference, and effects of variability are not considered here. At this stage, the line managers and industrial engineers from the manufacturer are consulted to verify that these upper bounds obtained are representative of the reality. Thus, in effect static model also works as a check on the input data to be used for simulation. Once the static model is verified, it is ready to be used for the validation and benchmarking of the simulation model. Scenario 1 is used to validate the simulation model by comparing its throughput with that given by corresponding simulation model. A part of the static model *details sheet* is shown in Table 4.2, whereas Table 4.3 shows the *input/output page* for one of the projects. The fields highlighted in yellow are the ones which the user can input or change easily.

4.3 Develop and Validate Simulation Model

The simulation model differs from static model by explicitly considering: a) capacitated buffers between each pair of processes, b) operator interference when operators are responsible for manual operations as well as for repair and replenishment of multiple processes, c) variability in terms of cycle time, failure/repair rates, and exhaust/replenishment rates, d) detection of board failures at various inspection stages and subsequent rework and e) exact flow logic of complex subsystems like serial-layout, parallel-operation. The PCB assembly template originally developed by Mukkamala, Smith & Valenzuela (2003) and enhanced as part of this thesis is used for modeling purposes along with the built-in templates available in software package Arena 7.01. All the components of the PCB assembly line, namely, processing and testing equipment, operator, conveyor, pallet, buffer, traverser, and component part inventory are modeled along with the corresponding flow logic. Subsystem configurations like parallel legs and even more complex ones like serial-layout, parallel-operation processes are also modeled. On most PCB assembly lines, the processes are connected by conveyors and the conveyor links typically provide the buffering between processes. The flow logic may depend on number of boards being conveyed on these conveyor links. The simulation model incorporates such flow logic. As a result of detailed modeling as mentioned above, simulation analysis can expose less apparent and complex interactions like the interdependence of processing steps due to capacitated buffers and operator interference caused due to multiple tasks being allocated to operators.

The model developed using the PCB assembly template is first verified by using an animation run. Here, the flow of entities is observed to make sure that the modeled

flow logic exactly resembles the one on the real line. The configuration used for validation is called the *Base Configuration*. It is modeled in line with Scenario 1 of the static model, where machine failures and part exhaustions are ignored, there is no operator interference (which effectively means that each process/task is assigned a separate operator, so that there is no time lost in waiting for an operator), detection of board failures at inspection stages and further rework is ignored, and mean values are used for cycle times (these are usually deterministic for PCB assembly line). Then, the simulation model is validated by comparing the throughput given by Base Configuration with that given by the Scenario 1 of the static model. Also, the line managers and industrial engineers from manufacturer are consulted at this point for validation. Model verification and validation is simplified as the template modules are pre-verified and pre-validated.

4.4 Run Simulation Model with Current Configuration

Current configuration resembles the design as proposed by the manufacturer. It incorporates variability in the processing times, failure/repair rates, exhaust/replenishment rates. The detection of board failures at inspection stages and the subsequent rework are modeled here. Also, it incorporates the operator assignment to multiple processes as proposed by the manufacturer. A trial run is made with the current configuration. Then, as Current Configuration involves sources of randomness, the number of replications to be run is decided upon by observing the 95% confidence interval half width for the throughput. Accordingly, the simulation is run for number of

replications decided upon as mentioned. The output statistic will then predict the ability to meet the planned production. The statistics obtained from the Current Configuration run are then used to develop the resource state graph. The resource state graph and its analysis to come up with improvement strategies are explained in next section.

4.5 Develop and Analyze Resource State Graph

The resource state report provided by the PCB assembly template gives significant insight into performance characteristics of the line. The Busy state essentially tells the effective utilization of the resource; this resource utilization is different from that given by Arena standard reports as it does not differentiate between the seven states as is done in this methodology. Arena combines Busy, Blocked, Failure, Repair, Exhaust, and Replenish states considered here into a single state to calculate the utilization. Idle and Blocked states signify the impact of starving and blocking caused by the inherent speed mismatch, machine failures, and component part exhaustions. Repair and Replenish states show the productive time lost due to machine failures and component part exhaustions. Failure and Exhaust states capture the complex interactions between the resources, namely, machines and operators. These two states combined represent what we call as “operator interference”. This is the time lost due to unavailability of the operators when the machine goes down. In case of the PCB assembly lines, which are characterized by highly automated machines, each operator monitors a group of machines (this group usually comprises five to six machines). The operator is responsible for attending the machine failures and component part exhaustions, apart from some manual operation in

some cases. If the line is understaffed or if tasks aren't allocated correctly, there could be many instances when one of the resources fails or exhausts the component part inventory, and waits for the operator to fix the machine (or replenish the component parts) as the same operator is tending some other machine. Thus, operator interference could have great impact on throughput of the line if the tasks are not allocated appropriately.

In the *resource state graph*, all the resources on the line are laid across the X-axis in the same sequence as they appear on the line. The percentage of time spent by each resource in each of the seven states is stacked along the Y-axis to form a stacked bar graph showing the percentage distribution of time for each of the seven states the resource was in over total simulation period. This Microsoft Excel graph is generated directly from Arena upon completion of simulation using a VBA project. All the states are color coded as follows: a) Busy – light slate blue, b) Idle – maroon, c) Blocked – beige, d) Failure – powder blue, e) Repair – green, f) Exhaust – orange red, and g) Replenish – blue. Other important factors to be understood about resource state graphs are explained as follows. The white bars in all the resource state graphs depict the resources for which there is no provision to collect the information about the states. For resources which have the characteristic configuration of serial-layout, parallel operation, for example, in-circuit testers and functional testers, the resource states are shown for all the resources in the subsystem. This is done so as to show the impact of such a design on the utilization of the resources in the subsystem. Whereas, for a subsystem configuration that works exactly in parallel, the resource states are shown for any one of the resource in the subsystem, as all the resources in such a subsystem have almost same percentages of time spent in each of the seven states. Thus, the difference between the two subsystem

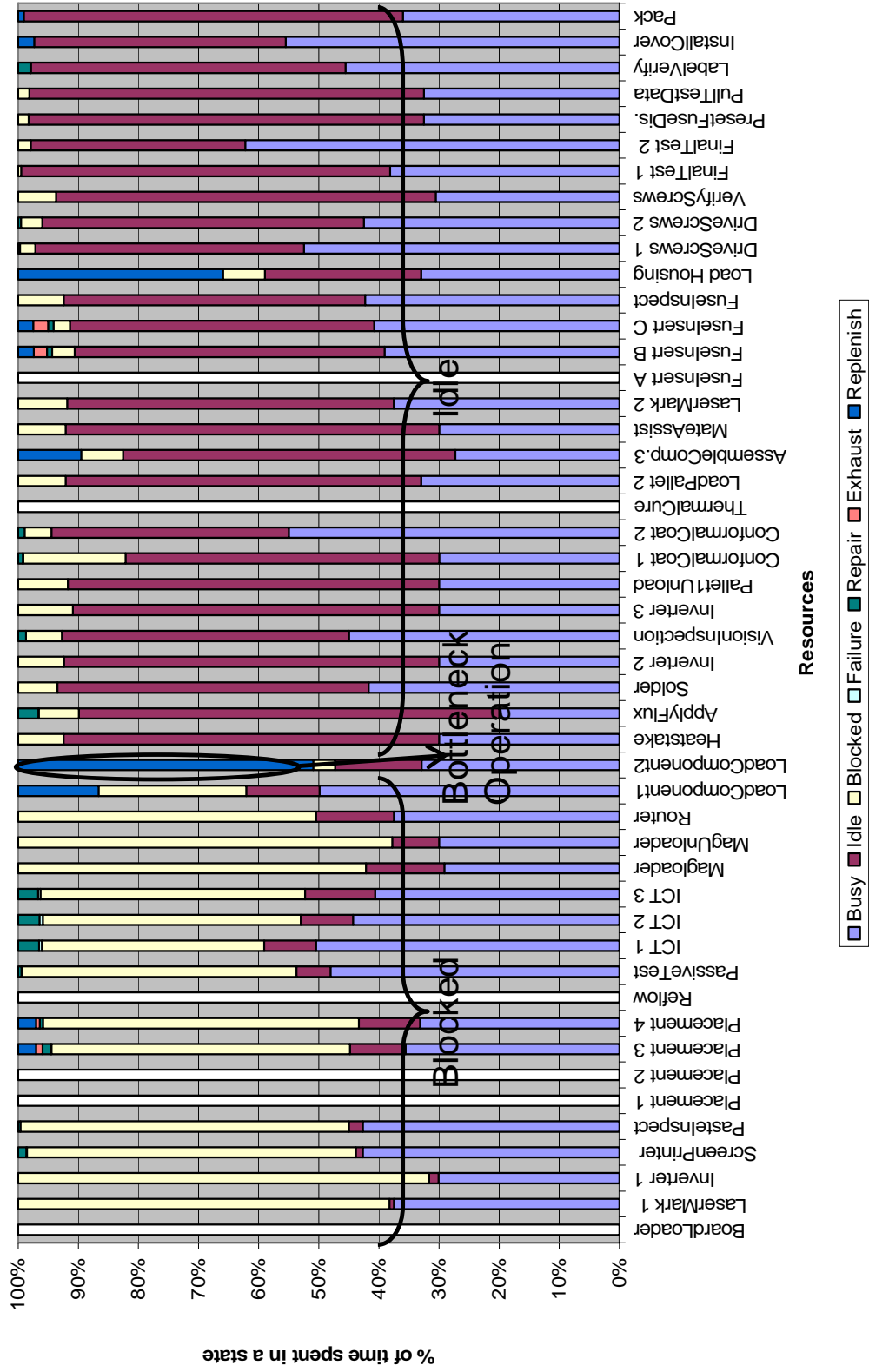


Figure 4.2: Example Resource State Graph

operations is made evident, with exact parallel configuration giving higher throughput rate than that of the serial-layout, parallel operation; supporting the observation presented in Jeong, Perry & Zhou (2005). This difference can be attributed partly to the delay in transfer in serial-layout, parallel-configuration subsystem. Lastly, the resource states for operators in the rework loop are not shown in the resource state graph as these are non-cyclic operations.

An example resource state graph is shown in Figure 4.2. Once the resource state graph is ready, it is analyzed to find the potential for improvement. For analyzing the Resource State Graph, specific patterns should be looked for. The patterns that one should look for in order to find the constraints or bottlenecks are explained as follows.

First we look for a specific pattern of significant blocking followed by significant starving. If this pattern is observed then the resource which separates the blocking and starving sub-patterns is identified as the bottleneck. This resource will block all the upstream resources significantly and at the same time it will starve all the downstream resources. For example, in Figure 4.2, the resources upstream of Load Component 2 are blocked for a significant amount of time (indicated in beige); whereas the resources downstream to Load Component 2 are starved most of the time (indicated in maroon). From the above observations it is pretty clear that Load Component 2 is the bottleneck for this scenario, as it is blocking the upstream resources and starving the downstream resources.

If such a pattern is not apparent, then, we look for pattern where there is remarkable spike in idle time from one resource to the other, with corresponding decrease in blocking. If such a pattern is seen, then, a constraint (it may not be a primary

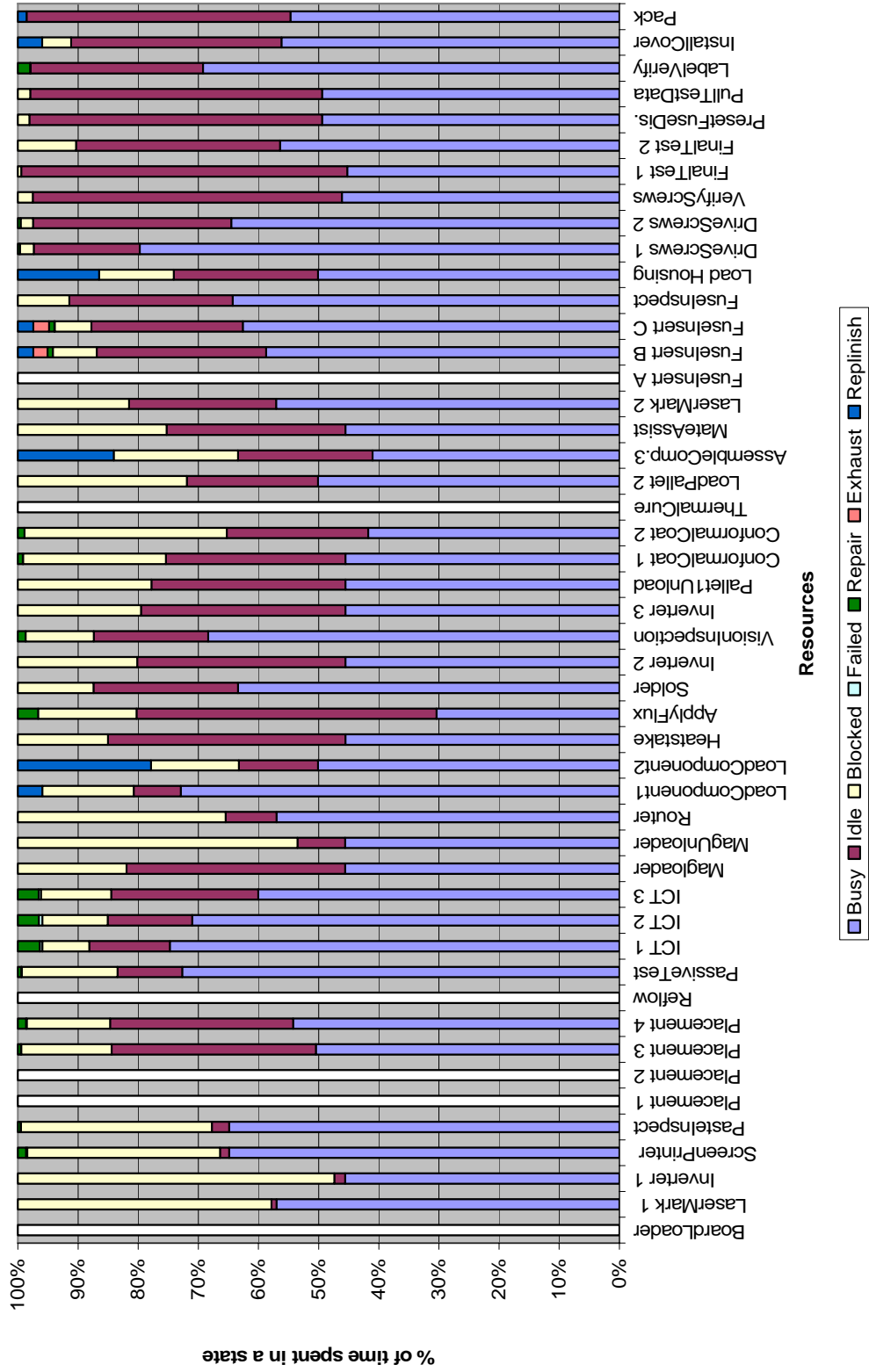


Figure 4.3: Example Resource State Graph 2

bottleneck by itself) is found, and overcoming this constraint will then potentially improve the throughput. For example, Figure 4.3 depicts resource state graph for such a scenario. A spike in idle time is seen from resource Paste Inspect to Placement 3, accompanied by decrease in blocking. This pattern indicates the location of a potential constraint and where the efforts should be directed for improvement.

If such a pattern is also not apparent then we look for the resources which spend highest percentage of time in states other than Idle and Blocked, because, more the percentage of time spent in these five states combined, lesser is the time the resource is available for further processing needs.

Once the constraint or bottleneck is identified, we look for the reasons for this resource to be the bottleneck. Then, depending upon the reason for a resource to be bottleneck, decision is made as to where to add capacity or some other appropriate remedy to overcome the constraint or bottleneck.

Next, one should look whether there is significant amount of operator interference. The operator interference is represented by the Failure and Exhaust states (shown in powder blue and orange red respectively). The set of machines or processes an operator is supposed to monitor is the *home work area* of that operator. If the home work area of a particular operator experiences a significant amount of operator interference (i.e. it spends significant time in Failure or Exhaust or both states), then one can conclude that this particular set of machines is understaffed, and some sort of operator reallocation is required. If, at the same time there is another work area where the effect of operator interference is insignificant, strategy of cross training of this operator for the tasks in work area with higher operator interference can be proposed. Observing the operator

interference can serve two purposes. If a stage is reached where the target throughput is met, then one can go on reducing the number of operators by increasing the number of tasks assigned to individual operators (which would essentially need cross-training from management point of view) such that the operator interference does not increase significantly, at the same time ensuring that the throughput does not decrease much. On the other hand, if significant operator interference is observed, re-allocation of the operators could be done so as to decrease the operator interference, increasing the throughput. In case of Figure 4.2, the impact of operator interference is negligible (indicating that the line is either overstaffed or it is staffed optimally). So, in such a case number of operators could be reduced, without losing throughput, by cross-training the operators looking at the operator interference in different work areas. Section 5 explains this procedure with an example from a case study.

Further, by looking at the time spent in Idle and Blocked states one can try to reallocate the buffers to reduce the blocking and starvation. For example, in Figure 4.4, resources upstream to Placement 5 are blocked for significant time, whereas the resources downstream are idle for most of the time. The bar showing resource states for ICT 1 renders ICT 1 as the bottleneck. But, from the layout shown in Figure II.1, ICTs have a characteristic configuration of serial-layout, parallel-operation which causes the utilizations of the subsequent resources in the subsystem to decrease from ICT 1 to ICT 4. Thus, though ICT 1 has the higher utilization than Placement 5, spare capacity is still available at ICT subsystem. Hence, Placement 5 is identified as the bottleneck resource. It can be observed that this resource is idle as well as blocked for noticeable amount of time. Adding buffers before and after this resource will reduce the amount of starvation

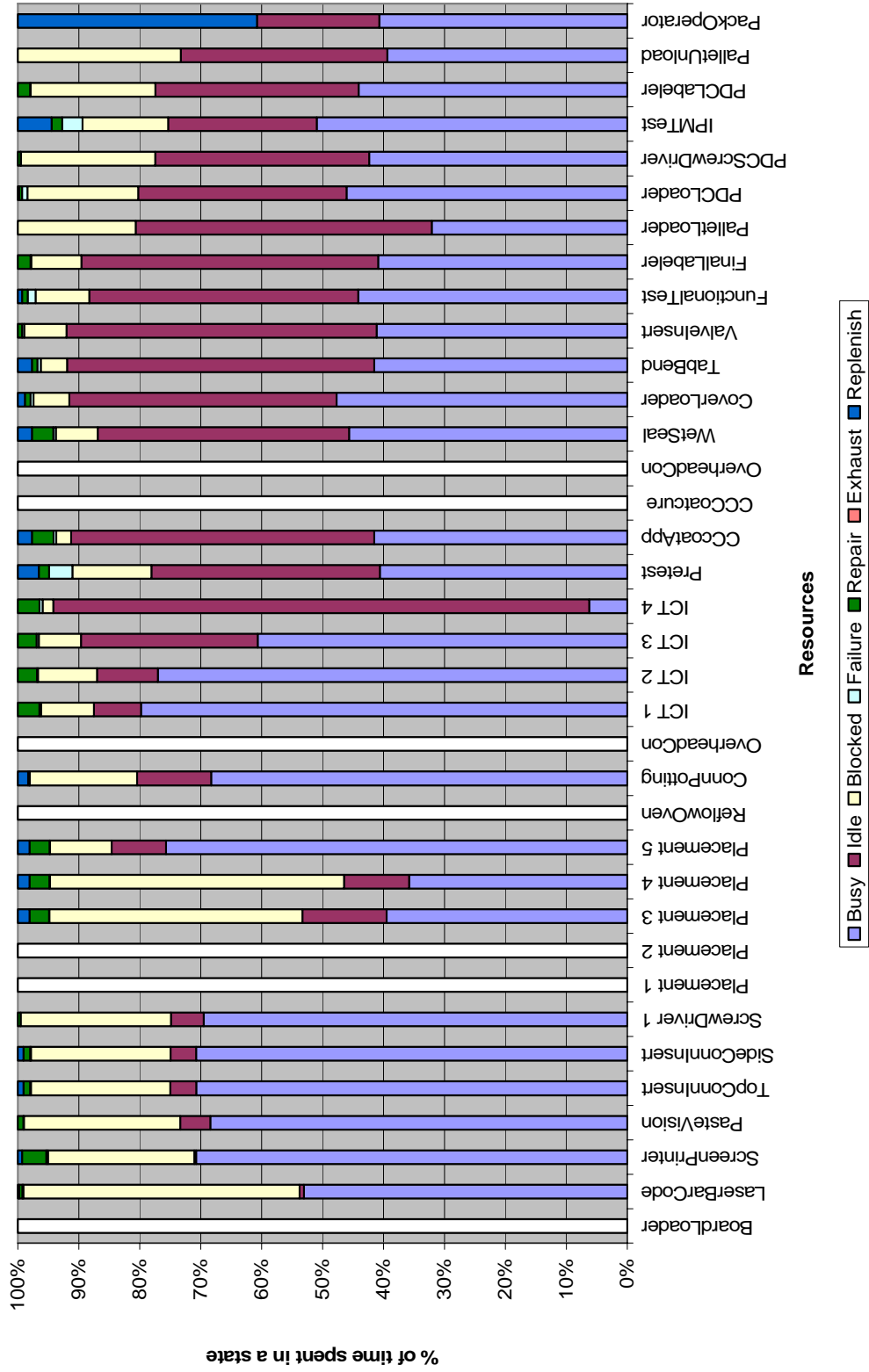


Figure 4.4: Example Resource State Graph 3

and blocking experienced by this resource. This will increase the availability of the bottleneck thereby increasing the throughput. All the above analysis steps are explained in Chapter 5 by means of examples from the case studies.

4.6 Modify Configuration and Repeat

The Resource State graph is analyzed as explained in the previous section. Based on this analysis, a set of improvement strategies is developed. Once we have the set of improvement strategies, the impact of implementing these strategies is then assessed by incorporating these changes in the simulation model. The line managers and decision makers are consulted on these improvement strategies and decision is made if some other improvement strategy needs to be considered and tested. Then, the most feasible improvement strategy is selected for further consideration. Once this process is over, the resource state graph is prepared for the modified configuration. This resource state graph is then analyzed to identify the new bottleneck, constraint or potential for further improvement. The improvement strategies are developed accordingly. Then, steps 5 and 6 are repeated iteratively until there are any significant discernable patterns shown by the resource state graph which could lead one to some improvement strategy.

CHAPTER 5

CASE STUDIES

This chapter presents case studies carried out for three of the PCB assembly lines for a local PCB manufacturer. The methodology as explained in previous chapter was used for simulation modeling and analysis of these lines. These lines are referred here as Product X line, Product Y line, and Product Z line. Note that the real data has been coded to protect proprietary information. The simulation modeling and analysis for each of these lines is explained in detail one by one.

5.1 Simulation Modeling and Analysis of Product X Line

This line did not exist when the analysis started. The manufacturer had an initial line configuration and was interested in determining whether the proposed configuration would meet planned production requirement of 1.5 million boards per year and in identifying ways to improve the line productivity.

The process started with the input data collection. As the line did not exist, historical data from other similar lines (which have many processes in common) was collected in the format as mentioned in Section 4.1. The failure data was collected from the machine logs; the data on stockouts was calculated from the planned inventory of component parts to be available at the machines. From the data points for time to failure

Table 5.1: Static Model Results for Product X Line

Scenario	Scenario 1	Scenario 2	Scenario 3
Expected boards per year	1,863,438	1,862,674	1,398,628

and time to repair, statistical distributions were fit using Arena Input Analyzer. Three different variants will be processed on this line with a product mix as follows: Variant A-34%, Variant B-39.5%, and Variant C-26.5%, with a changeover time of 10 minutes between different variants. It is assumed that the bare boards are always available for production. The process flow chart for this line is shown in Figure I.1. The total production time available per year was considered to be 347,616 minutes. Based on the input data collected, the static model was developed and the total throughput as given by the static model for three scenarios is depicted in Table 5.1. After verifying the results of the static model with the line managers and industrial engineers from manufacturer, the simulation model was developed and two configurations inline with Scenario 1 and Scenario 3 were run. The results are summarized in Table 5.2.

For the Base Configuration (which resembles Scenario 1 of the static model), the results of the simulation model and static model match closely, and hence we have strong evidence that the simulation model is valid. The difference between the throughputs

Table 5.2: Comparison of Static and Simulation Model Results For Product X Line

Scenario	Boards produced per year	
	Static model	Simulation model
Scenario 1	1,863,438	1,862,600
Scenario 3	1,398,628	1,046,817

given by the static model and the simulation model for the Current Configuration (which resembles Scenario 3 of the static model) is due to capacitated buffers, operator interference, variability in the failure and exhaust data, and detection of board failures ignored at various inspection stages and subsequent rework, which are not considered by static model. Despite the difference, the comparison verifies that the boards produced for Current Configuration are within upper bound set by the static model.

After validating the simulation model, the resource state graph is prepared based on the resource state statistics provided by the modules from PCB assembly template. The resource state graph for Current Configuration for Product X line is shown in Figure I.2. As indicated in Figure I.2, we can observe the pattern in which the resources upstream to Load Component 2 are blocked for significant amount of time whereas the resources downstream are starved for most of the time. Hence, Load Component 2 was identified as the bottleneck for the Current Configuration. As is observed from the graph, the resource (an operator in this case) which is responsible for this operation is spending almost 50% of the total time in Replenish state. We distinguish the inventory of component parts as either *on-hand inventory* (available in small totes at the workstation) or *off-hand inventory* (available in large containers away from the workstation). As this resource is rendered as bottleneck due to the replenishment process, three improvement strategies were proposed as follows: a) reducing time required for replenishment of Component 2, b) assigning Component 2 replenishment task to some other operator, and c) increasing on-hand and off-hand inventory of Component 2. After consultation with the line managers and decision makers, the strategy of assigning the task of replenishing the off-hand inventory of Component 2 to a material handler was selected so that the

operator who was doing it originally can devote his time in value added task of loading Component 2. The configuration of the simulation model was modified accordingly. This configuration is named Alternative 1.

From the simulation run for Alternative 1, the throughput increased to 1,352,515 – an increase of 29.2% over Current Configuration. Resource state graph for this configuration is shown in Figure I.3. This resource state graph was then analyzed to find potential for further improvement. Though the blocking and starvation has reduced as compared to Current Configuration, the discernable pattern of blocking of a group of resources and starvation of rest of the processes could still be observed. The processes upstream to Load Component 1 are blocked for significant time and processes downstream are starved for most of the time. This indicates that the bottleneck has now shifted to the process Load Component 1. Upon consultation with the line managers and decision makers, the on-hand as well as off-hand inventory of Component 1 was increased. This is referred to as Alternative 2; the corresponding resource state graph is shown in Figure I.4. Alternative 2 resulted in a throughput of 1,476,071 boards per year, an increase of 9.13% over Alternative 1.

At this stage, manufacturer decided to move to 15 shifts per week schedule from 17 shifts per week schedule. The target still remained at 1.5 million boards per year; hence, the manufacturer was interested in determining how they can achieve this target and where they need to add capacity. Also, due to design changes, conveyor segment data was updated. These changes were incorporated in the model, which resulted in change in the pattern of blocking and starvation. After incorporating these changes in the configuration of Alternative 2, this is considered as Current Configuration 1. Simulation

was run with Current Configuration1 and the corresponding throughput was 1,168,061. The resource state graph for this configuration is shown in Figure I.5. Upon experimentation, it was found that achieving the target production of 1.5 million boards per year by just adding capacity was not feasible, and hence it was decided that buffers be added along the line. But, due to process requirement there were only a few selected places where the buffers could be added. Hence, data was collected as to where the buffers could be added and what could be feasible sizes of these buffers. Accordingly simulation model was modified to incorporate these buffers; this is regarded as Scenario 1. Throughput for Scenario 1 is 1,235,055. The resource state graph for this scenario is shown in Figure I.6.

From Figure I.6, significant blocking of resources upstream to Final Testers and starvation of the downstream resources can be observed. Hence, the Final Testers are identified as the bottlenecks. The configuration of Final Testers resembles that of serial-layout, parallel-operation, but, is more complex. As shown in Figure 5.1, an incoming board first goes to Final Tester 1. If, by the time the first board reaches Final Tester 1, there is one more arriving board at Final Tester 2, i.e. both the testers have boards available for processing, two boards will be processed simultaneously. Otherwise, if, as shown in Figure 5.2, only one board is available for testing at Final Tester 1 and there is no board at Final Tester 2, then, testing will commence at Final Tester 1. As shown in Figure 5.3, during this testing, no further board is allowed to get into Final Tester 2. Thus, there could be instances when Final Tester 2 sits idle due to nature of operation.

This methodology, by observing the blocking and starvation pattern enabled detection of the bottleneck even with this complex configuration. Whereas, the

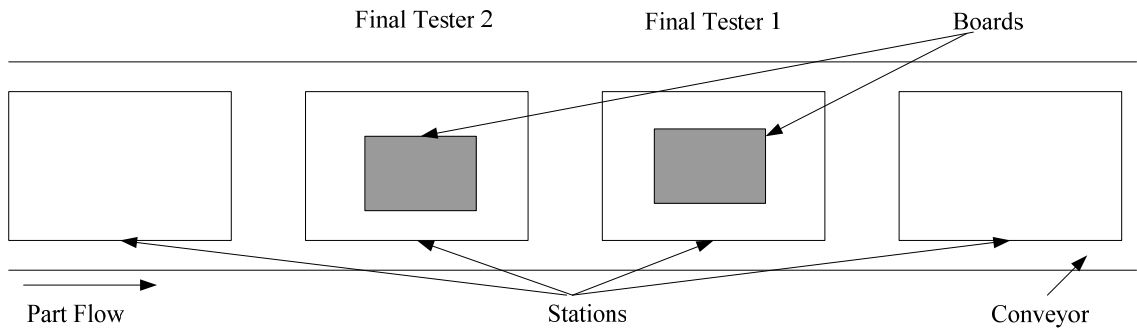


Figure 5.1: Final Testers for Product X Line - Two Boards Available for Processing Simultaneously

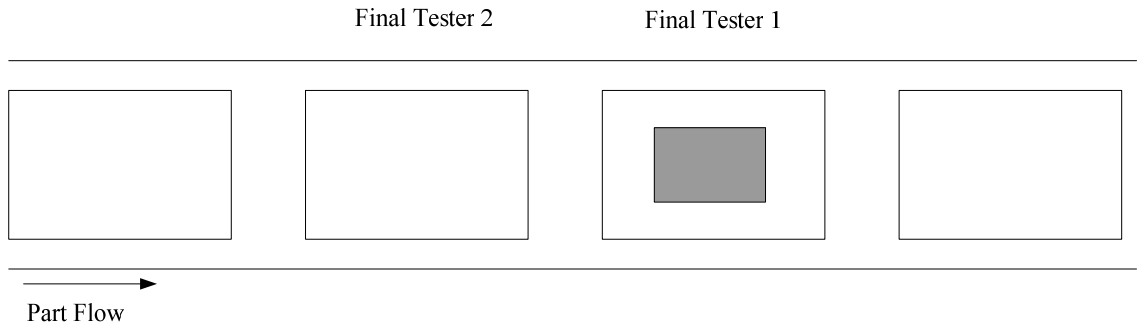


Figure 5.2: Final Testers for Product X Line - Only one Board Available for Processing

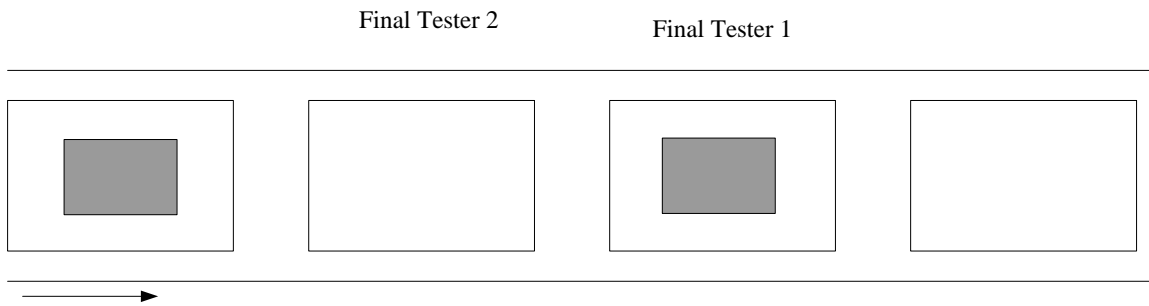


Figure 5.3: Final Testers for Product X Line – Final Tester 1 Busy, Final Tester 2 Idle

conventional method like observing utilization would have resulted in indicating Install Cover Operators as the primary bottlenecks. This demonstrates the ability and simplicity of the proposed methodology. By adding another Final Tester the throughput increased to 1,284,733 boards per year. This configuration is referred to as Scenario 2. The resource state graph for Scenario 2 is shown in Figure I.7. Here, by observing blocking and starvation pattern in Figure I.7, Install Cover Operator is identified as the bottleneck. To overcome this bottleneck, one more operator was assigned to Install Cover operation; this is Scenario 3 and the resource state graph for this scenario is shown in Figure I.8. Again, by observing the blocking and starvation pattern, Conformal Coat 2 is identified as the bottleneck, so, another copy of this resource is added. This is named Scenario 4; the corresponding resource state graph is shown in Figure I.9.

Additional Install Cover operator resulted in increase of 2.13% in throughput, and subsequent addition of another copy of Conformal Coat 2 resulted in increase of 6.5% in throughput. So, further experimentation was done to determine relative impact of adding capacity at these resources; Table 5.3 depicts the results of this experimentation. In Table 5.3, “FT” stands for Final Testers; “IC” stands for Install Cover Operators, “Yes” stands or addition of corresponding resource, and “No” for not adding the resource. It can be

Table 5.3: Relative impact of adding capacity at Final Tester, Install Cover, and Conformal Coat 2 for Product X Line

	FT-Yes		FT-No	
	IC-Yes	IC-No	IC-Yes	IC-No
Conformal Coat 2 -Yes	1,397,566	1,300,011	1,241,004	1,239,892
Conformal Coat 2 - No	1,312,156	1,284,733	1,233,310	1,235,055

observed that addition of copy of Final Tester results in an increase of 4% in the throughput. After the addition of Final Tester, additional copy of either Install Cover operator or Conformal Coat 2 results in slight increase in production. But, adding both Install Cover operator and Conformal Coat 2 results in approximately 9% increase in production and it was concluded that to get substantial increase in throughput, for all three resources, namely, Final Tester, Install Cover Operator, and Conformal Coat 2, capacity needs to be added.

Now, in Figure I.9, no discernible pattern of significant blocking followed by significant starvation is observed; instead, blocking and starvation is observed throughout the line. But, a remarkable spike in idle time is observed going from Paste Inspect to Placement 3. The blocking reduces and the starvation increases significantly moving from Paste Inspect to Placement 3. This indicates the direction for further experimentation. In between Paste Inspect and Placement 3 there are two multi-stage placement machines. For the multi-stage placement machines, there is no provision to track the states, so, these are shown as white bars in the resource state graph. By little experimentation, it was found that Placement 1 was one of the causes of the spike in the starvation. Another copy of this resource was added, this is Scenario 5. Addition of this resource results in a throughput increase of 3.5% over Scenario 4. Resource state graph for Scenario 5 is shown in Figure I.10.

At this stage, it can be observed from Figure I.10 that, there is no single characteristic pattern of significant blocking followed by significant starving. Instead, the starvation is observed at various locations along the line. Also, no significant spike in idle time is observed. Now, we look for resource which spends highest time in states other

than Idle and Blocked. But, due to the configuration of the line with Scenario 5, the difference between the percentages of time spent in states other than Idle and Blocked for different resources are very less. Table 5.4 depicts these percentages. With so many resources having high percentages of time spent in states other than Idle and Blocked, and the difference between these percentages for different resources being small, the benefit of adding any single resource is very less. This can be attributed to the fact that, at this point the line becomes fairly balanced. Hence, adding capacity at one of the resources on this balanced line results in negligible increase in throughput. With experimentation it was determined that, adding all thirteen resources mentioned in Table 5.4 results in an increase of 66,268 boards per year, i.e. an increase 4.6% over Scenario 5.

Table 5.4: Percentage Time Spent in States Other Than Idle and Blocked for Product X Line Scenario 5

Resource	% time in states other than Idle/Blocked	Difference
Drive Screws 1	82.71	N/A
Load Component 1	79.52	3.19
Stinger	75.73	3.78
ICT 2	74.83	0.90
LabelVerify	73.59	1.24
Vision Inspection	71.88	1.71
Fuse Insert C	70.81	1.07
Screen Printer	68.50	2.31
Paste Inspect	67.45	1.05
Drive Screws2	67.14	0.31
Fuse Inspection	66.42	0.72
Fuse Insertion B	66.39	0.03
Solder	65.51	0.88

Resource state graph for this Scenario 6 is shown in Figure I.11. Adding these many resources to get an increase of merely 4.6% is not economically feasible for the manufacturer, and hence it was concluded that achieving the target production of 1.5 million boards per year with working on a 15 shifts per week schedule is very difficult and would require adding capacity for almost one third of the line. Hence, further experimentation was focused on Scenario 5. But, to demonstrate the effectiveness of the proposed methodology, the resource state graph was further analyzed as follows.

In Figure I.11, characteristic pattern of significant blocking followed by significant starvation is observed. The resources up till Conformal Coat 2 are blocked significantly and the resources Load Pallet 2 onwards are starved for most of the time. In between these two resources there is an in-line type Thermal Cure Oven for which there is no provision for reporting states. As seen in Figure I.11, the percentage time spent in Busy state is more for Load Pallet 2 than that for Conformal Coat 2, also, the cycle time of Load Pallet 2 operation is more than that of Conformal Coat 2. Hence, this spike in idle time is attributed to the operation of Thermal Cure Oven. By experimentation it was found that increasing the speed of the Thermal Cure Oven results in reduction of the starvation of the downstream processes and blocking of the upstream processes. Increasing the speed of the Thermal Cure Oven by 60%, results in an increase of approximately 50,000 boards per year, i.e. an increase of 3.3% over Scenario 6.

As mentioned earlier, adding of the resources as done in Scenario 6 to get 4.6% increase in throughput is not economically feasible. Hence, further experimentation was done with Scenario 5 in consultation with the line managers and decision makers. The original configuration of Final Testers rendered them as the bottlenecks, because, if two

boards do not arrive simultaneously at the two Final Testers, then, one of the testers has to sit idle till the other one finishes testing. This idle time was forced due to the logic designed for the operation of Final Testers. Arranging these Final Testers to operate exactly in parallel solves the problem and yields the same throughput as that given by Scenario 2. Also, by discussion with the line managers it was observed that the cycle time for the Install Cover operation could be reduced by some amount by re-designing either the elemental tasks or the station layout. Experimentation was done to determine percentage reduction in cycle time of Install Cover operation that would yield the same throughput as adding another operator. It was found that reducing the cycle time by 10% results in the same throughput as that achieved by adding another operator. So, Scenario 5 stands for configuration where three buffers and copies of machines Conformal Coat 2 and Placement 1 is added; the Final Testers are arranged exactly in parallel and the cycle time for the Install Cover operation is reduced by 10%.

5.2 Simulation Modeling and Analysis of Product Y Line

This is existing line which was to be modified by removing one of the two parallel legs of placement machines. The manufacturer was interested in determining the impact of removing one of the placement legs. The process flow chart for Product Y line

Table 5.5: Comparison of Static and Simulation Model Results for Product Y Line

Model	Static Model	Simulation Model
Number of boards produced per year	100,174	100,152

is shown in Figure II.1 and Figure II.2. Table 5.5 shows the comparison of the throughputs given by the static and simulation models. As the throughputs predicted by the two models match closely we have evidence that the simulation model is valid. The resource state graph for the Current Configuration is shown in Figure II.3. The resources upstream to Placement 5 are blocked for significant time followed by starvation of downstream resources, which suggests that this is the bottleneck for the Current Configuration. To overcome this bottleneck, another copy of Placement 5 was added; this resulted in 4.5% increase in throughput. This is referred as Scenario 1; Figure II.4 shows the corresponding resource state graph. One of the goals of this analysis was to effectively allocate 2 buffers of size 12 boards each. From Figure II.3, it can be observed that Placement 5, which is identified as the bottleneck, spends approximately 10% of the total time in each of the states Idle and Blocked. Hence, by locating the 12 piece buffers before and after Placement 5 would increase the availability of the bottleneck resource, thereby increasing the throughput. This configuration is named as Scenario 2 which yields the same throughput as given by Scenario 1. The resource state graph for this scenario is shown in Figure II.5. Thus, by analyzing the resource state graphs two alternative strategies, namely, adding an additional copy of Placement 5 or adding the 12 piece buffers before and after Placement 5 were proposed. This is important, because, without simulation it would have been difficult for the line managers to assess the impact of adding buffers to overcome the bottleneck. As two piece buffers were already available, strategy of adding buffers was chosen. From Figure II.5, no discernable pattern of significant blocking followed by significant starving is observed. A spike in idle time is seen from Screen Printer to Placement 3. But, moving from Screen Printer to

Placement 3 the amount of blocking remains almost the same and the percentage of time spent in Busy state reduces significantly; hence the spike in the idle time cannot be attributed to the resources between Screen Printer and Placement 3. Next, we compare the time spent by the resources in states other than Idle and Blocked. Screen Printer is then identified as the next bottleneck as it spends maximum percentage of time in states other than Idle and Blocked. Addition of this resource does not result in a significant increase in throughput as there are many resources which have similar percentage of time spent in states other than Idle and Blocked. Considerably increasing the throughput would require the addition of all these resources. As the target throughput was met with Scenario 2, no further experimentation was done.

5.3 Comparison of Bottleneck Detection Methods

For the Product Y line, the proposed bottleneck detection methodology was compared with the methodology developed by Roser, Nakano & Tanaka (2001). As per their method, the states a resource can be in were divided into active states and inactive states. Idle and Blocked states were identified as inactive states, the rest of the states being active. The Current Configuration was run again for Product Y line. Using a VBA project, time stamps for all the state changes were written for each resource along with the current state and state before the state change. Then, average active duration is calculated for each resource; the resource with longest average active duration being identified as the bottleneck. The average active durations and the corresponding resources are listed in Table 5.6. Third column in Table 5.6 depicts whether the processing type is

parallel processing, batch processing or single board processing; fourth column depicts the number of parallel processes or batch size for a process depending upon whether resources work in parallel or process boards in batches. As shown in Table 5.6, ICT is the bottleneck; whereas the methodology proposed in this thesis identifies Placement 5 as the bottleneck. This is so because, the methodology as proposed in Roser, Nakano & Tanaka (2001) does not consider parallel operations and batch processing. If these are considered and the average active duration is divided by corresponding number of parallel resources or the batch size, similar results as given by the proposed methodology are obtained.

Table 5.7 shows the effective average active durations calculated in this manner. Now, as

Table 5.6: Average Active Durations for Product Y Line Current Configuration

Resource	Average active duration	Processing Type (P, B or SB)	Number of parallel processes / batch size
ICT	1.0093	P	4
PreTest	0.9342	B	5
FT	0.9161	B	5
IPMTest	0.4727	B	2
WetSeal	0.4120	P	2
CoverLoader	0.4027	P	2
PDCLoader	0.3840	B	2
ConfCoatApp	0.3803	P	2
Placement 5	0.3232	SB	1
ScreenPrinter	0.3038	SB	1
SideConnInsert	0.2938	SB	1
TopConnInsert	0.2937	SB	1
ConnPotting	0.2841	SB	1
Screw Driver1	0.2818	SB	1
PasteVision	0.2809	SB	1
LaserBarCode	0.2175	SB	1

shown in Table 5.7, Placement 5 is identified as the bottleneck, which is the same as the one identified by the methodology proposed here. Similar procedure was followed for Scenario 1 for Product Y line. The resources and corresponding effective average active durations are listed in Table 5.8. As shown in Table 5.8, Screen Printer is identified as the bottleneck, which matches with the results obtained using the proposed methodology. It can be observed in both the cases above, that the resource that spends highest percentage of time in states other than Idle and Blocked is identified as the bottleneck. It would be interesting to see how this method performs in a scenario like Scenario 1 for the

Table 5.7: Effective Average Active Durations for Product Y Line Current Configuration

Resource	Effective average active duration
Placement 5	0.3232
Screen Printer	0.3038
SideConnInsert	0.2938
TopConnInsert	0.2937
ConnPotting	0.2841
Screw Driver1	0.2818
Paste Vision	0.2809
ICT	0.2523
IPM Test	0.2363
LaserBarCode	0.2175
Wet Seal	0.2060
CoverLoader	0.2014
PDCLoader	0.1920
ConfCoatApp	0.1901
PreTest	0.1868
FT	0.1832

Product X Line, where Final Testers are identified as bottlenecks and which are not the resources that spend highest percentage of time in states other than Idle and Blocked.

However, this analysis is beyond the scope of this thesis.

5.4 Simulation Modeling and Analysis of Product Z Line

Figure III.1 shows the process flow chart for Product Z line. Manufacturer was interested in assessing the impact of moving from a 17 shifts per week schedule to a 15 shifts per week schedule. Part of the analysis performed is presented here. As this is an

Table 5.8: Effective Average Active Durations for Product Y Line Scenario 1

Process	Effective average active duration
ScreenPrinter	0.3036
TopConnInsert	0.2956
SideConnInsert	0.2937
ScrewDriver	0.2846
ConnPotting	0.2840
PasteVision	0.2812
ICT	0.2417
IPMTest	0.2365
LaserBarCode	0.2177
WetSeal	0.2065
CoverLoader	0.2022
PDCLoader	0.1924
PreTest	0.1904
ConfCoatApp	0.1902
FT	0.1839
TabBend	0.1814

existing line, the data pertaining to this line was collected from the historical databases. Then, static model was developed for this line, followed by developing, verifying, and validating the simulation model. Table 5.9 compares the throughputs given by the static model Scenario 1 and the corresponding simulation model. As seen from Table 5.9, the throughput given by the two models closely matches, and hence we have a strong evidence that the simulation model is valid.

Figure III.2 shows the resource state graph for the Current Configuration for the Product Z Line. All the resources upstream to Packout are blocked for significant amount of time and Packout, which is the last operation on the line, is starved for most of the time. This indicates that the bottleneck is located towards the end of line between Final Testers and Packout. From the simulation results it was observed that, the operator responsible for reworking the rejects from the in-circuit testers and the functional testers (not shown in resource state graph as it is in rework loop) has utilization of more than 98%. The rejects from the ICT and FT are carried over a common conveyor to rework station. When this conveyor becomes full, it blocks all the upstream processes. Until this conveyor clears up, no processing takes place; which causes blocking of the upstream resources and starvation of the downstream resource, Packout. Another operator was assigned for reworking the rejects from ICT and FT. This is referred to as Scenario 1. Addition of another operator resulted in 100% increase in the production. The resource state graph for this configuration is shown in Figure III.3.

From Figure III.3, blocking of resource upstream to ICT and starving of the downstream resources can be observed, indicating that ICT is the bottleneck for this configuration. A copy of ICT was added to overcome this bottleneck. The new

Table 5.9: Comparison of Static and Simulation Model Results for Product Z Line

Model	Simulation Model	Static Model
Number of boards produced per year	600,330	600,331

configuration is called Scenario 2. By adding another ICT, the throughput increased by 3.1%. The corresponding resource state graph is shown in Figure III.4.

Next, as discussed in Section 4.5, operator allocation based on the resource state graph is demonstrated for this line.

Figure III.5 shows resource state graph for Scenario 3, which has the same line configuration as Scenario 2 except that, the operator assignment was changed arbitrarily to show the ability of the resource state graph to the use of operator interference for operator allocation. In Scenario 3, there are a total of six operators, the throughput remaining at same level as that of Scenario 2. The operators responsible for repair or replenishment tasks for various resources are shown on each the stacked bar for the resource states. The group of machines allocated to an operator is called the home work area for that operator. The method involves observing the operator interference, represented by Failure and Exhaust states in the resource state graph, so as to make better decisions regarding operator allocation or re-allocation.

From Figure III.5 it can be observed that the operator interference in home work areas of operators 1, 2, 5, and 6 is negligible. But, home work areas of operators 3 and 4 experience some operator interference. So, in the next scenario, i.e. Scenario 4, the work areas of operators 1 and 2 are combined into one work area and operator 1 is assigned to this new work area. Similarly, work areas of operators 5 and 6 are combined into single work area and operator 5 is assigned to this new work area. Thus, the total number of

operators was reduced to 4 and throughput still remained at the same level. The resource state graph for Scenario 4 is shown in Figure III.6.

It can be observed from Figure III.6 that, for work areas of operators 1 and 5, the operator interference is negligible; whereas work areas of operators 3 and 4 experience some operator interference. Next, operator re-allocation was done by combining one work area with some interference and the other with negligible operator interference. Thus, the work areas of operators 1 and 3 were combined to form a new work area, and operator 1 was assigned to this work area. Similarly, work areas of operators 4 and 5 were combined to form a new work area and operator 4 was assigned to this newly formed work area. In this step, the total number of operators was reduced to 2 and the throughput still remained at the same level. This is Scenario 5 and the corresponding resource state graph is shown in Figure III.7. As observed from Figure III.7, both the work areas experience noticeable operator interference and further reducing the number of operators can result in decrease in throughput. A confirmation run was made and it was found that reducing the number of operators to 1 results in 9.5% decrease in throughput.

Next, Figure III.8 shows resource state graph for Scenario 7, which has the same line configuration as that of Scenario 6 with a total of two operators, but, the composition of the two work areas is changed from that of Scenario 6. This resulted in 3.5% decrease in throughput. It can be observed from Figure III.8 that, the operator interference in home work area of operator 1 is significant. But, the operator interference in home work area of operator 3 is much less as compared to that in home work area of operator 1. In such a situation, operator cross training can be proposed so that both the operators can do the

tasks in either of the work areas, resulting in reduction in operator interference. In next scenario, i.e. Scenario 8, the two work areas were combined into a new single work area and two cross-trained operators were assigned to this work area. This was implemented in simulation model by assigning operator 1 with a capacity of 2 to the newly formed work area. This resulted in 3.5% increase in throughput.

Thus, it is shown that, analysis of the resource state graph can be successfully used to guide the managers in operator allocation or re-allocation. The operator allocations considered here were arbitrarily designed and effect of operator walk times was not taken into consideration while combining the work areas, but, nevertheless this method can be used in close consultation with line managers by discussing the feasibility of combining particular work areas.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This thesis studied the problem of simulation modeling and analysis of PCB assembly lines. This is an important problem as line managers and decision makers of PCB assembly lines often face situations like designing configuration of entire line for new product, altering line configuration to meet the changes in the demand, changes in the policy decisions like number of work shifts in a week. The task of the line managers and decision makers is to assess the ability to meet the target production quantities/rates and accordingly decide upon allocation/re-allocation or addition of resources like equipments, operators, and buffers. Analytical models available need simplifying assumptions, rendering the model unrepresentative of the real line behavior. Whereas simulation studies have focused on some isolated problem like scheduling policies, bottleneck detection or buffer allocation and do not capture all the complex interactions resulting from dynamics of the line behavior.

In this thesis, a six-step methodology was proposed for simulation modeling and analysis of PCB assembly lines capturing complex interactions between various resources. PCB assembly template developed in Mukkamala, Smith & Valenzuela (2003) and Mukkamala (2003) was enhanced and used for modeling in Arena 7.01. Use of PCB assembly template simplified model development, verification, and validation for PCB

assembly lines. Importance of ability to distinguish between seven different resource states is discussed. A resource state graph is developed using the resource state statistics provided by template modules. This resource state graph gives significant insight into the complex interaction between resources within a manufacturing system. This graph is then analyzed by observation of specific patterns to identify the problem areas, bottlenecks, and constraints and improvement strategies are developed accordingly. We strongly believe that proper application of this methodology leads to developing improvement strategies which could fetch significant improvement in terms of throughput and operator allocation. The ability of the method to identify bottlenecks and constraints was demonstrated with the help of three case studies. It was shown that the method successfully identifies the bottlenecks or constraints even with complex subsystem configuration; for example, Final Testers were identified as bottleneck in Product X line, where other bottleneck detection methods like highest utilization, longest queue length, and longest average active duration would have failed to identify the true bottleneck. Also, the ability of the resource state graph to expose the complex interactions like operator interference was demonstrated for making decisions pertaining to operator allocation/re-allocation or operator cross-training. Thus, the resource state graph proves to be an excellent tool for analysis of a manufacturing system and which does not involve complexities of math or exhaustive statistical analysis of the results.

An area which needs further investigation is buffer allocation. It was observed from the case studies that the difference between the upper bound on throughput as obtained using the static model and the throughput as obtained using simulation model is mainly due to capacitated buffers. Buffer allocation in this thesis was mainly based on

allocating buffers by identifying the bottleneck and observing its blocking and starvation; thereby improving the availability of the constraining resource. Due to the size of the lines under consideration, existence of subsystems like serial-layout, parallel-operation and even more complex ones like Final Testers, operator interference, machines with non-identical processing times, times to failure and time to repair, buffer allocation methods available cannot be applied directly to the PCB lines under consideration. Further investigation needs to be done to properly adapt the available buffer allocation methods and develop rules-of-thumb for allocating buffers so as to regain part of the lost capacity due to capacitated buffers.

The specifics provided in this thesis pertain to PCB assembly lines. But, with proper adaptation of the template, the methodology can be equally applicable to serial production lines other than the PCB assembly lines. This could be another potential extension of the work described in this thesis.

Additionally, work in this thesis can be extended to facilitate an environment to simplify the simulation model development and modification. The static model as described in this thesis has all the input data required for the template modules. This static model can be modified to include the data on the conveyors that link these processes. A field could be included for each process which defines the type of the subsystem configuration, e.g. exact parallel-operation, serial-layout, parallel operation or more complex configuration like that of Final Testers described for Product X line. Another field will define the subsystem selection logic. Then, a VBA project can be used to generate the simulation model. A provision can be made to switch the failures and exhausts on and off. Thus, the static model spreadsheet can work in effect as the *line*

000definer; the VBA project being *code generator* as described in Farrington, Rogers, Schroer et al (1995). Similarly, user interface can be embedded in static model, which will enable setting the simulation parameters and running of the simulation model without user actually interacting with the simulation model. The idea is to enable managers with working knowledge of simulation develop and modify simulation models with minimal efforts and to reduce the repetitive model development and modification time. The generation of the resource state graph has already been automated; by automating the generation of simulation model, modeling efforts required will be limited to avoiding potential deadlock situations and complex flow logic and valuable time can be devoted in analyzing the resource state graph

REFERENCES

- Abdul-Kader, W. and Gharib, A. (2002). Capacity estimation of a multi-product unreliable production line. *International Journal of Production Research*, 40(18), 4815-4834.
- Brown, E. (1988). IBM combines rapid modeling technique and simulation to design PCB factory-of-the-future. *Industrial Engineering*, 20(6), 23-26.
- Capillo, C. (1990). *Surface Mount Technology: materials, processes, and equipment*. McGraw-Hill Publishing Company.
- Chiang Shu-Yin, Kuo Chih-Tsung, and Meerkov, S.M. (2000). DT-Bottlenecks in serial production lines: theory and application. *IEEE Transactions on Robotics and Automation*, 16(5), 567-580.
- Chiang Shu-Yin, Kuo Chih-Tsung, and Meerkov, S.M. (2001). c-Bottlenecks in serial production lines: identification and application. *Mathematical problems in engineering*, 7, 543-578.
- Conway, R., Maxwell, M., McClain, J.O., and Thomas, J.L. (1988). The role of work-in-process inventory in serial production lines. *Operations Research*, 36(2), 229-241.
- Dallery, Y. and Gershwin, S.B. (1992). Manufacturing flow line systems: a review of models and analytical results. *Queuing Systems*, 12, 3-94.

- Doss, D.L. and Ülgen, O.M. (2004). A case for generic, custom-designed simulation applications for material handling and manufacturing industries. *Brooks Automation's Worldwide Automation Symposium*, 1-5.
- D'Souza, R.C. (2004) *A throughput-based technique for identifying production system bottlenecks*. M.S. thesis. Mississippi, MI: Mississippi State University.
- Enginarlar, E., Li, J., Meerkov, S.M., and Zhang, R.Q. (2002). Buffer capacity for accommodating machine downtime in serial production lines. *International Journal of Production Research*, 40(3), 601-634.
- Farrington, P.A., Rogers, J.S., Schroer, B.J., Swain, J.J., and Evans, J.J. (1995). Simulation environment for electronics manufacturing. *Proc. of 1995 Winter Simulation Conference*, 917-924.
- Gebus, S., Martin, O., Soulas, A., and Juuso, E. (2004). Production Optimization on PCB assembly lines using discrete-event simulation. Technical Report, A No 24. Linnanmaa, Finland: University of Oulu.
- Goss, G.B. (2000). *Measuring machine interference to evaluate an operator cross-training program*. M.S. thesis. Massachusetts, MA: Massachusetts Institute of Technology.
- Govil, M.K. and Fu, M.C. (1999). Queuing theory in manufacturing: a survey. *Journal of Manufacturing Systems*, 18(3), 214-240.
- Harris, J.H. and Powell, S.G. (1999). An algorithm for optimal buffer placement in reliable serial lines. *IIE Transactions*, 31, 287-302.

- Heavy, C., Papadopoulos, H.T., and Browne, J. (1993). The throughput rate of multistation unreliable production lines. *European Journal of Operational Research* 68(1), 69-89.
- Hegde, G.G., Ramamurthy, K., Tadikamala, P.R., and Kekre, S. (1988). Capacity choice, work-in-process inventory and throughput: a simulation study. *Proc. of 1988 Winter Simulation Conf.*, 662-666.
- Hillier, F.S. and So, K.C. (1991). The effect of machine breakdowns and interstage storage on the performance of production line systems. *International Journal of Production Research*, 29(10), 2043-2055.
- Hollomon, James K. Jr. (1989). *Surface mount technology for PC board design*. Indianapolis: Howard W. Sams & Company.
- Huang, S.H., Dismukes, J.P., Shi, J., Su, Q., Razzak, M.A., Bodhale, R., and Robinson, E.D. (2003). Manufacturing productivity improvement using effectiveness metrics and simulation analysis. *International Journal of Production Research*, 41(3), 513-527.
- Huettner, C.M. and Steudel, H.J. (1992). Analysis of manufacturing system via spreadsheet analysis, rapid modeling, and manufacturing simulation. *International Journal of Manufacturing Systems*, 30(7), 1699-1714.
- Ingalls, R.G. and Eckersley, C. (1992). Simulation issues in electronics manufacturing. *Proc. of 1992 Winter Simulation Conference*, 861-864.
- Jeong, Myong-Kee, Perry, M., and Zhou Chen. (2005). Throughput gain with parallel flow in automated flow lines. *IEEE Transactions on Automation Science and Engineering*, 1-3.

- Kamath, M. (1999). Recent development in modeling and performance analysis tools for manufacturing systems. In: S. B. Joshi and J. S. Smith, *Computer Control of Flexible Manufacturing Systems*, (pp.231-263). Chapman and Hall.
- Kavusturucu, A. and Gupta, S.S. (1999). Analysis of manufacturing flow lines with unreliable machines. *International Journal of Computer Integrated Manufacturing*, 12(6), 510-524.
- Kiran, Ali S., Kaplan, Celal, and TamerUnal, A. (1993). Simulation of electronics manufacturing and assembly operations. *Proc. of the 1993 Winter Simulation Conf.*, 773-779.
- Kotcher, R.C. (2001). How “overstaffing” at bottleneck machines can unleash extra capacity. *Proc. of the 2001 Winter Simulation Conference*, 1163-1169.
- Kouikoglou, V.S. (1994). Discrete event modeling and optimization of unreliable production lines with random rates. *IEEE Transaction on Robotics and Automation*, 10(2), 153-159.
- Kuo, C.-T., Lim, J.-T., and Meerkov, S.M. (1996). Bottlenecks in serial production lines: a system theoretic approach. *Mathematical problems in engineering*, 2, 223-276.
- Law, A.M. and McComas, M.G. (1997). Simulation of manufacturing systems. *Proc. of 1997 Winter Simulation Conf.*, 86-89.
- Law, A. M. and Kelton, D. W. (2000). *Simulation Modeling and Analysis*. McGraw-Hill Companies, Inc.
- Lawrence, S.R. and Buss, A.H. (1994). Shifting production bottlenecks: causes, cures, and conundrums. *Production and operations management*, 3(1), 21-37.

- Mahadevan, S. (2004). *Automated simulation analysis of overall equipment effectiveness metrics*. M.S. Thesis. Cincinnati, CN: University of Cincinnati.
- Miller, S. and Pegden, D. (2000). Introduction to manufacturing simulation. *Proc. of 2000 Winter Simulation Conference*, 63-66.
- Mukkamala, P.S., Smith, J.S., and Valenzuela, J.F. (2003). Designing reusable simulation modules for electronics manufacturing systems. *Proc. of 2003 Winter Simulation Conf.*, 1281-1289.
- Papadopoulos, H.T. and Heavy, C. (1996). Queuing theory in manufacturing systems analysis and design: a classification of models of production and transfer lines. *European Journal of Operations Research*, 92, 1-27.
- Papadopoulos, H.T. and Vidalis, M.I. (2001). A heuristic algorithm for the buffer allocation in unreliable unbalanced production lines. *Computers and Industrial Engineering*, 41, 261-277.
- Phillis, Y.A., Kouikoglou, V.S., Sourlas, D. and Manousiouthakis, V. (1997). Design of serial production systems using discrete event simulation and nonconvex programming techniques. *International Journal of Production Research*, 35(3), 753-766.
- Roser, C., Nakano, M., and Tanaka, M. (2001). A practical bottleneck detection method. *Proc. of 2001 Winter Simulation Conf.*, 949-953.
- Roser, C., Nakano, M., and Tanaka, M. (2002). Shifting bottleneck detection. *Proc. of 2002 Winter Simulation Conference*, 1079-1086.
- Roser, C., Nakano, M., and Tanaka, M. (2002). Throughput sensitivity analysis using a single simulation. *Proc. of 2002 Winter Simulation Conference*, 1087-1094.

- Roser, C., Nakano, M., and Tanaka, M. (2003). Buffer allocation model based on a single simulation. *Proc. of 2003 Winter Simulation Conference*, 238-1246.
- Smith, J.S. (2003). A survey of the use of simulation for manufacturing systems design and operation. *Journal of Manufacturing Systems*. 22(2), 157-171.
- Systems Modeling. ARENA Professional Edition Reference Guide. Systems Modeling Corporation, Sewickley, PA, 2003.
- Tempelmeier, H. and Bürger, M. (2001). Performance evaluation of unbalanced flow lines with general distributed processing times, failures and imperfect production. *IIE Transactions*, 33, 292-302.
- Tempelmeier, H. (2003). Practical considerations in the optimization of flow production systems. *International Journal of Production Research*, 41(1), 149-170.

APPENDICES

APPENDIX I
PROCESS FLOWCHART AND RESOURCE STATE GRAPHS FOR PRODUCT
X LINE

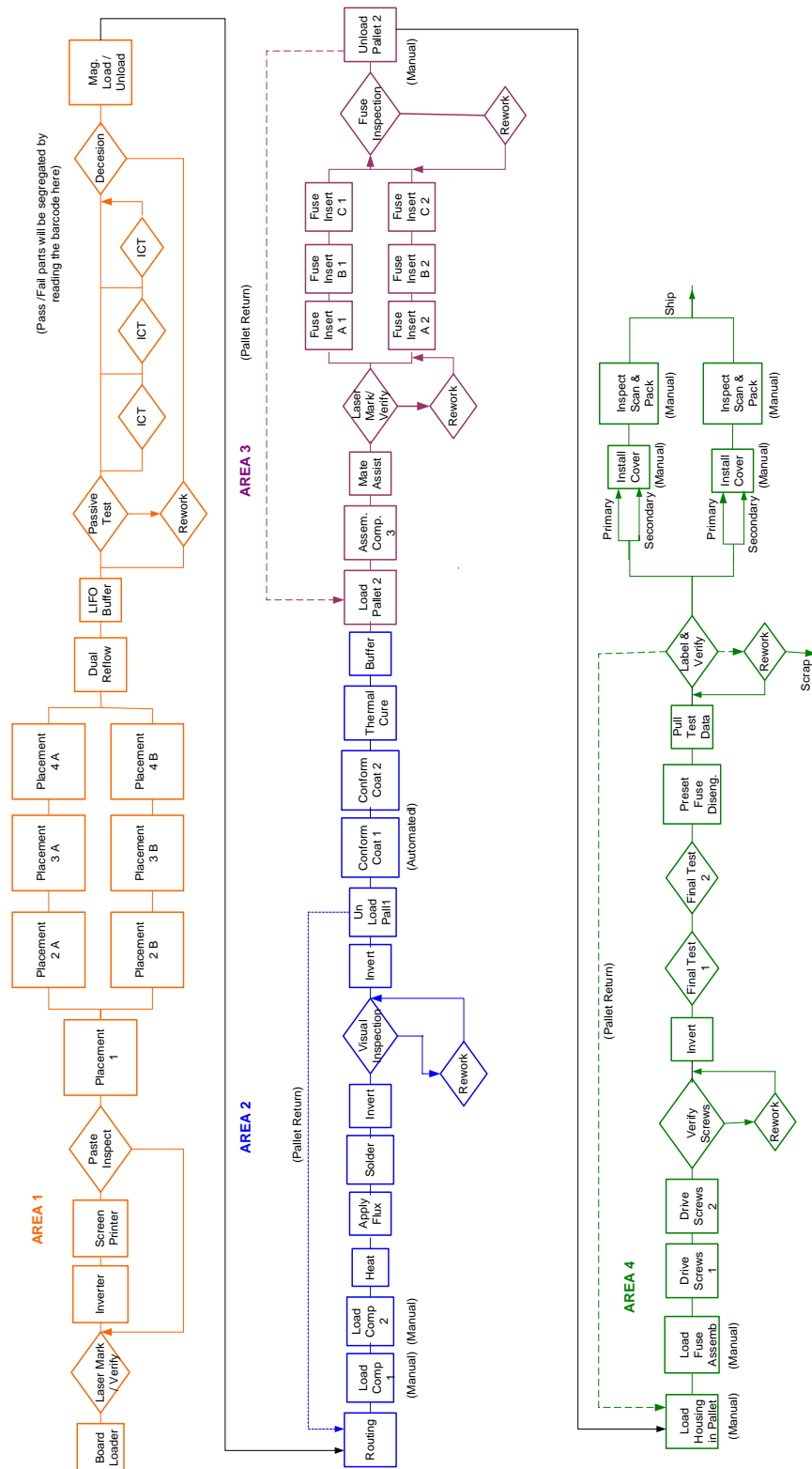


Figure I.1: Product X Line - Process Flow Chart

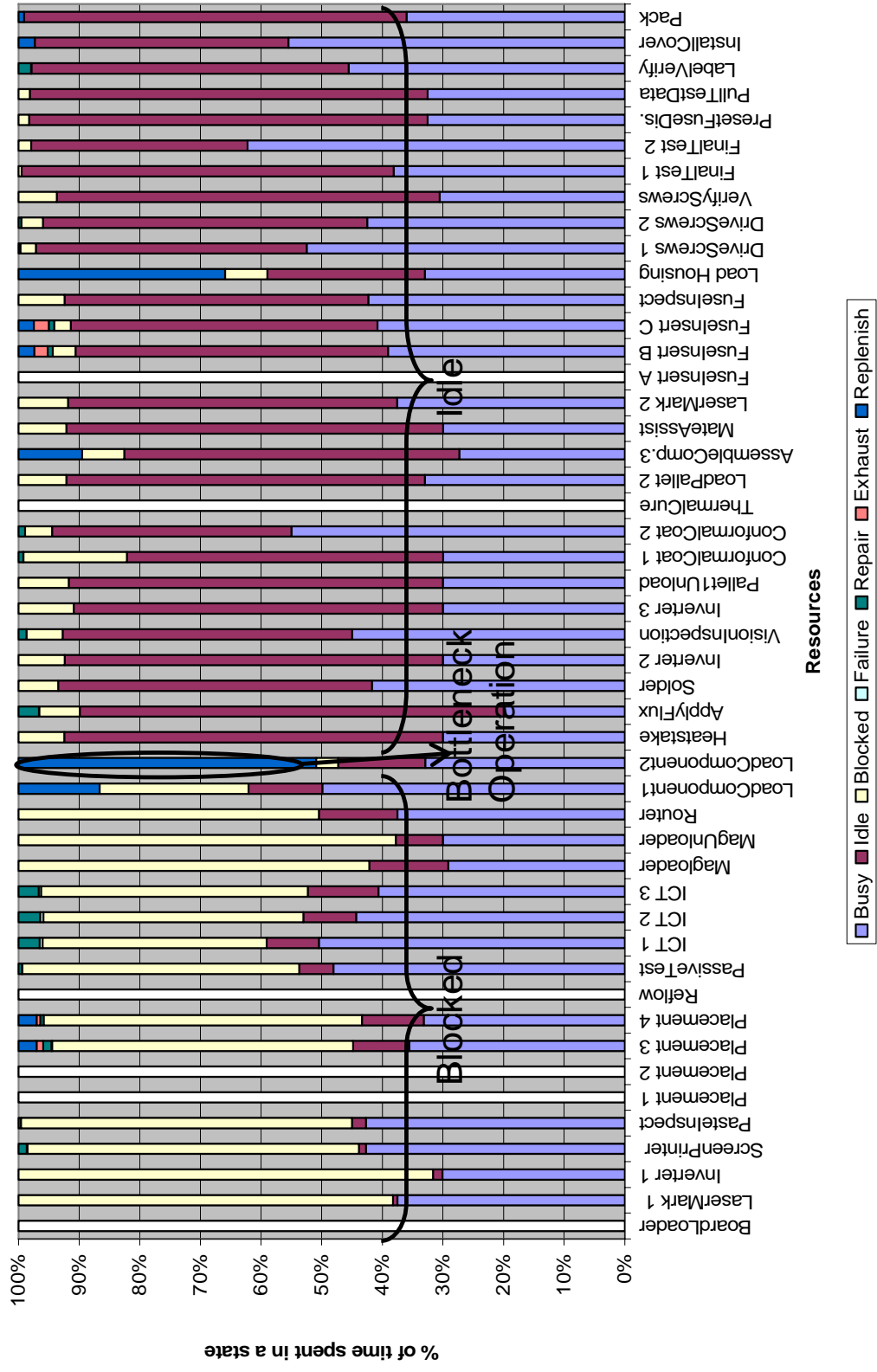


Figure I.2: Product X Line - Resource State Graph for Current Configuration

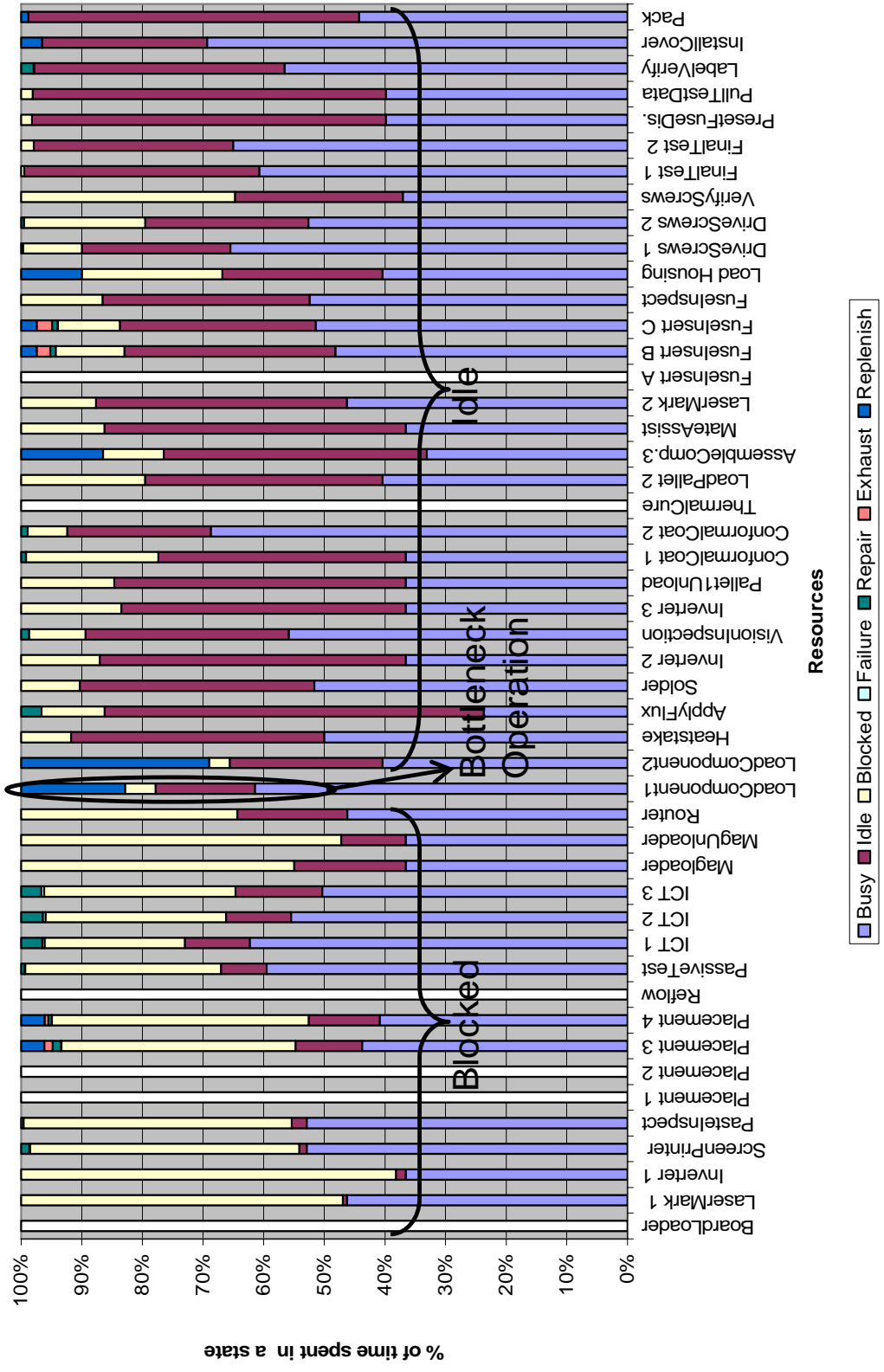


Figure I.3: Product X Line - Resource State Graph for Alternative 1

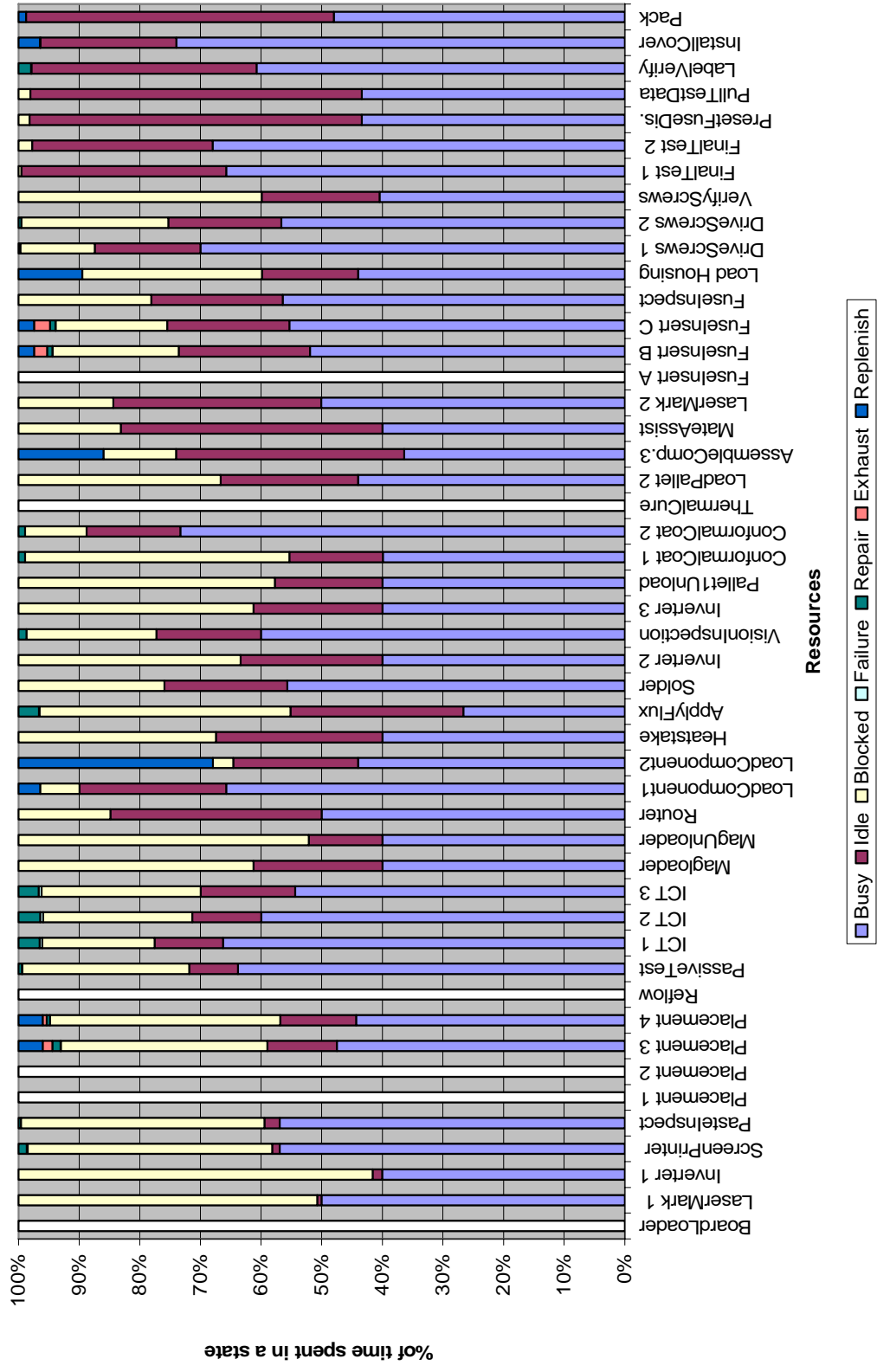


Figure I.4: Product X Line - Resource State Graph for Alternative 2

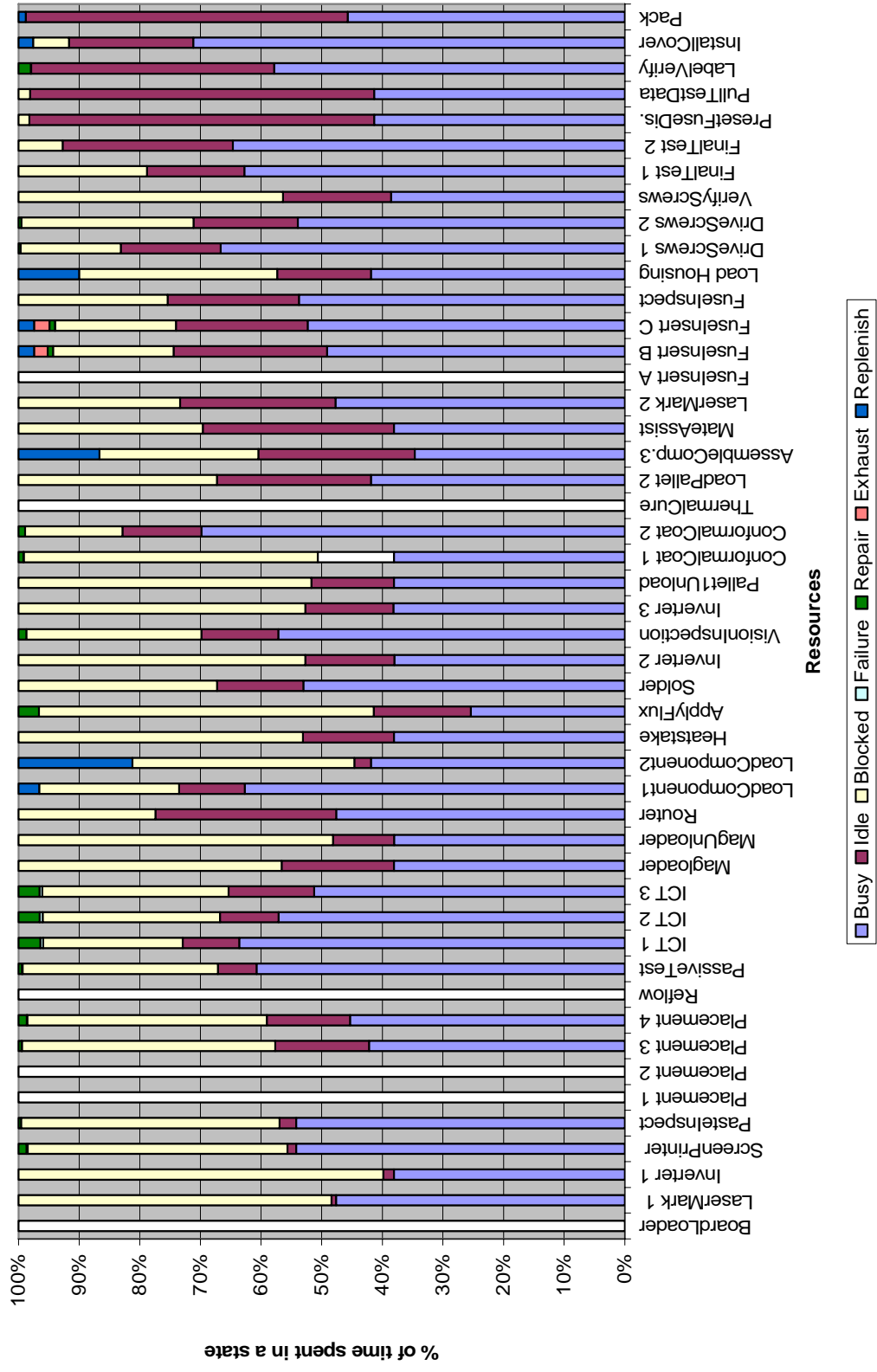


Figure I.5: Product X Line - Resource State Graph for Current Configuration 1

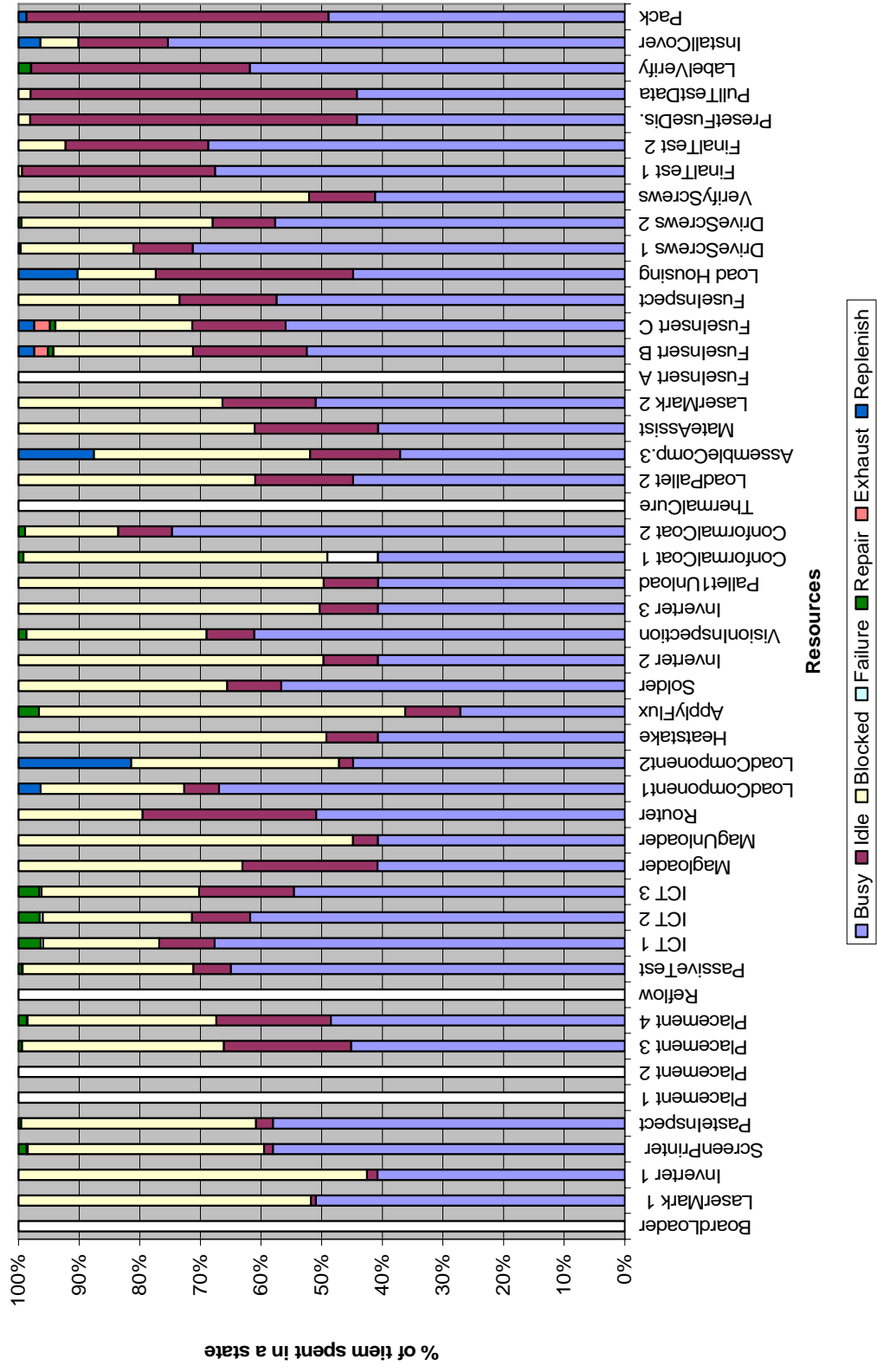


Figure I.6: Product X Line - Resource State Graph for Scenario 1

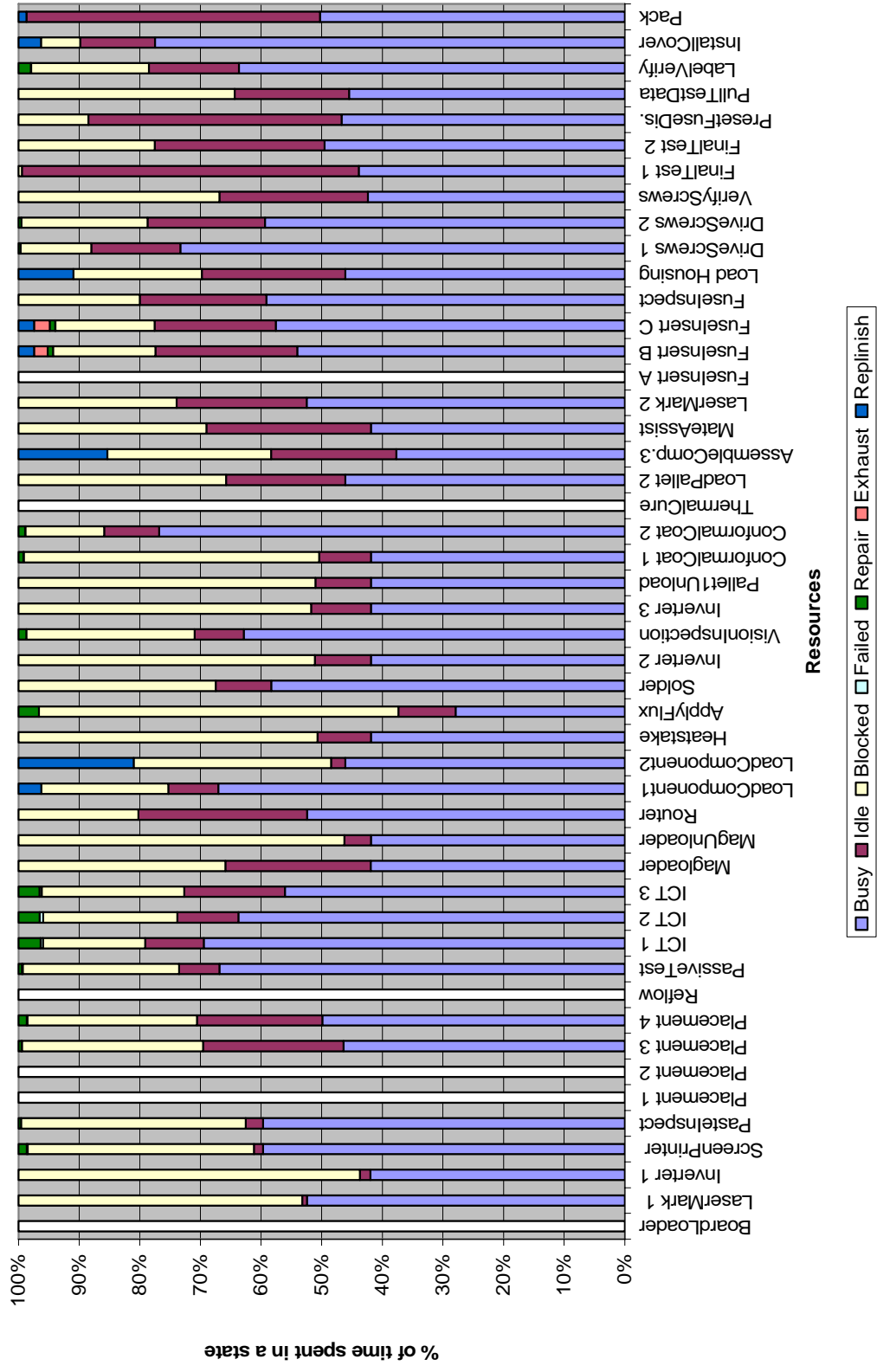


Figure I.7: Product X Line - Resource State Graph for Scenario 2

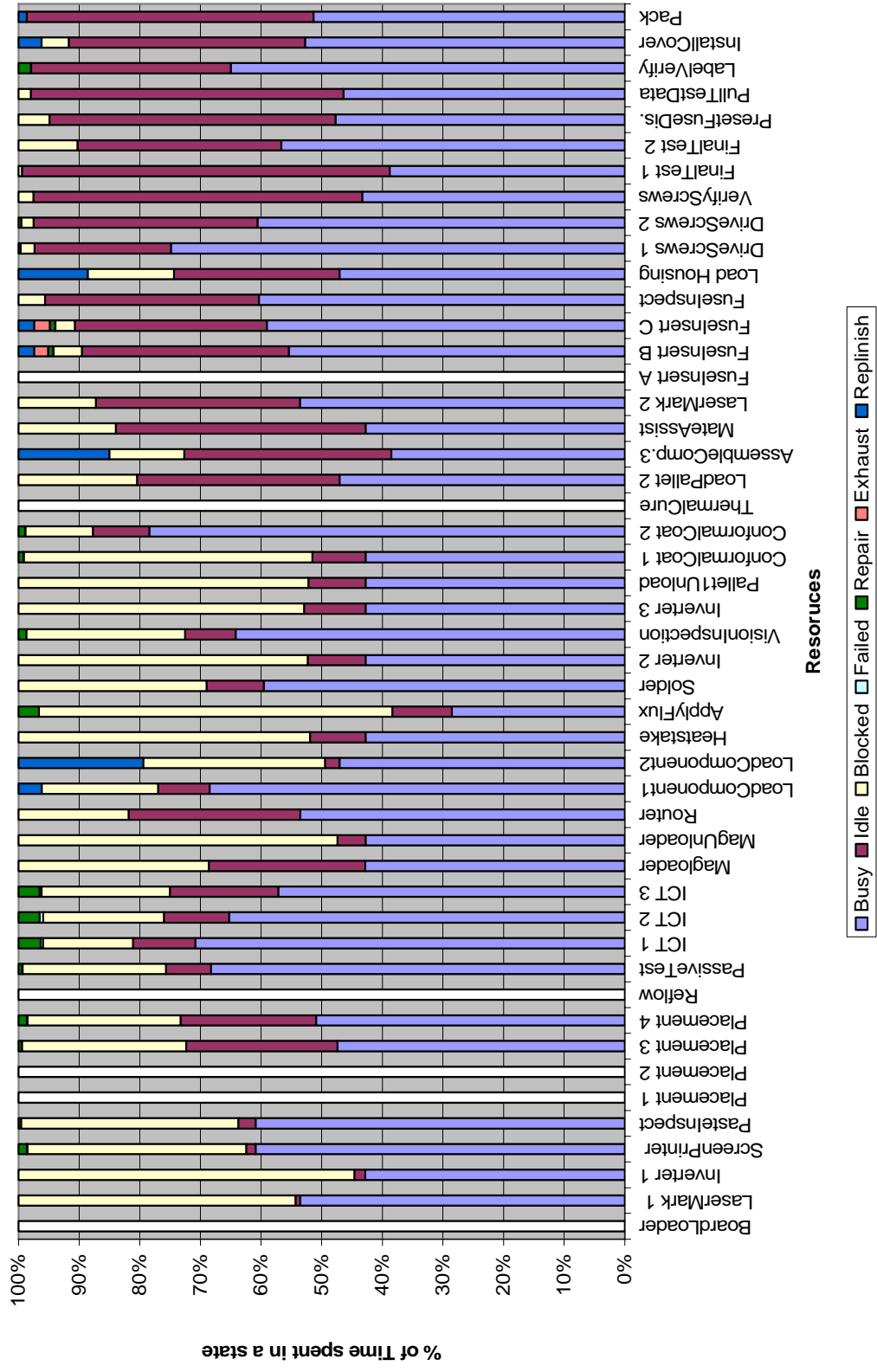


Figure I.8: Product X Line - Resource State Graph for Scenario 3

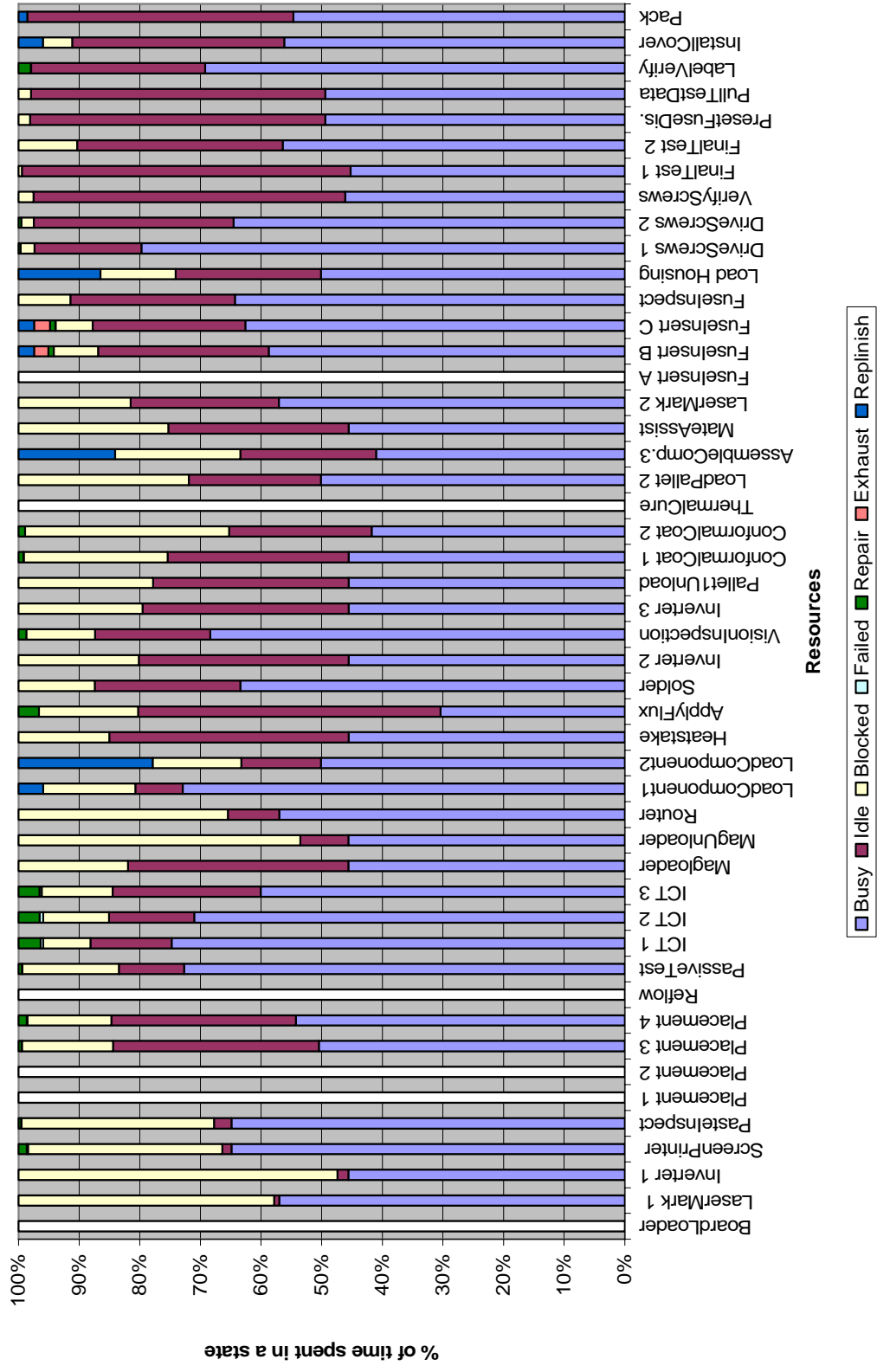


Figure I.9: Product X Line - Resource State Graph for Scenario 4

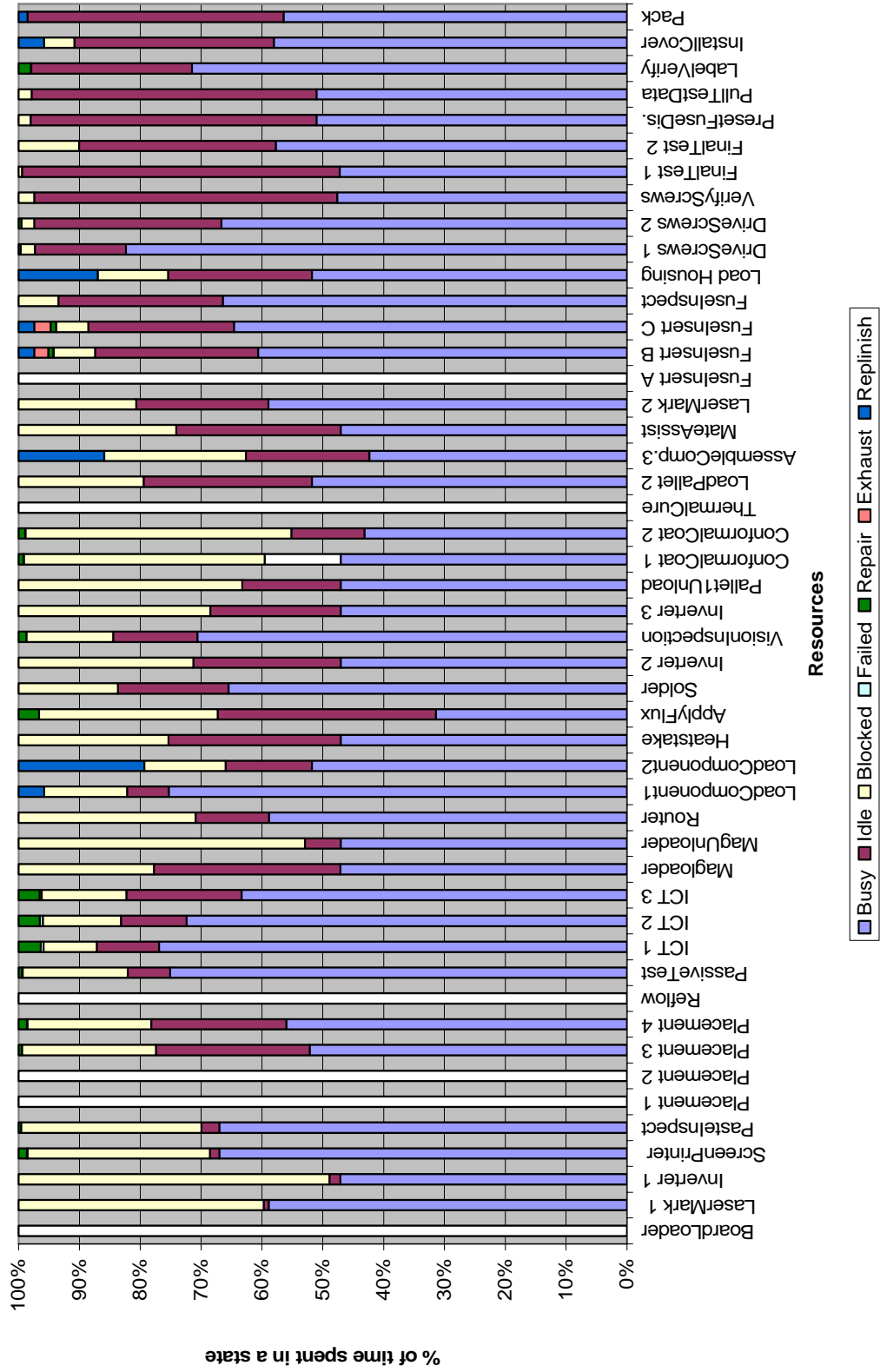


Figure I.10: Product X Line - Resource State Graph for Scenario 5

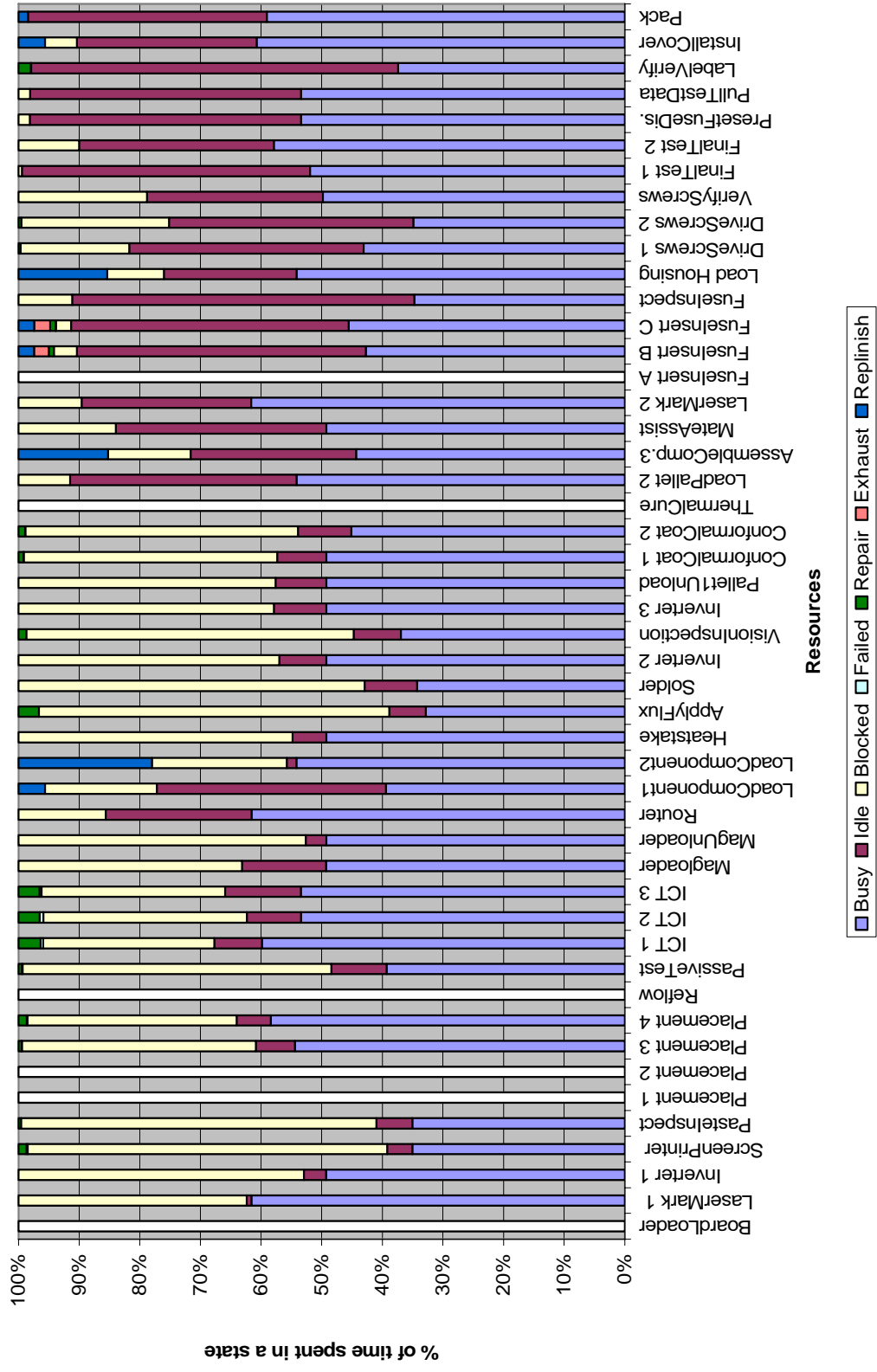


Figure I.11: Product X Line - Resource State Graph for Scenario 6

APPENDIX II
PROCESS FLOWCHART AND RESOURCE STATE GRAPHS FOR PRODUCT
Y LINE

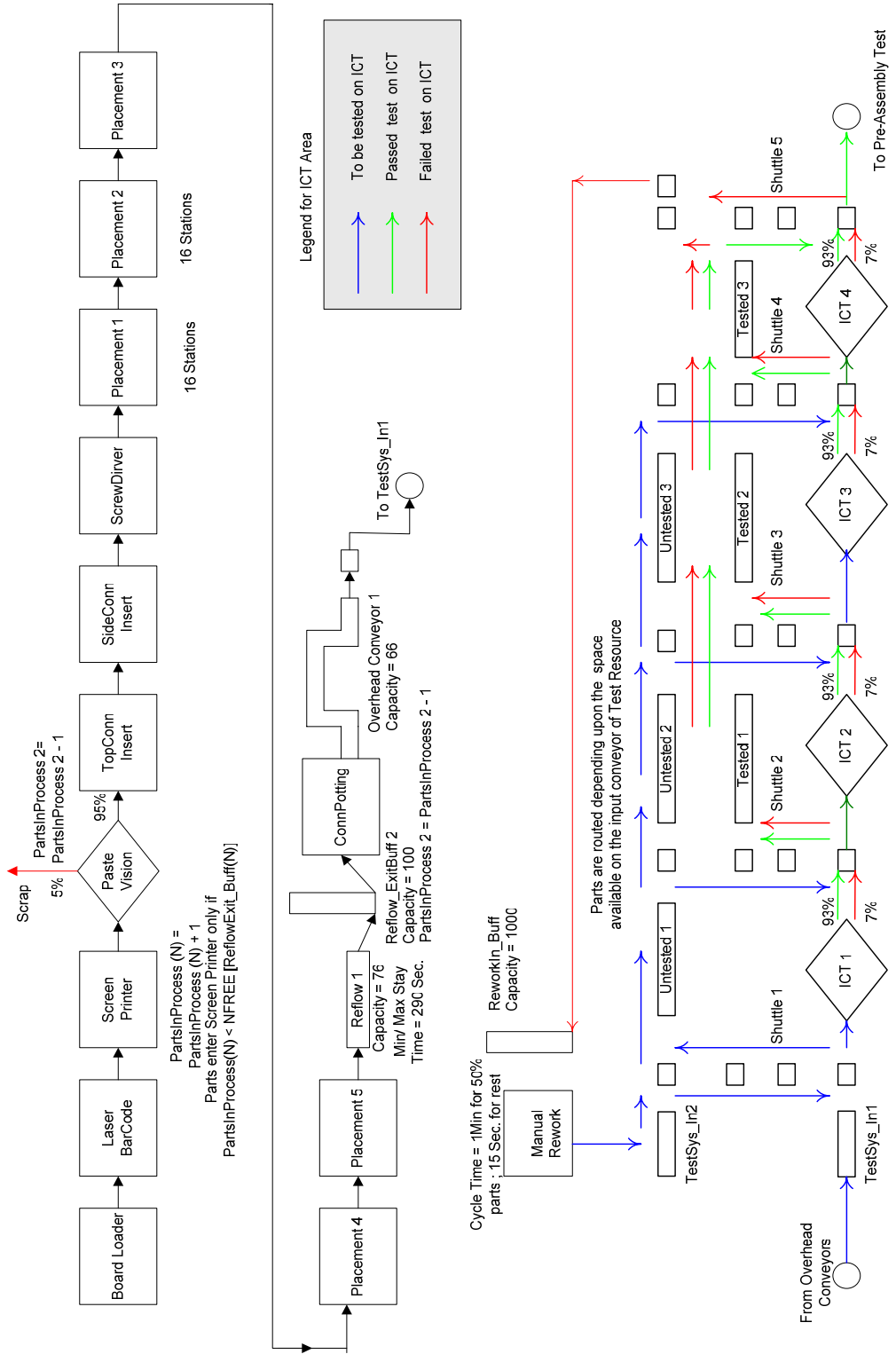


Figure II.1: Product Y Line - Process Flowchart (Contd.)

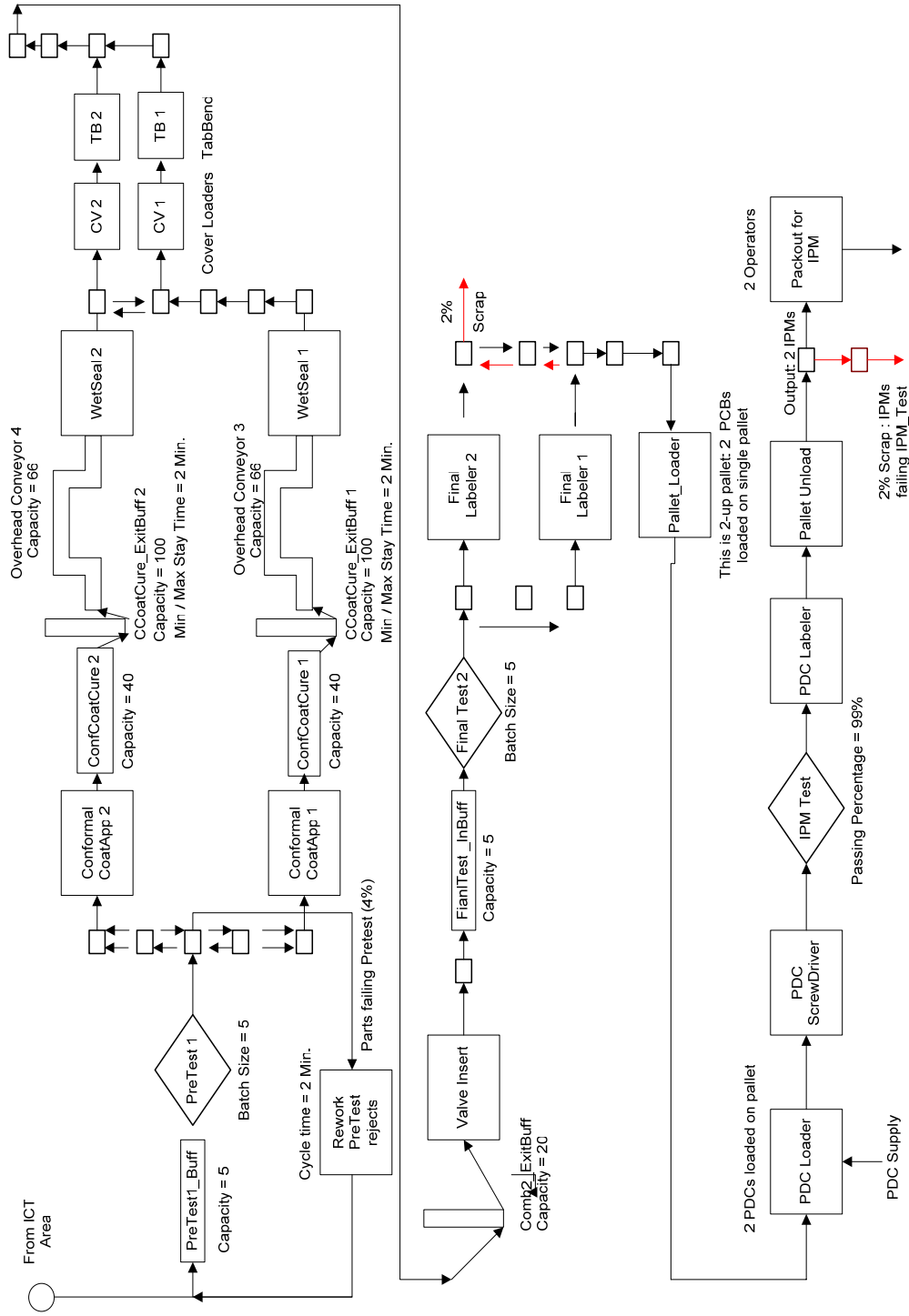


Figure II.2: Product Y Line - Process Flowchart

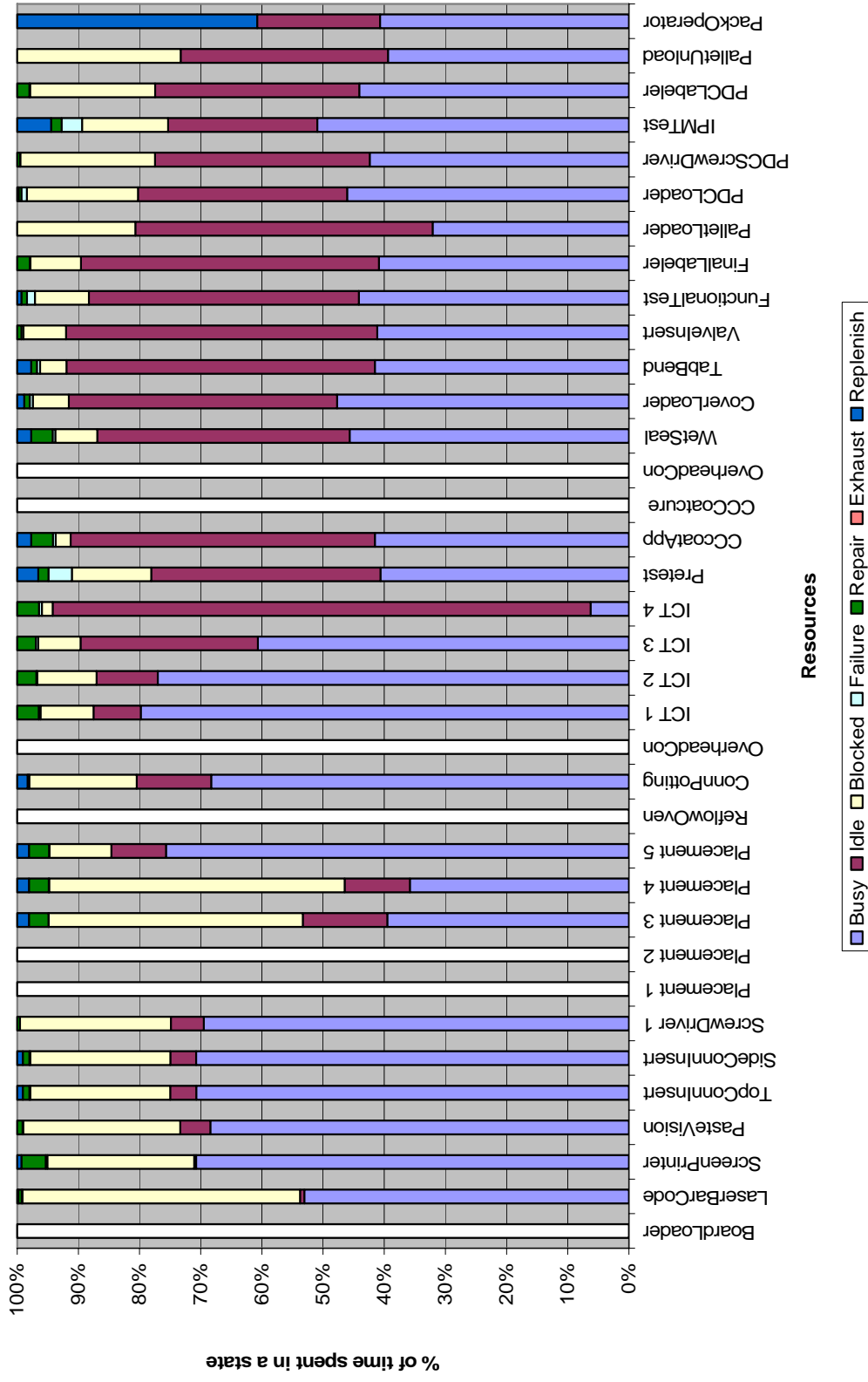


Figure II.3: Product Y Line – Resource State Graph for Current Configuration

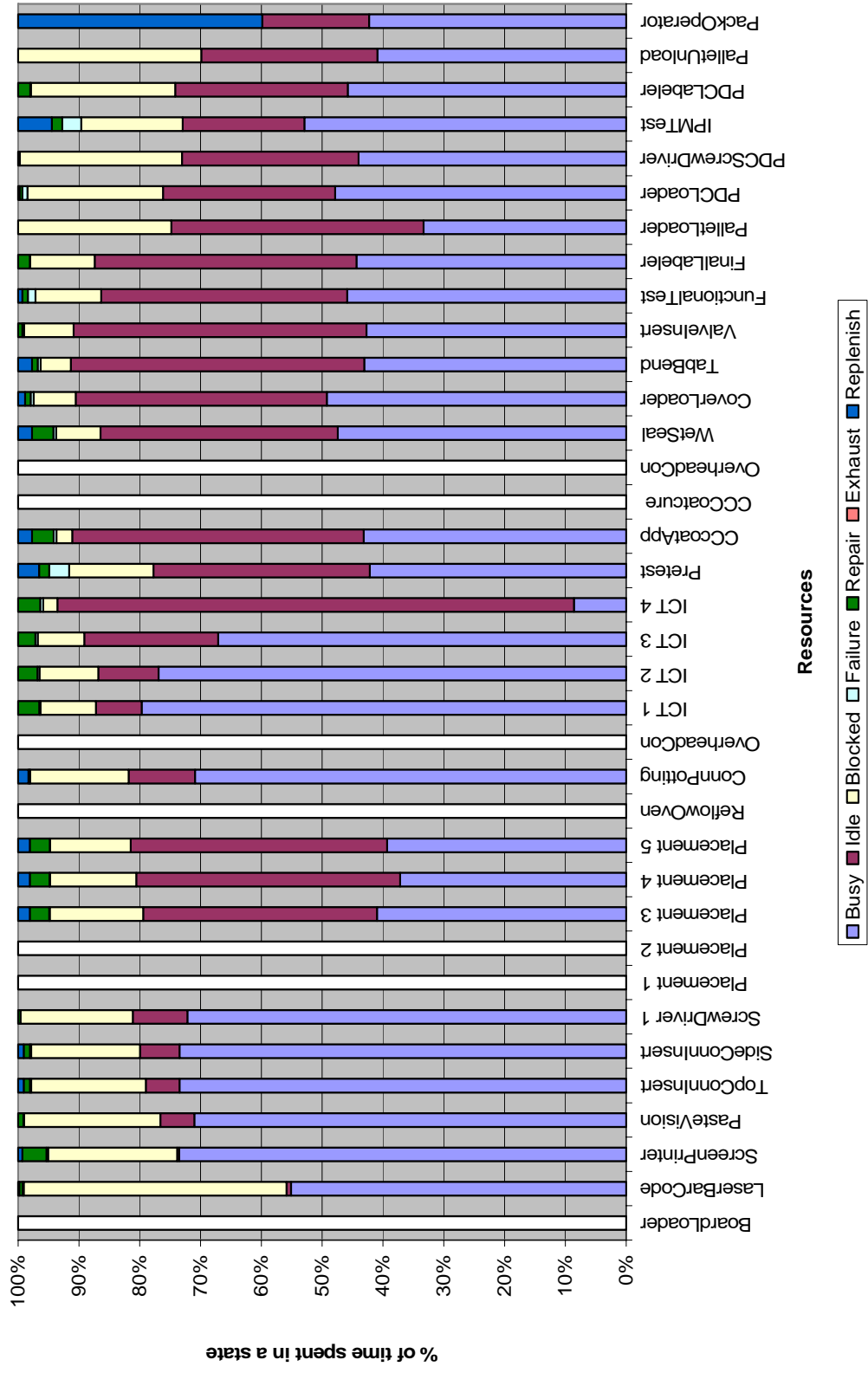


Figure II.4: Product Y Line - Resource State Graph for Scenario 1

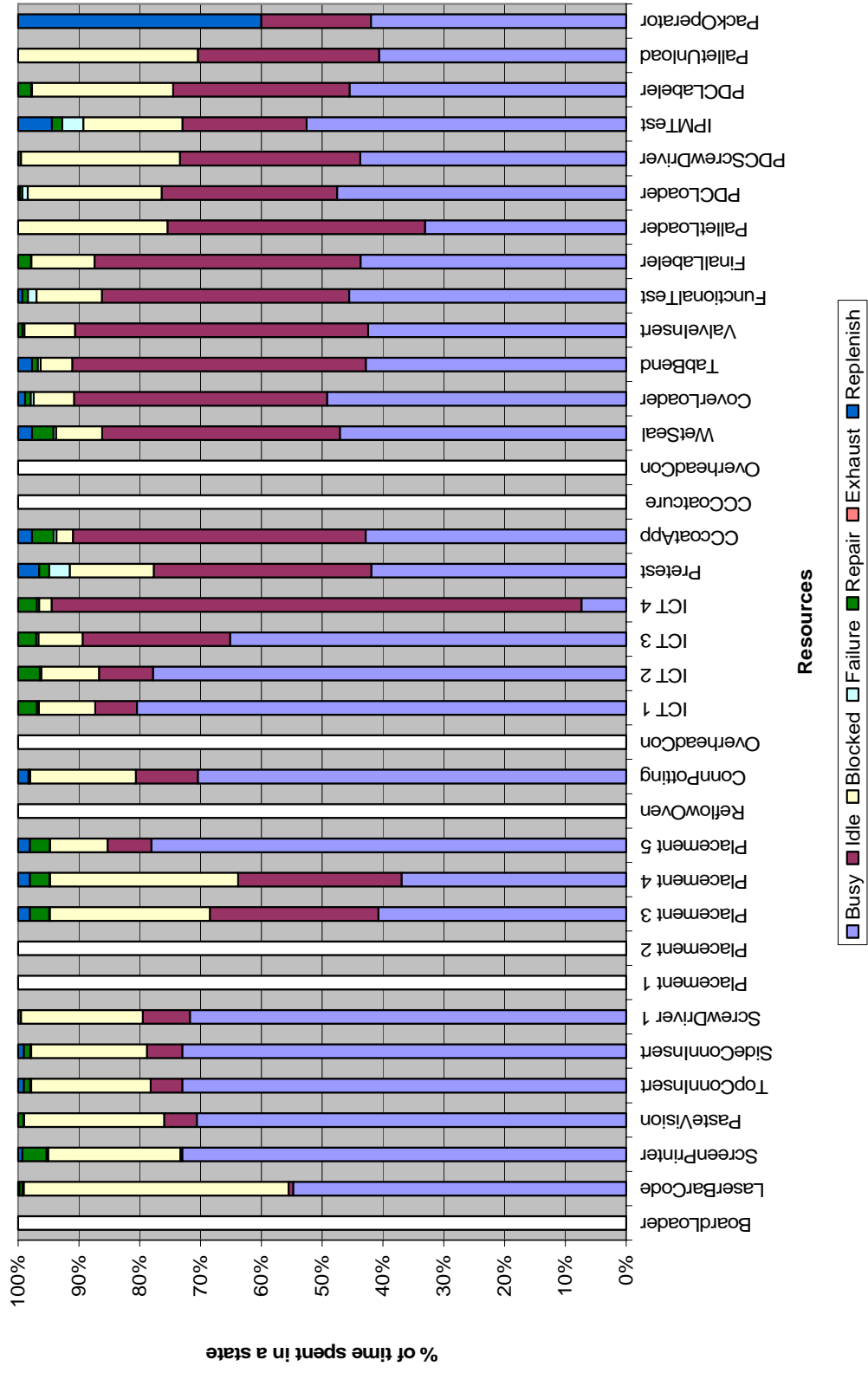


Figure II.5: Product Y Line - Resource State Graph for Scenario 2

APPENDIX III
PROCESS FLOWCHART AND RESOURCE STATE GRAPHS FOR PRODUCT
Z LINE

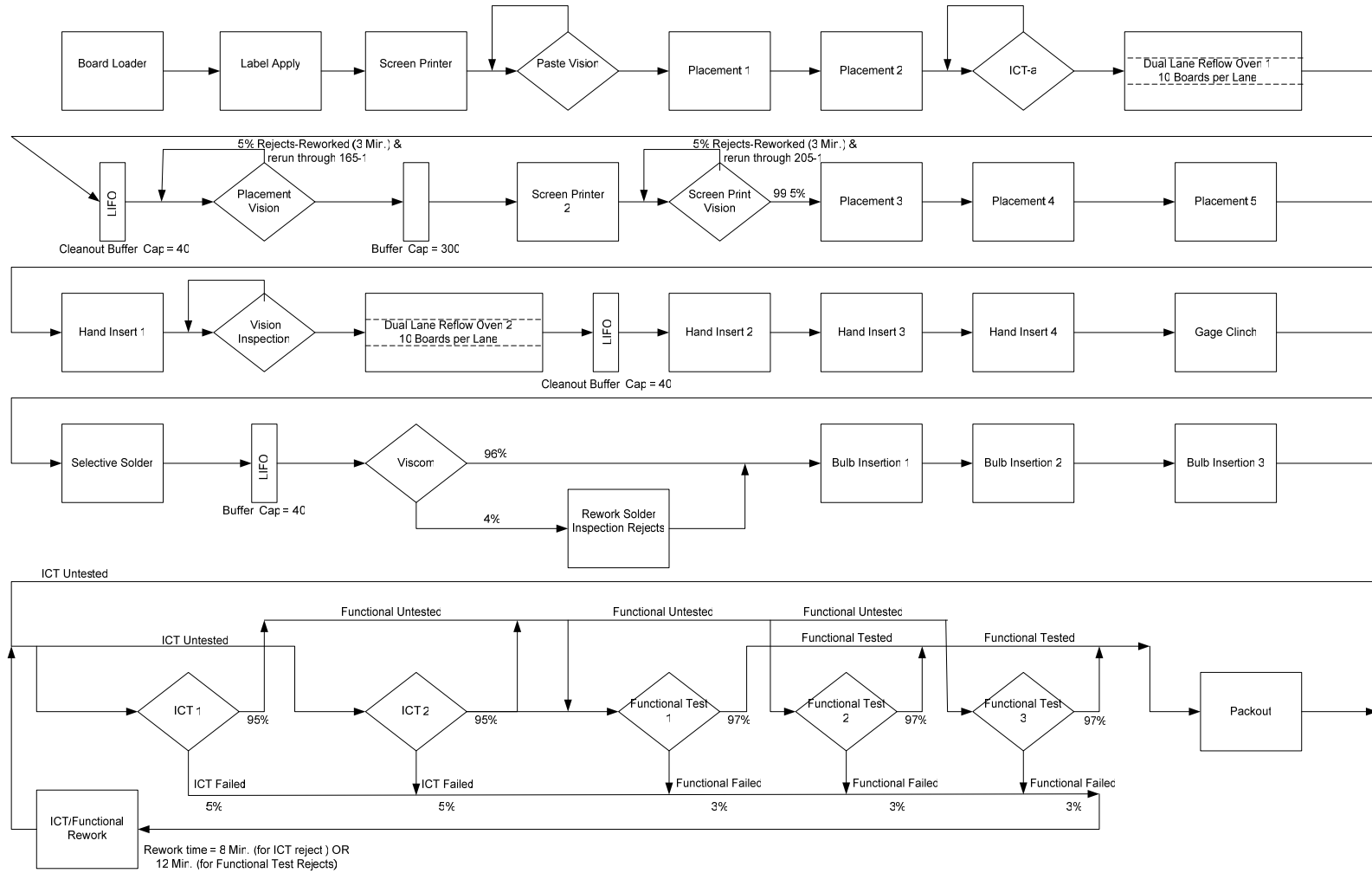


Figure III.1: Product Z Line - Process Flow Chart

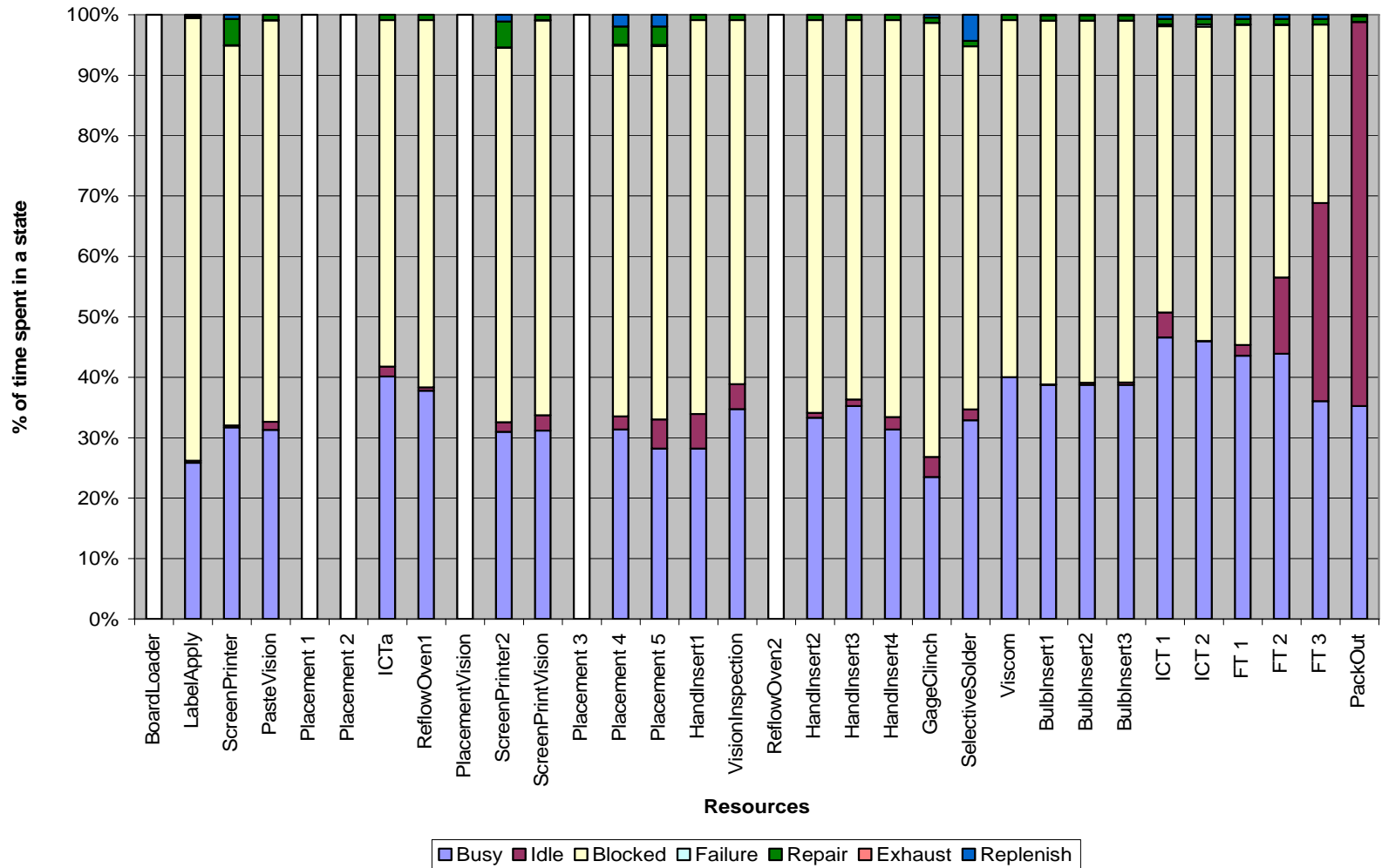


Figure III.2: Product Z Line Resource State Graph - Current Configuration

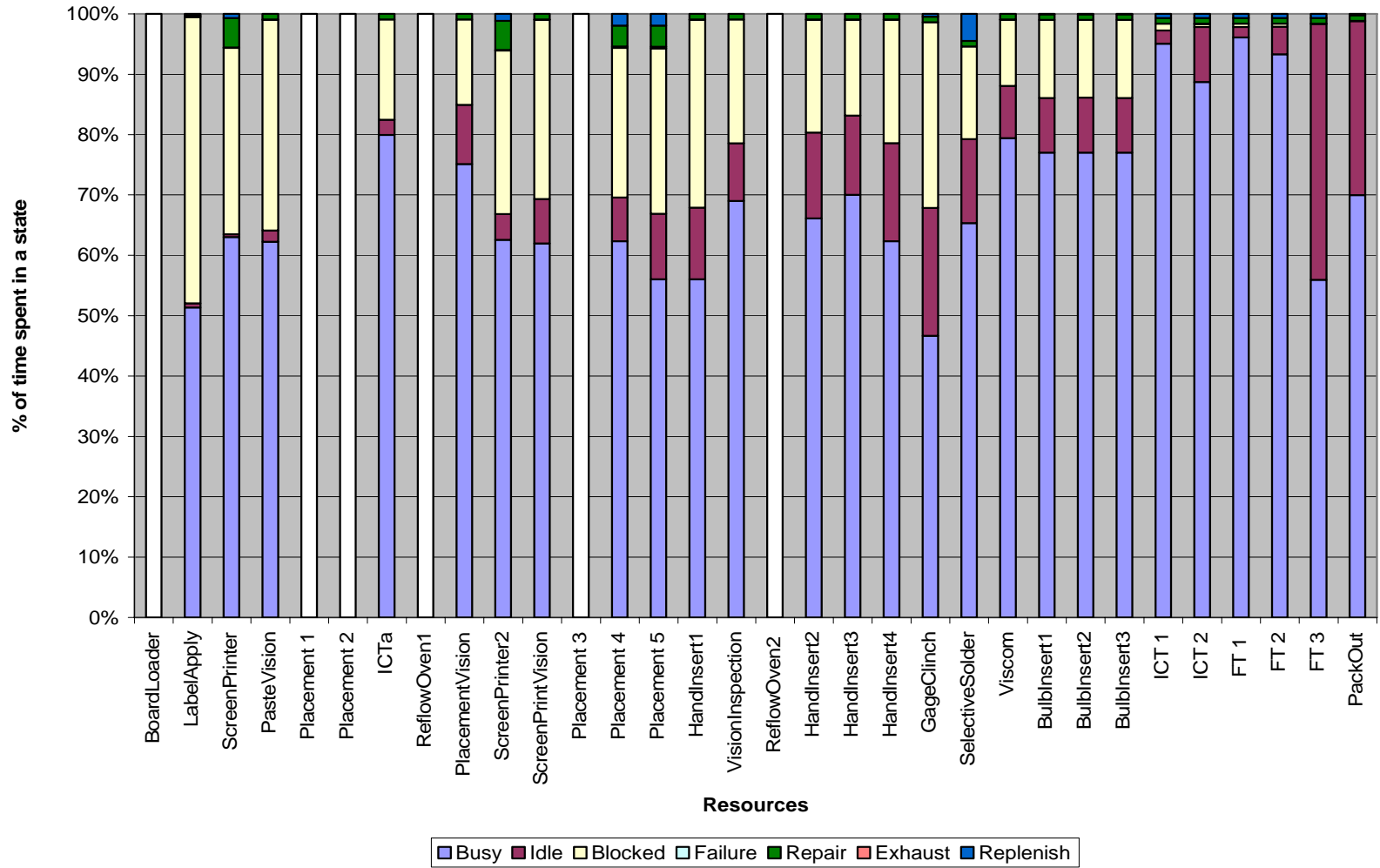


Figure III.3: Product Z Line - Resource State Graph for Scenario 1

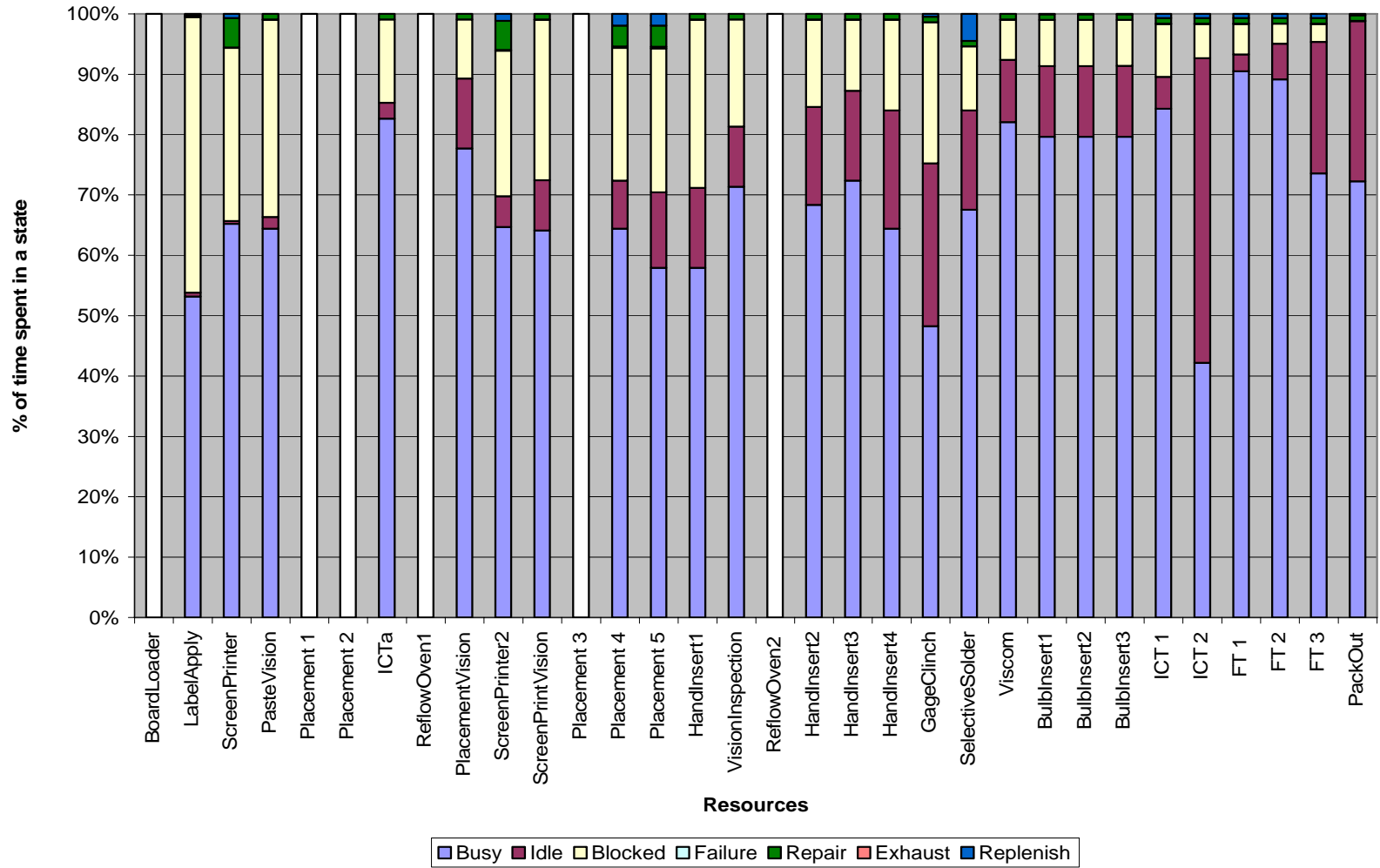


Figure III.4: Product Z Line - Resource State Graph for Scenario 2

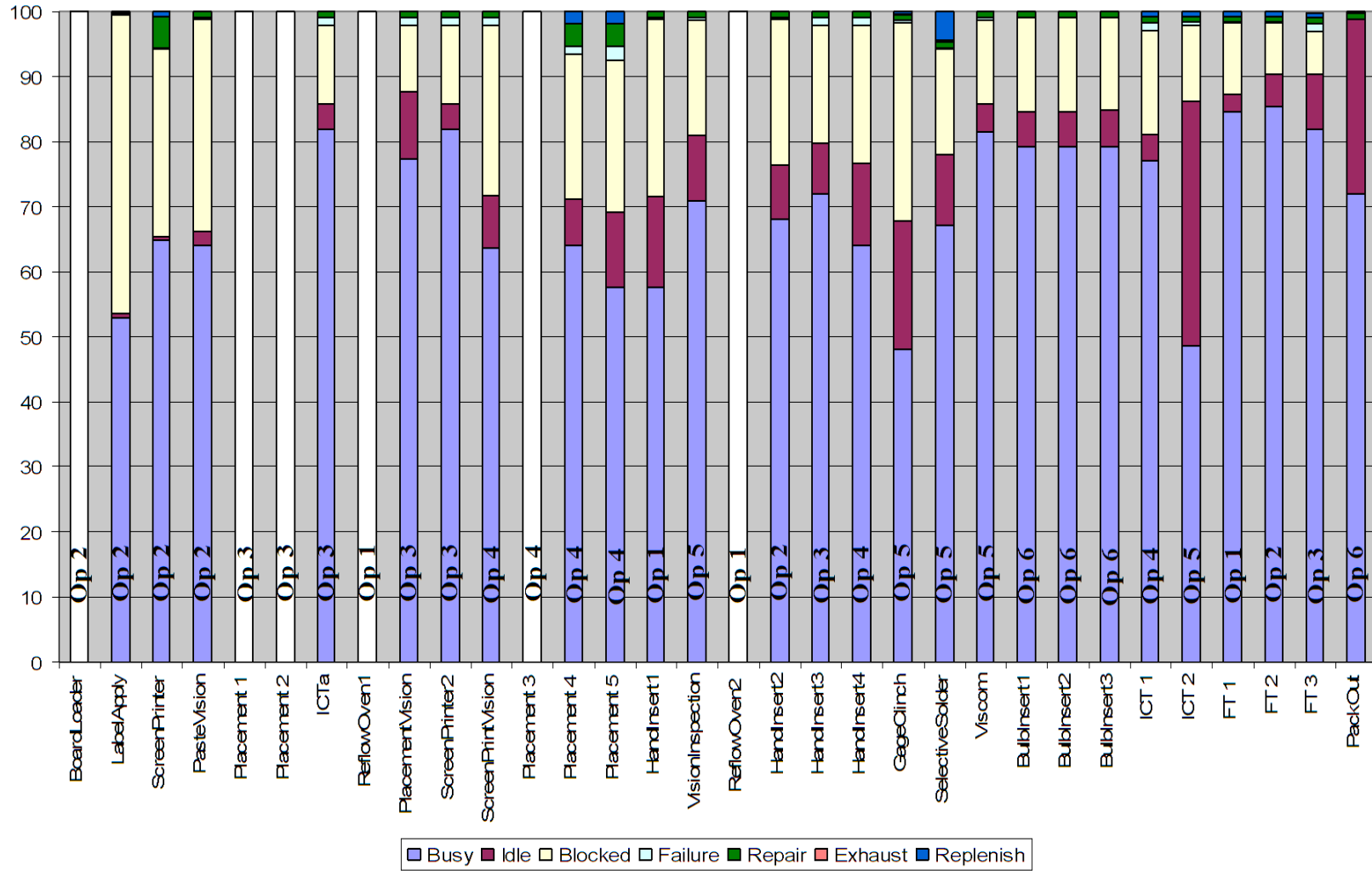


Figure III.5: Product Z Line - Resource State Graph for Scenario 3

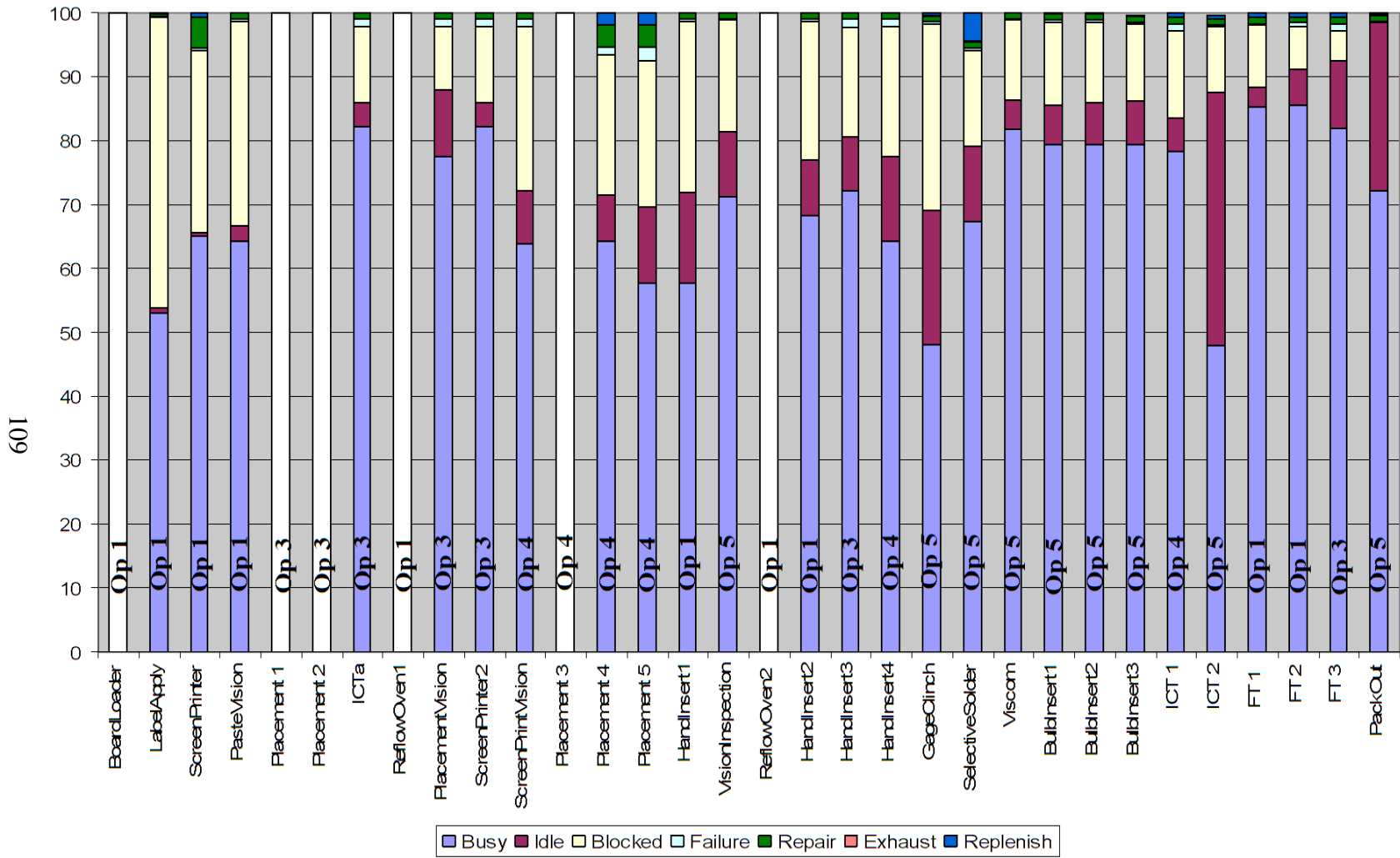


Figure III.6: Product Z Line - Resource State Graph for Scenario 4

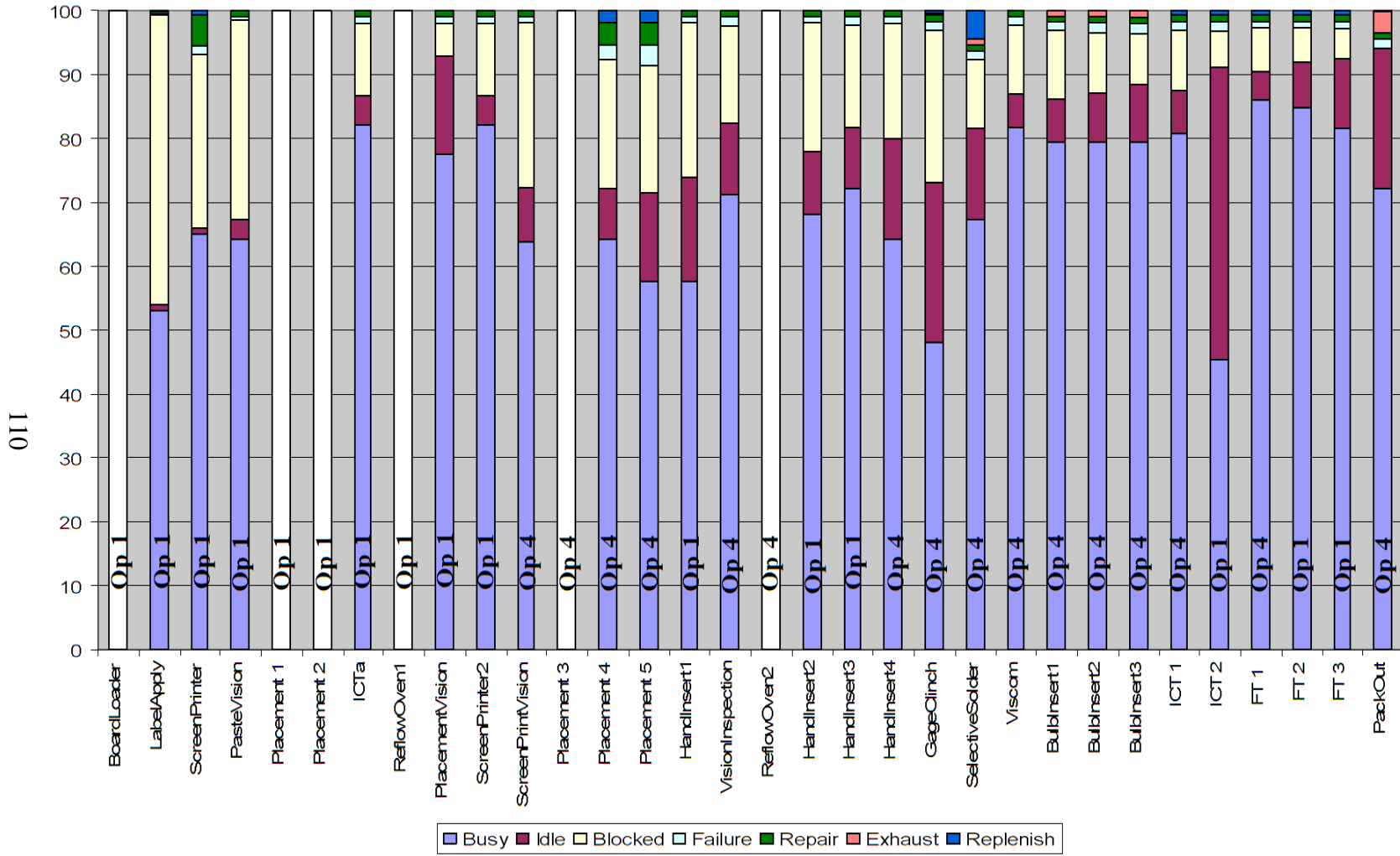


Figure III.7: Product Z Line - Resource State Graph for Scenario 5

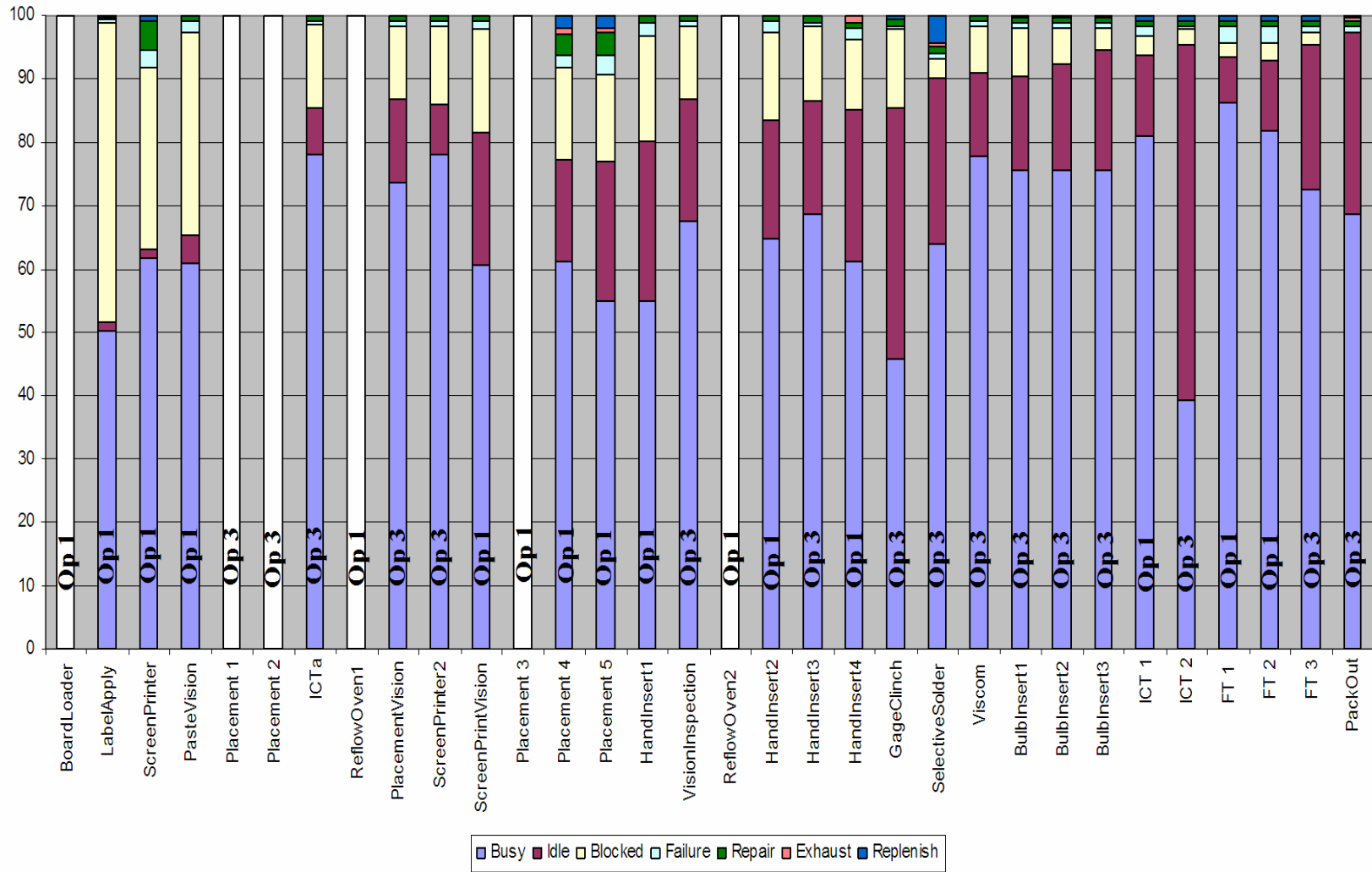


Figure III.8: Product Z Line - Resource State Graph for Scenario 6

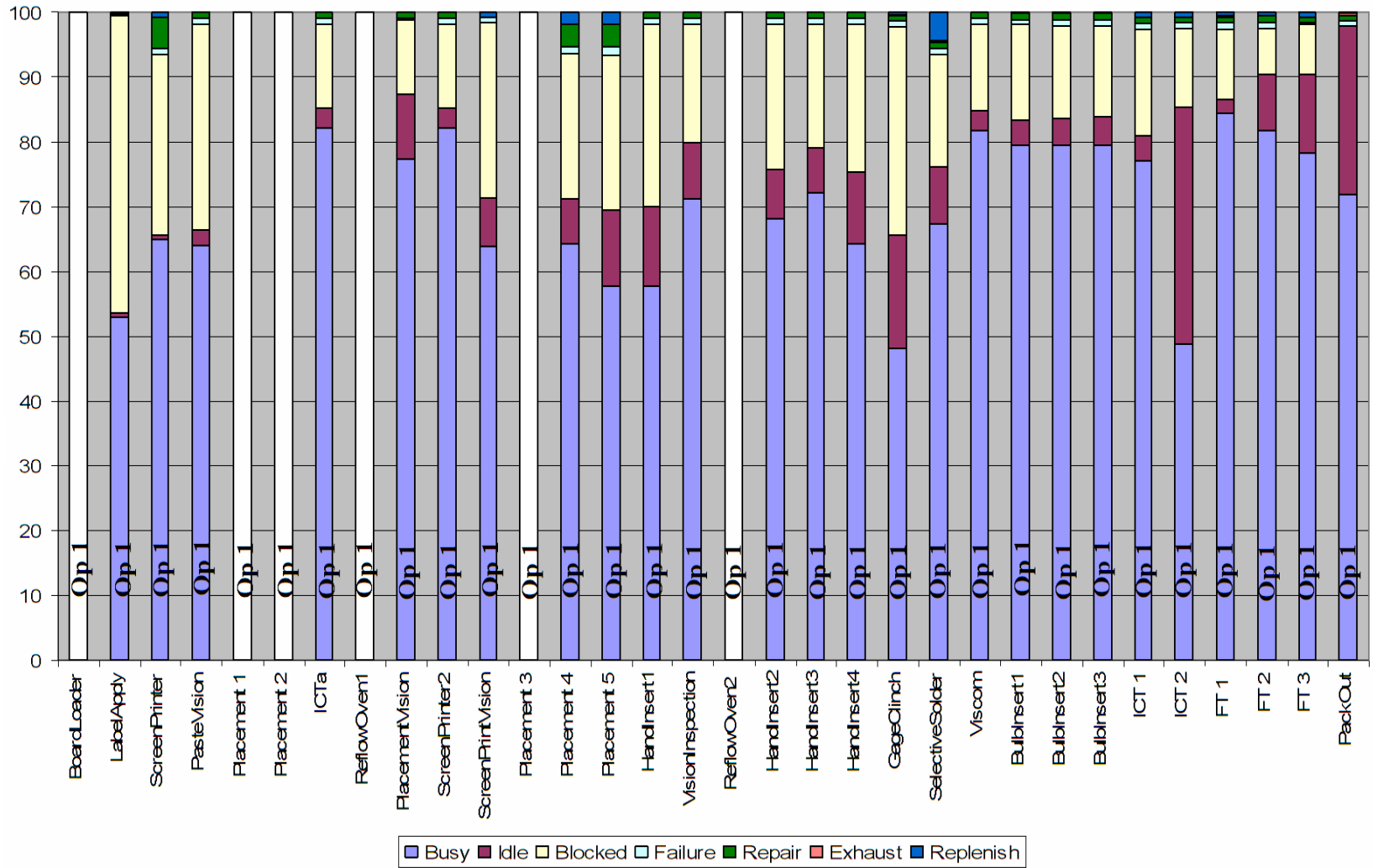


Figure III.9: Product Z Line - Resource State Graph for Scenario 7