Production Line Design and System Analysis for New Product Introduction in Electronics Manufacturing

by

Vrushank S. Phadnis
B.E. in Production Engineering
University of Mumbai, 2010

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Engineering in Manufacturing

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

Signature of Author
....
Vrushank Shripad Phadnis
Department of Mechanical Engineering
August 21st, 2013

Certified by
....
Stanley B. Gershwin
Senior Research Scientist
Thesis Supervisor

Accepted by
....
David E. Hardt
Ralph E. and Eloise F. Cross Professor of Mechanical Engineering
Chairman, Committee for Graduate Students

This page is intentionally left blank

Production Line Design and System Analysis for New Product Introduction in Electronics Manufacturing

by
Vrushank S. Phadnis
B.E. in Production Engineering
University of Mumbai, 2010

Submitted to the Department of Mechanical Engineering
on August 21, 2013 in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Manufacturing

Abstract

In this research, a system level analysis was performed on a one piece flow layout design for a factory manufacturing AC motor control drives. The requirement was to create a layout for a new product while integrating existing manufacturing lines. Analytical methods and simulation techniques were used to validate the performance of the proposed line.

The thesis outlines the manufacturing processes required to make the new product and presents an approach used to predict the new line's performance. A hybrid approach was employed wherein analytical methods were used to create a baseline plan of the production line. The baseline requirements were refined by modeling the line in a discrete event simulation to emulate existing factory constraints. The use of simulation enabled definition of lower level details like shift breaks times, machine breakdown trends and product batching policies. The simulation model was used to predict the impact of factory scenarios that were determined as necessary milestones in transforming the factory layout from existing layout to the proposed layout.

The concept of changing the layout from process based to one piece flow was validated through a Kaizen event. The event resulted in floor space savings of 500 sq. ft. and a reduction in work in progress inventory of 2160 drives. The Kaizen event facilitated in familiarizing with the existing factory processes which was essential in creating the new factory layout.

Thesis supervisor: Stanley B Gershwin

This page is intentionally left blank

Acknowledgements

My journey here at MIT has undeniably been the most exciting time of my life till now. I got the opportunity to enhance my technical skills under the guidance of the most revered faculty members of this institution, but more importantly it has been a self-revealing experience. A part of me is now represented by the experiences and conversations I have had at MIT.

I thank Stanley Gershwin for all his advice and support to enable me in successfully completing this thesis. The weekly meetings with him were certainly looked forward to and brought new insights and infused enthusiasm in me to keep moving forward. My team mates David McCalib and Joseph Falvella were an integral part of the experience and I am confident that the relationship will not end at this project. I thank the entire MEngM cohort and MEngM team for accepting me into the community and all my time spent with them was gratifying. I would like to thank Jose Pacheco and Jennifer Craig for believing in me and helping me write this document.

This project would not have been possible without the support from Lenze Americas, MA. I would like to thank Chuck Edwards, Michele Dauphinee and Nicholas Siciliano for the opportunity to work with them in solving real world factory problems. Meetings and discussions with Lenze's manufacturing team provided insights which were indispensable to the solutions provided in this thesis.

I would finally like to thank my family from back home for their overall support and trust in me throughout this journey. I would like to dedicate this thesis to my parents, sister and family who were a constant source of motivation in accomplishing this degree and will continue to be my drive in the journey here after.

This page is intentionally left blank

Table of Contents

List of figures

List of Tables

1. Introduction

1.1 Project introduction

The purpose of this thesis document is to describe the process used to design a new factory layout that will be capable of manufacturing a new product without losing the ability to produce existing products. The goal of the research presented in this thesis is to investigate the impact of changing a factory layout from process based to one piece flow. New layout designs were proposed and tested using analytical methods and simulation techniques. The final results in this thesis provide a full factory blueprint which maps the new equipment locations and material flow path.

1.2 Company Overview

Lenze is a global manufacturer of motion control solutions with a customer base ranging from new product designers to system integrators [1]. Lenze Americas manufactures frequency invertors, servo drives, gearbox, motors, motion controller clutches and brakes.

Lenze Americas was formally known as AC-Tech before its acquisition by Lenze-Germany. It's facility in Uxbridge Massachusetts still brands its product under the Lenze AC Tech trademark to appeal to its older customer base. The company has a strong customer focus and Lenze Americas continues to produce and support products from the AC-Tech portfolio to maintain this relationship. The company strives to innovate and meet customer expectations by constantly updating existing products and investing in new product development. Lenze products have been used in highly integrated environments where they communicate with other machinery like motors, sensors and user interfaces.

An example of such an application is a bottling plant where Lenze drives control motors that run conveyors, actuate pushers and apply labels. The drive needs to run synchronous with other system components and interact with other drives to successfully execute its function.

The motion control industry is driven by customer requirements like higher reliability and faster response times. The company's products are known to deliver value to their customers by providing user specific functionality bundled with reasonable cost. Lenze is in the process of introducing a revolutionary product in the year 2014 and the work presented in this thesis will contribute in realizing the manufacturing of the new product.

1.3 Motivation

The company had planned a new product introduction with a compact electrical architecture and a new mechanical structure.The new product introduction brought a set of challenges to the manufacturing department on the lines of capacity planning and creating a manufacturing plan for the new product. The department was in the process of reorganizing the factory floor to streamline operations and the new product added unknowns to this project. It was important to be able to predict the impact of integration of the new product into the planned factory layout.

The project charter for this thesis project was to create a new factory layout and validate the design using analytical methods and simulation techniques. It was important to understand and model the behavior of existing equipment as the new line would use most of the existing equipment. The proposed production line was designed to produce four product types from the frequency invertor family.

1.4 Generic Product architecture

Frequency invertor (herein referred as drive) construction can be broken down into three basic modules: PCB, heat sinks and enclosures. The PCB has the highest number of parts and takes the most time to manufacture.

1.4.1 Printed circuit board (PCB)

Figure 1 represents an illustrative image of the three basic drive modules. The image on the left is a representation of a control drive PCB (also referred as board). A typical PCB consists of a mounting surface (PCB laminate) and electrical components that are linked with each other through electric tracks printed on the laminate. The drive relies on two fundamental architectures to perform its function: the control board and the power board. The control board receives user input and processes it using a pre-programmed logic sequence. The logic circuitry is made up of numerous ICs (integrated circuits) and other electrical components like resistors, transistors and diodes. The programmed logic sequence determines the input signals sent to the power module. The power module is a rectifier that stores the incoming AC (alternating current) signal and generates a modulated power output which is used to drive the coupled motor. The size of capacitors and number of electric coils determine the power handling capability of a drive.

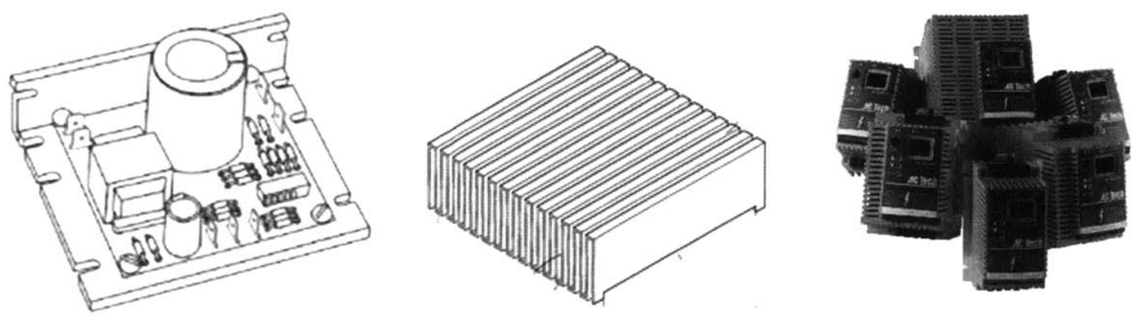


Figure 1: Basic drive modules: printed circuit board, heat sink and enclosures [2]

1.4.2 Heat sink

The heat sink is a reservoir attached to the power module to dissipate heat generated during power transformation. It is attached to the integrated rectifier (IR) module, and the interface between the two is sealed with thermal grease to minimize leakage losses. The IR module generates heat as it contains multiple transistors which step down the voltage for speed control and dissipate heat in the process. The heat sink is made of aluminum as aluminum is a good thermal conductor. The heat sink also has a number of fins (as seen in Figure 1) to increase surface area thus dissipating heat faster. It is made in long sections through a die extrusion process and then cut to length as per specification. The heat sink may be coated to enhance moisture repellent property, a requirement for drives functioning outdoors or in hazardous environments.

1.4.3 Enclosures

The prime function of a cover (enclosure) is to protect the drive from the environment and hold the I/O (input output) keypad. The body of the cover is a single piece injection molding. The cover body also holds other auxiliary components like electric filter boards, serial number bar code stickers and company logos.

1.5 New product architecture

As mentioned in Section 1.2, the thesis project involves creating a manufacturing plan for a new product herein referred as SMX. The proposed line layout will be capable of handling SMX production alongside an existing product herein referred as SMV. This section will outline the construction of SMX drives and provide the necessary background to understand the process plans presented in Chapter 3.

Figure 2 shows the assembly tree diagram for SMX drives. The SMX architecture was a new concept at the time this document was written and part drawings had to be studied to create an assembly sequence. The bill of material (BOM) for SMX drives was split based on manufacturing processes and is presented in Figure 2. The assembly sequence is presented on a vertical line on the far right of Figure 2. The build process starts with PCB at its core and thus, the PCB is displayed as the first component on the assembly sequence. Other drive components are listed in circles on the left of the PCB laminate circle. The vertical lines extending from the top row circles have boxes with numbers written in them. The numbers denote the quantity of components (in top row circles) in the drive assembly. The boxes at the end of each vertical line mention the manufacturing process used to attach the component to the main assembly. The letters inscribed in circles on the assembly sequence line (rightmost vertical line) denote the type of manufacturing process (A: automated assembly, M: manual assembly). Operation times are listed in boxes on the right side of the circles on the assembly sequence.

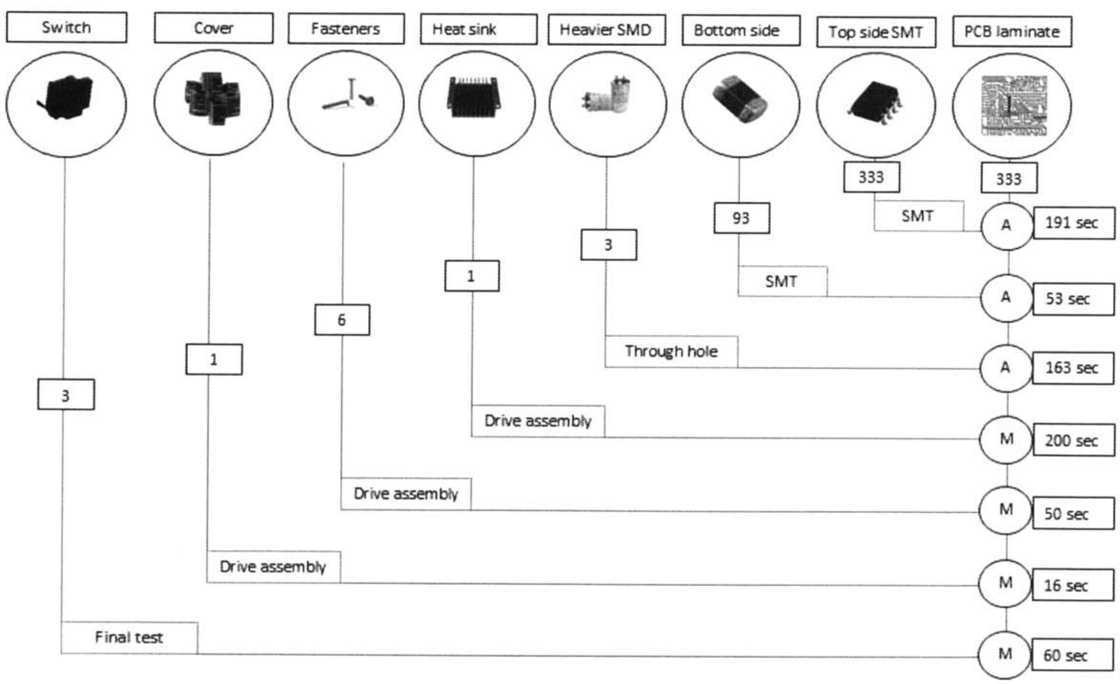


Figure 2: Assembly tree diagram for SMX drives

The construction of SMX drives is similar to SMV drives but the SMX drives use a double sided PCB which makes it more compact. The compatibility between SMX and SMV drive build sequence is important to ensure the smooth functioning of the proposed production line.

2. Problem Statement

2.1 Problem statement

Problem statement: To create a new factory floor layout with one piece flow and streamlined material movement for a drive manufacturing facility. The new line should have the capability to process SMX drives but should also be able to produce the existing SMV drives. The new process is expected to run with minimum work in progress inventory and balanced operation times at each work station.

The new line's design process began with understanding the company's existing shop floor constraints. Section 2.2 addresses the current factory situation and its findings were used as design input for the new line. The analysis of the existing factory conditions was done using a hypothesis driven methodology which ensured that the proposed line does not have a relapse of problems observed in the existing factory.

2.2 Hypothesis tree methodology

The tree diagram in Figure 3 is based on a hypothesis driven methodology that helped in identifying the root cause of issues faced by the manufacturing department of the company. The leftmost block states the overall problem faced by the manufacturing department and each subsequent tier is developed as a possible cause for the hypothesis stated in the tier to its left. The rightmost tiers present root causes for the overall problem and they were used as design requirements for the new line. The tiers of the hypothesis tree were framed after observing the shop floor operations over the first weeks at the company and based on interviews with shop floor personnel.

2.2 Description of Hypothesis tree

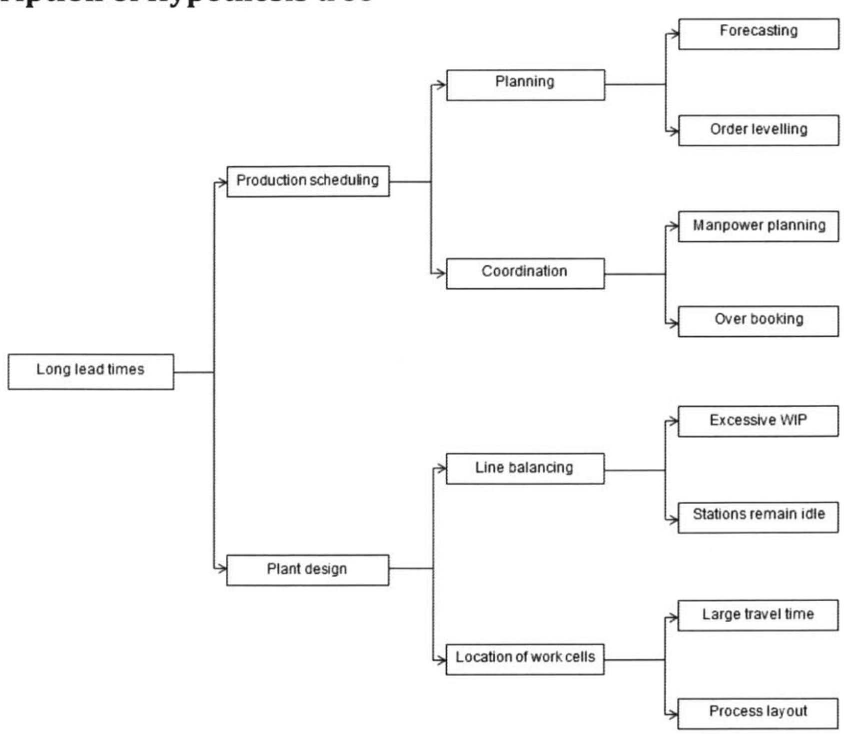


Figure 3: Hypothesis tree analysis of the company's manufacturing challenges

As seen in Figure 3, the overall problem identified by the company was long lead times on customer orders.

The current manufacturing setup required a lead time of up to eight weeks to satisfy customer orders. However, an addition of build time at each station making drives was less than an hour. This difference was partly because of suboptimal production ordering and partly due to an inefficient plant design.

2.3 Production scheduling

Production scheduling decides the daily line production and it is also used to decide the resource capacity every month. The manufacturing lead times depends on the production quantities ordered and product mix. Ideally a uniform ordering policy is required and it will help run the shop without any restructuring of work cell capacities.

2.3.1 Planning

Interviews with the manufacturing department and shop tours showed that the shop floor loading was affected by an erratic scheduling policy. Customer orders were released without leveling production quantities and without grouping similar orders, this gave rise to excessive setup changes on machines. This also affected the lot sizes sporadically and daily production targets were unpredictable. Rush orders from customers could not be anticipated, and products had to be delivered at a short notice. The capacity to process a rush orders would be pulled from planned production thus disrupting the scheduled output.

2.3.2 Coordination

Manpower planning had to be done at least a month in advance to provide time for sourcing. The manufacturing and planning departments collaboratively worked on monthly schedules beforehand to mitigate the loss of production due to unbalanced ordering. Rush orders often overbooked the plant capacity and the factory had to run overtime shifts to accommodate the additional orders.

2.4 Plant design

The company started operations with a few customer orders and a small set of plant machinery that was arranged in a layout which was efficient at that time. But over the years new products were added and order quantities multiplied. This put a lot of strain on the plant capacity and new equipment was bought to satisfy the increased demand. However, the additional capacity was placed on the shop floor as an extension to the first layout. All the similar equipment was placed together in a shared floor space to form work cells. This resulted in a factory layout which was not necessarily the best layout for the current production requirements.

2.4.1 Line balancing

The operation time of a work cell depends on the work content and number of machines available to perform the task. An analysis of the current work cell structure suggests inconsistency in operation times at work cells. This resulted in a mismatch in build times at work cells and WIP (work in progress) inventory would build up over time. Machines or manual assembly lines had to remain idle due to excessive production and manpower would be shifted to run other machines. This created confusion in the assembly operators and the area managers were constantly busy as they had to rework their manpower assignments frequently.

2.4.2 Location of work cells

Plant equipment was located in clusters of machines that performed the same operation. Suboptimal placement of work cell clusters in the factory floor resulted in redundant material movement. The work cell structure created siloes in operation by limiting the interaction across work cells. Chapter 3 presents a detailed analysis of existing layouts and the problems in it.

2.5 Project objectives

The project charter was to create a factory layout capable of meeting anticipated customer demand for SMX drives and at the same time retain the ability to produce SMV products. The strategy to produce both products was crucial to the company as the market reaction for SMX was unknown and the ability of the new line to efficiently switch to existing products was important to mitigate any forecasting error losses.

2.5.1 Forecasted sales for SMX

Figure 4 shows the expected sales numbers for the SMX product line over a ten year horizon. However the sales numbers are changed to maintain confidentiality. The sales are anticipated to rise in the first few years and then remain stable after year five. Years 8, 9 and 10 show a drastic fall in sales and therefore it is a likely time for another new product introduction. It should be noted that existing products will be manufactured alongside the SMX and the expected output from the production lines will be aggregated.

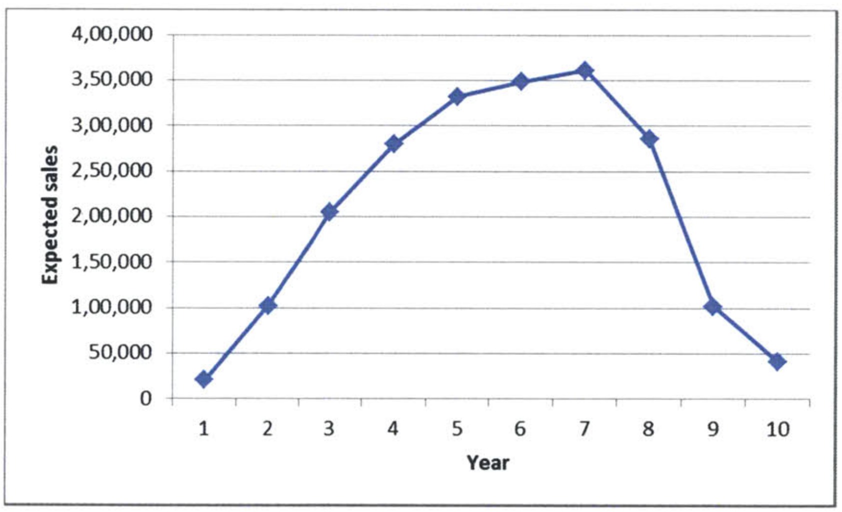


Figure 4: Ten year forecast of SMX sales

2.5.2 Takt time

Takt time is the time it should take to process a drive at each work station to satisfy expected customer demand. The forecasted maximum of customer demand for SMX drives is 360,000 drives/year (from the maximum value on graph in Figure 4).Takt time from this section will be used in Chapters 7 and 8 to develop the new floor factory plan.

The formula for takt time is as shown in equation (1) [3]:

$$\text{Takt time} = \frac{Total\ production\ hours\ available\ in\ a\ year}{Forecasted\ customer\ order\ for\ the\ year} \quad (1)$$

The factory operates on a two shift schedule and daily working hours are split as per Table1.

Table 1 : Daily shift timings

Break schedule per day (in min)	
Morning coffee break	10
Lunch break	30
Shift change	30
Second shift coffee break	10
Second shift dinner break	30
Total break time	110

Available time per day	1050

Total production time per day	940

Using data from Table 1 and assuming 50 five day work weeks per year,

$$\text{Takt time} = \frac{940*5*50}{3{,}60{,}000} = 0.65 \text{ minutes}$$

The takt time calculations tell us that the new layout needs to produce a drive every 41 seconds to meet customer demand. All our proposals will be tested against the takt time value of 41 seconds.

2.6 Requirements for the new line

A summary of the new line's requirements based on the discussion in this chapter is as follows:

- The line should support production of the new SMX drives.
- Production of existing SMV drives should run in parallel to SMX drives.
- The new line design should function on minimum work in progress inventory.
- Material flow through the line should be linear and material travel time should decrease.
- Existing production equipment should be used to build the new line.

3. Drive manufacturing processes and layout proposal

This chapter describes the existing shop floor operations and explains the in-house processes used to manufacture drives.

3.1 Existing Process flow

The company facility operates on a 50,000 square foot floor space located in Uxbridge, Massachusetts. The facility is set up on a process based layout wherein machines performing similar operations are part of one work cell. All equipment in a work cell is physically located in a confined floor area[4]. As seen in Figure 5, the entire shop floor is segregated into distinct work cells. The work cell construction encourages isolated functioning where each work cell strives to achieve individual objectives without attaining the overall system efficiency.

Figure 5 shows the footprint of the facility with each block representing a work cell. Each work cell is explained in the subsequent sections and the section numbers are displayed in parenthesis besides the work cell name in Figure 5. The SMV build plan shown in Figure 9 was used to create the travel path seen in Figure 5 [5]. Material movement is not streamlined and flow lines can be seen crisscrossing the facility from one corner to the other. Inventory is not located near the point of use and this increases the travel time to replenish parts at each work cell.

3.1.1 Incoming Warehouse

The company holds all its drive build material at a central warehouse known as the incoming warehouse. Material is pulled from the incoming warehouse to restock SMT and bulk item warehouses. The inventory level is monitored using SAP[1] and is managed jointly by the Supply Chain department and warehouse personnel. Some of the build material is sourced from overseas and is stocked in advance. Incoming inspection is required on certain vendor supplied parts like heat sinks. The manufacturing work cells pull material from the warehouse according to the daily production schedule.

The manufacturing shop floor places a kitting order which initiates a picking sequence. The warehouse personnel gather all the required components on a rack based on the pick list. Loaded kits are then transferred to the shop floor and stored in a kit area. Generally, orders with same customer due date were loaded on a single rack depending on the order size.

The bulk item warehouse contains all C category (low value) items and the inventory in this warehouse is vendor managed. The SMT warehouse pulls SMT machine consumables, raw material like PCB laminates, resistors, ICs and other electronic components from the incoming warehouse and puts them in a secondary storage location near the SMT area.

[1] Enterprise resource planning software used by the company

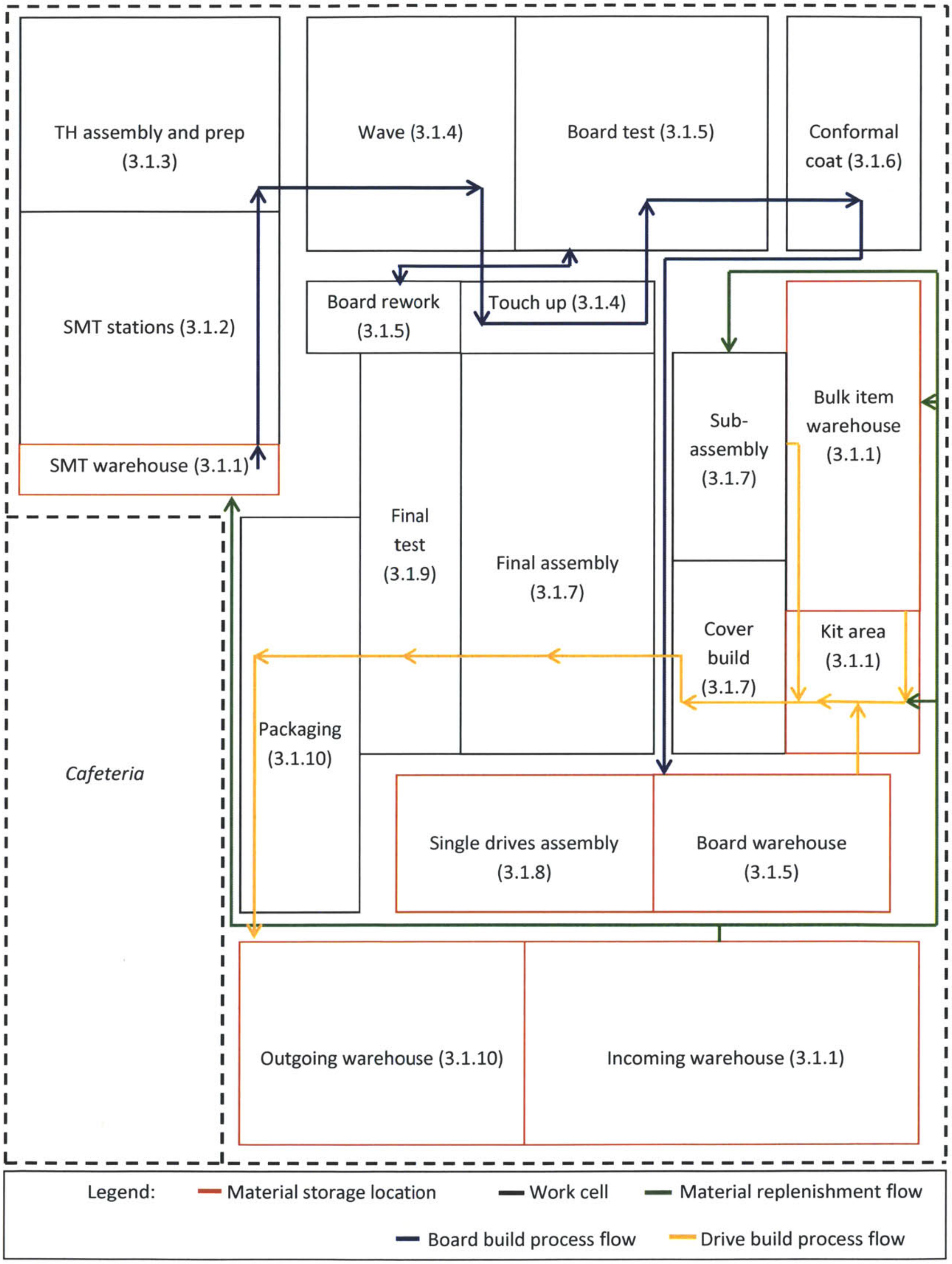


Figure 5: Existing shop floor layout

3.1.2 SMT stations

SMT technology was a major leap in PCB manufacturing in the 1960s. The process replaced through-hole solder joints by planar placement of components on laminates. Components are positioned using high speed automatic pick and place robots. A SMT operation requires a set of equipment as shown in Figure 6.

Figure 6: SMT area equipment [6]

The stacker machine holds multiple PCB laminates and loads the stencil machine with one laminate at a time. The stencil machine lays solder paste over the electric track terminals on the laminate. The next machine in line is a high speed gantry-type pick and place robot that populates the laminate with SMD (surface mounted device) components. Machine speeds in the range of 30,000 to 50,000 components per hour are available today. The populated PCB laminates pass through a 10 stage oven which exposes the PCB to a heating and cooling cycle to form the solder joints. The final equipment in the SMT area is an AOI (automated optical inspection) machine. The AOI machine uses image processing techniques to verify component part number and SMD placement accuracy. PCBs rejected by the AOI station are reworked for minor defects on a work bench located near the machine. Severe SMD placement failures like wrong component use are not repairable and the PCB has to be scrapped. The pick and place stations have an inbuilt vision system to detect faulty components at the source thus reducing major PCB failures at the AOI station.

Setup times on SMT machines can take anywhere between 5 minutes to an hour. An SMED (single minute exchange of dies) project is underway in this work cell to bring down the setup time to less than 10 minutes.

3.1.3 TH (through hole) assembly and prep

Oversize components cannot be placed at the SMT station due to the high forces generated during rapid movements of the placement arm. These components are manually assembled at the TH station. The assembly station inserts components in hole locations on the board and places the assembly in a fixture. The TH prep station prepares electronic components and transfers them to the assembly station. The TH station consists of two work benches with hand tools for assembly and component prep.

3.1.4 Wave and Touch up

Wave solder is a simultaneous multiple joint soldering process that connects the TH components to the circuit track on the PCB laminates. The wave solder machine operates off a belt conveyor with a capacity to hold about 10 board fixtures depending on the fixture dimensions and weight. The PCB are loaded onto a fixture that masks specific areas on the PCB that should not be exposed to the solder bath. The fixture is set on a conveyor that takes the PCB through a three stage process as seen in Figure 7. The PCBs are sprayed with flux at the first station (a). The second station (b) preheats the PCBs to avoid thermal shocks during soldering. The final station (c) dips the electrical joints in a liquid solder pot which establishes an electrical contact between the laminate tracks and the component leads.

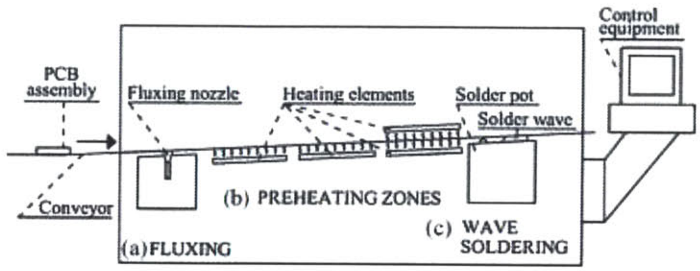


Figure 7: Wave solder machine [7]

The touch up station is placed at the output end of the wave solder and it is a manual inspection where the PCB is checked for faulty solder joints and component alignment. Any inconsistencies are reworked by the touch up operator and qualified PCBs are transferred to board test.

3.1.5 Board test

PCBs are tested for component functionality, power characteristics, logic and short connections. The board test fixture is designed on the 'bed of nails' concept that uses pogo pins to establish contact with the SMD components. The test is partly manual and test times vary based on product type and operator skill. David McCalib's thesis is focused on developing an automated testing strategy that involves designing a fully automated test fixture[8]. The new lines will have fully automated test equipment.

Failed PCBs are sent to a board rework station. The rework station rectifies the defect and records board failure data into a non-conformance report which is used to generate monthly quality data.

3.1.6 Conformal coat (CC)

Conformal coating is a process that is used to seal PCB components using a transparent layer of plastic to isolate the PCB from moisture, dust and hazardous chemicals. The coating can be applied using spray, dip or brush technique. The existing machine uses a spray based technology. The coating is then exposed to UV (ultraviolet) radiation to cure the spray and create a seal. Conformal coat is a special treatment and is only applied on drives used in hazardous conditions.

3.1.7 Final assembly

Final assembly (also referred as drive assembly) is a critical step in the entire build process as it pulls material from all the other work cells in the factory. The assembly area consists of nine lines which are categorized based on lead content in the solder joints. Three out of the nine lines run a low product mix and have specialized equipment to increase the build efficiency. Each assembly line consists of three work benches: assembly, solder and inspection. The first bench integrates the heat sink, sub-assembly and other manually assembled components like fans, IR modules and transistors to the PCB. The second bench solders capacitors, IR modules and transistors to the power board. The third desk is an inspection station that visually checks the assembly before fastening the cover.

Figure 8 shows the physical arrangement of benches (in solid lines) and material flow between drive assembly and other work cells (in dotted lines). Figure 8 is drawn to illustrate the interaction of assembly area work benches with other work cells in the factory. All the material required for drive assembly is introduced at the kit staging area and transferred along the line by the assembler.

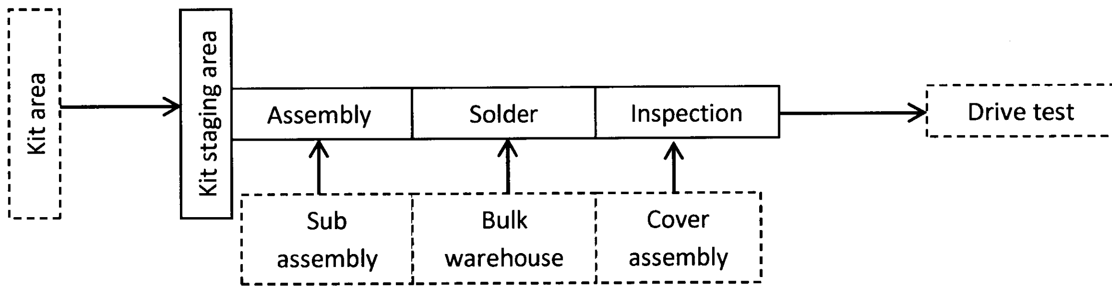


Figure 8: Interaction of final assembly with the other shop floor work cells

3.1.8 Single drive assembly

This is a self-sustained assembly station used to manufacture large drives. Only a few products are made at this station and all the work is done by a single person. The drives made in this section are used in high power applications and are heavy in construction. The operators working in this area are highly experienced as the built is complicated and there is frequent product changeover.

3.1.9 Final test

The assembled drives go through a sequence of tests based on the product test strategy: hipot, functional test, burn-in and power-up. The hipot test measures leakage current through the insulation in the event of an over-capacity load. The functional test simulates on-site performance of the drive. Measurements are recorded after the drive is coupled to a motor. Functional test is automated and is the final activity in most test strategies. The power up and burn in tests measure drive reliability under high loads and high temperatures.

Final test (also known as drive test) requires the longest operation time amongst all the work cells in the existing factory. The final test work cell was determined to be the bottleneck of the existing build process and inventory staging areas were located before final test. This information will be useful in comparing and validating the simulation logic when the simulation model is set up to emulate the existing factory conditions.

3.1.10 Packaging and Outgoing warehouse

The packaging operator kits the assembled drives and documents like a user manual in a box. Packaging cycle times are short and a single packaging area handles output from multiple assembly lines. Ready-to-ship drives are released to the outgoing warehouse after an SAP update. The packaged boxes await shipment from the outgoing warehouse to customers based on delivery commitments.

3.2 Product process plan

The proposed line was designed to produce four product families; two of which are from the SMV family and the remaining two are from the SMX family. The SMV build sequence is bifurcated due to a two board construction but the SMX drives have a linear build sequence. The two process flows are described in Section 3.2.1 and 3.2.2 respectively.

3.2.1 SMV

SMV drives have been part of the company's product range for a long time and the manufacturing team was well versed with the manufacturing of these drives. The SMV build sequence branches out at the SMT warehouse as the product has two PCB that are manufactured using processes which are identical (as shown in Figure 9). The ready boards are stored in a temporary warehouse until they are ordered by the final assembly station. The remainder of the drive build process is similar to the SMX build process presented in 3.2.2.

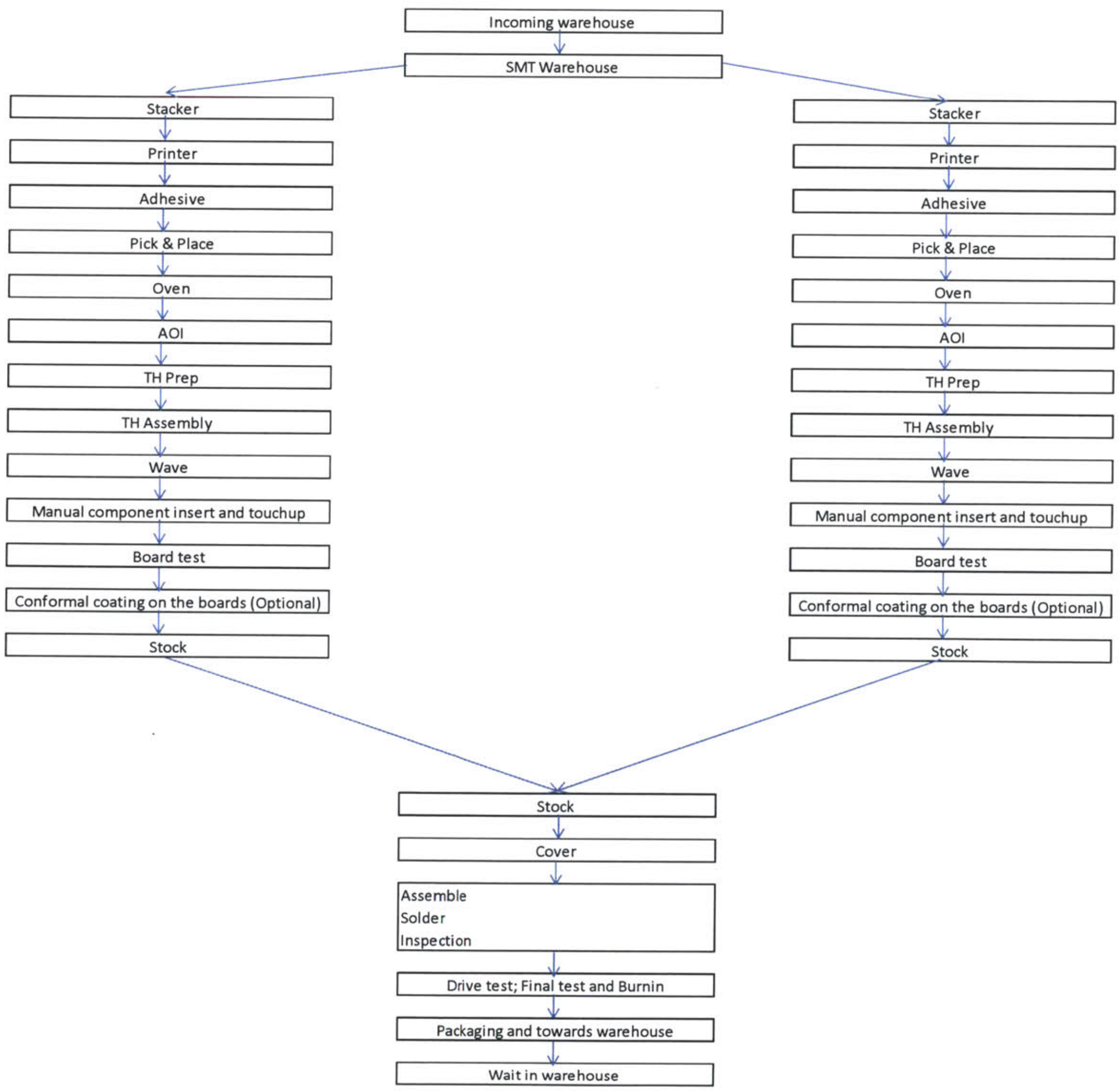


Figure 9: SMV drive manufacturing process flow

3.2.2 SMX

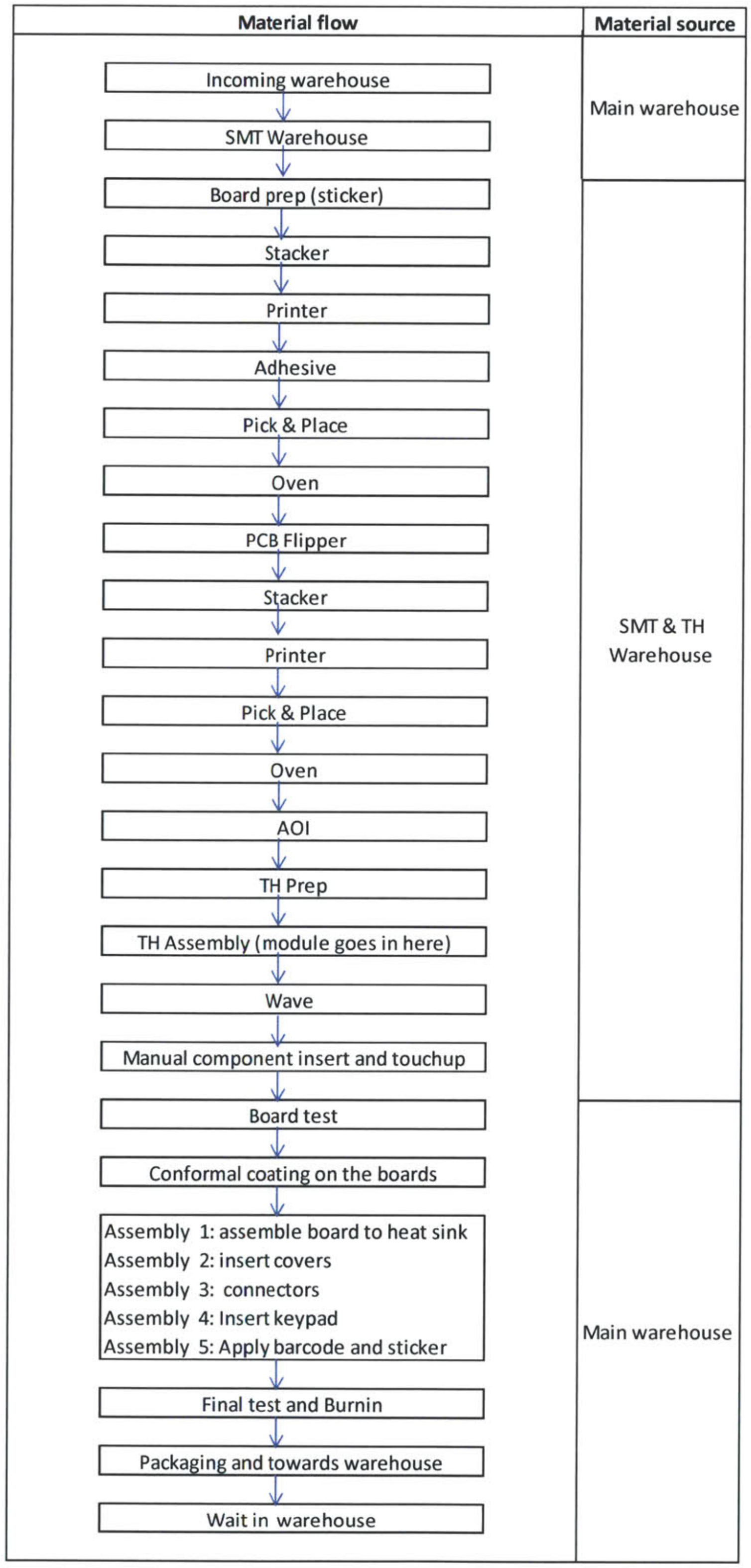


Figure 10: SMX drive manufacturing process flow

Figure 10 shows all the processes used to manufacture a SMX drive. SMX drives have a linear build sequence as shown in the process flow. The process flow in Figure 10 is created after analyzing SMX part print drawings and consulting the manufacturing engineering team to understand possible selections of manufacturing processes used to make the drive. The machines arrangement in Figure 10 was developed by placing processes in a sequence required for the SMX build and to decrease material movement [9]. The SMX build process begins at the SMT stations and terminates at the packaging station. The SMX SMT area has three pick and place stations due to the large number of SMD components. The first pick and place station inserts components on the bottom side of the board and the other three pick and place machines populate the top side of SMX boards along with SMV boards.

3.3 Proposed factory floor map

A new factory layout is proposed in this section and it will be developed further in chapters 5 and 6 using analytical methods and simulation techniques[3]. The new layout will be based on the sequence shown in Figure 11. However, Figure 11 is an illustration and does not show buffer locations.

The company has a strong focus on lean manufacturing and has taken initiatives to educate their employees about one piece flow. One piece is the arrangement of equipment to streamline material flow and balance operation times to produce as per the takt time[10]. One of the main objectives of one piece flow is to process one part at a time on each station. Benefits of one piece flow have been enumerated in many factories most notable in the Toyota Production system[11]. One piece flow was one of the requirements stated by the company for the design of the new line. The proposed layout in Figure 12 has equipment arranged in a linear layout to encourage one piece flow. Figure 12 shows the overall shop floor with four new lines. Material is introduced to each line at the SMT area and as the build progresses bulk items are pulled from temporary storage racks. This is a schematic of the proposed line layout and a detailed capacity plan and shop floor prints will be developed in chapters 8 and 9. The layout in Figure 12 is designed to function as a pull system where each machine is working on material placed on a buffer conveyor.

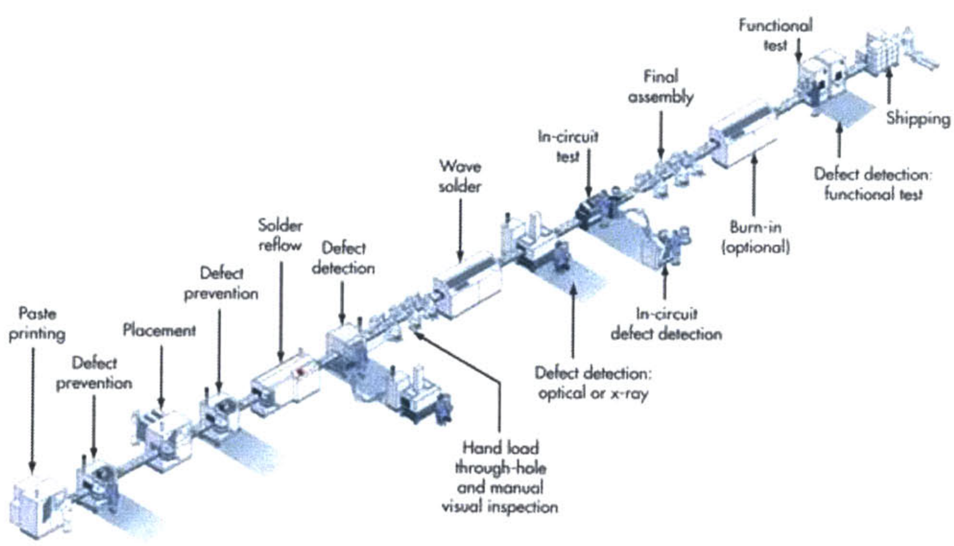


Figure 11: Representative image of one piece flow in board manufacturing[12]

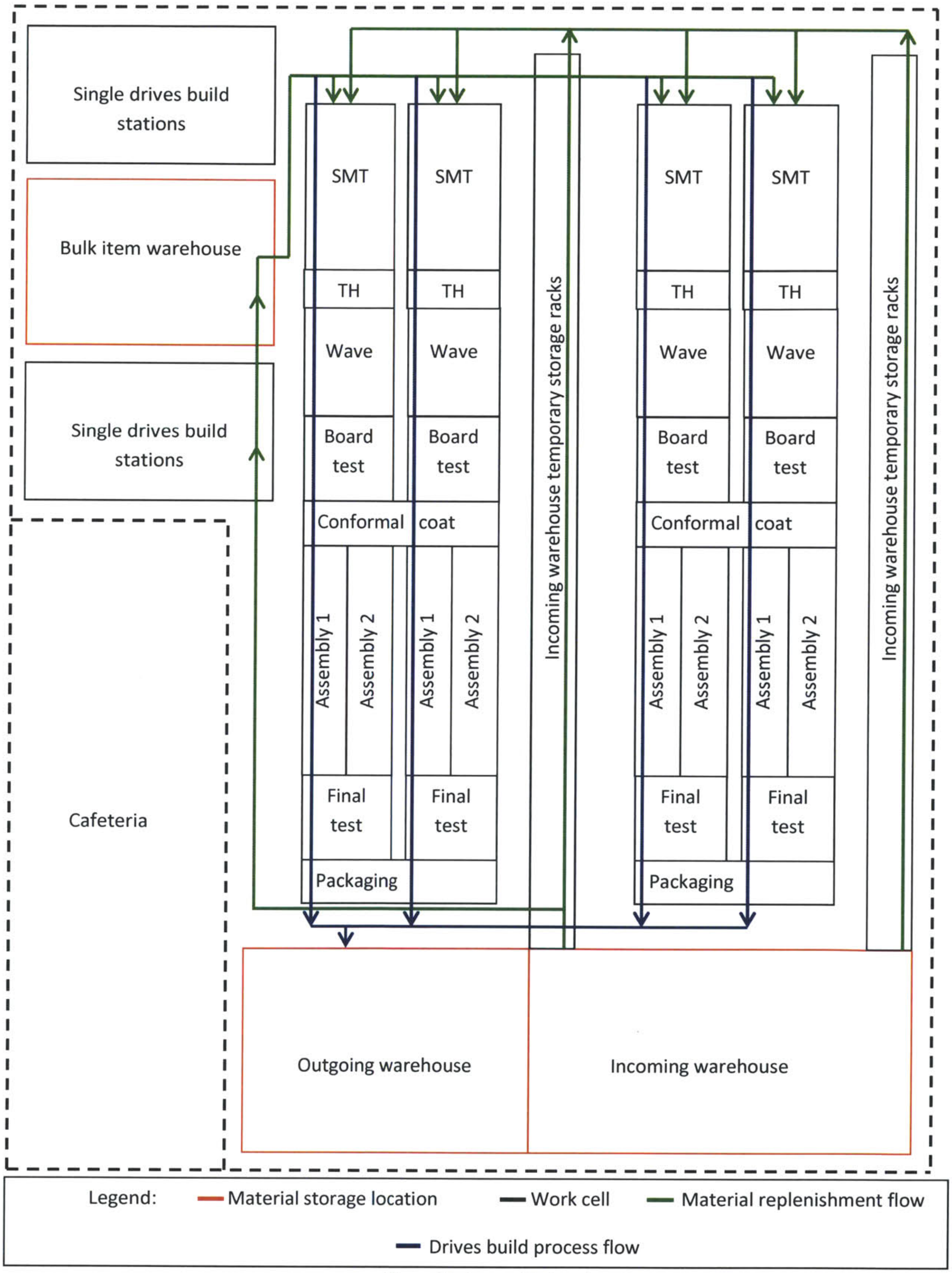


Figure 12: Proposed factory layout

3.3.1 Expected improvements in the proposed layout

A list of expected benefits in changing from the existing to proposed layout (Figure12) is as follows:

- Work cells function cohesively to optimize overall line production and do not strive to maximize the local output.
- Clearer communication between work stations as they are located in each other's vicinity.
- Material travel paths are decreased and there is better traceability of customer orders.
- WIP buffers are limited to conveyor space between stations and thus promoting one piece flow.
- It is easier to change floor capacity by shutting down individual lines as opposed to calculating the number of active stations needed at each work cell in the previous layout.

4. Pilot project implementation through a Kaizen event

This section presents the implementation of a Kaizen event that involved re-layout of a small section in the factory. The Kaizen was conducted to demonstrate the concept of using one piece flow to streamline drive assembly process. The event was instrumental in demonstrating the advantages of using one piece flow to improve existing factory conditions.

The DMAIC (Define-Measure-Analyze-Improve-Control) methodology is an approach used to improve processes and is often used in six sigma projects[13]. The following sections are structured on the DMAIC methodology and they will document the different phrases of the project [13]. The event was executed over a three day period with a 6 member team, in which the author and Joseph Falvella were responsible for coordination of the entire event.

4.1 Define

The define phase describes events from the first day of the event. The intention of the activities on the first day was to familiarize the team with the general concept of Kaizen and discuss the agenda for the next two days. The Kaizen goal was to streamline drive assembly operations based on one piece flow. The next step was to create the value stream map of existing drive build process. A series of shop floor tours and interviews were part of the process to create the value stream map. The objective set forth to the team was to develop a plan to revise drive assembly operations so that there is a decrease in material travel time and lower waste in the drive build process.

4.2 Measure

Figure 13 shows the value stream map drawn for the drive assembly area. The waste identification on Figure 13 is done on callouts alongside the material flow lines. As mentioned in Section 4.1, the value stream map was used to measure the existing scenario. Value stream mapping has evolved to be a preferred tool for improvement projects and waste identification[14]. The team traced the drive build process from the inspection desk to the first assembly station to create the value stream.

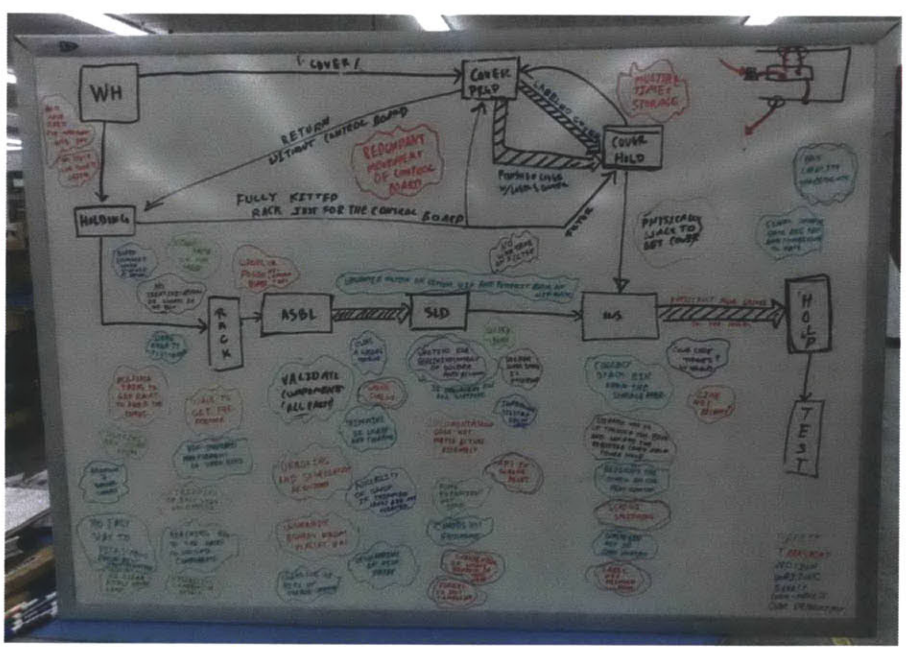


Figure 13: Value stream map of the existing drive build process

Time studies and travel path measurements were recorded by the team to capture the existing process parameters. Figure 14 shows the material travel path in the current drive build process. The red lines show material transported from the warehouse to drive assembly and the orange lines show material flow for cover builds. Cover assembly is a subset of drive assembly and the covers need to be available on time at the last drive assembly bench for the final integration.

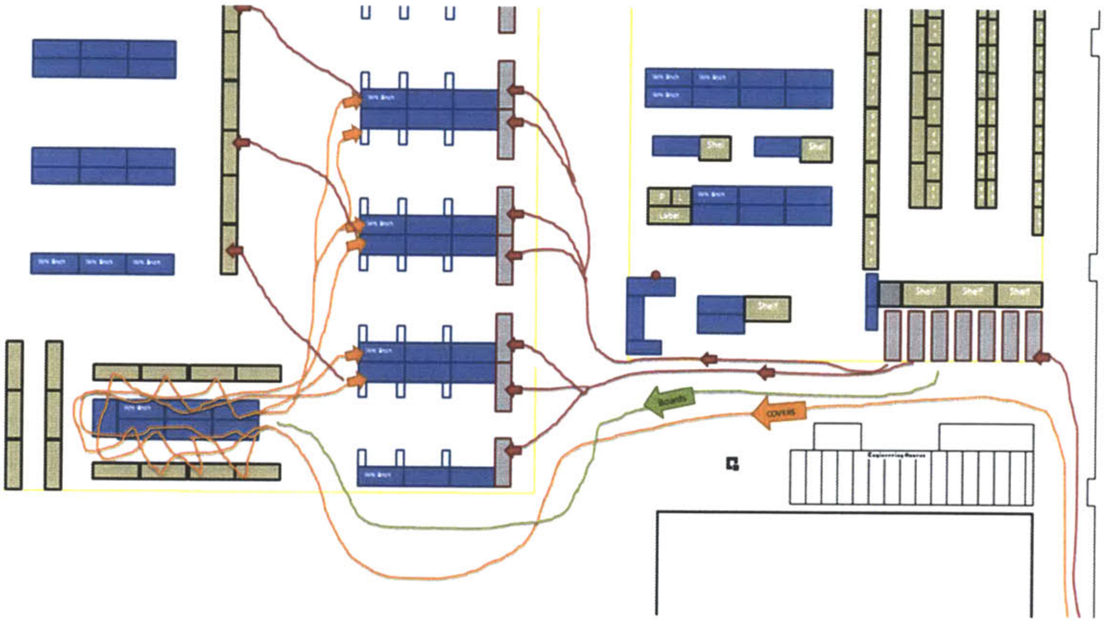


Figure 14: Floor plan of cover build area before the Kaizen event [15]

4.3 Analyze

The flow lines drawn in Figure 14 suggest multiple redundant loops in the cover build process. Cover WIP was one of the prime wastes identified in the measure phase too. Covers were assembled in a standalone area away from the main assembly line. This required that the cover raw material be transported from the warehouse to the cover build area and back. Finished covers were stored in stock once the cover build was complete. Figure 15 captures the interaction between the cover build area and the temporary material storage. There is excessive material movement between the parking lot and the cover area. However the covers are used at the inspection station on the main assembly line. Covers had to be built ahead in time anticipating the drive assembly requirements. A brainstorming session to create a new cover build sequence resulted in ways to reduce the wastes in building covers and they will be discussed in Section 4.4.

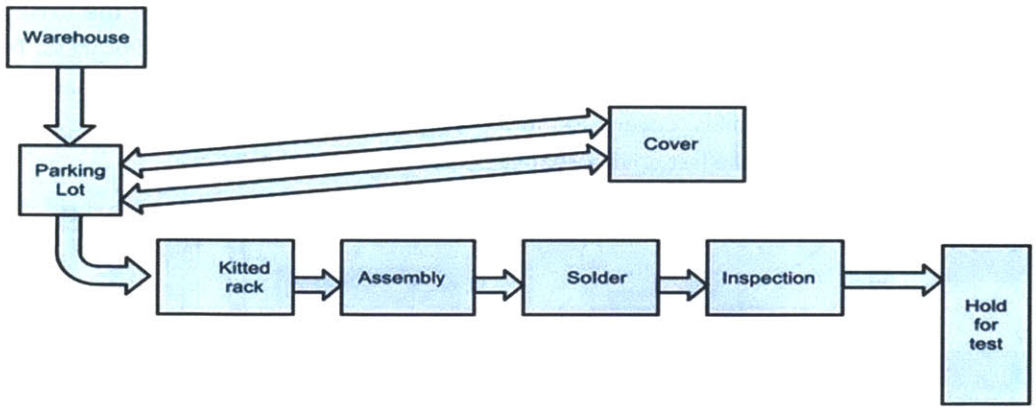


Figure 15: Material flow in existing cover build sequence

4.4 Improve

Figure 16 shows the new cover build material flow path. A new cover build sequence was suggested wherein the cover raw material was part of the drive assembly racks, which would now stop by the cover station on their way to drive assembly. The cover station would process the covers and place them back on the rack for drive assembly. The prepped covers reached the inspection work bench without being stored. The new process was expected to decrease material travel times and WIP of covers.

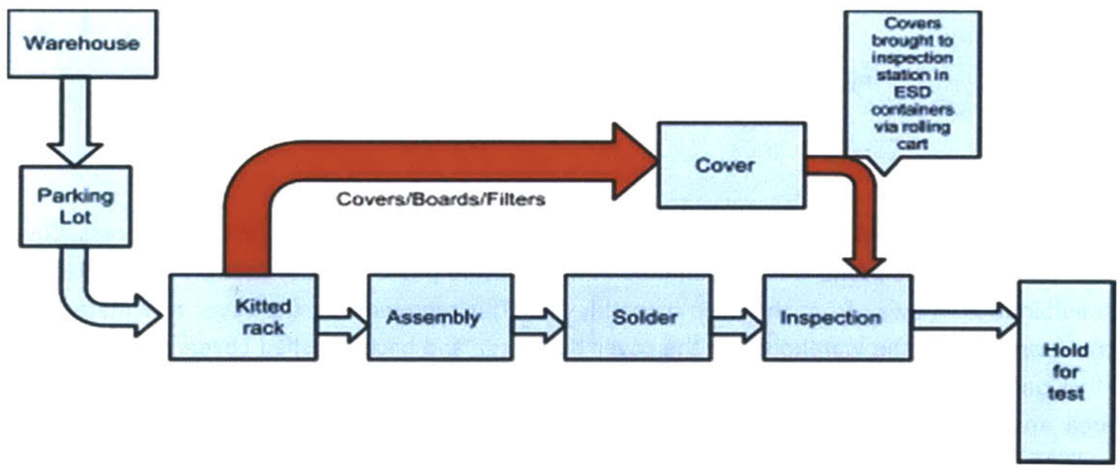


Figure 16: Material flow in proposed cover build sequence

The changes to the cover build sequence were discussed with the concerned departments and the cover build equipment was re-arranged as per Figure 17. The old cover area previously occupied a floor space highlighted in yellow in Figure 17. The new cover area operated on less than half that space. The project evolved over time and the cover build had become part of the drive assembly line. The new material flow is shown in red in Figure 17 and a comparison to Figure 14 shows the reduction in material travel time. The new cover build sequence exemplified the one piece flow concept and was successful.

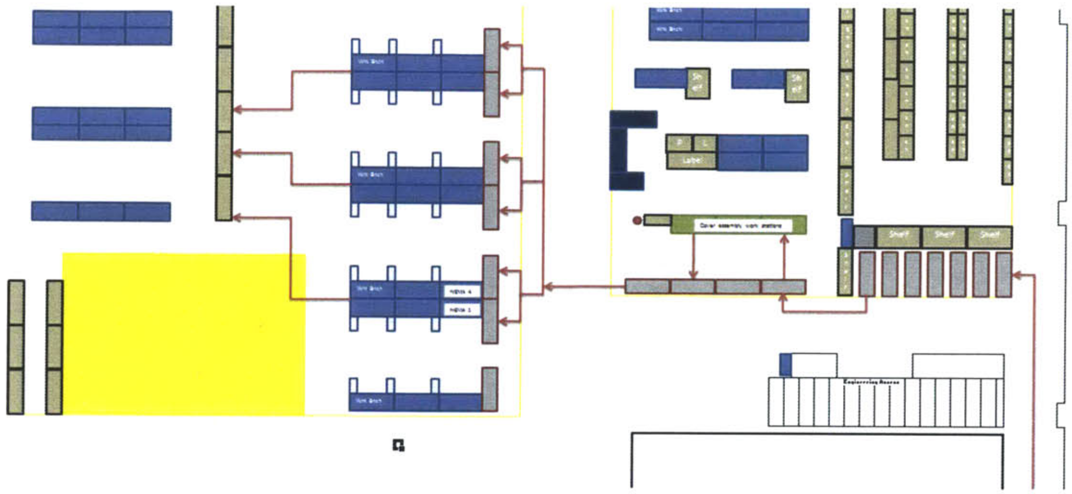


Figure 17: Floor plan of new cover build area after the Kaizen event[15]

4.5 Control

The new cover build process was shared in team meetings and communicated to the personnel in the cover build area. The warehouse department was familiarized with the new kitting process. The new process was monitored for the next few weeks to ensure sustenance of the new process.

4. 6 Outcomes of the Kaizen event

The two major savings from the project came in the way of reduction in number of covers in WIP and floor space use.

- Reduction in cover build WIP inventory=2160 covers.
- Floor space savings= 500 sq. ft. (Figure 18).
- New kitting procedure was implemented to decrease de-trashing time loss.
- Work bench bins were numbered on each station for standardization.

Figure 18: Cover build station in its old location and floor space after it was moved

Figure 18 shows the photos of the old cover build station and Figure 19 shows images of the new cover prep station. The new cover prep area was laid out in line with drive assembly stations and material flow was streamlined. The cover prep happened in tandem with drive assembly and the new process ensured better traceability of covers and drives.

Figure 19: Photographs of new cover assembly area in line with drive assembly

5. Review of theoretical background

This chapter presents material reviewed by the author that is pertinent to the concepts used in developing the factory layout and provides a framework that was used in the simulation study.

5.1 Factory layout types

Figure 20 shows the three types of manufacturing environments and the overlaps between them. Manufacturing operations can be classified based on product mix and production quantity as: mass production, batch production and job shop production. Equipment purchases, level of automation and production costs are influenced by the factory layout which depends on the company's manufacturing operation.

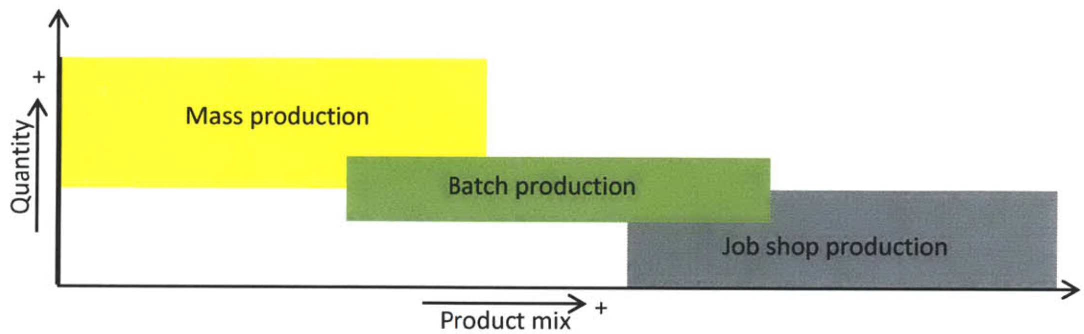


Figure 20: Different types of manufacturing environments

There are multiple ways to lay out a factory floor but most factory floor layouts conform to one of the three types: product layout, process and fixed position layout[5].

5.1.1 Product layout

Figure 21 shows a product layout with an equipment arrangement to produce two product types. Product layouts are typically used in high volume production and it uses high level of automation and special purpose machines that are tuned to perform specific operations. While product layout runs with a high efficiency, the customized equipment demands a high initial investment and is not easily reconfigurable. Equipment breakdowns affect other stations drastically due to limited in-process inventory and line stoppages incur strong penalties.

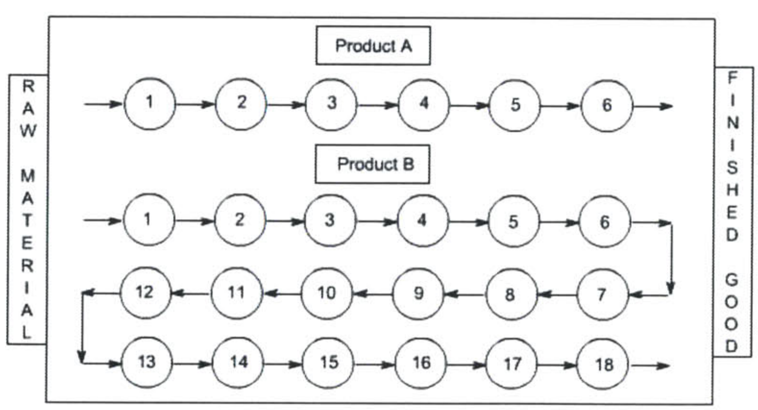


Figure 21: Product layout with two product lines[16]

5.1.2 Process layout

Figure 22 shows material flow through a process layout with batch production. In a process layout, all the equipment performing similar operations are located in a shared factory floor space. Process layouts are suitable in factories with a medium product mix [5]. Most of the equipment used in a process layout is general purpose and this provides flexibility to process multiple product types. However work cells cannot be located to optimize material flow for any single product and this results in a lot of redundant material movement. Frequent inspection checks are a required as there is high chance of errors resulting from frequent product changeovers.

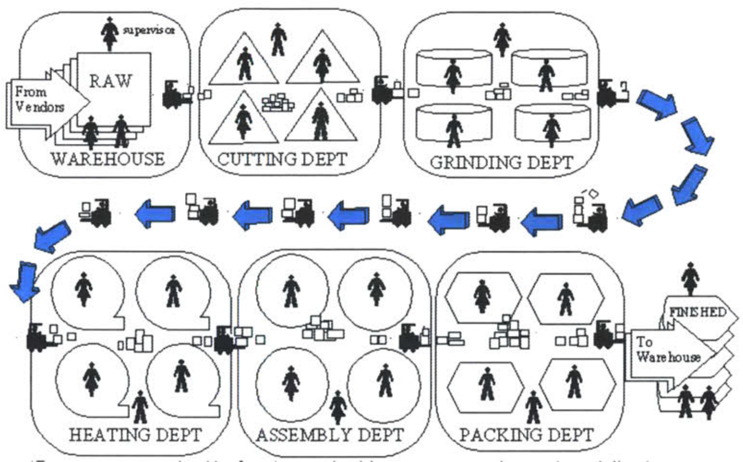


Figure 22: Process-type factory layout[17]

5.1.3 Fixed position layout

A fixed position layout is designed with the job as the center piece, as seen in Figure 23.Large-size products like ships and aircrafts are manufactured in one-off numbers and with limited or no movement of the job from one location to other. Manufacturing equipment is modular and mobile to enable placement near the point of use. Fixed position layouts are highly flexible as all the machinery is general purpose and easy to move. Multiple operations are executed at the same time and there is scope for confusion and conflict, thus posing a project management challenge to deliver the job on time. The lead times on customer orders are high and products in industries like oil and gas can sometimes take several years to complete.

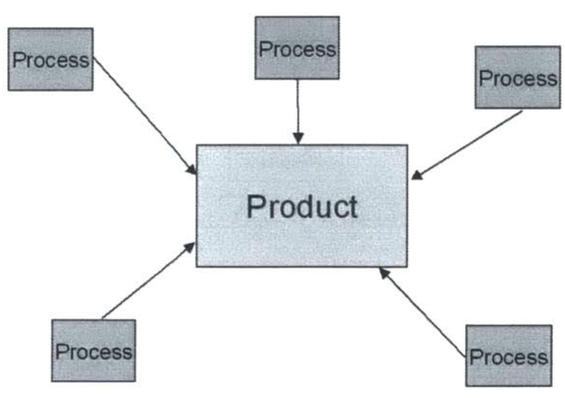


Figure 23: Fixed position factory layout[18]

5.1.4 Group technology

Figure 24 shows a schematic of a factory layout designed on GT (group technology) principles wherein each cell (dotted line) has equipment working to process a product family. GT is a fairly new technique used to decompose a large set of product types into small sub categories based on similarities in design and manufacturing techniques. GT factory layout is developed by clustering equipment to efficiently manufacture a product sub-category [19]. Various clustering methods like matrix formulation and mixed integer programming have been used to create part families [19]. However this thesis will not explore product clustering in detail and will focus on designing a layout for a pre-defined cluster of four product types. Group technology offers advantages like decreased waiting time, lower operating costs and streamlined material flow [5].

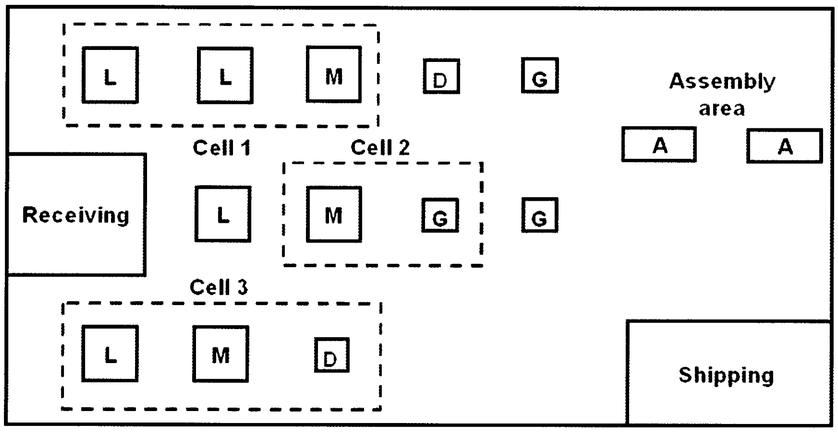


Figure 24: Group technology based factory layout[18]

5.2 Factory layout design framework

The flow chart in Figure 25 is adapted from a research paper describing the approach used by a team of consultants working to expand factory operations for a company with effective use of simulation tools[20]. However in this thesis, the approach has been modified to use a hybrid model which uses analytical methods to create a baseline design before creating a simulation study. The layout design methodology used in this thesis is based on the material presented in this section.

The first three phases of the flow chart in Figure 25 are used to focus the involved department members towards a common project objective. The process begins with need analysis and creating proposals of the new line design. The proposed design is refined through an iterative loop to find an optimal solution. The optimal solution is then converted to an usable action document like factory floor plans[20]. Project kick-off meetings, department interviews, and project outline discussions are framed at this stage. Once the project charter is ready the management and involved members sign-off on the implementation plan.

The next phase involves data collection which consumes the most time but is crucial to the success of the simulation study. Company records like monthly quality reports, machine breakdown logs, company labor standards and machine capacities are known data sources. Other techniques like time trials and value stream mapping are used to generate data for sections where company records do not exist. This information is used as input for the analytical as well as simulation model[20].

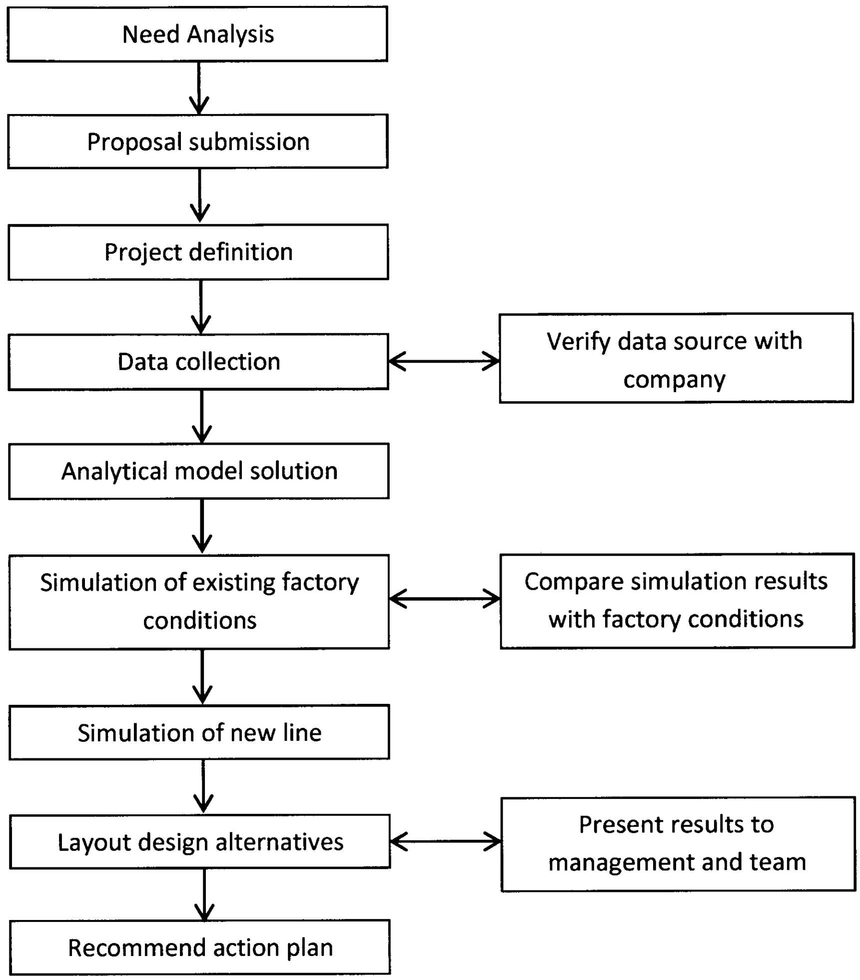


Figure 25: Factory layout project framework [20]

The next step in the process is simulation of new line. Factory simulation is a complicated process that takes several iterations before any usable results emerge. However, simulation techniques are a reliable tool for factory layout design as the process relies on scientific methods and not the intuition of a designer [21]. In fact, some organizations will not make important production decisions without simulating the impact on key factory parameters [22].

Figure 26 lists all the steps needed to execute a successful simulation study. We need to first define the simulation goals like target process cycle times, work scheduling policy, changeover time, etc. [4]. The next step involves creation of the baseline model and collecting data. There will be instances where no data is available and decisions have to be made based on the customer's intuition about the system. A sensitivity analysis of such decisions should be done to understand its impact on the output. Once the simulation model is ready it should be set up to emulate existing shop floor conditions and the results should be checked for consistency with actual factory conditions[23].

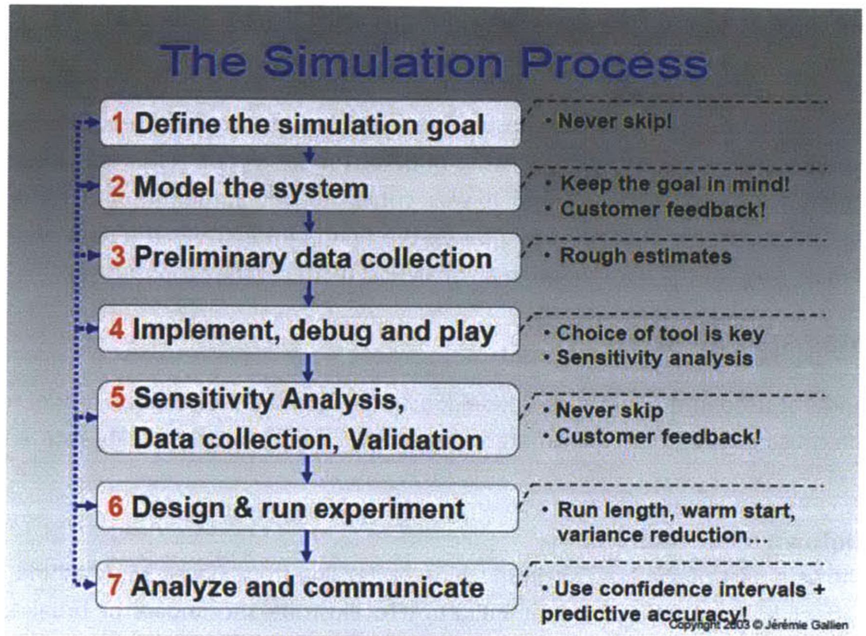


Figure 26: Simulation study steps [24]

The next step involves iterating different scenarios (experiments) in the simulation. It should be noted that customer feedback is crucial in each steps described in Figure 26 and more so in designing the scenarios. Parameter settings for the optimal solution should be noted and the results should be presented in a usable document like factory floor plans. The findings should be communicated to the customer in all member presentations and closing meetings.

6. Machine specification and analytical methods

This chapter outlines the process of data generation and the use of data to create system level definitions of stations used in the proposed line. Machine logs, quality reports, company labor standards and time trails were primary sources of data. The simulation relies on work presented in this chapter for its model accuracy and it was critical to get authentic data for all important stations[25]. The chapter is divided in two parts, the first half describes the data analysis and the second half makes use of the data to frame an analytical model of the proposed line.

6.1 Machine specification

All the equipment used in the analytical model (Section 6.2) and the simulation model from Chapter 7 are specified using three basic parameters: breakdown trend, output quality rate and operation time.

6.1.1 Breakdown trend analysis

Figure 27 shows a part of the long chain machine buffer line from Figure 35. Machine breakdowns drastically affect line output and buffer build-up. To illustrate the impact of breakdown on line behavior, consider SMT2 shown in Figure 27 to be under breakdown.

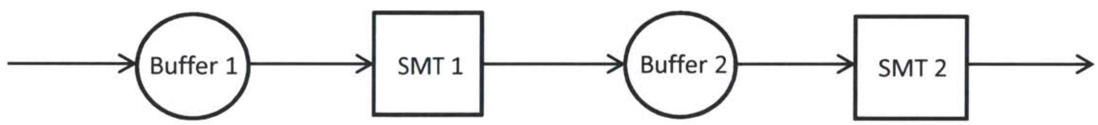


Figure 27: A section of the long machine buffer line from section 6.2.1

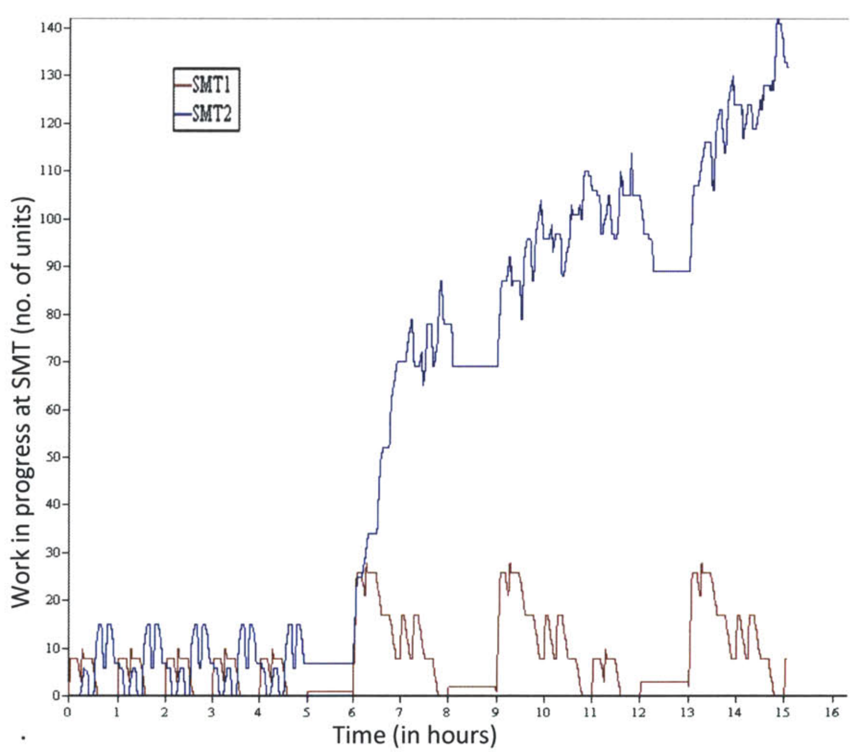


Figure 28: Effect of SMT machine breakdown on WIP inventory

Figure 28 is a snapshot (from the simulation in chapter 7) showing number of parts in buffers before each SMT machine. The blue line in the graph is inventory in Buffer 2 and the red line is showing inventory in Buffer 1. The graph shows that parts get accumulated in buffers 2 when SMT 2 goes down at 6 hours and stays down until about 7 hours.

Machine breakdown induces variability in the system and breakdown times are almost always randomly distributed. As mentioned in the introduction of this chapter, machine logs were used to access breakdown data for the wave solder. A record with such metrics for the wave solder machine is seen in Table 2. The data was compiled to generate metrics like mean time to repair (MTTR) and mean time to fail (MTTF). MTTF and MTTR numbers are important in defining machine reliability in the analytical model in Section 6.2. The values in Table 2 were validated during project meetings with the manufacturing engineering team. A collection of machine breakdown data table for other equipment is available in the Appendix Section A.

Table 2: Wave machine breakdown record

Breakdown calculations for Wave (in minutes)			
Date	**Time**	**Downtime (MTTR)**	**Uptime (MTTF)**
20-12-2012	08:30	25	
26-12-2012	07:30	30	3000
28-12-2012	10:45	45	1215
21-01-2013	07:00	110	12015
25-01-2013	10:30	50	2250
29-01-2013	03:00	30	1590
14-02-2013	07:35	15	8435
20-02-2013	11:00	90	3265
27-02-2013	10:00	90	3510
07-03-2013	13:15	45	4275
11-03-2013	07:00	45	1665
14-03-2013	13:20	35	1910
15-03-2013	07:00	30	130
17-03-2013	07:10	30	1030
19-03-2013	09:15	105	1145
18-04-2013	07:00	120	15165
18-04-2013	09:00	60	120
09-05-2013	07:00	450	10590
14-05-2013	07:00	40	2550
03-06-2013	10:00	81	10380
08-06-2013	07:00	210	2370
13-06-2013	11:00	90	2790
13-06-2013	12:30	180	90
18-06-2013	11:00	90	2460
		90	**3997.8**

The uptimes from Table 2 were inputted to a curve fitting tool called 'Arena input analyzer' to identify a trend in the data that matches with the standard random distributions[2]. The Input analyzer uses a mean square calculation to compare the data trend with standard random distributions. The input analyzer identifies the best fit and outputs statistics like fit error and the curve fit expression.

Formula (2) was used for mean square error calculations by the Input analyzer

$$\text{Mean Square error} = \sum_{i=1}^{n} r^2 \qquad (2)$$

Where, $r = y_i - f(x_i, \beta)$ and 'f' is the model function or the true exponential curve in this case

A snap shot of the curve fit for data in Table 2 (wave solder uptimes) is seen in Figure 29 and it is exponentially distributed with a mean of 15 hours. The graph in Figure 29 was generated using JMP[26]. The box plot represents the machine data and the red curve refers to a standard fit (in this case exponential) curve being superimposed on the data.

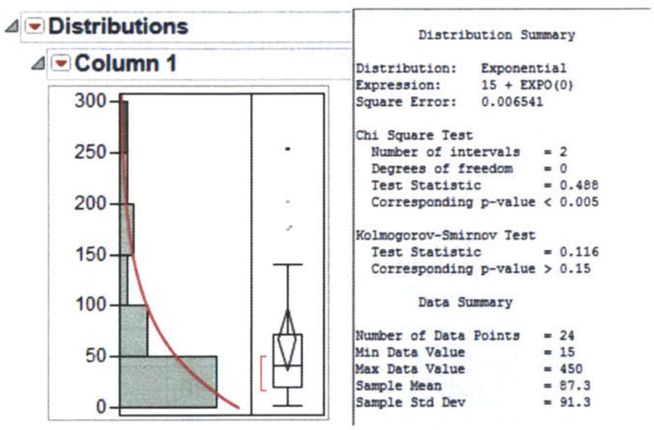


Figure 29: Curve fitting and goodness of fit test for the wave solder breakdown times

A similar analysis was carried out for the other equipment and fit expressions for all the machines were obtained Figure 30 shows the input screen for a failure data set that drives the machine availability of all machines in the simulation model from chapter 7.

Failure - Advanced Process

	Name	Type	Up Time	Up Time Units	Count	Down Time	Down Time Units
1	SMTfailure	Time	2+EXPO(42.4)	Hours	2+EXPO(42.4)	5 + EXPO(44.6)	Minutes
2	Wavefailure	Time	1 + EXPO(65.6)	Hours	1 + EXPO(65.6)	15 + EXPO(72.3)	Minutes
3	CCfailure	Time	TRIA(8, 23.9, 170)	Hours	TRIA(8, 23.9, 170)	TRIA(15, 34, 210)	Minutes
4	Boardtestfailure	Count	1.0	Hours	1000	0.83	Minutes
5	Finaltestfailure	Count	1.0	Hours	1000	0.83	Minutes

Figure 30: Breakdown expressions inputted into the simulation package

[2] The curve fitting software uses Beta, Empirical, Erlang, Exponential, Johnson, Gamma, Lognormal, Normal, Poisson, Triangular, Weibull and Uniform distributions in its standard set of random distributions.

6.1.2 Quality rates at test stations

The proposed line has three test stations which is consistent with what exists on the actual factory floor. The company maintains weekly test data at the board and drive test stations which is used to create a consolidated monthly quality report. A sample of test reports for the AOI and board test stations is seen in Table 3 and Table 4.

Table 3: Consolidated AOI inspection quality rates

AOI Yield

Date	Board Failures	Total Boards	Yield %
Jan-2013	58	3705	98.4%
Feb-2013	85	2362	96.4%
Mar-2013	144	3347	95.7%
Apr-2013	165	3230	94.9%
May-2013	161	3453	95.3%

The data from Tables 3 and 4 was processed using the input analyzer mentioned in Section 6.1.1 and the resulting curve fit expressions were used in the simulation study. Defect rates drove the test station behavior in the simulation and the test data was also used to define yield rates for machines in the analytical model in Section 6.2.

Table 4: Board test and drive test quality reports

	Board Test	Average reject rate		Drive Test	Average reject rate
Month	Total	%	Month	Total	%
Jun-12	11858	2.50%	Jun-12	6343	2.90%
Jul-12	13296	1.90%	Jul-12	6493	2.20%
Aug-12	12625	2.20%	Aug-12	7835	2.20%
Sep-12	15517	2.10%	Sep-12	7416	1.90%
Oct-12	9289	1.90%	Oct-12	5827	1.90%
Nov-12	12004	1.60%	Nov-12	6091	1.60%
Dec-12	10755	1.45%	Dec-12	4297	1.40%
Jan-13	14207	1.39%	Jan-13	6392	2.38%
Feb-13	12700	1.40%	Feb-13	5923	2.40%
Mar-13	14536	0	Mar-13	6942	1.69%
Apr-13	11858	0	Apr-13	6365	1.80%
May-13	16191	0	May-13	7345	2.46%

Unlike quality data, operation time for each machine could not be used directly from the company data logs, it had to be calculated using a different approach for each station.

6.1.3 Final assembly operation time

Final assembly process consists of multiple small steps that integrate the different drive components manually. Final assembly operation times depend on product construction and operator skill. SMX final assembly was anticipated to be quicker than SMV final assembly because of an easier cover assembly and the double-sided PCB. Time estimates for SMX assembly times were obtained by analyzing data on existing products which were similar in construction to the SMX drives. Some SMX assembly processes were simulated and timed to get precise time numbers. A summary sheet presenting SMX build times is available in the Appendix Section B. The assembly times for the two products, SMX1 and SMX2, were identical because of the similarity in mechanical assembly of the drives.

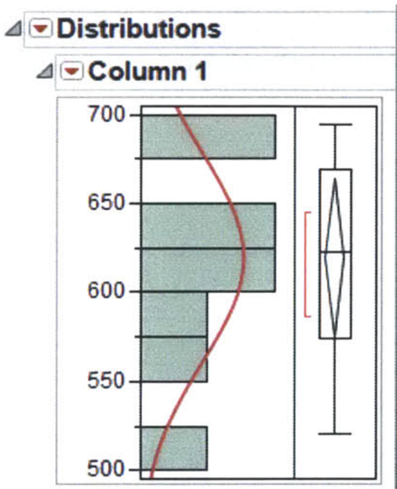
Table 5: Drive assembly time-study summary

	Process step	Average time in seconds
Assembly work bench processes	Apply Silicone	18.3
	Apply Ins. Patch	12.1
	Install spacers/collars/Zip tie holders	44.0
	Install bottom components	22.6
	Place board and place screws onto board	33.9
	Tighten screws	31.8
	Install bracket	11.1
	Clip leads	33.3
	Dump leads	9.4
	Inspect	5.4
Solder work bench processes	Pick drive	8.0
	Solder (33 joints)	101.1
	Inserting capacitor (small orange)	7.8
	Cleaning with alcohol	10.0
	Insert tie wrap x2	14.2
	Apply silicon (2 spots)	10.2
	Insert capacitors (2 no)	18.7
	Soldering caps and orange caps (4 joints)	32.4
	Clip leads of orange cap	3.7
	Tighten and clip tie wrap	9.1
Inspection work bench processes	Pick drive	3.8
	Inspection	38.2
	Cover assembly	22.2
	Insert fixture and topple	5.8
	Color machined surf	21.2
	Insert screw and torque	60.6
	Read bar code	8.6
	Print	5.7
	Stick label	10.4
	Scan new bar code	7.1
	Put finish part in bin	4.4

The SMV product line is divided into categories based on power handling capacities and the product application. The company had been selling SMV drives for a long time and company labor standards were available for each model. The company labor standards for the past two months were used to create two representative *entities*: SMV1 and SMV2 that will be used in the simulation in chapter 7.

The company labor standards provided constant operation time values and time trials had to be undertaken to understand the variability in assembly times. The assembly time data trend is expected to be normal because assembly is an addition of small independent steps. The data presented in Table 5 was recorded during one such time study during a Kaizen event described in Chapter 4. A more detailed split of the drive assembly times is available in the Appendix Section B.

The data from Table 5 was processed using the input analyzer to identify a fit from standard random distributions. Figure 31 shows the result of the curve fit that suggests the data confirms to a normal distribution with a mean of 620 seconds and standard deviation of 54.7 second.




```
Distribution Summary

Distribution:   Normal
Expression:     NORM(620, 54.7)
Square Error:   0.009299

Kolmogorov-Smirnov Test
  Test Statistic          = 0.134
  Corresponding p-value > 0.15

Data Summary

Number of Data Points     = 9
Min Data Value            = 521
Max Data Value            = 695
Sample Mean               = 620
Sample Std Dev            = 58
```

Figure 31: Curve fitting and goodness of fit test of the drive assembly operation times

6.1.4 Wave solder and Conformal coat operation time

Operation times on wave solder and conformal coat stations were obtained from the company labor standards. The wave solder machine is capable of processing multiple parts at a time but the labor standard data specified the total time a part spends in the machine. This meant that the labor times had to be revised to represent the actual operation time per drive. Figure 32 and Equation (3) explain the fixture placement used to arrive at operation times for Wave and CC machines.

$$\text{Wave solder machine capacity} = \frac{\text{Bed length}}{\text{Pitch of fixture placement}} * 2 \qquad (3)$$

(We multiply by two in the above formula as each fixture houses two PCBs.)

$= \frac{180*2}{36}$ = 10 boards per bed length

∴ Revised wave solder operation time as per equation (3) = 5/10= 0.5 minutes per board.

Operation time calculations on wave solder and CC:

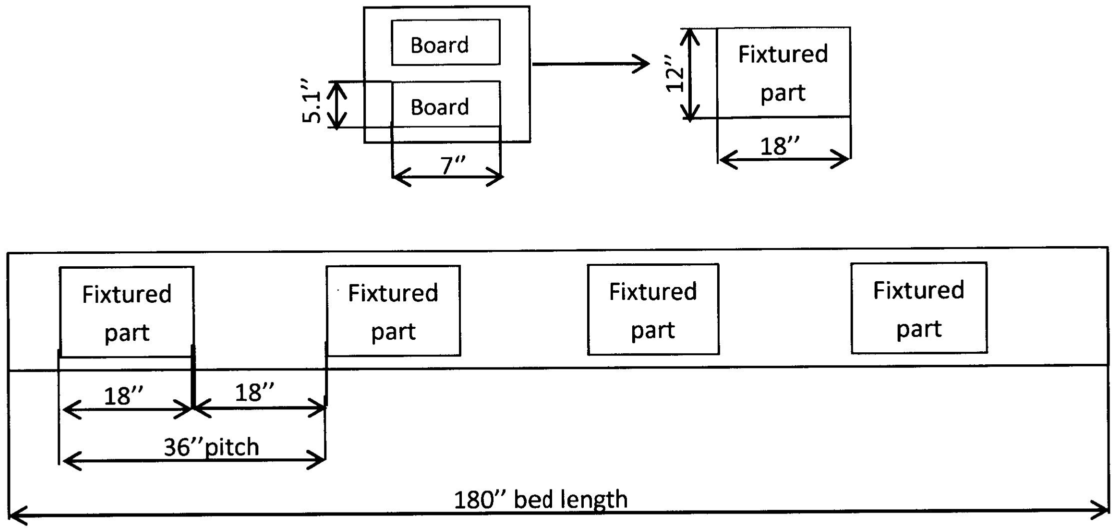


Figure 32: Component loading and wave solder bed dimensions

The CC machine operates similar to the wave solder and the calculations below will obtain the operation times on CC.

Conformal coat bed length= 193 inches

Pitch of fixture placement= 30 inches

∴ Revised CC operation time = 0.416 minutes per board

6.1.5 SMT operation time

SMT machines are highly automated and the operation times do not vary much. The placement times (operation time for SMT) for SMV drives were obtained from company labor standards and empirical values were used for SMX drives.

The total SMT processing times for SMX drives was anticipated to be 4.07 minutes for 430 part placements (337 on top and 93 on bottom side of the PCB). The SMX boards were processed on two SMT pick and place machines and the estimated operation time had to be split between the two machines. The top side of SMX boards is processed at SMT1 and the bottom side is processed at SMT2. The calculations presented below describe operation times at each SMT station. Equation (4) is used to determine the time required to process each side of the PCB.

$$\text{Operation time for SMT1} = \frac{Total\ SMT\ cycle\ time * Number\ of\ parts\ on\ top\ side}{Total\ number\ of\ parts\ on\ the\ PCB} \quad (4)$$

$$= \frac{4.07 * 333}{430} = 3.18 \text{ minutes}$$

(There are 337 components on the top and 93 on the bottom side of the SMX PCB.)

Operation time for SMT2= Total processing time - SMT 1 operation time = 4.07-3.18= 0.88 minutes

6.1.6 Summary of operation times

Operation times for through hole, touch-up and packaging stations were obtained from company labor standards for SMV drives and empirical values were assumed for SMX drives as there was no labor standards available for SMX drives. Table 6 presents a summary of all the operation times of processes used to manufacture SMX and SMV drives.

Table 6: Summary of operation times and breakdown expressions for all stations

			SMX1	SMX2	SMV1	SMV2
Sr no.	Machine name	Break down expression (in hours)	Operation time (in minutes)	Operation time (in minutes)	Operation time (in minutes)	Operation time (in minutes)
1	SMT top	5 + EXPO(44.6)	3.18	3.18	-	-
2	SMT bottom	6 + EXPO(44.6)	0.89	0.89	1.39	2.33
3	AOI	-	1.00	1.00	1.00	1.66
4	AOI rework	-	1.00	1.00	1.00	1.00
5	TH prep	-	0.13	0.15	0.69	2.16
6	TH assembly	-	2.72	3.00	2.72	2.72
7	Wave	15 + EXPO(72.3)	0.50	0.50	0.51	1.01
8	Touchup	-	1.00	1.00	6.97	13.64
9	Board test	1000 runs	3.00	3.00	3.72	5.48
10	CC	TRIA(15, 34, 210)	0.42	0.42	-	-
11	Drive assembly	-	4.44	5.00	9.89	12.92
12	Final test	1000 runs	10.00	10.00	7.00	7.00

6.1.7 Break schedule

The shop floor operates on a two shift per day schedule with five breaks per day. All equipment and machine operators stop production during break times, and it was important to model this detail in the simulation. Break hours affect resource behavior and daily output numbers as seen in Figure 33. The graph shown in Figure 33 is the output of the simulated line from Chapter 7 and the effect of breaks can be seen as zero slope lines depicting no production during break hours.

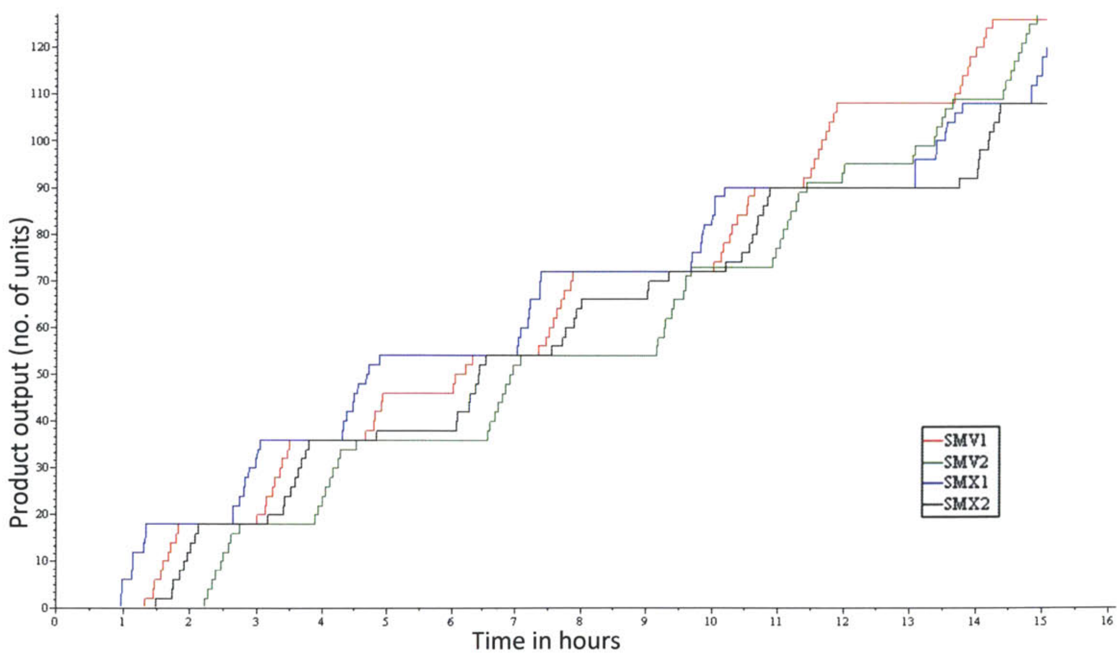


Figure 33: Simulation output over a period of two shifts

The simulation software makes use of a data module by the name 'schedule' (Figure 34) which freezes machine availability during break hours.

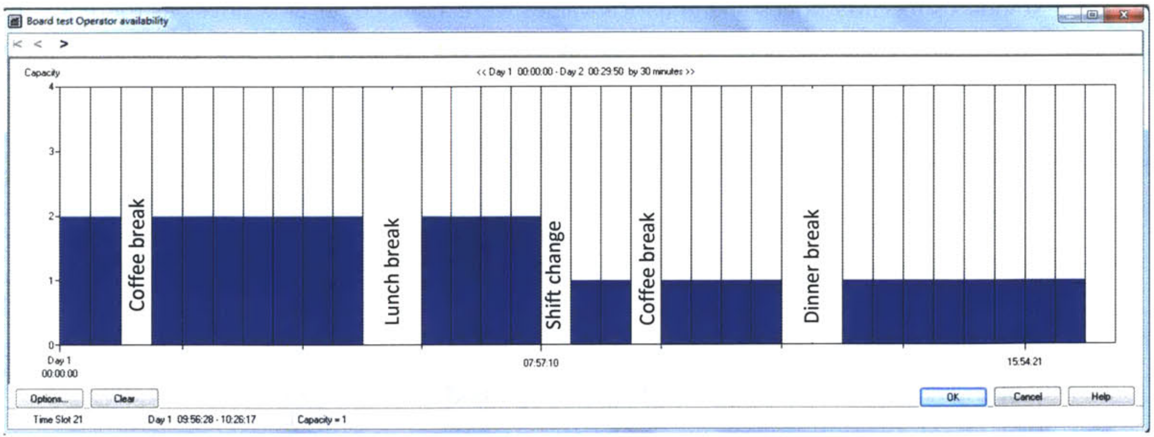


Figure 34: Break time modeling using a *schedule* data module

6.2 Analytical methods

This section describes an analytical solution to a simplified long machine buffer line problem. A long chain machine line was modeled in Microsoft Excel and Cell1[3] to test different line configurations. The baseline capacities for number of equipment were developed from the analytical solution and the values were then inputted into a simulation model (chapter 7) [27].

Figure 35 shows the machine buffer lines designed using the operation plan for SMX and SMV drives discussed in Section 3.2.1 and 3.2.2. Squares represent machine stations and circles show inter-stage buffer space. The entire system had to run on an extremely low WIP and the buffer sizes were to be restricted by the amount of space on the conveyor between machines.

6.2.1 Long chain analysis in Excel

A Microsoft Excel sheet (Table 7) was set up using MTTF, MTTR and defect rate data from Section 6.1. Table 7 shows the Excel sheet with additional columns that were created using formulae (5) to (10). The right most column 'Isolated production rate' was used to determine if the line is balanced. Values in Table 7 are derived using a single resource at each station.

$$\text{Processing rate} = \frac{1}{Operating\ time} \quad (5)$$

$$\text{Yield} = \frac{1}{Defect\ rate} \quad (6)$$

$$\text{Failure rate} = \frac{1}{MTTF} \quad (7)$$

$$\text{Repair rate} = \frac{1}{MTTR} \quad (8)$$

$$\text{Isolated efficiency} = \frac{MTTF}{MTTF+MTTR} \quad (9)$$

Isolated production rate = Isolated efficiency x Processing rate (10)

[3] Developed by Stanley Gershwin, Laboratory for Manufacturing and Productivity, MIT.
Refer [28] and [29] for a detailed description of the decomposition technique and long transfer line analysis

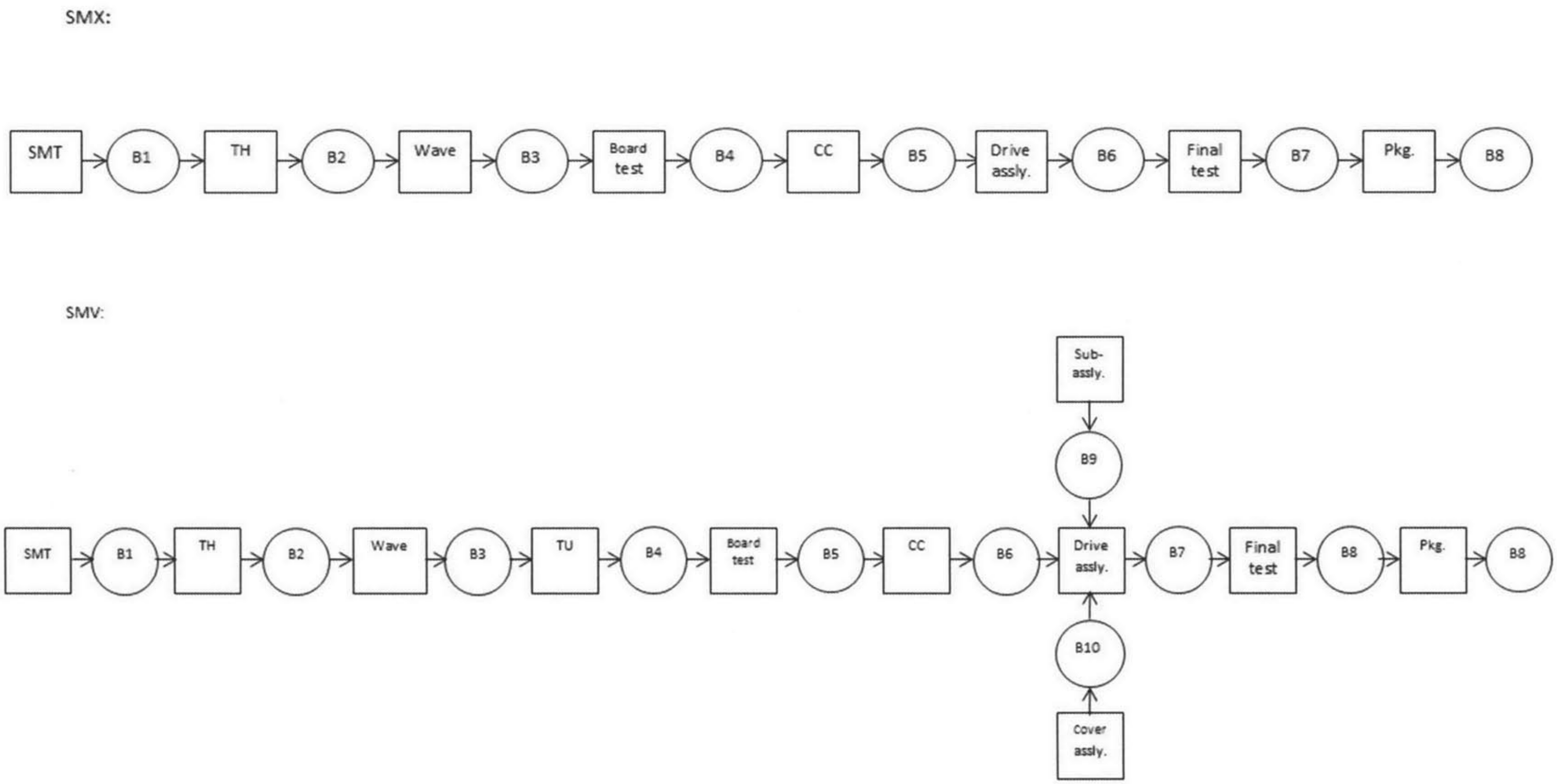


Figure 35: Long line design for SMX and SMV production

Table 7: Initial long chain table values

Name of station	Operation time (mins)	Processing rate (per min)	Defect rate	Yield rate	MTTF (mins)	Failure rate (per min)	MTTR (mins)	Repair rate (per min)	Isolated Efficiency	Isolated Prod. rate
SMT	4.07	0.25	0.01	0.99	2665	0.0003752	71.3	0.014025	0.97	0.24
TH	2.85	0.35	0.0001	1	999999	0.00001	0	0.9999	1.00	0.35
Wave	3	0.33	0.01	0.99	3998	0.0002501	82.67	0.012096	0.98	0.33
Board Test	3	0.33	0.015	0.985	1560.6	0.0006408	49.8	0.02008	0.97	0.32
CC	2	0.50	0.0001	0.999	4038	0.0002476	86.33	0.011583	0.98	0.49
Drive Assembly	4.44	0.23	0.0288	0.971	999999	0.00001	0	0.9999	1.00	0.23
Final test	4.69	0.21	0.0073	0.993	1560.6	0.0006408	49.8	0.02008	0.97	0.21
Pkg	2.5	0.40	0.0001	1	999999	0.00001	0	0.9999	1.00	0.40

Analysis of Table 7:

The isolated production rate is a station's production rate when it is running independently or decoupled from other equipment on the line. The entries highlighted in yellow indicate a low isolated production rate and the next table will be framed with revisions to the capacity numbers of the highlighted stations.

Proposed changes and their effect:

Every change made to capacity numbers in Table 7 and 8 was checked for feasibility by discussing the results with the manufacturing team at the company during project meetings. Table 7 recommends an increase in drive assembly and final test capacities. Both the stations are manually operated and a change in capacity was feasible. The final test station will need additional test fixtures in addition to manpower. The SMT station operation times can be decreased by splitting the number of component placement amongst multiple pick and place machines. Table 7 was revised with revised capacities at the three highlighted stations and the result is seen in Table 8.

Table 8 shows the final equipment capacities with almost uniform isolated production rates at each station. This table was generated with adding capacities at SMT, drive test and final test. TH operation time was modified slightly as it is a manual process and some of the tasks can be transferred to TH prep. It was not possible to increase capacity at any other stations and the equipment capacity numbers were ready for input to Cell1. Board test stands out as a strong bottleneck in Table 8.

Table 8 : Final long chain table values

Name of station	Operation time (mins)	Processing rate (per min)	Defect rate	Yield rate	MTTF (mins)	Failure rate (per min)	MTTR (mins)	Repair rate (per min)	Isolated Efficiency	Isolated Prod. rate
SMT	2.035	0.49	0.01	0.99	2665	0.0003752	71.3	0.014025	0.97	0.48
TH	2.5	0.40	0.0001	1	999999	0.00001	0	0.9999	1.00	0.40
Wave	2	0.50	0.01	0.99	3998	0.0002501	82.67	0.012096	0.98	0.49
Board Test	3	0.33	0.015	0.985	1560.6	0.0006408	49.8	0.02008	0.97	0.32
CC	2	0.50	0.0001	0.999	4038	0.0002476	86.33	0.011583	0.98	0.49
Drive Assembly	2.22	0.45	0.0288	0.971	999999	0.00001	0	0.9999	1.00	0.45
Final test	2.345	0.43	0.0073	0.993	1560.6	0.0006408	49.8	0.02008	0.97	0.41
Pkg	2.5	0.40	0.0001	1	999999	0.00001	0	0.9999	1.00	0.40

6.2.2 Baseline solution using an analytical tool: Cell 1

Cell 1 is manufacturing systems software capable of analyzing two-machine systems and long transfer lines[28]. Two machines can be evaluated exactly however; there is no exact analytical method to evaluate long lines. Evaluation of long lines is done using decomposition techniques to represent the long lines as a two machine line [29]. The 'continuous material long line' model from Cell1 fits our requirements and a screen shot of the input data page can be seen in Figure 36. Metrics like buffer size, overall production rate and individual station efficiency are seen in the output screen in Figure 36.

Assumptions in Cell1 Continuous material long line model:

- Exponentially distributed failure and repair times
- Finite buffer sizes
- Constant processing times
- Yield rates are randomly distributed

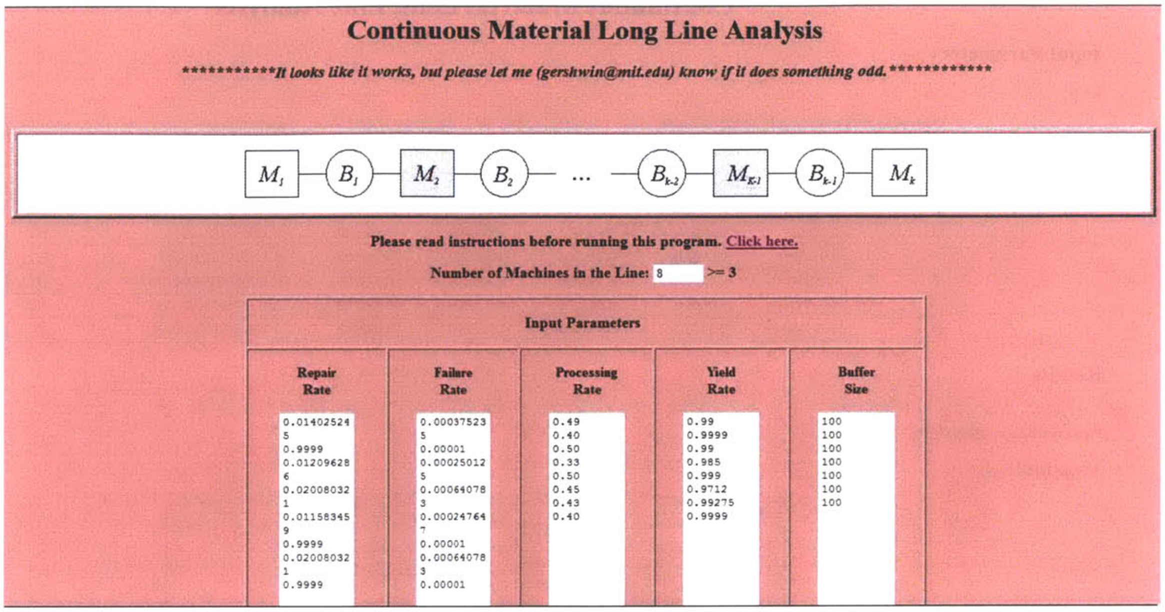
Exponential repair times will mean that the repair times are assumed to hold a constant value most of the time[30]. This is not true in the actual line operation and the analytical tools result may vary a little bit due to this discrepancy.

Figure 37 shows the results from Cell1 using the data in Table 8 as input. The results show that all buffers until buffer 3 (highlighted in Figure 37) are full. This indicates that machine 4 is a bottleneck and its slower operation is blocking all buffers prior to it. This result matches with the Excel model in Section 6.2.1 where board test (machine number 4) was identified as a bottleneck. This verifies the Cell1 setting and we can proceed to test different station capacities in Cell1.

As station 3 was the bottleneck, the capacity numbers at station 3 were revised and the next set of results was obtained. The process of revising capacity numbers was continued until no more changes were feasible.

Figure 38 shows results with the final line capacity numbers. The results show only one buffer reaching its full capacity. However, the line cannot be changed further as alterations would mean adding a new SMT line or modifying the sequence of operation, both of which were not possible. The input values for the results presented in Figure 38 are presented in the top half of Figure 38 and the new line production rate is 0.362 parts per minute.

It should also be noted that the buffer sizes have now been decreased to 12 parts. The model in Figure 37 was run with excessive buffer space to find the optimal solution for equipment capacities without the influence from blockages due to buffer size limitations. The final buffer sizes were determined later by keeping the equipment capacities constant.



Continuous Material Long Line Analysis

Input Parameters

k = 8

Machine	r	p	e	mu	y	iso. prod. rate	N
1	0.014025	0.000375	0.973943	0.490000	0.990000	0.4724671875	100
2	0.999900	0.000010	0.999990	0.400000	0.999900	0.399956000040004	100
3	0.012096	0.000250	0.979741	0.500000	0.990000	0.484976510610724	100
4	0.020080	0.000641	0.969076	0.330000	0.985000	0.31499464311568	100
5	0.011583	0.000248	0.979068	0.500000	0.999000	0.489029541036261	100
6	0.999900	0.000010	0.999990	0.450000	0.971200	0.437035629206629	100
7	0.020080	0.000641	0.969076	0.430000	0.992750	0.413676975049467	100
8	0.999900	0.000010	0.999990	0.400000	0.999900	0.399956000040004	0

Figure 36: Initial Cell1 values

Results

Production Rate = 0.3031785670

Average Buffer Levels:

Buffer	n bar
1	98.7129436872
2	99.9844511174
3	99.0259524608
4	0.9568532147
5	0.0006220959
6	1.3894853293
7	0.0000135992

Figure 37: Results from baseline model in Cell1

Continuous Material Long Line Analysis

Input Parameters

k = 8

Machine	r	p	e	mu	y	iso. prod. rate	N
1	0.014025	0.000375	0.973943	0.490000	0.990000	0.4724671875	12
2	0.999900	0.000010	0.999990	0.400000	0.999930	0.399967999919993	12
3	0.012096	0.000250	0.979741	0.500000	0.990000	0.484976510610724	12
4	0.020080	0.000641	0.969076	0.660000	0.985000	0.62998928623136	12
5	0.011583	0.000248	0.979068	0.500000	0.999000	0.489029541036261	12
6	0.999900	0.000010	0.999990	0.450000	0.971200	0.437035629206629	12
7	0.020080	0.000641	0.969076	0.430000	0.992750	0.413676975049467	12
8	0.999900	0.000010	0.999990	0.400000	0.999900	0.399956000040004	0

Results

Production Rate = 0.3616852080

Average Buffer Levels:

Buffer	n bar
1	11.6049528149
2	0.6379022150
3	0.4357408441
4	0.4227493495
5	0.4365088256
6	0.9073859078
7	0.2995950965

Figure 38: Final Cell1 values

After the final equipment capacities were obtained, an analysis was done to determine the inter-stage buffer sizes. The buffer sizes had to be restricted below 20 as buffer space was limited by conveyor length between machines. The buffer size was varied to monitor the effect of buffer sizes on the line's production rate. Figure 39 shows the change in the overall line production rate with increasing buffer size. It can be seen that the line production increases considerably until about 12 part buffer but after that the increase in line production is less and we would be investing in excessive buffer space to gain marginal benefits on line efficiency.

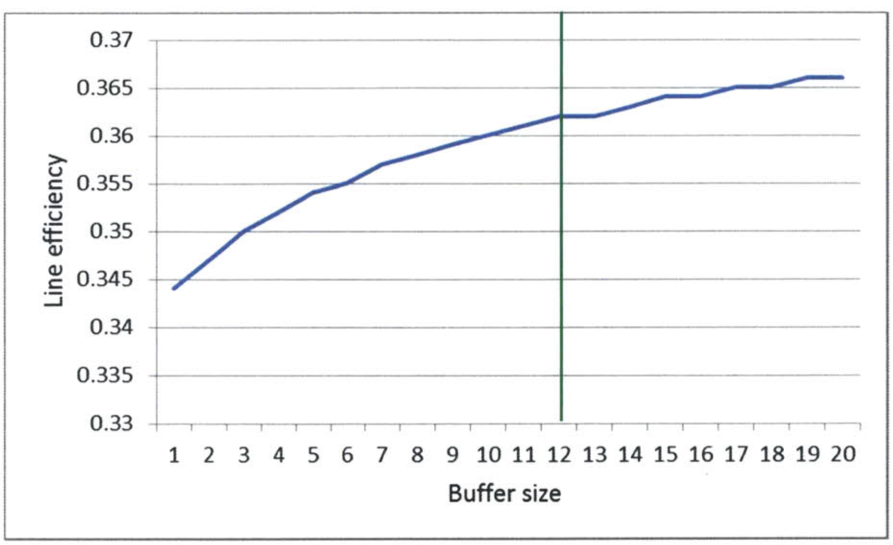


Figure 39: Line efficiency vs. buffer size to determine optimal buffer size

The final baseline plan derived from Figure 38 is shown in Table 9. The baseline capacities defined in the table serve as input to the simulation model in the next chapter.

Table 9: Baseline capacities from the analytical solution

	Name of station	Resource capacity
1	SMT	3 Pick and place machines
2	TH	2 Operators
3	AOI	1 AOI machine
4	Wave	1 Wave machine
5	Board Test	2 Board test fixtures per product
6	CC	1 CC machine
7	Drive Assembly	5 Assembly stations
8	Final test	2 Final test fixtures per product
9	Packaging	1 Operator

7. Simulation

The goal of this chapter is to educate the reader about the use of simulation techniques to create a virtual representation of an actual factory. The development of a simulation model was one of the main expectations from the thesis project. The simulation model will be used to predict the impact of changes to the factory layout and the impact of introduction of SMX. The company had planned to buy a license of Arena DSS to maintain and further develop the model.

This chapter will describe in detail the development of an Arena (software used for simulation) model and its overall structure. Every simulation package is developed uniquely and some topics presented in this chapter require a basic understanding of Arena. A glossary of basic Arena features that are frequently used in this chapter is available in Appendix Section C. The author found the book "Simulation with Arena" a great resource in working on the model [31].

7.1 Overview of discrete event simulation

Simulation is a tool for analyzing system behavior by emulating machines, operators and other factory resources in a virtual environment. Simulation models fall into the category of discrete-time simulations or discrete-event simulations. Discrete-time simulations are slower because the system state is recorded at every time unit of the system clock. Discrete-event simulation on the other hand, record and update the system state only when an event (like machine failure or part completion) occurs. In general, discrete event simulations are faster and more flexible but are more difficult to code.

Simulations tools have evolved over time and have found extensive use in creating long term plans and design decisions[32]. The tool used in this project is based on discrete event time. The software utilizes flow chart modules to create the logic of material flow and control strategies. A description of different modules used to design the model is available in the Appendix Section C.

Our objective is to create a representation of the planned factory layout and validate the baseline capacities defined in the analytical model. The model should be able to test various scenarios that will predict the impact of milestone factory configurations. The use of simulation will helps us building lower level factory details like shift timings, running multiple products with different process plans and assembly of products.

7.2 Assumptions in the simulation model

As with every simulation model our model was based on the following assumptions. Cover assembly happens asynchronously with drive assembly.

- All SMVs bypass the CC station.
- The first SMT machine will be run with a low utilization to accommodate production of boards for other lines.
- The wave solder is shown as one resource but the machine is capable of handling multiple products at a time The operation time of this machine is modified to accommodate the machines ability to process multiple parts.
- There are zero breakdowns at manual stations like through hole and drive assembly.
- All final assembly operators are multi-skilled to process both SMV and SMX drives.

7.3 Different blocks of the simulation model

Arena has two modes of operation: flowchart mode and 3-D animation. This section describes the use of basic flow chart elements (herein referred as modules) to create a model of the proposed line and then use it to control a 3-D animation presented in Section 7.4. A module is a basic building block that is used with other modules to define a station on the proposed line[31]. Arena uses two types of modules: data and flowchart module. A flowchart module is dynamic in nature and it will determine the path of an *entity* (product representation in simulation) through the model[31]. A data module is used to characterize machine behavior and machine availability [31]. A glossary of basic Arena modules is available in Appendix Section C and it is recommended to get familiarized with the basic modules as they will be extensively used in the following sections. Each station on the proposed line was modeled using a set of flowchart modules and such a cluster of flowchart modules is herein referred as a block.

After the entire line is modeled, *entities* are introduced into the system. An *entity* is used to represent the four products that will run on the proposed line. As the *entities* travels through different blocks they records data on system performance. The data recorded by *entities* is used to compile a final report available at the end of the simulation.

The proposed line is made up of multiple processes arranged in a sequence as presented in Figure 40 (based on analytical model from Chapter 6). Sections 7.3.1 to 7.3.10 will present a comprehensive description of how simulation logic was created to represent each box in Figure 40. *Entities* are introduced at the first block of the model and they get assembled to compatible *entities* at different locations in the simulation model based on the product construction.

Figure 40: Blocks of the simulation model

7.3.1 SMT area

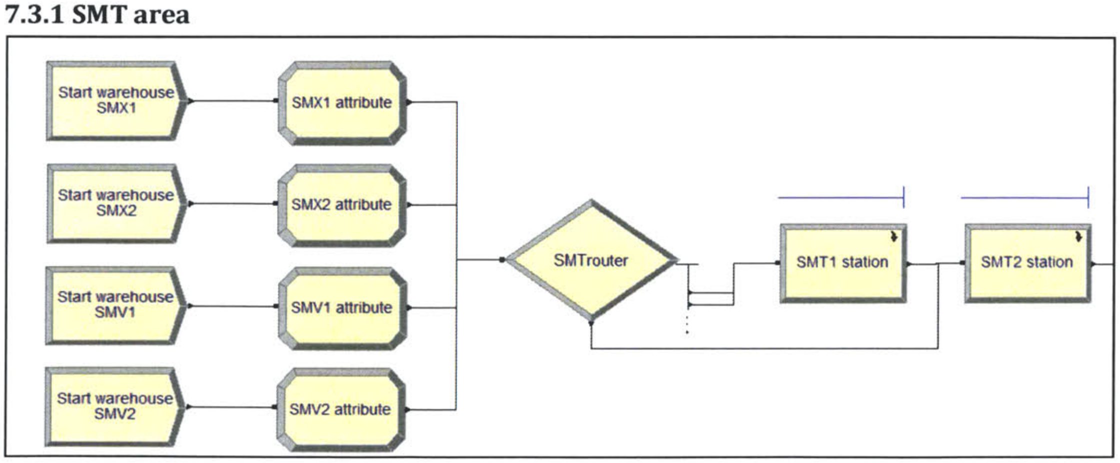


Figure 41: SMT area block logic

The block in Figure 41 is set up to emulate a SMT processing area and will be referred as SMT area. SMT area is the first block in the model and it is thus responsible for *entity* creation. Four 'Start warehouse SMV/SMX' *create* modules are used to generate *entities* known as 'board entity'. After creation, the board *entities* pass through a 'SMV/SMX attribute' *assign* module which assigns operation time and test failure values to each *entity*.

All the *entities* then converge into a 'SMT router' *decide* module which directs *entities* to their appropriate 'SMT station'. The SMV drives bypass 'SMT1 station' and are received directly at 'SMT2 station'. The large number of SMD components on the SMX boards requires two machines to process the boards and thus the SMX *entities* pass through 'SMT1station' and 'SMT2station'.

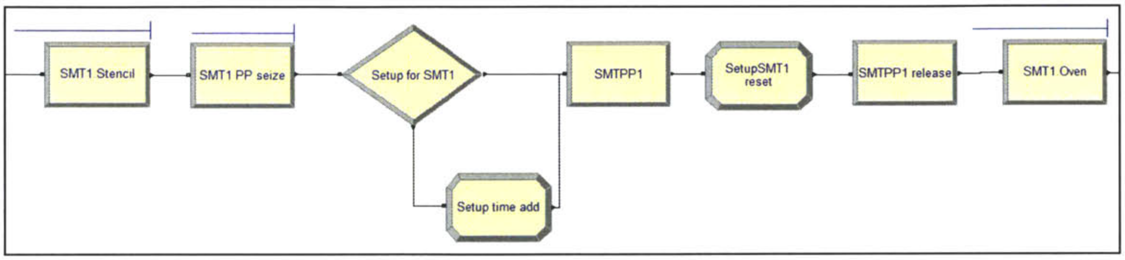


Figure 42: SMT machine *sub-model* showing the setup time logic

An SMT station consists of a series of machines as explained in section 3.1.2. It can get confusing to include each machine's simulation logic on the main simulation screen. Arena tackles this problem with the use of *sub-models*. A *sub-model* is represented by a single module on the simulation screen but any *entity* that enters the *sub-model* is processed through a nested logic that is designed within the *sub-model*. 'SMT1station' and 'SMT2station' in Figure 41 are two such *sub-models* which represent a built-in simulation logic shown in Figure 42.

All the *entities* entering the *sub-model* pass through a *process* module by the name 'SMT1.stencil' which emulates the stencil operation from the SMT1 group. The next module 'SMT1 PP seize' holds parts in a queue until the *resource* PP (pick and place machine) is available. Once released from the 'SMT1PP seize' queue the *entities* arrive at a 'Setup for SMT1' *decide* module which checks the incoming *entities* product type for consistency with the last product processed by the 'SMT1PP1' *process* module. If the *decide* module does not find a match in the *entity* types it passes the *entities* through a 'setup time add' module which adds setup time to the operation time at 'SMTPP1'. This increases the *delay* time at the 'SMTPP1' *process* module. Machine setup is an important aspect in modeling as it drives the data on machine downtime and affects product queues. This data is then used to determine the optimum product mix by running scenarios in chapter 8[33].

Once an *entity* passes through 'SMTPP1', it is transferred to the 'SMTPP1.release' module which releases the 'PP' *resource* for the next *entity's* use. The 'setupSMT1 reset' module is next in line and it resets the setup time to zero and ensures that the system is set to its initial condition.

The 'SMT2station' *sub-model* is based on the same logic as 'SMT1station' however it makes use of three 'SMTPP' stations to provide increased capacity, SMT2 has to processes SMV and SMX drives together and needs this extra capacity.

7.3.2 AOI

AOI testing determines the quality of boards produced in the SMT area. Figure 43 shows the simulation logic for AOI herein referred as AOI block. The AOI block is made of three modules, two in series and one in parallel (as seen in Figure 43). The entering *entities* pass a *process* module 'AOI' which holds the *entity* for *resource* 'AOImachine' to be available. After *seizing* the *resource*, the *entity* is further held for a time equivalent to the 'AOI.testingtime' (defined in the *assign* module in section 7.3.1). The action of an *entity* capturing a *resource* is refered to as *seize* in the simulation. The *entity* is then transferred to a 'AOIcheck' *decide* module which is designed to emulate rejection rate at the AOI station. The expression for "Percent true" in Figure 44 was developed from the analysis in chapter 6 (section 6.1.2). The expression will be used to sort *entities* at the 'AOI check' module. In this case the quality data at the AOI station was found to be triangular in distribution. The software randomly samples a data point from the triangular distribution as mentioned in Figure 44 and then routes the *entities* to either of the branches based on the value of the sample.

The rejected *entities* are routed to the 'AOIrework' *process* module which is used to represent the rework time involved in rectifying the rejected SMT boards.

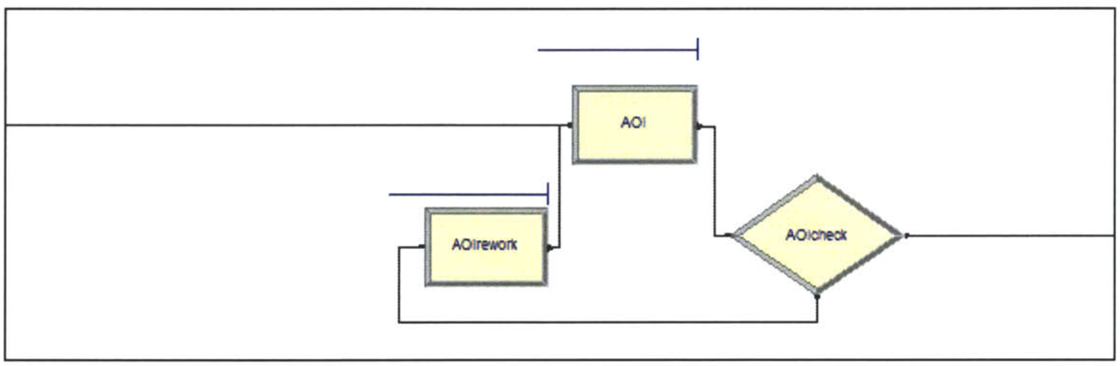


Figure 43: AOI station block logic

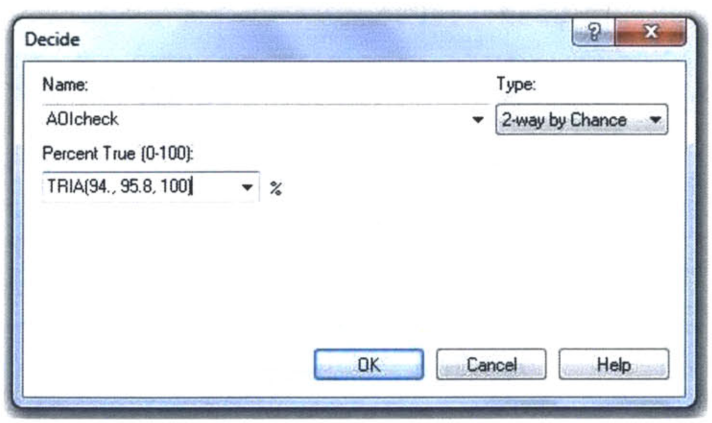


Figure 44: AOI station data module

7.3.3 TH prep and TH assembly

TH assembly is a *batch process* and it integrates 'TH prep' (a new *entity*) with the board *entity* arriving from the 'AOI station'. Figure 45 shows two branches entering the 'TH assembly' *batch* module.

The TH block logic is designed to represent the manual assembly of TH prep components with ST boards. The 'TH assembly' *batch* module combines *entities* regardless of their *entity* types. It was important to ensure that no two similar *entity* types are *batched* together. For example a board *entity* should not be *batched* with another board *entity*. To ensure there is no *batching* of similar *entities*, *entities* are first placed in *hold* modules. The *entities* are released from the *hold* module after scanning the 'TH assembly' *batch* module and verifying if a counterpart *entity* is available.

Figure 45 shows the simulation logic designed to represent the TH operations. Once a board *entity* enters the top chain it is captured in 'TH assembly drive hold' module and is not released until the 'TH assembly hold counter' is zero. As the initial value of the 'TH assembly hold counter' is zero, the board *entity* is released to the 'TH assembly' *batch* module. This *entity* sets the 'TH assembly counter' to one and stops any subsequent board *entity* from entering the 'TH assembly' *batch* module. The board *entity* continues to remain in the *batch* module until a 'TH prep' *entity* reaches the 'TH assembly' *batch* module to satisfy the assembly requirement. The *batched entities* remain together through the rest of the simulation.

The *batched entities* passes through the 'TH assembly counter reset' module and set the *hold counters* on each branch to zero signaling that the *batch* module is ready for the next pair of *entities*. The last module in the chain is a *process* module 'TH assembly process time' which accounts for TH assembly time. It is to be noted that the attributes defined in the *assign* module (Section 7.3.1) determine the *delay* time at the 'TH assembly process time' *process* module.

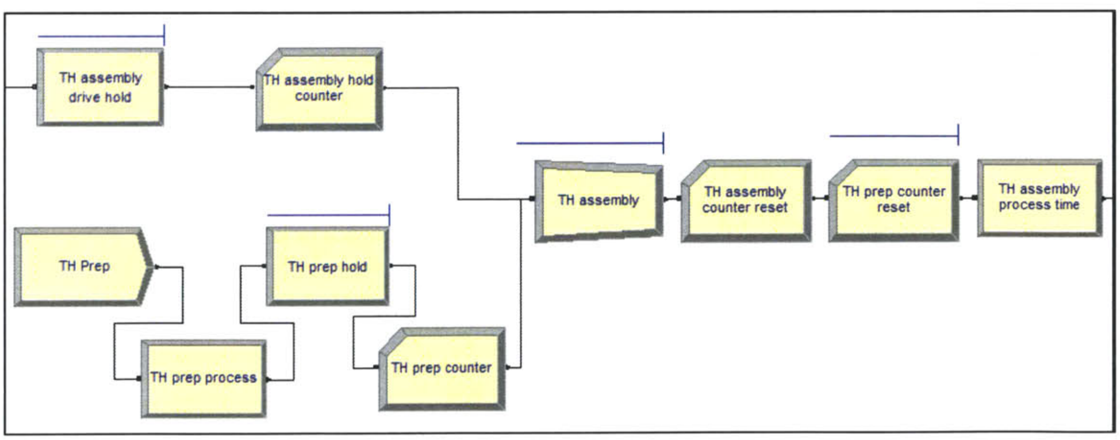


Figure 45: TH assembly and prep block logic

7.3.4 Wave/Touch up (TU)

Figure 46 shows the simulation logic that emulates the wave solder machine behavior. The simulation logic shown in Figure 46 is similar in construction to the 'SMT1station' *sub-model* described in Section 7.3.1. Entering 'board' *entities* are queued at 'Wave seize' awaiting availability of the 'wave' *resource*.

After *seizing* the *wave resource* the *entities* pass through the 'Setup for Wave' *decide* module. The 'Wave w setup' module *delays* progression of the *entities* based on the processing time and setup time. The next module by the name 'Wave release' releases the *wave resource* for the next *entity*.

The last module 'Touch up' is used to represent manual touch up boards exiting the wave solder machine. It is a *process* module and it *delays entities* by a time equivalent to touch-up operation time.

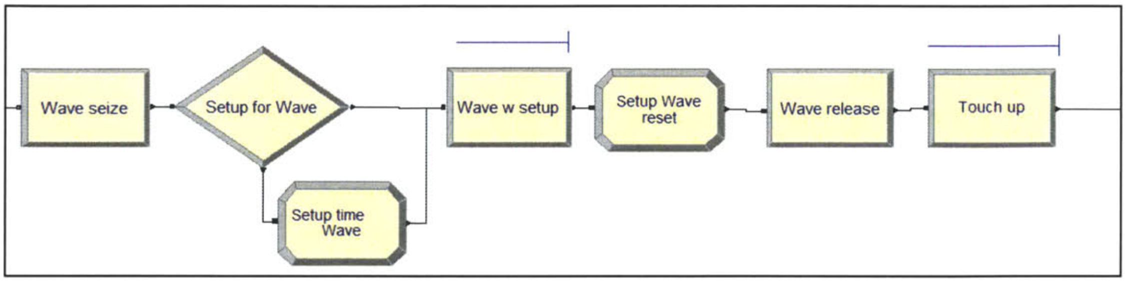


Figure 46: Wave and touch-up block logic

7.3.5 Board test

Figure 47 shows the simulation logic used to represent the board test area operations. SMX and SMV drives use a different board test strategy. The board test block logic shown in Figure 47 has a bifurcated structure to define the test strategies for SMX and SMV drives on each branch.

The incoming *entities* first arrive at the 'Test router' *decide* module and are channeled to a branch based on the *entity* type. Each branch contains a set of modules arranged in a sequence similar to the 'AOI' block described in Section 7.3.2. Test operating times on 'SMX/SMV Test' *process* modules are pulled from the *entity's attribute* list defined in Section 7.3.1.

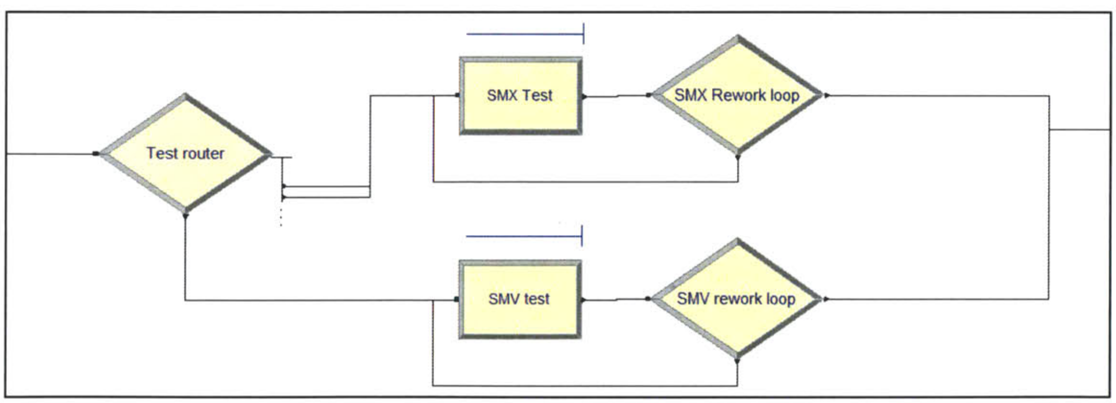


Figure 47: Board test block logic

7.3.6 CC

Figure 48 shows the simulation logic used to represent CC operations. The logic in Figure 48 is similar in arrangement to the wave block from 7.3.4. However, the CC operation time specified in the 'CC work' *delay* module differentiates the CC block from the wave block. The *delay* times for each *entity* are derived from calculations in Section 7.1.4. CC is not part of the SMV build sequence and all SMV *entities* are routed directly to the next block from the 'CC router' *decide* module.

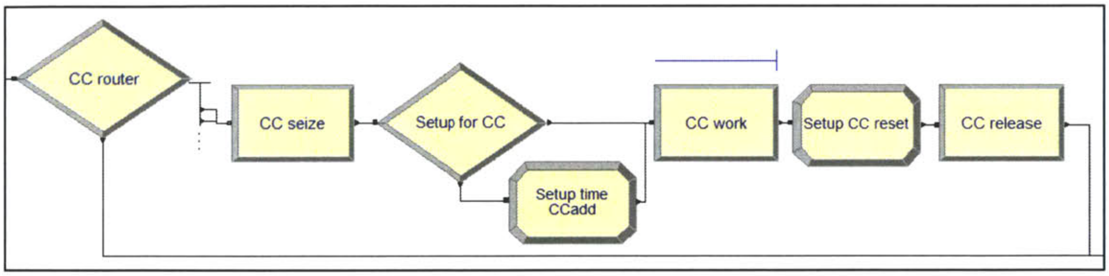


Figure 48: CC block logic

7.3.7 Cover assembly

Figure 49 shows the arrangement of flow chart modules to represent cover assembly operations. Cover assembly area is an offline process that feeds into the main line. As seen in Figure 49 the cover assembly begins with four 'Cover SMX/SMV warehouse' *create* modules that generate a 'cover' *entity* pertaining to each product type. The created *entities* pass through a 'Cover SMX attribute' *assign* module which allocates attributes like 'drive assembly time' and 'entity name'. All 'cover' *entities* then move to the Drive assembly block.

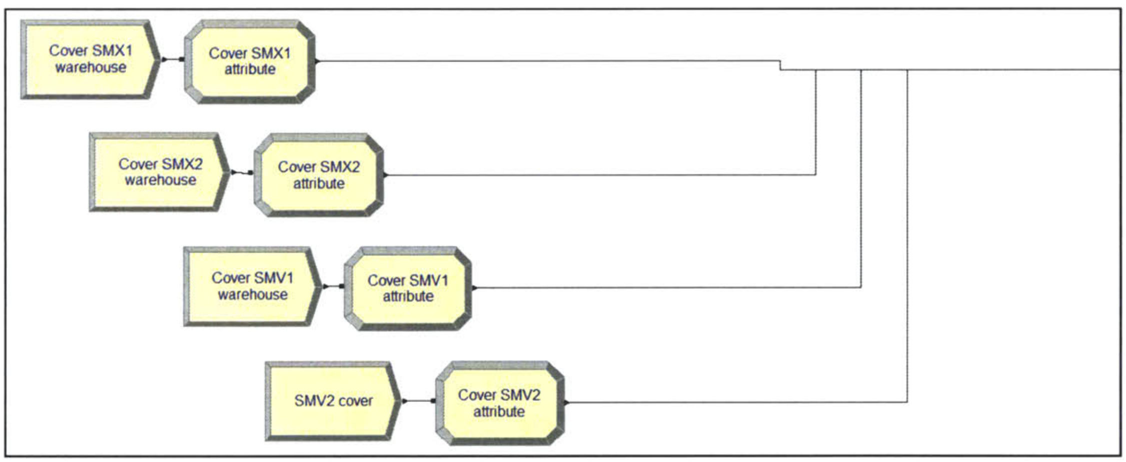


Figure 49: Cover assembly block logic

7.3.8 Drive assembly

Figure 50 shows the *sub-model* used to represent drive assembly on the simulation screen. The drive assembly block logic is intricate and will look cluttered on the simulation screen. The *sub-model* module nests the intricate logic and keeps the simulation screen readable. The 'Drive assembly' *sub-model* receives board *entities* from the main line and cover *entities* from the cover assembly area.

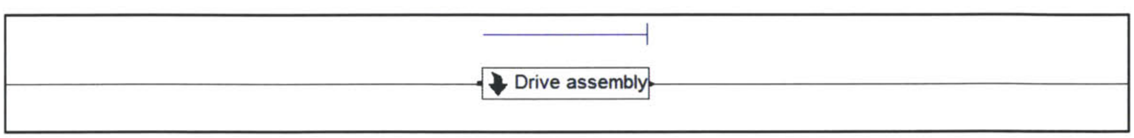


Figure 50: Drive assembly ***sub-model***

The nested *sub-model* logic for 'Drive assembly' is seen in Figure 51. It can be divided into four distinct module clusters; each cluster is designed to *batch* a 'board' *entity* from a product family to its compatible 'cover' *entity*.

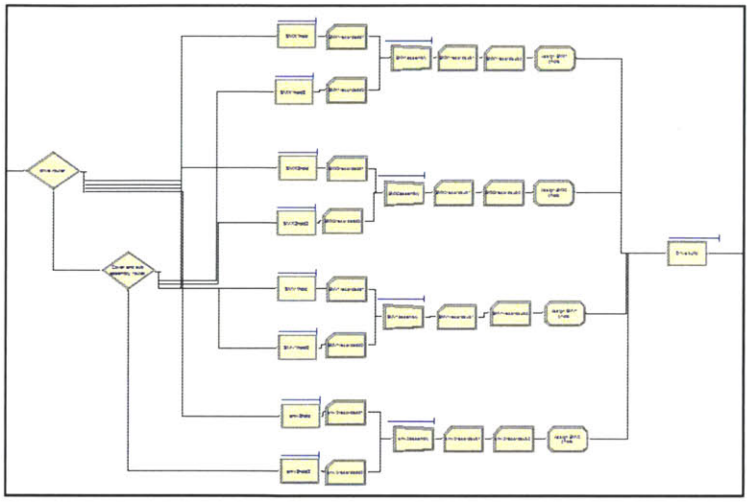


Figure 51: Overview of drive assembly *sub-model*

Figure 52 shows a magnified view of the *entity* sorting logic from Figure 51. The entering *entities* pass through two decide modules that are responsible for sorting *entities* and routing them to their appropriate assembly cluster.

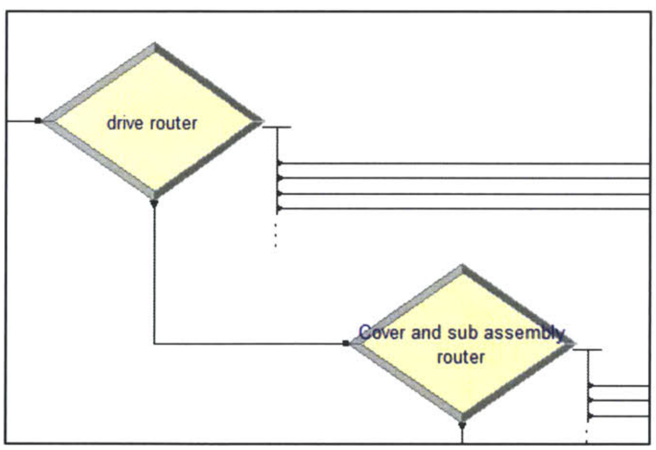


Figure 52: Close-up of *entity* segregation logic in drive assembly *sub-model*

Figure 53 shows the construction of one of the drive assembly module clusters from Figure 51. The arrangement of modules in Figure 53 is similar to 'TH assembly' presented in Section 7.3.3. Cover *entities* and board *entities* enter the 'SMV2assembly' *batch* module through two branches. Each branch has a 'SMV2holdhold' module which releases *entities* referencing the'SMV2assembly' *batch* module. The *entities* are *batched* permanently and remain a single unit for the rest of the line. The *batched entities* pass through a 'Drive build' *process* module which emulates the assembly time required to put together a drive. All *entities* exiting the drive build module are transferred to the final test block.

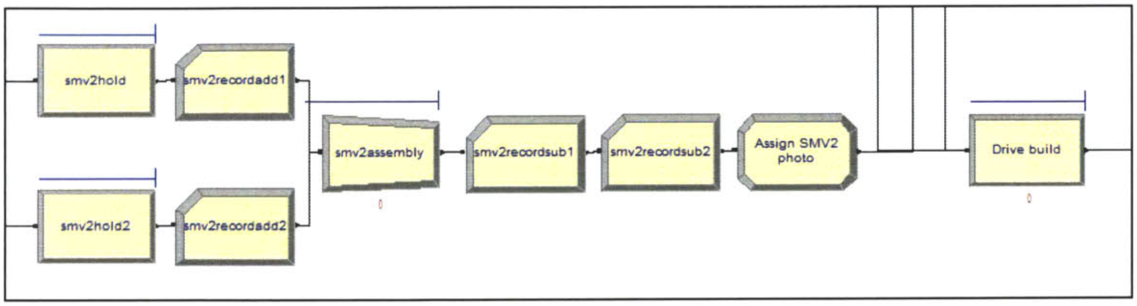


Figure 53: Close-up of drive assembly logic in the drive assembly *sub-model*

7.3.9 Final test

Figure 54 shows the simulation logic used to represent a final test station on the proposed line. The 'Final test' block shown in Figure 54 is alike in construction to the 'Board test' block presented in Section 7.3.5. The *entities* entering from drive assembly block are routed by a 'Final test router' *decide* module to a branch based on the *entity's* product family. Each branch *processes entities*, based on test times and rejection rates obtained from Section 6.1.2.

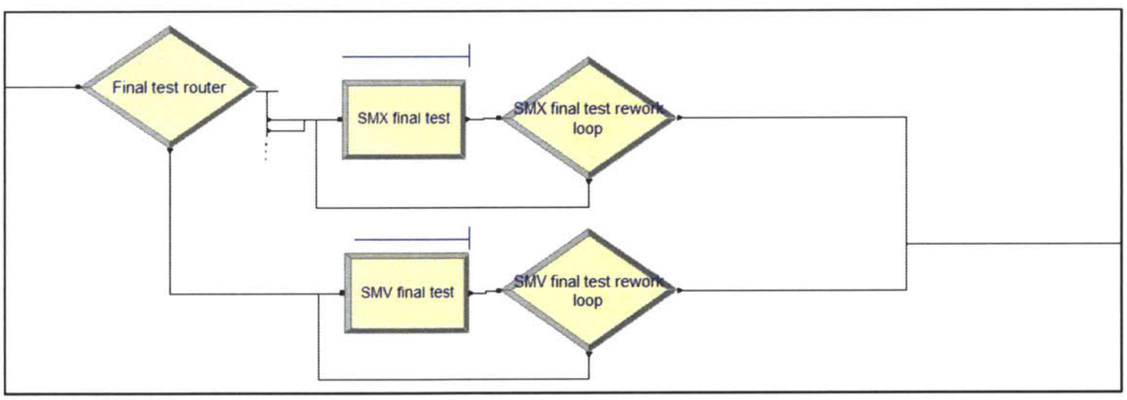


Figure 54: Final test block logic

The major difference between board test and final test is in the form of *resources* and test operation times.

Figure 55 shows the dialog box for final assembly test times. The *resource* list and operation time expressions in Figure 55 differentiate this block from the board test block explained in Section 7.3.5.

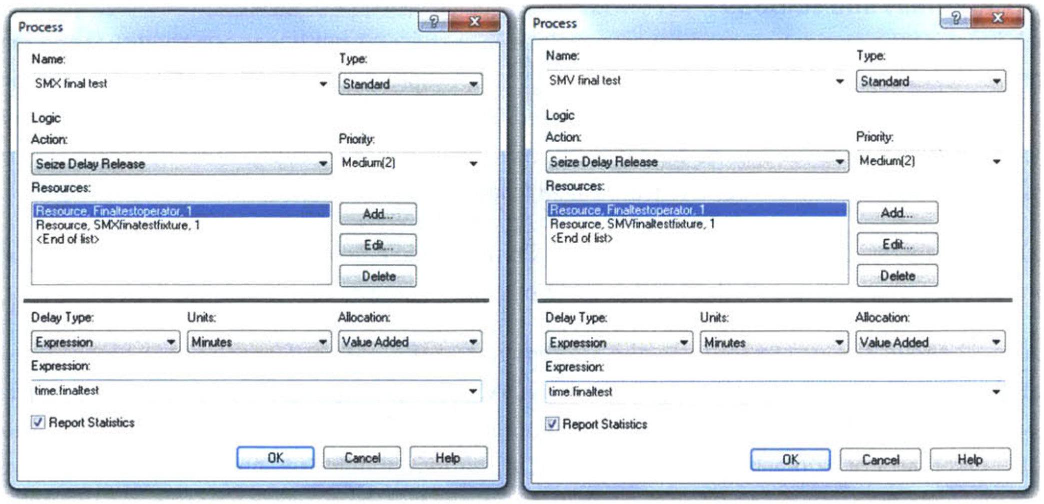


Figure 55: Final test *process* module input

7.3.10 Packaging area

Figure 56 shows the simulation logic used to represent the packaging station's operations. This 'Packaging' block is the last step in the simulation model. The packaging block is made of a 'Packaging final' *process* and 'Outgoing warehouse' *dispose* module. All entering *entities*, first pass through 'Packaging final' which emulates the packaging station operation times. The *entities* then pass through 'Total drive' which is a *counter* that keeps track of the product mix and total line production. The *entities* are finally captured by the 'Outgoing warehouse' *dispose* module and this marks the end of the simulation model. Various statistics on system performance are recorded at this module and all attributes assigned to the *entity* are reset.

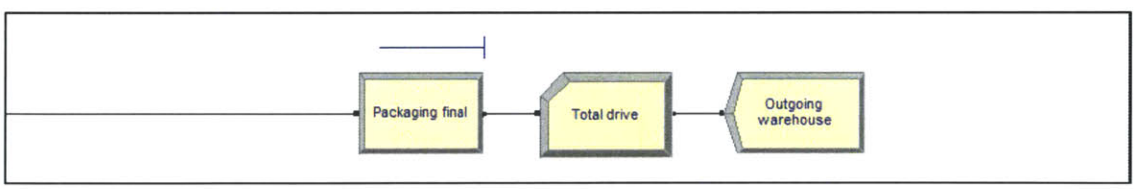


Figure 56: Packaging area block logic

7.4 Integrating the simulation blocks and creating a 2-D animation

All the blocks explained till now function independently but the *entities* need to move from one block to another for the simulation model to represent the entire proposed line. This section describes the use of a *routing* module which integrates all the blocks already described to function as a complete simulation model of the proposed line. The latter half of the section explains setup of a display panel that was used to monitor system behavior using visuals like WIP graphs and *resource* utilization monitors.

7.4.1 Routings and stations

Each block of the simulation model was explained in section 7.3 but it was assumed that the *entities* transfer seamlessly across blocks in no time. This is not representative of parts actually move on the on the factory. For example, PCBs are transferred from wave solder to board test on a belt conveyor and the process requires a finite time period. *Entity* movement from one block to another is made possible with the use of an Arena feature called 'routings' and 'stations'.

Routings transfer an *entity* between blocks. A *station* module defines a location in the block where *entities* can be sent or received. A station functions like a gateway to a simulation block

Figure 57 shows the use of *routings* and *stations* to transfer *entities* from the SMT block. The 'SMT station' module is used as an entry point and the SMT block is referenced throughout the simulation model using this 'SMT station' module. The *routing* module 'Route to AOI' sends *entities* to the AOI block. Once an *entity* reaches the 'Route to AOI' *routing* module it is placed in a temporary hold emulating the time required to transport parts n conveyors.

Routings brings flexibility in modeling and makes the simulation model modular.

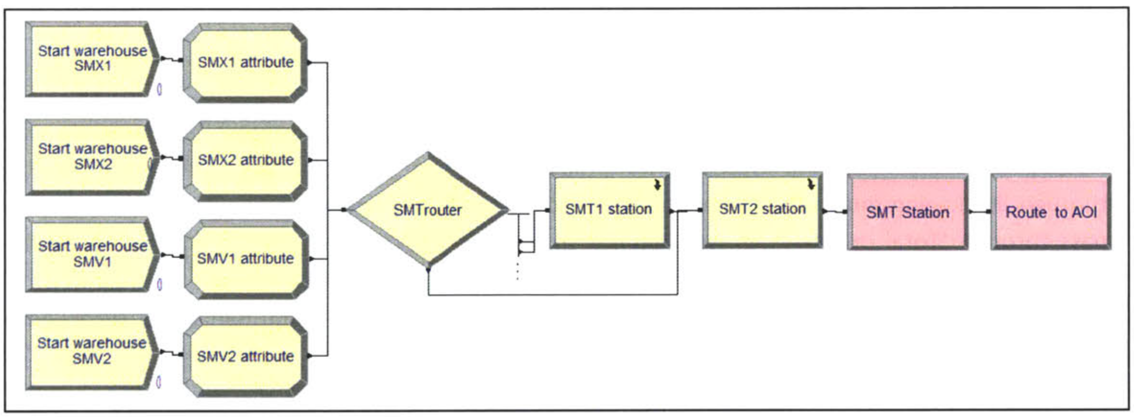


Figure 57: SMT area *routing* and *station* modules

7.4.2 2-D animations

Animations in simulations are useful in communicating the model functioning in a more visual and easy way. The 2-D animations were particularly useful in validating the model behavior and troubleshooting.

There are four *entity* types in our simulation and they were assigned distinct images to trace their movement in the model. Similarly, all *resources* were assigned images to understand *resource* state changes as the model runs. A legend of the images used to represent *entities* is shown in Table 10. The different *entity* pictures shown in Table 10 were helpful in validating simulation logic for setup time. *Entity* pictures were important to understand how a machine reacts to product changeovers.

Table 10 : *Entity* legend map

Name	Picture	
	Original *entity*	With cover
SMV1	SMV 1	SMV 1
SMV2	SMV 2	SMV 2
SMX1	SMX 1	SMX 1
SMX2	SMX 2	SMX 2

Figure 58 shows a snapshot of the simulation run where a product is being processed by both the SMT machines. Most images used to depict *resources* (Figure 58) are representative of the actual equipment. The *entities* in Table 10 can be seen moving along different paths in the simulation in Figure 58 and 59. The SMT machine images in Figure 58 are used to show a *busy* SMT *resource*.

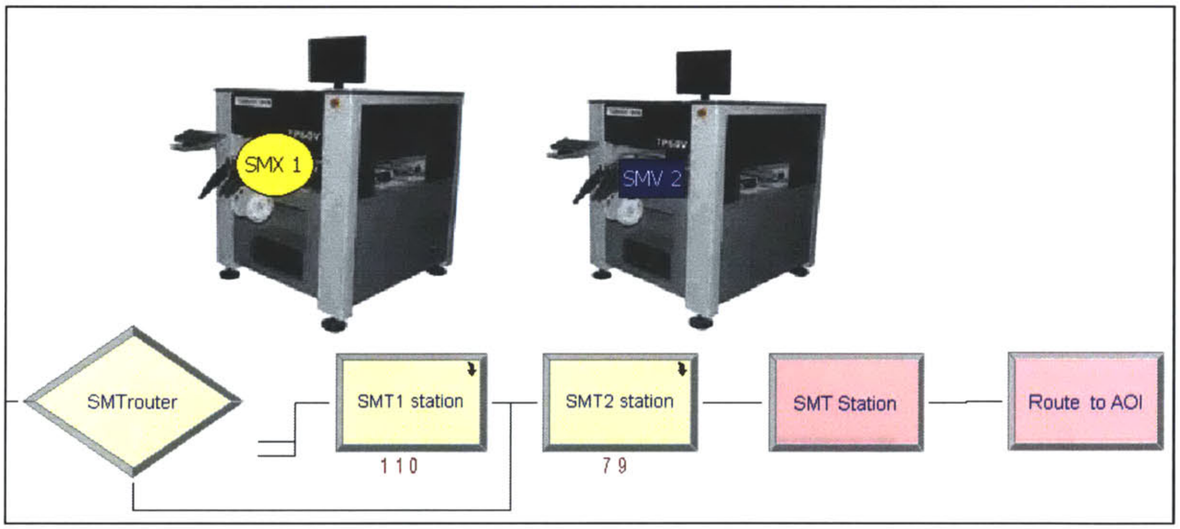


Figure 58: Busy SMT *resource* image and routing logic

Figure 59 shows the complete simulation model with blocks arranged to represent the proposed line. The image is a snapshot of the model while it is running. Animations were used to debug the model and explain the model working to a new user. Animations add to the simulation's computing requirements and slow the performance of the model. However the model analysis was done after switching off animations to ensure faster performance of the simulation model.

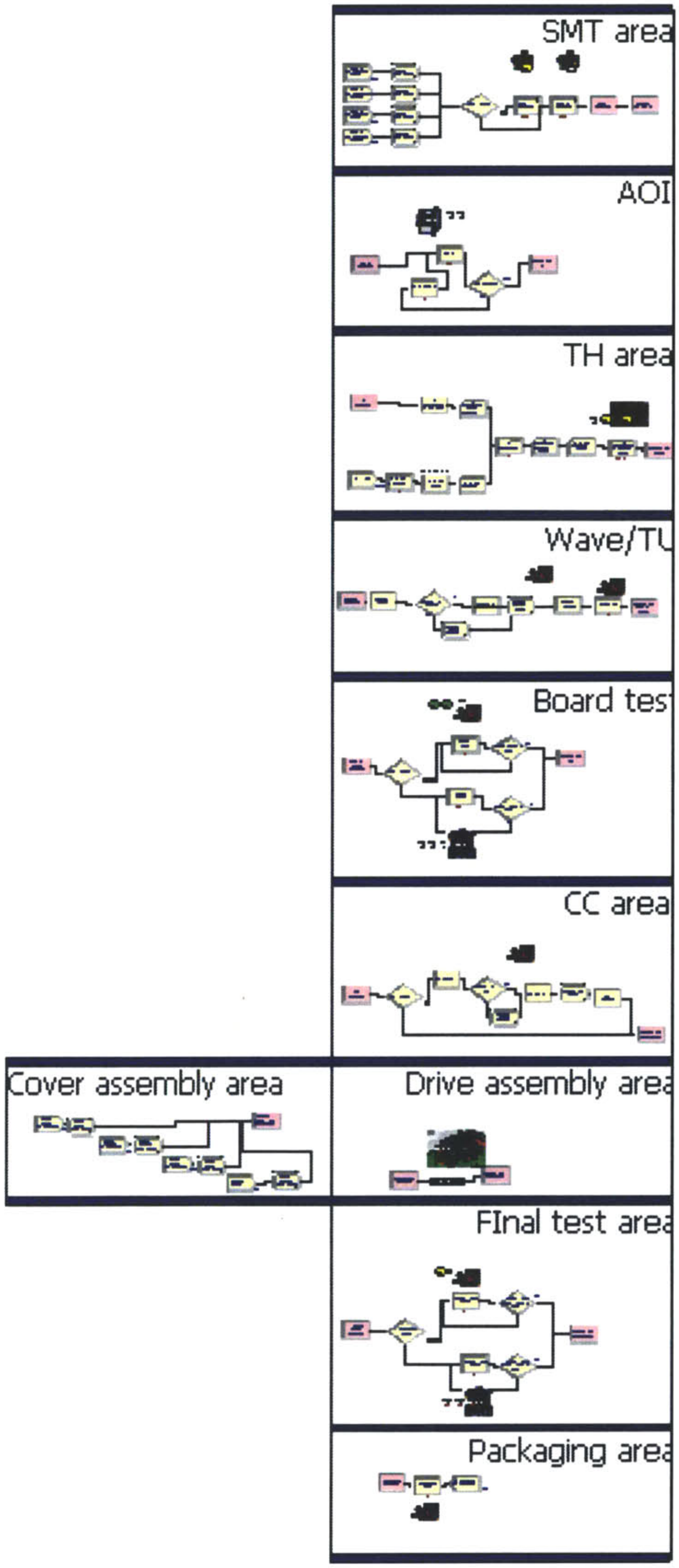


Figure 59: Integration of blocks to create the final line simulation

7.4.3 Graphs and *resource* utilization levels

Every time an *entity* flows through the system it creates a record in the output summary report. The record is modified as multiple *entities* interacts with different modules in the simulation model [32]. It was difficult to monitor the trend of variables like WIP inventory, *resource* utilization and final output over a period of a day as the simulation provided aggregate values in the final report [22]. To overcome this problem, a display panel was set up to observe changes in behavior of significant parameters during a simulation run equivalent to a day (two shifts).

Figure 60 shows the new model screen which includes WIP graphs and *resource* monitors that display the instantaneous model state. The first two columns in the Figure 60 show the simulation logic which is identical to Figure 59. The third column consists of a series of graphs that monitor WIP. The WIP graphs were used to identify trends of buffer build up and a more detailed use of this feature will be seen in Chapter 8.

Machine utilization is an important system parameter that was monitored through the red bar graphs in column four of Figure 60. A white rectangle represents zero *resource* utilization and the red bars tell the user the number of *resources* in use. The behavior of *resource* monitors was useful in deciding capacities at the drive assembly and both the test stations.

The line graph in the right most column shows a plot of the cumulative system production. The four lines represent the product mix generated by the line. The number displayed under the graph is readout of the instantaneous product mix (it is the Y coordinate value of each of the four graphs). Looking at the product mix helps in understanding the compatibility of the products running on the proposed line. A more detailed use of the display panel from Figure 60 will be seen in chapter 8.

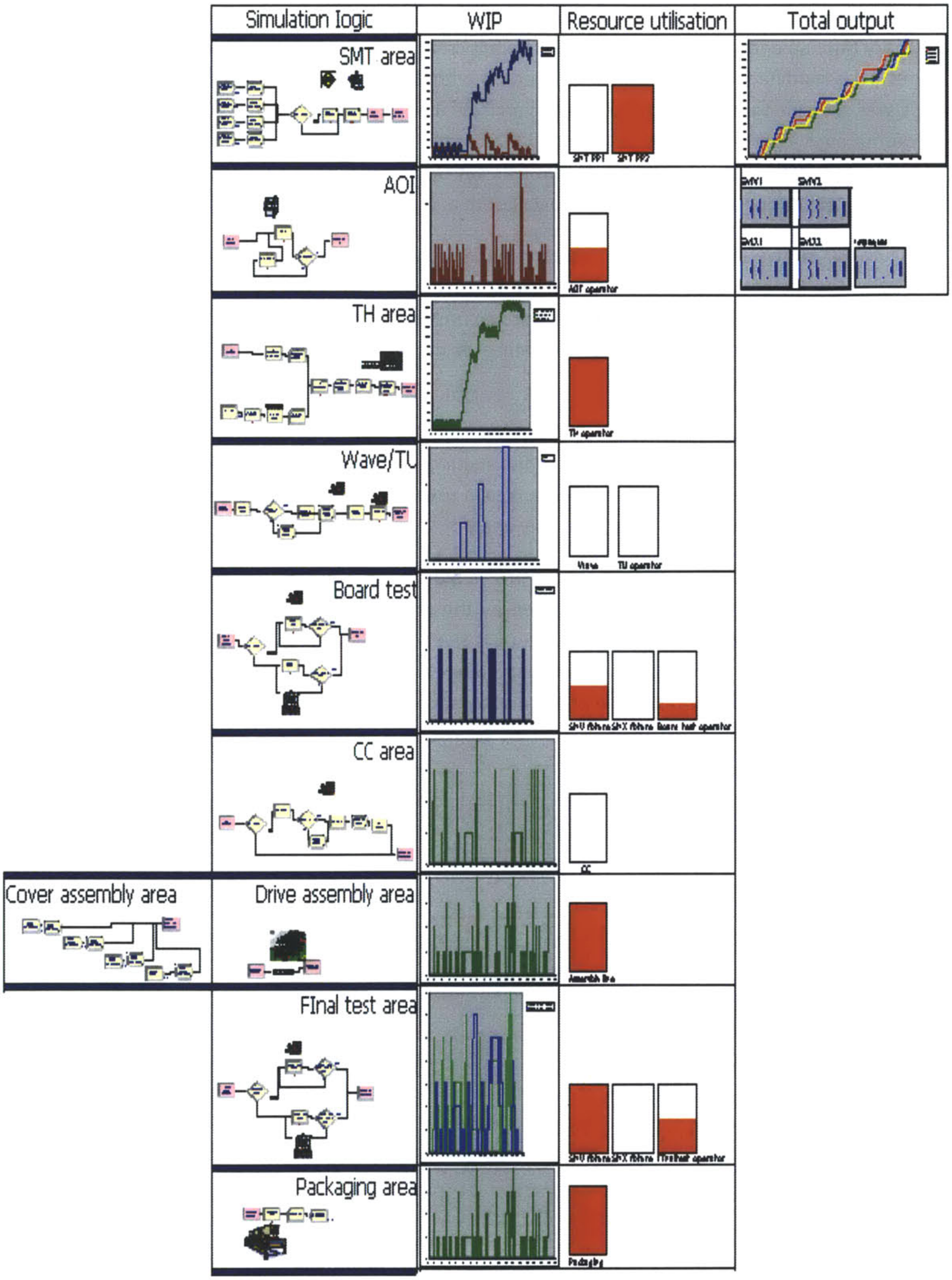


Figure 60: Final simulation with display panel and animation

7.5 3-D animations

Figures 61 to 68 show a 3-D rendering of the proposed line. All images were captured from the 3-D animation mode of Arena visual designer. 3-D animations help to visualize the proposed equipment would fit together in the physical shop floor. Images like Figure 61 were vital to communicate the proposed line's design to the manufacturing personnel. CAD software has long been using 3-D rendering to effectively test and optimize mechanical design, and the use of 3-D modeling serves a similar purpose in our simulation study [34]. The 3-D animation was used to verify the flow chart module's functioning.

The 3-D animation process began by arrangement of equipment CAD files in 2-D as shown in Figure 62. The 2-D layout is not to scale, but placement of conveyors, machines and storage racks are oriented relative to each other and proportionate in size.

Each CAD model had to communicate with its flow chart module. For example the storage racks in Figure 63 have arrows marked on them signaling that the rack houses an *entity* from TH assembly. The 3-D racks will fill up material based on the queue length of the 'TH assembly process time' module from Figure 45. Once all the CAD files are linked to their flow chart modules, camera placements were set (Figure 64) to observe the animation from strategic locations. Various camera placements ensured the entire line was monitored when the model is running. A snapshot from camera placements in Figure 64 is seen in Figure 65.

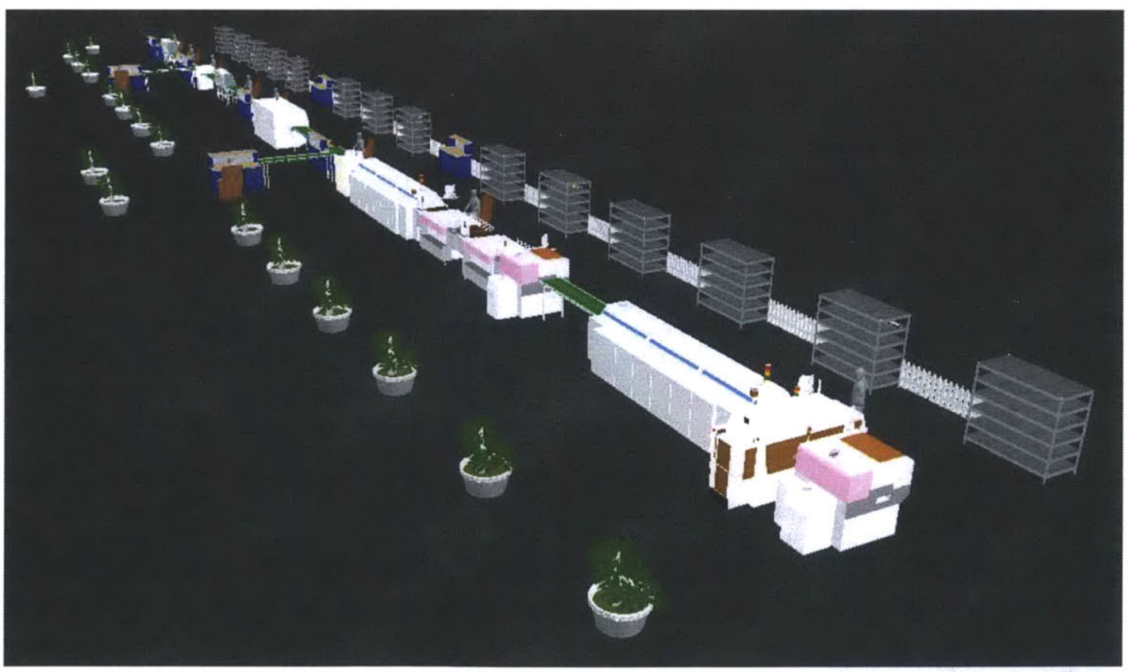


Figure 61: Overview of the 3-D animation

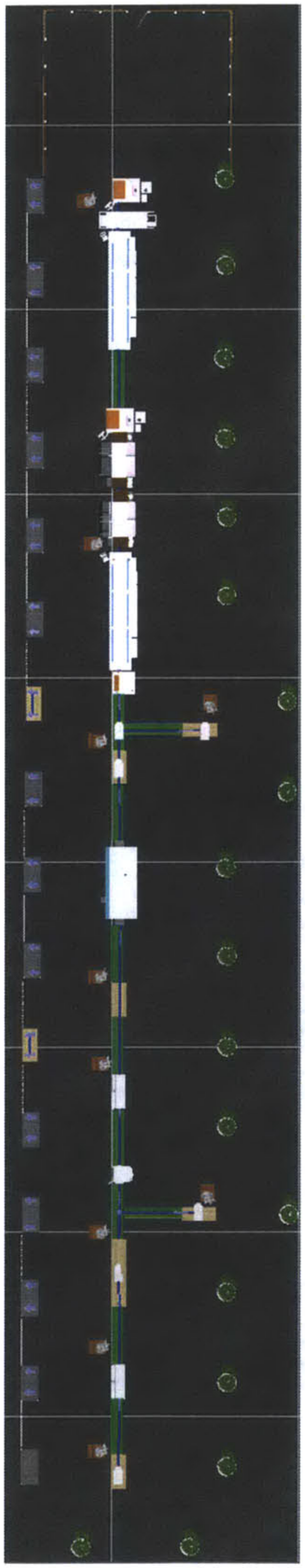


Figure 62: Top view of the final line

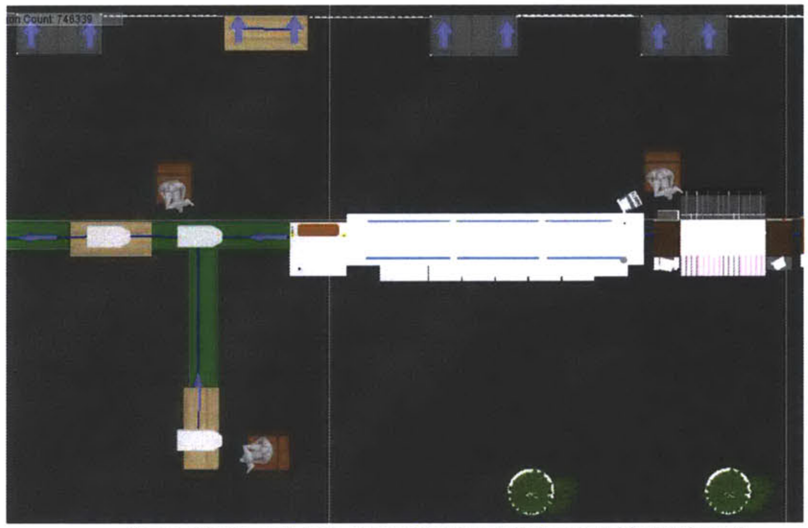


Figure 63: Through hole prep and through hole assembly 2-D layout

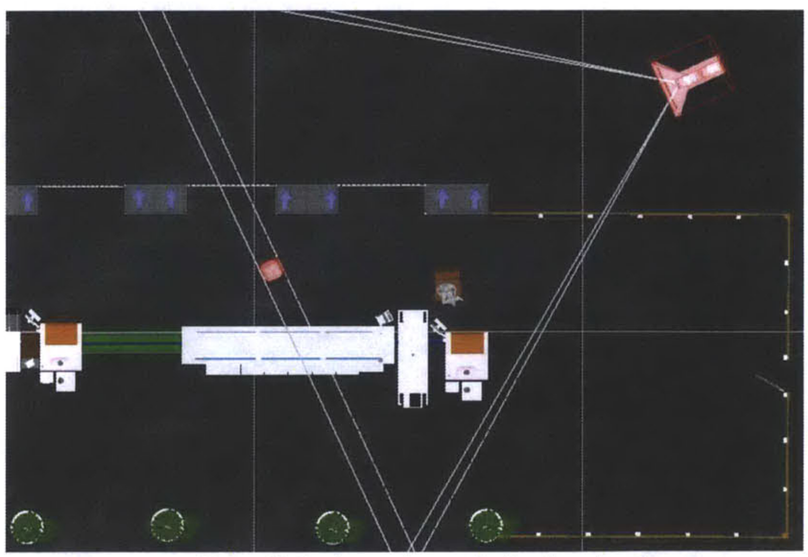


Figure 64: Setting up view angles for the animation

Figure 65: Dynamic operator and storage rack animation

The next step was to make the 3-D animation to represent part (*entity*) movements in the factory. Part transfers were made possible by the use of *stations* and *routings* defined in Section 7.4.1. *Stations* were placed on equipment CAD files, to represent the start point of an *entity*. Each *entity* would travel on *routing* lines placed over a conveyor CAD file. Figure 66 shows the conveyor CAD file and a *station* placement.

Storage rack CAD files were placed behind each operator CAD files (Figure 67) to represent buffers. Storage rack CAD files are dynamic and rack shelves would fill up with the *entity* CAD files as the simulation progressed.

Figure 68 shows the overall 3-D animation of the line wherein equipment is arranged to form a linear layout which is connected by two auxiliary lines (through hole assembly and cover assembly). Although the 3-D animation can represent the factory, it could not be used as a factory blue print. Actual factory floor plans had to be created to implement the findings of this thesis project and a discussion about the same will be done in Chapter 9.

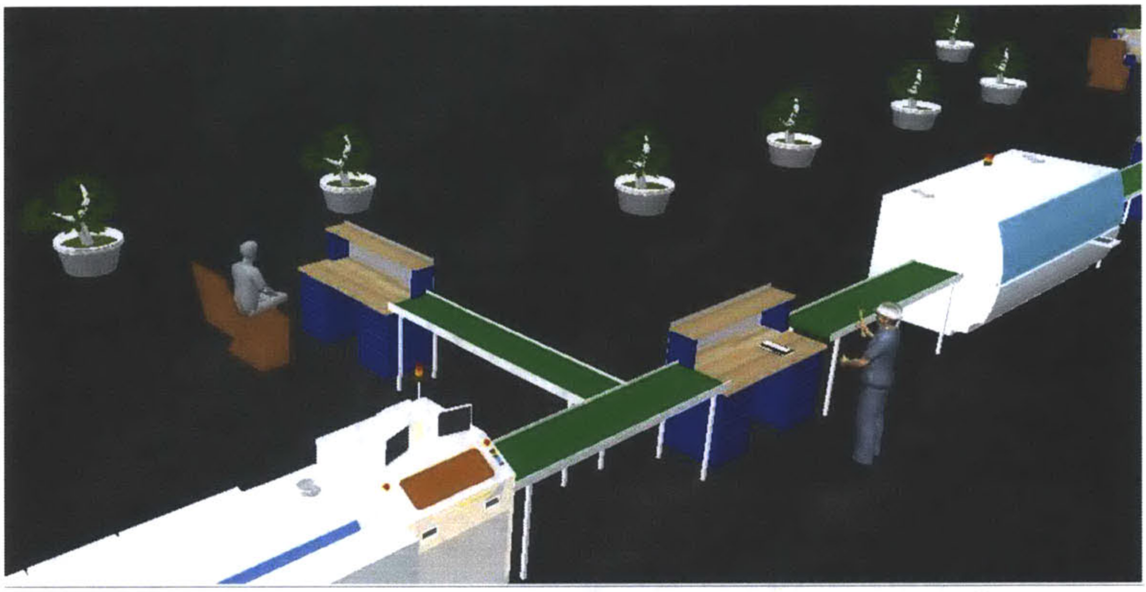


Figure 66: Through hole prep and through hole assembly

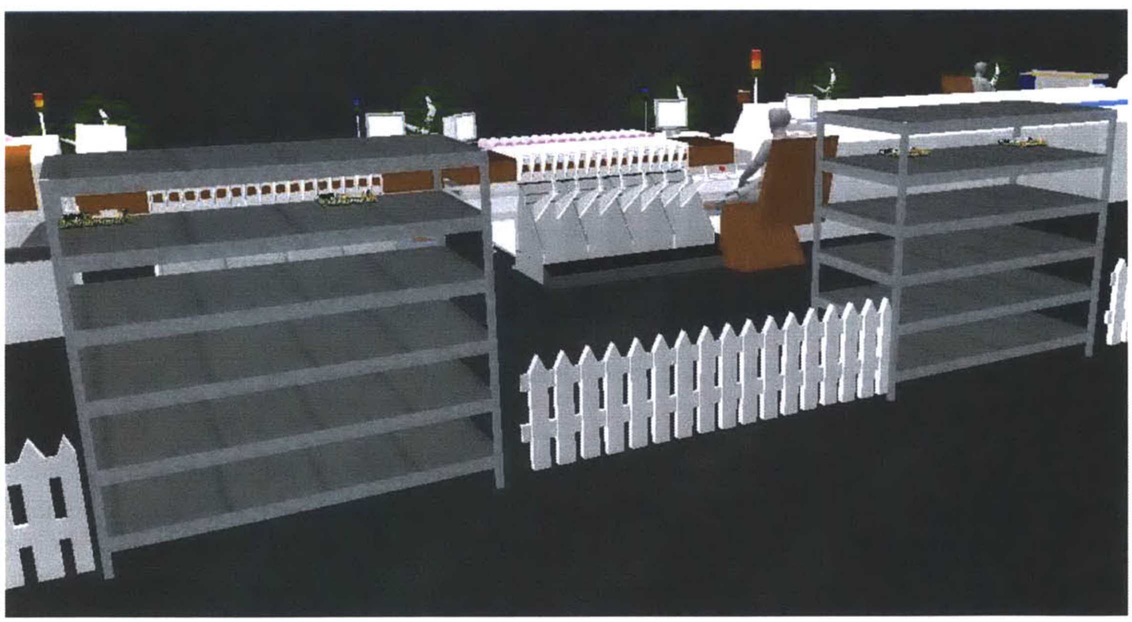


Figure 67: Machine loading racks and equipment CAD files (SMT PP2 and its buffer can be seen)

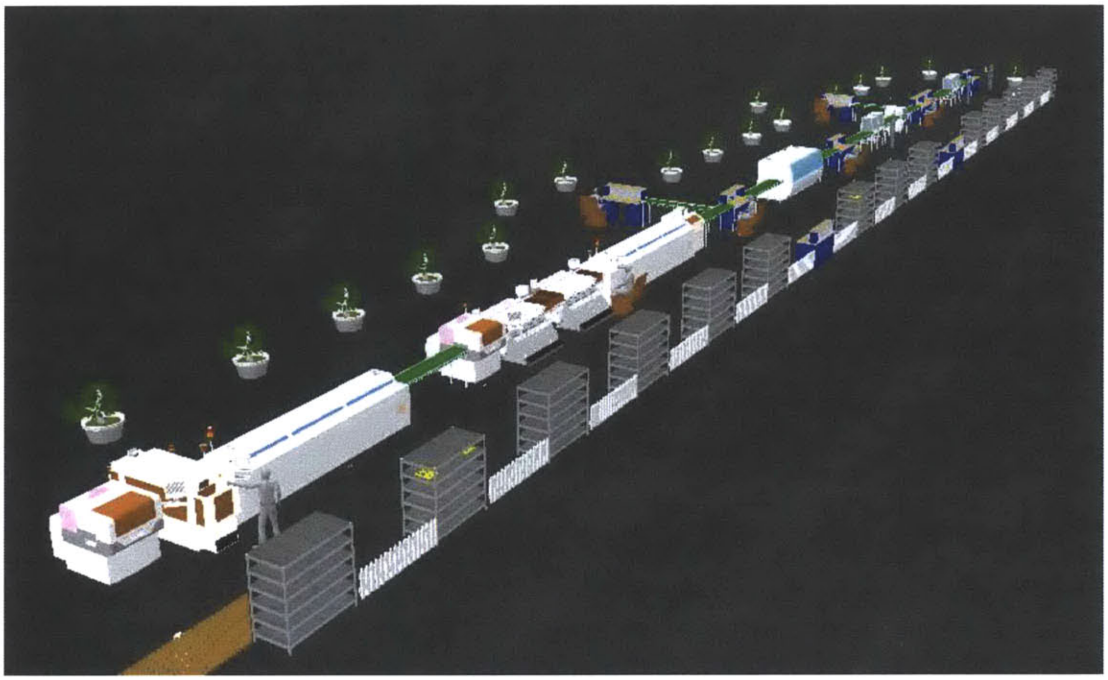


Figure 68: Overview of the 3-D line animation

7.6 Initial run settings

This section describes the initial setup of the simulation model required to run scenarios in Chapter 8. The initial setup includes specifying the number of products running on the line, simulation run length and number of replicates.

7.6.1 Number of *entities* in the simulation model

The line output and *resource* behavior depends on the rate of incoming *entities*. The number of *entities* in the model was determined by first flooding the system by introducing excessive *entities* at the 'SMT area' block. After about two hours of system clock time, the line metrics stabilized and the line production rate was noted. The steady state production time of the line came out to be 1.72 minutes per drive. This value was then used to determine the number of *entities* at each *create* module of the 'SMT area' block. The outcome of this analysis was to introduce an *entity* every 6.7 minutes to prevent system instability.

7.6.2 Run lengths and number of replicates

Simulation run length is the number of time units the simulation is run before the simulation metrics are reset. A replicate is the repetition of a run length to mitigate the effect of erroneous sampling from the defined random distributions (outliers). Multiple replicates will average out any unrealistic readings and ensure an unbiased result.

A series of simulation run were tried out to determine the optimum run length and the number of replicates. Figure 69 shows a graph of the system output as we increase the simulation run length. Each simulation run was executed by keeping all the simulation parameters constant. The graph shows an increasing trend until a simulation run length of 5 days. The simulation results don't change a lot if we increase the simulation run length beyond 5 days.

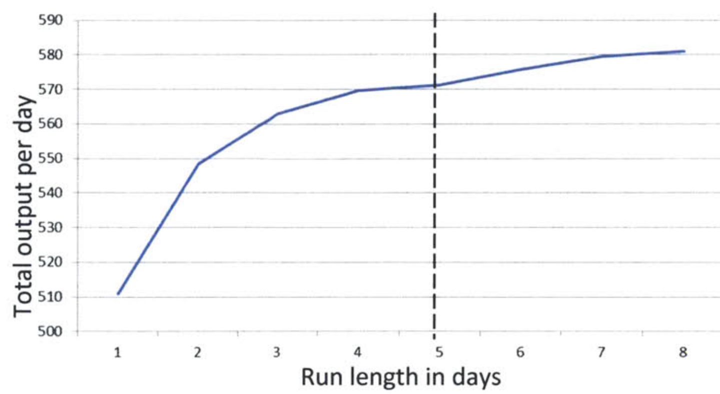


Figure 69: Total output of line vs. run length

A warm up period in simulation is the time period (at the start of every simulation run) when the system statistics are not noted [31]. Table 11 shows the simulation output values as we increase the run length from one day to five days. The steady state line output is stated as 571.2 drives for the current setting.

Table 11: Sensitivity of run length to total drives production

Days per replicate	Total production units per day on each line
1	511
2	548.5
3	563
4	569.75
5	571.2

Table 11 and Figure 69 suggest that the system attains steady state after 4 days and thus it was decided to use a run length of 5 days.

This concludes the initial run settings of the simulation and the model is now set up to validate and test line configurations in chapter 8.

8. Results and conclusions

The results presented in this chapter were obtained by running different configurations of the simulation model developed in chapter 7. A simulation model configuration designed to represent a specific factory condition is called a scenario. Scenarios are a powerful resource that will enable us to understand the impact of various decisions that need to be made to evolve from the existing factory layout to the proposed line design [31].

8.1 Scenarios

8.1.1 Running SMV drives only

The first scenario was set up to emulate existing factory conditions and the resulting output was compared to actual factory conditions to validate the simulation logic [35]. The results from this section confirm the accuracy of the simulation logic and are an important stepping stone to the other scenarios.

Simulation model:

The existing factory produces SMV drives only and the simulation had to be set up so that the simulation model was exclusively running SMV *entities*. This requirement was achieved by setting the 'SMX create' module to zero but this also meant the simulation will run at half its capacity. To compensate for the loss of SMX *entities*, the number of SMV *entities* was doubled. Figure 70 shows the *create* module dialog boxes for SMV and SMX drives set up for this scenario.

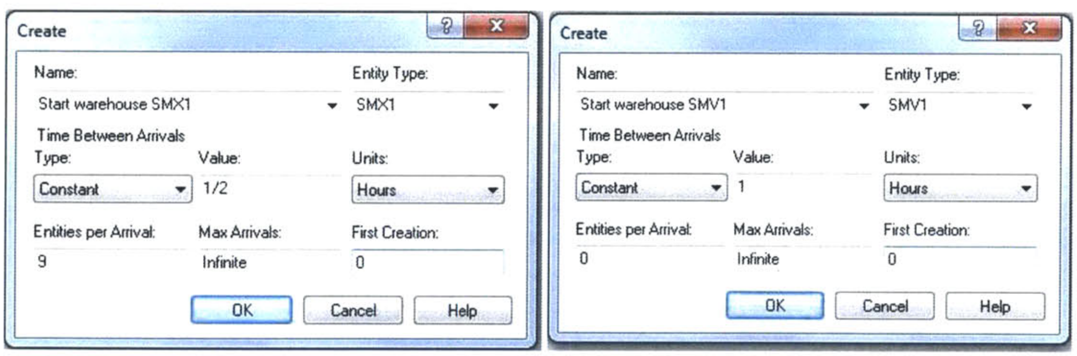


Figure 70: *Create* module setting for Section 8.1.1

Results:

Table 12 shows the output report for Section 8.1.1. As the model is now running with twice as many SMV *entities*, all the SMV processing equipment are utilized more and the results from Table 12 suggest high utilisation at drive assembly and final test. As per the SMV process plan, the SMV *entities* bypass 'SMT1', 'CC' and 'SMX test fixtures' resulting in zero utilization at those stations.

It was observed that drive assembly line and final test stations were bottlenecks in the current factory operation as well. Inventory staging points existed before both the stations and these stations would often run overtime shifts to keep up with the other stations. This scenario confirms the simulation model logic and validates the simulation logic so that it is ready to test the other scenarios.

Table 12: Results from simulation setting for Section 8.2.1

Type	Average Work in progress per week	Average output per week	Resource utilization		Bottleneck machine	Average product mix per week	
Just SMV	249.21	2274.6	Drive assembly line	0.9956	Drive assembly	SMV1	1123.2
			SMT2	0.962		SMV2	1151.4
			SMT1	-			
			SMV board test fixture	0.8523			
			SMV final test fixture	0.9846			
			SMX board test fixture	-			
			SMX final test fixture	-			

8.1.2 Running SMX drives only

Simulation model:

One of the problem areas identified in the decision tree analysis in Section 2.1 was the disruptions to production caused by rush orders. This scenario is set up to test the line's ability to run one product type (SMX drives) only. A single product type may be produced for an entire day during rush order processing. The simulation setting in Figure 71 will emulate such a situation, by running just SMX *entities* in the model. The *create* modules for SMV *entities* is set to zero and twice as many SMX *entities* are running in the model.

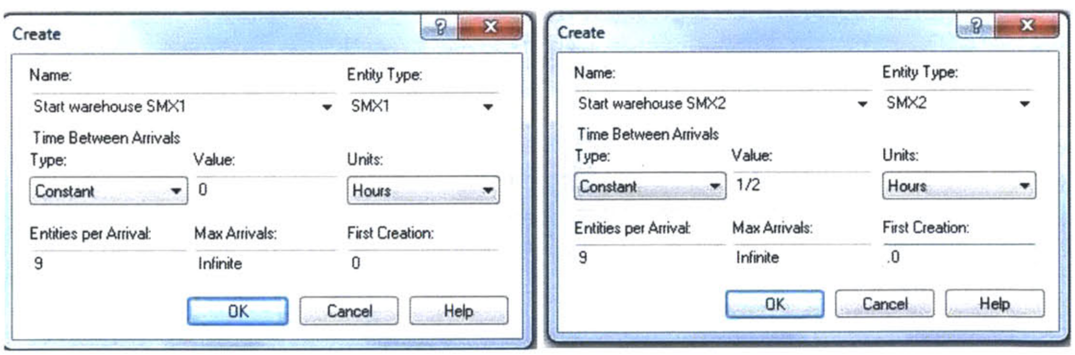


Figure 71: *Create* module setting for Section 8.1.2

Results:

Table 13 shows the output report for scenario 8.1.2. The results presented in Table 13 are in line with the results obtained from the analytical model described in Chapter 6 as both the models are set up to run SMX drives only. As per SMV process plan, SMV *resources* are not used and their utilization is zero. One of the bottlenecks has now shifted from drive assembly to SMT1. The second bottleneck remains at final test as indicated by its high *resource* utilization. Drive assembly is not a bottleneck anymore because of the low operation time of the SMX drives. Overall the SMX drives are faster to produce and run with 40% less average WIP compared to SMV drives.

Table 13: Results from simulation setting for Section 8.1.2

Type	Average Work in progress per week	Average output per week	Resource utilization		Bottleneck machine	Average product mix per week	
Just SMX	138.46	2397	Drive assembly line	0.3964	SMT1 and final test	SMX1	1204.6
			SMT2	0.7211		SMX2	1192.4
			SMT1	0.9974			
			SMV board test fixture	-			
			SMV final test fixture	-			
			SMX board test fixture	0.5292			
			SMX final test fixture	1			

8.1.3 Without grouping production orders

Simulation model:

This scenario will test the impact on line output if no systematic order is maintained in products running on the line. The goal of this scenario is to understand the change in WIP and line output due to a variation in product mix. Figure 72 shows the *create* module setting for this scenario and the settings ensure that at any given time the line does not receive two similar *entities* back to back. As seen in Figure 72, the *create* module is set up to introduce one *entities* every 6.6 minutes (1/9 hours). This setting will ensure that *entities* are *created* in a random order, for example a SMX1 *entity* is *created* following a SMV2 *entity creation* and the SMV2 *entity* is *created* following a SMV1 *entity*.

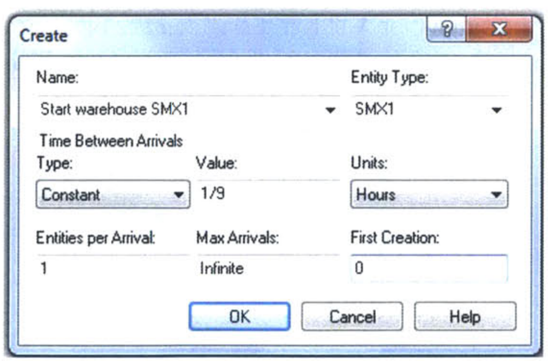


Figure 72: *Create* module setting for Section 8.1.3

Results:

Table 14 shows the output report for scenario 8.1.3. We can see an unbalanced mix of products coming out of the line from the last column of Table 14. The inconsistency of *entity* types adds setup time to the operation time at the *process* modules of each block.

The average WIP is six times of WIP in Section 8.1.2. This can be traced back to the drive build sequence of SMX which uses two extra *resources* and hence there are two more stations with setup. The SMV drives are skipping these stations and are able to *seize* shared *resources* along the much more frequently. The SMT *resources* are first in line to be affected by the product mix and SMT queue lengths are notable larger. The total weekly output of this scenario is 100 drives fewer than in Section 8.1.4.

Table 14: Results from simulation setting for Section 8.1.3

Type	Average Work in progress per week	Average output per week	Resource utilization		Bottleneck machine	Average product mix per week	
No grouping	828.9	2805.6	Drive assembly line	0.8407	SMT 2	SMV1	1125.2
			SMT2	1.01		SMV2	1120
			SMT1	1		SMX1	280.4
			SMV board test fixture	0.6471		SMX2	280
			SMV final test fixture	0.9774			
			SMX board test fixture	0.1028			
			SMX final test fixture	0.2307			

8.1.4 Grouping production orders

This scenario will test the impact of grouping similar production orders on the output of the proposed line.

Simulation model:

Figure 73 shows the *create* module setting to enable grouping of similar *entities*. As per 'Entity per arrival' box in Figure 73, the simulation will run a single *entity* type for one hour before changing over to the next *entity* type. The simulation will continue running the next *entity* type for another hour and this pattern will continue periodically.

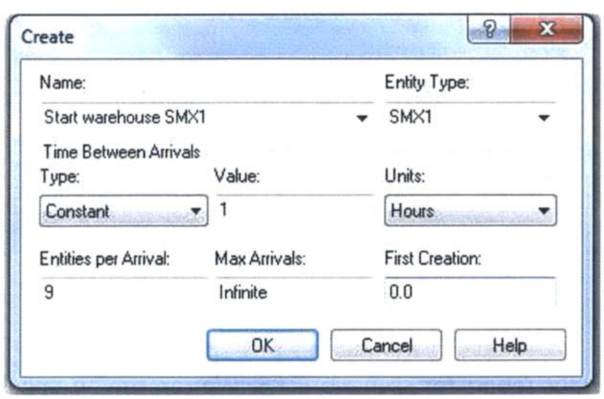


Figure 73: Create module settings for Section 8.1.4

Results:

As seen in Table 15, the line output for this scenario is highest amongst all the scenarios discussed till now. The output product mix is balanced but SMX production is slightly higher. The bottleneck remains at SMT2. This is because SMT2 has the highest operation time and is prone to breakdown in comparison to manual stations which are modeled with zero breakdowns. The average WIP has decreased to 185 because of shorter queue lengths at *process* modules due to lower operation time as there are fewer setups.

This scenario defines the proposed line's default configuration. Thus, the results from this scenario will be used to compute the number of lines required to satisfy the takt time calculated in Section 2.2.2. The weekly output of the line averages at 2959 drives which is 1.72 minutes to produce a drive. Results of Equation (11) will determine the number of lines required to satisfy the forecasted SMX demand.

As mentioned in 2.2.2, the Takt time for the SMX demand is 0.65 minutes

$$\therefore \text{Number of lines} = \frac{time\ to\ produce\ a\ drive}{takt\ time} = \frac{1.72}{0.65} \simeq 3 \tag{11}$$

Hence, we conclude that, on an average, 3 lines are needed to satisfy the forecasted SMX demand.

Table 15: Results from simulation setting for Section 8.1.4

Type	Average Work in progress per week	Average output per week	Resource utilization		Bottleneck machine	Average product mix per week	
Grouping	184.6	2959.6	Drive assembly line	0.6936	SMT 2	SMV1	748.4
			SMT2	0.9994		SMV2	744
			SMT1	0.7334		SMX1	737.2
			SMV board test fixture	0.413		SMX2	730
			SMV final test fixture	0.6382			
			SMX board test fixture	0.2727			
			SMX final test fixture	0.607			

8.1.5 Various CC configurations

The scenarios discussed in this section were designed to predict the impact of temporary CC arrangements that were identified as necessary milestones to reconfigure the line from existing layout to proposed layout.

Case1. One CC, but away from the main line

The CC machine is currently located at the corner of the factory floor as seen in Figure 5. It was not feasible to move the machine immediately to fit into the proposed layout any sooner. The first scenario will simulate the line with the CC operations happening at the machine in its existing location. This would mean parts have to be transported from board test to CC and back to *drive assembly* as seen in Figure 74. Furthermore, as the SMX boards are two sided they have to pass through the machine twice. The second side CC can begin only after a *batch* of ten one-side-CC boards is ready.

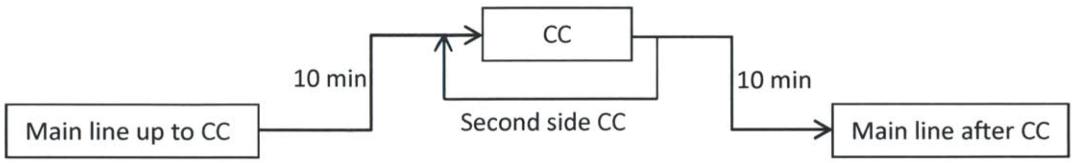


Figure 74: One CC machine, located offline

Simulation model:

The board *entities* have to pass through the 'CC work' module twice and the simulation setting to enable this is shown in Figure 75. The simulation logic is similar to what was presented in the original CC logic in Section 8.3.6, but differences can be seen after the 'CC release' module.

The *decide* module 'CC twice check' is used to *route entities* for a second round through the 'CC work' module. The *decide* module routes *entities* to a 'Batch for second side' *batch* module which releases *entities* once a *batch* of 10 *entities* is available. The *entities* then pass the 'CC work' module a second time and the module 'reset CC counter' keeps *count* of the number of times an *entity* has passed the 'CC work' module. The *entities* which have passed the 'CC work' module twice are routed to the next station by the 'CC check twice' *decide* module.

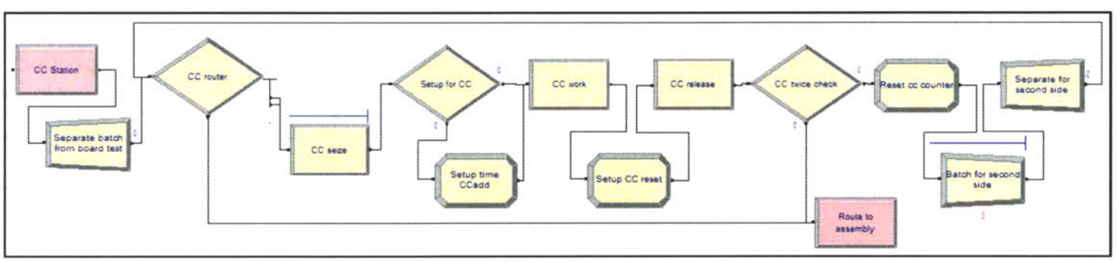


Figure 75: Simulation setting for one CC machine placed away from main line

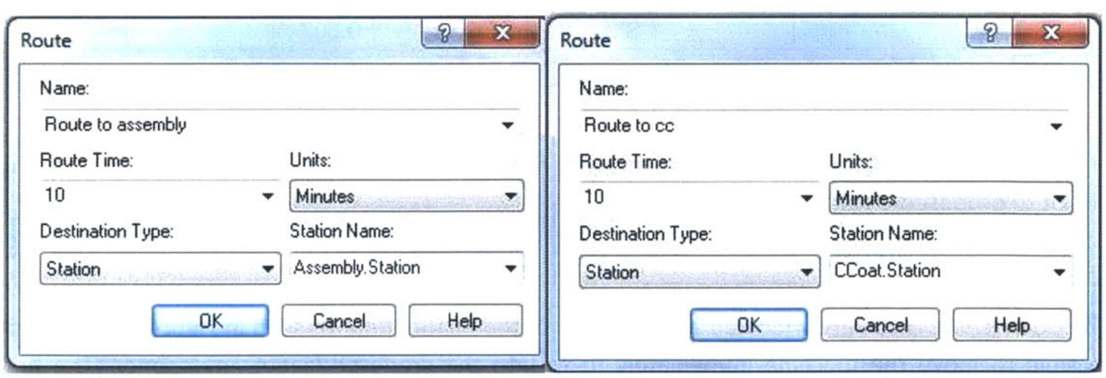


Figure 76: *Route* module setting for Section 8.1.5 Case 1

Figure 76 shows the *routing* module setting to emulate the transportation time between the main line and CC. The *entities* move from the main line to the CC block after a time *delay* to emulate the action of manual board transfers on trolleys.

Results:

Table 16 shows the line parameters and CC utilizations from the simulation model set up for Case 1. The average queue length of CC has gone above 4 because the board *entities* are running through the 'CC work' *process* module twice. The product mix column shows the overall SMX output to be decreased by 45 drives and no affect to SMV output, this is because the SMV *entities* skip the CC station and are not affected by this scenario. The average inventory in the system has gone up as *entities* are waiting in transit between CC and the main line.

Table 16: Results from simulation setting for Section 8.1.5 Case1

Type	Average Work in progress per week	Average CC queue per week	CC utilization	Average product mix per week	
No batching during transport	198.4	4.72	0.7155	SMV1	750.2
				SMV2	744.6
				SMX1	752.6
				SMX2	760

The graph in 77 shows the CC buffer build-up over a day. Parts are seen entering the CC station one hour into the shift and the queue length is 3 at the beginning. After the boards pass the machine once and are ready for the second side, the queue length is increased to 12. This happens because 'CC work' module has to *process* the second round of board *entities* and at the same time there is a influx of new board *entities* from the main line.

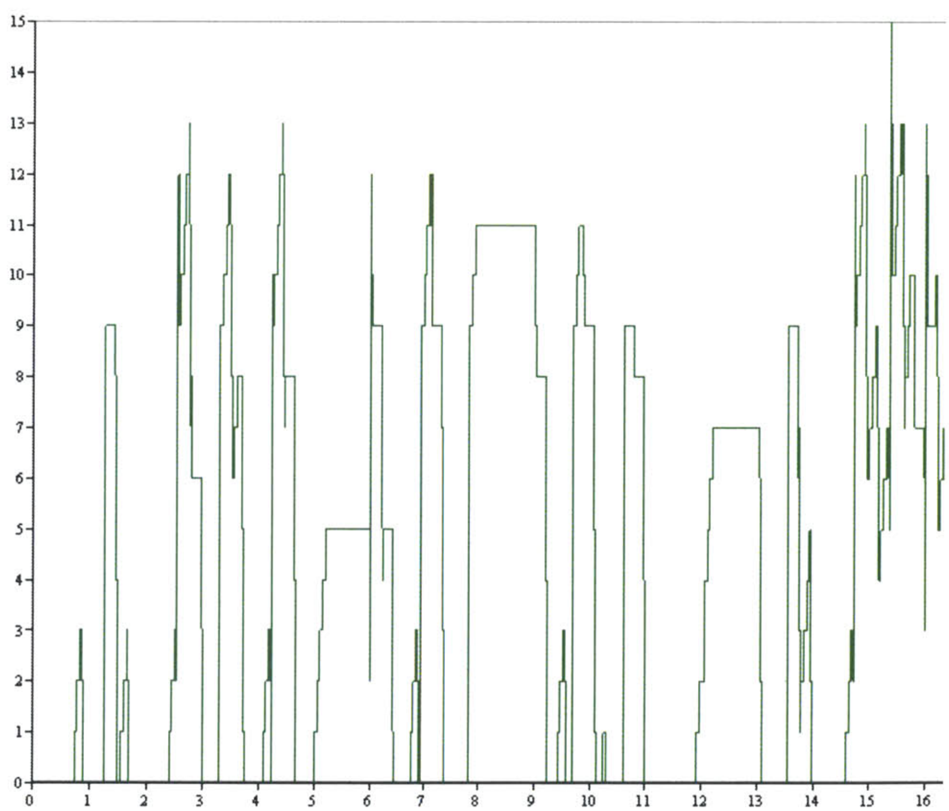


Figure 77: WIP graph for CC in Section 8.1.5 Case1 (one-product transfers)

The above discussed scenario assumes that *entities* are transferred to the CC machine one at a time. However it might not be possible to do so in the actual line if the production order has a higher percentage of SMX drives. The results in Table 17 were complied with a simulation setting to emulate the transfer of board *entities* in a *batch* of ten. As seen in Table 17 the parameters are not affected drastically but the product mix shows that SMX productivity has gone up. A closer look at the WIP graph in Figure 77 also shows a different trend in the way the CC *resource* is loaded.

Table 17: Results from Section 8.1.5 Case 1, but with batch transport

Type	Average Work in progress per week	Average CC queue per week	CC utilization	Average product mix per week	
batch during transport	192.9	4.7	0.7068	SMV1	749
				SMV2	744.8
				SMX1	785.6
				SMX2	797.6

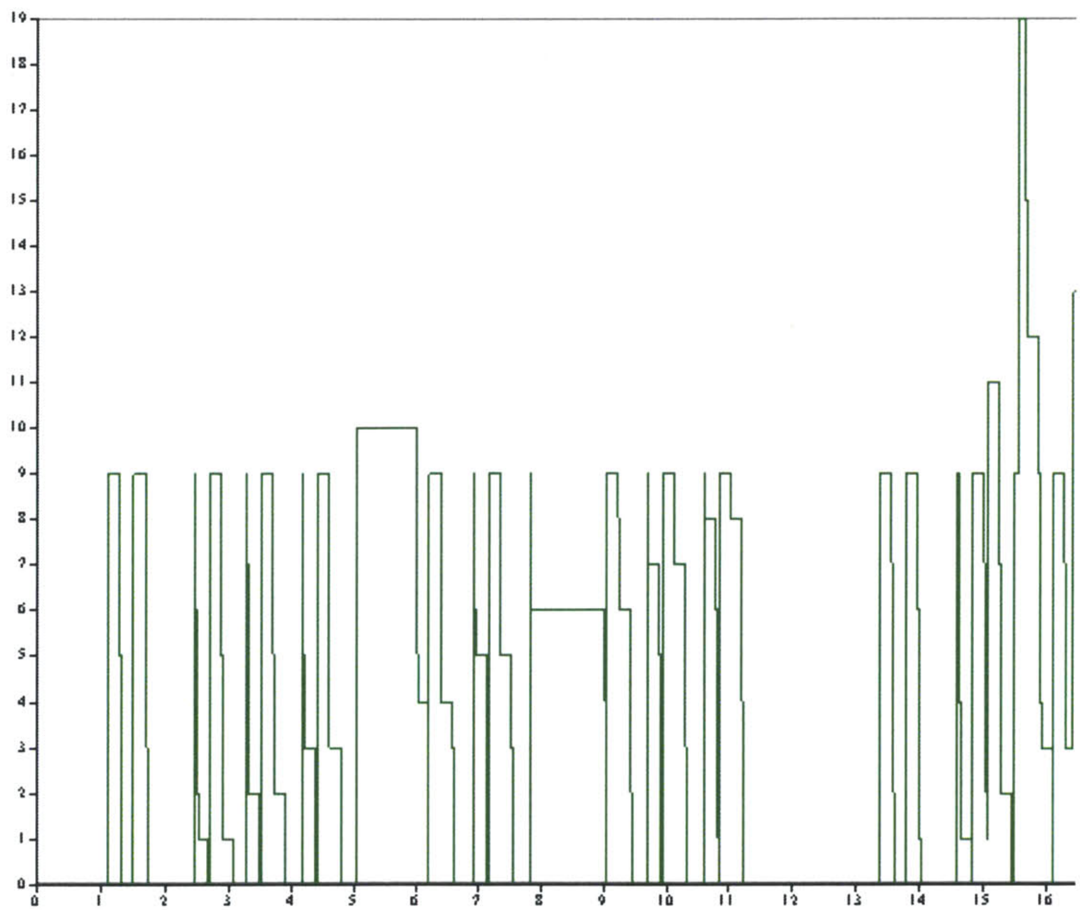


Figure 78: Graph showing CC WIP for Section 8.1.5 Case 1, with *batch*ed transport

The graph in Figure 78 shows the machine buffer reaching ten parts one hour into the shift. This confirms that the *batching* module is working and parts are being transferred in batches of 10. The graph is marked with spikes of ten parts periodically but around 15 hours there is a sudden spike. This can be attributed to a breakdown that reduced the machine availability. A comparison of Figure 78 with Figure 77 shows that the WIP in this scenario has decreased and there is lesser production fluctuation. In all, it seems to be a better strategy to batch products before they are transported to the CC area.

Case 2. One CC in line with loop flow of products

Eventually the CC machine will be relocated to the main line and parts will be transferred on conveyors. This scenario simulates the CC machine working in line with other equipment as seen in Figure 79. The simulation logic is same as in Case 1 but with a revised *routing* time of 10 seconds to emulate part transfers using conveyors.

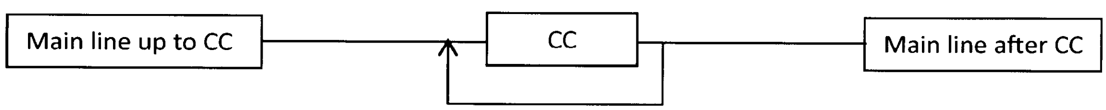


Figure 79: One CC machine, located in line with other equipment

Results:

Table 18 shows the results derived from the simulation setting in Section 8.1.3 Case2. The results show a decrease in WIP in comparison to Case1 and it can be traced to the lower number of *entities* waiting in transit. The CC utilization has not changed much as the underlying process structure is the same. The SMX production has decreased and batching of drives for CC has proven to be beneficial.

Table 18: Results from Section 8.1.5 Case2

Type	Average Work in progress per week	Average CC queue per week	CC utilization	Average product mix per week	
CC in line with two loops	187.17	4.46	0.7187	SMV1	749.2
				SMV2	745.8
				SMX1	758.4
				SMX2	760.6

The graph in Figure 80 shows a trend similar to the graph in Figure 79; this tells us that the line will not behave differently based on the location of the single CC machine.

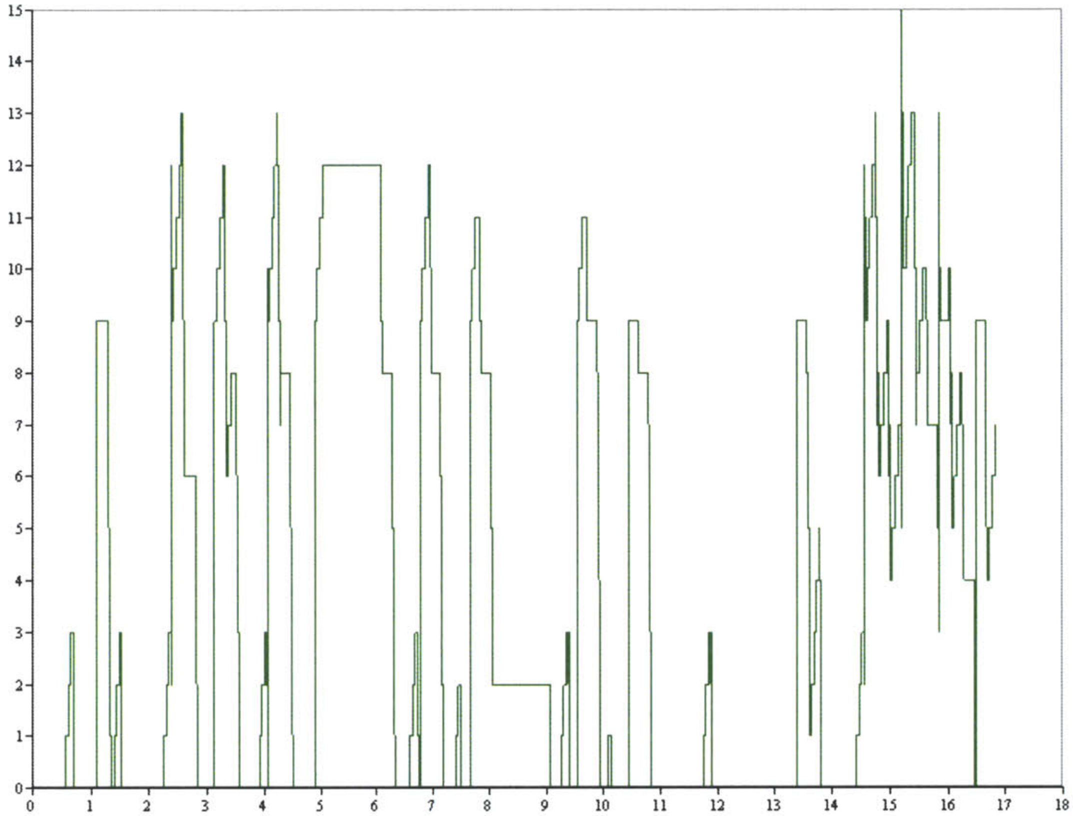


Figure 80: WIP graph for CC from Section 8.1.5 Case 2

Case 3. Two CC lines in line

The final vision of the proposed line is to include two CC machines in the main line. Each CC machine will coat one side of the board. This will hugely decrease the processing times of the CC operation and the machine utilizations are expected to decrease. The lower utilization at CC is required as this additional capacity will be shared with other lines to process boards of other products.

The final scenario with two CC stations in the main line is seen in Figure 81. There is no self-loop on the CC machine boxes anymore, and the two CC machines are placed back to back.

Figure 81: Two CC machines located in line with other equipment

Simulation model:

The simulation logic in Figure 82 represents a line configuration with two CC stations on the main line. The logic is revised with an addition of a *delay* module to represent the second CC machine but the rest of the logic remains the same as explained in Section 8.3.6.

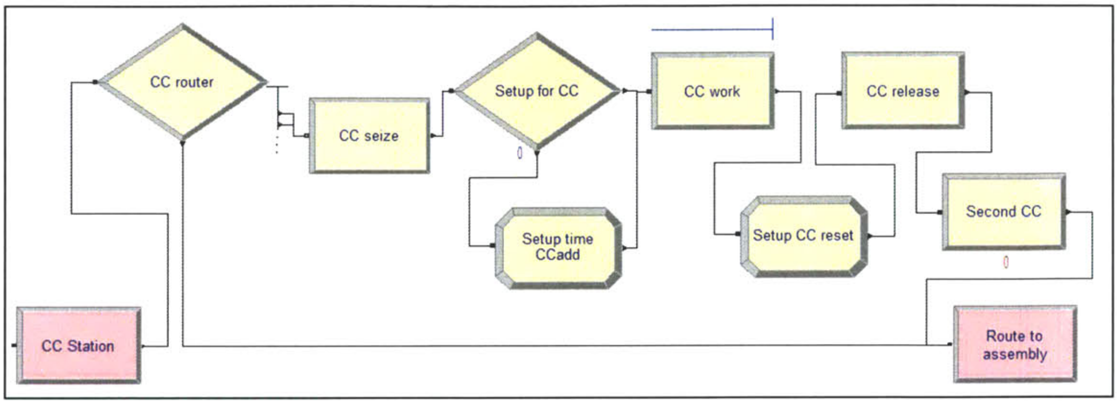


Figure 82: Simulation logic for two CC machines on the main line

Results:

Table 19 shows the CC output for the simulation setting for Case 3. The results show that utilization of the CC *resource* has decreased nearly 25% and the average inventory has gone down by a factor of four. These changes were anticipated as the line was running underutilized capacity in the CC area (Case2). The overall mix of products is balanced and not very different from Case 2.

Table 19: Results from Section 8.1.5 Case3

Type	Average Work in progress per week	Average CC queue per week	CC utilization	Average product mix per week	
Two CC in line	183.85	0.56	0.2848	SMV1	748.8
				SMV2	741.2
				SMX1	734.4
				SMX2	727.2

Figure 83 shows the WIP inventory awaiting CC operation. The spikes in Figure 83 are more spaced out than in any of the previous cases, and the queue length does not increase beyond 3 parts. This behavior is the result of lower operation times on the CC module. Overall the two CC machines are not improving the line's production but the two CC machines create ample room for resource sharing with other lines.

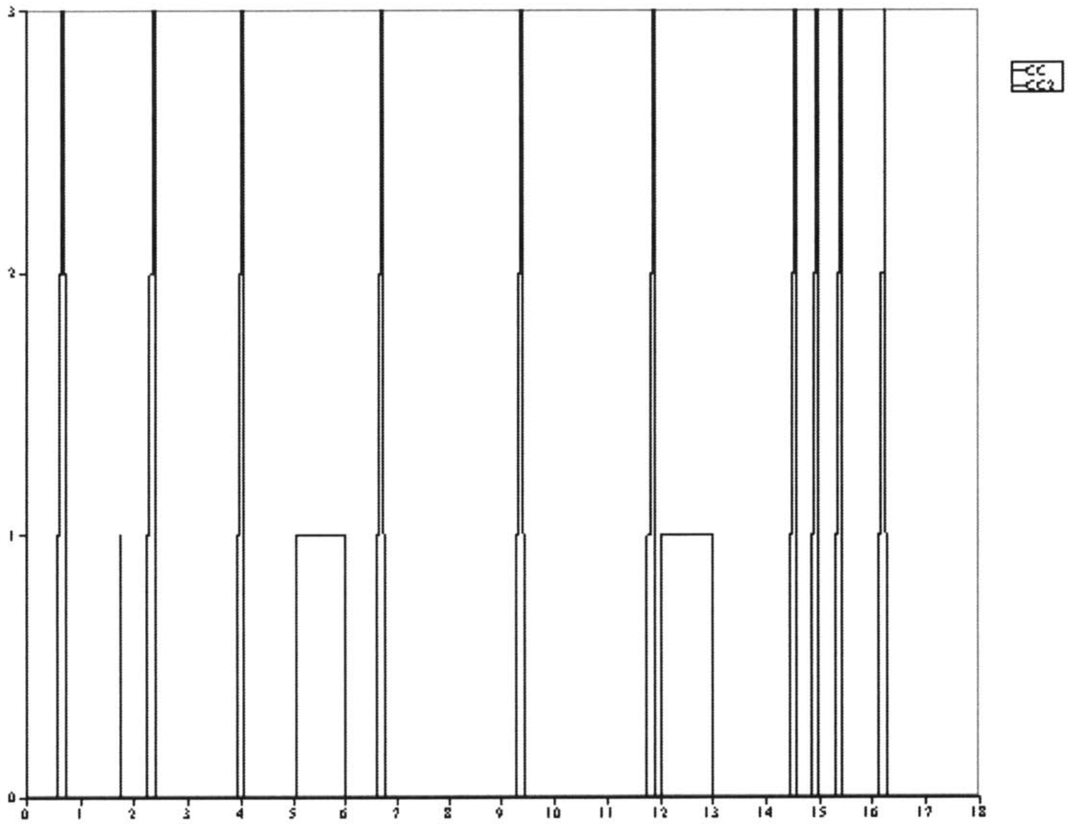


Figure 83: CC WIP graph for Section 8.1.5 Case 3

8.1.6 Eliminate board test from the SMX process flow

SMX drives use a double sided board which eliminates any manual board to board assembly. This means the SMX drives will have higher board reliability and the drive quality can be judged on the drive test only. Thus, there is potential to eliminate board test from the SMX build sequence. This scenario is designed to predict the impact of eliminating board test from the SMX drive build sequence.

Simulation model:

The 'SMX board test times' were set to zero (as seen in the expression field of Figure 84), denoting that the SMX *entities* would bypass the board test area without any *delay*.

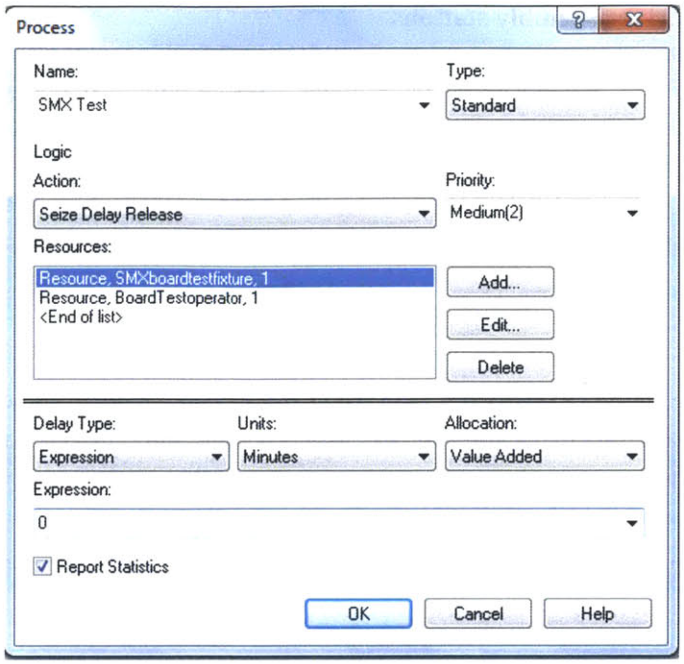


Figure 84: *Process* module setting to show board test elimination

Results:

Table 20 shows the results of the simulation setting for Section 8.1.6 Case3. Results in Table 20 show that board test is not the bottleneck of the new line and hence changes made to the station have a marginal effect on the line's output. The average WIP and average weekly production show a small improvement but it is not substantial. Although the results do not show any advantage in eliminating board test it will be considered for removal because of the high maintenance involved with the up-keeping of test fixtures.

Table 20: Results for simulation setting from Section 8.1.6

Type	Average Work in progress per week	Average output per week	Resource utilization		Average product mix per week	
Eliminate board test	177.7	2962.2	Drive assembly line	0.6936	SMV1	747.8
			SMT2	1	SMV2	745.6
			SMT1	0.733	SMX1	738
			SMV board test fixture	0.4147	SMX2	730.8
			SMV final test fixture	0.6431		
			SMX board test fixture	-		
			SMX final test fixture	0.6085		

8.1.7 Number of drive assembly stations

As pointed out in the decision tree analysis from Section 2.1, manpower planning was one of the major concerns of the manufacturing department. The current man power requirements are planned on a monthly basis. The number of people in the plant depends on the forecasted sales for the next month. Drive assembly is a labor intensive process and uses the most number of line operators amongst all the stations on the existing shop floor. A sensitivity analysis of the line output to the number of drive assembly stations will build an intuition about the manpower required for a certain output.

Simulation model:

This scenario is set up by changing the capacity of the *resource* module shown in Figure 85. The value in the field 'schedule name variable' was varied starting with one drive assembly station per line to emulate the number of corresponding drive assembly stations and the drive assembly stations utilization were noted.

Resource - Basic Process

	Name	Type	Schedule Name
1	Aoi1	Based on Schedule	Normal single person availability two
2	Stencil1	Based on Schedule	Normal single person availability
3	Oven1	Based on Schedule	Normal single person availability
4	Stencil2	Based on Schedule	Normal single person availability
5	PP2	Based on Schedule	Normal single person availability three
6	Oven2	Based on Schedule	Normal single person availability
7	PP3	Based on Schedule	Normal single person availability
8	THoperator	Based on Schedule	Normal single person availability
9	Wave	Based on Schedule	Normal single person availability
10	Touchup	Based on Schedule	Normal single person availability
11	BoardTestoperator	Based on Schedule	Board test Operator availability
12	SMVboardtestfixture	Based on Schedule	Normal single person availability two
13	SMXboardtestfixture	Based on Schedule	Normal single person availability two
14	CC	Based on Schedule	Normal single person availability
15	Coveroperator	Based on Schedule	Normal single person availability
16	Resource 1	Based on Schedule	Normal single person availability
17	Driveassemblyline	Based on Schedule	Final assembly Operator availability

Figure 85: Part of the *resource* data set showing drive assembly capacity

Table 21 shows the line output and the drive assembly utilization for a varying number of drive assembly stations. The values from output column of Table 21 are plotted in the graph in Figure 86. As seen in Table 21 and Figure 86 the drive assembly stations have 100% utilization for two drive assembly stations per line. This tells us that the drive assembly will be a bottleneck if we run the line with just two drive assembly stations. As we increase the number of drive assembly stations, the output increases until it reaches a limit around 3-4 assembly stations.

Table 21: Line output variation due to change in number of drive assembly station

Number of assembly stations	Drive assembly utilization	Output/week
1	1	1107.8
2	1	2181.2
3	0.8945	2935.4
4	0.6706	2951.8
5	0.5361	2953.2
6	0.4468	2949
7	0.3829	2952.6
8	0.335	2953
9	0.2978	2952.8
10	0.268	2952.8

This is the optimal setting for the number of drive assembly stations per line. It is also seen that the drive assembly station utilization erodes rapidly after four drive assembly stations per line, and it is not productive to increase the number of drive assembly stations per line beyond four.

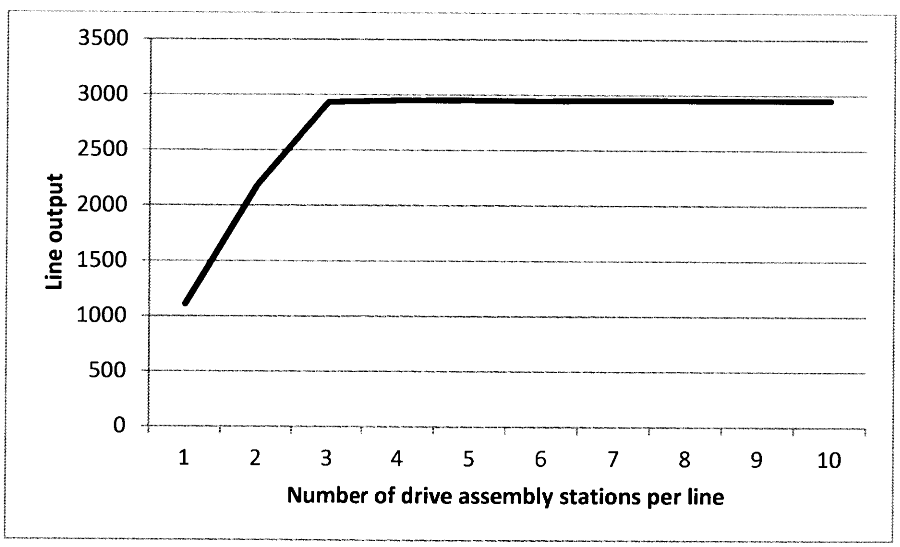


Figure 86: Line output variation due to change in number of drive assembly stations

8.2 Conclusions

Table 22 shows the equipment capacities for the new line derived from the simulation results in this chapter.

Table 22: Equipment capacities from the simulation results

	Name of station	Equipment Capacity
1	SMT	4 Pick and place machines
2	TH	2 Operators
3	AOI	1 AOI machine
4	Wave	1 Wave machine
5	Board test fixture	2 Board test fixtures per product
6	CC	1 CC machine
7	Drive Assembly	5 Assembly stations
8	Final test fixture	2 Final test fixtures per product
9	Packaging	1 Operator

A comparison of Table 22 values with Table 9 shows that the analytical solution provided in chapter 6 is at par with the simulation scenarios. The only change is seen at the SMT capacity which now has an extra pick and place station as per Table 22. The additional pick and place station was required as SMT2 was consistently identified as a bottleneck and addition of four more products to the simulation made the bottleneck more severe.

Additional observations:

- SMT2 has consistently shown to be a bottleneck and it should not be used as a shared resource.
- Temporary storage racks are necessary at SMT and final test station as these stations have shown to accumulate inventory sporadically due to breakdown.
- Manpower needs to be shifted from drive assembly to final test if we plan to run the line to produce SMX drives only.
- All three CC configurations are feasible but it is beneficial to run batches of similar products if we are operating with one machine per line.
- Eliminating board test from the SMX build does not affect the line output drastically.
- We need at most five drive assembly stations per line to produce a maximum output and any additional drive assembly stations do not add value.

9. Recommendation of revised floor plan and future work

This chapter summarizes findings from all the previous chapters to provide an implementation plan that will help the company layout their factory floor on one piece flow. The material presented in this chapter is based on work done by Joseph Falvella and the author together. Sections on current factory floor layout and milestone layouts are based on work done by Joseph Falvella[15].

9.1 Factory floor plans

This section describes the arrangement of equipment and material flow in the existing factory floor. The development of the new factory floor plan is based on the learning from this section.

9.1.1 Existing factory layout

Figure 87 shows the blue print of the existing shop floor. The layout in Figure 87 was created after numerous shop floor tours and interviews with shop personnel to understand material movement and location of equipment[15]. The layout in Figure 87 was important in understanding existing shop floor constraints like location of warehouse, electrical routings and office space. Figure 87 is a stepping stone to the next sections and helped build an intuition about the factory's operations.

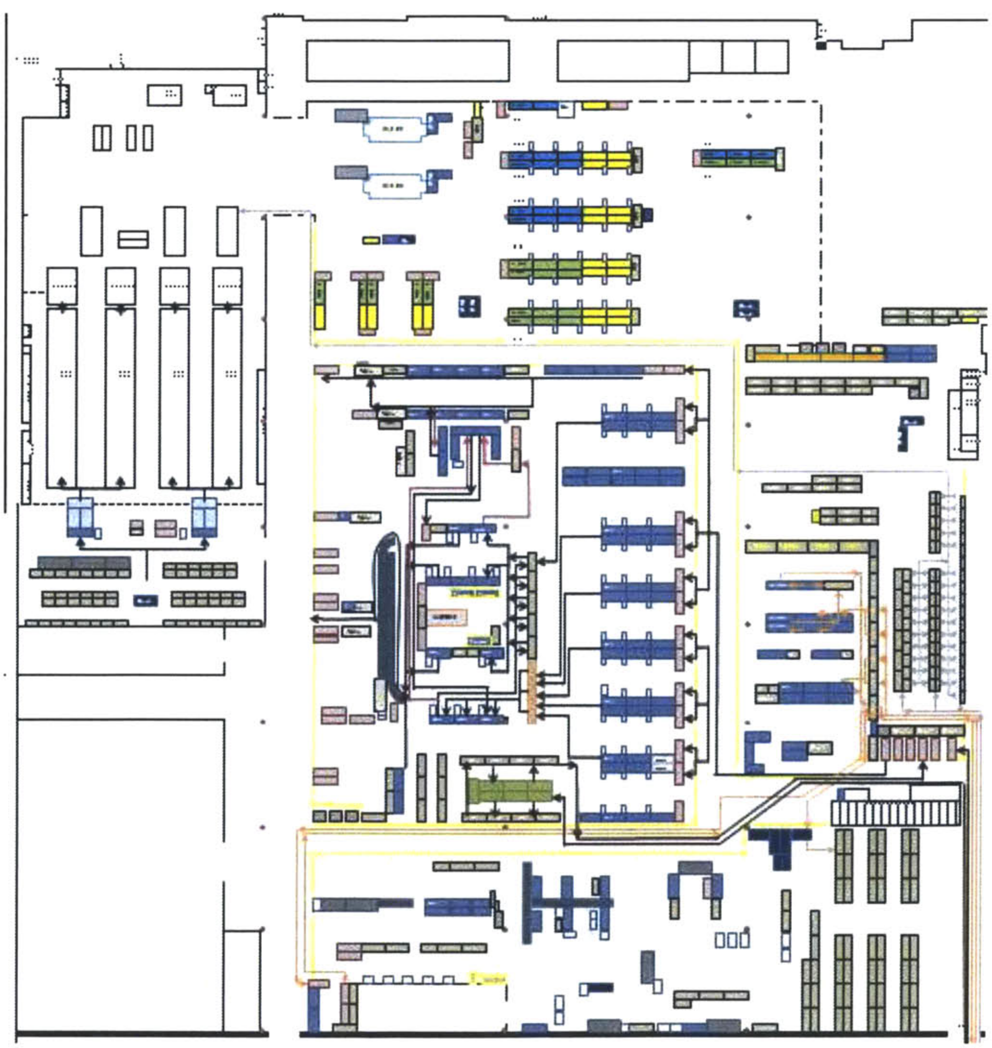


Figure 87: Arrangement of equipment in the existing factory floor [15]

9.1.2 Shop floor footprint of the proposed line

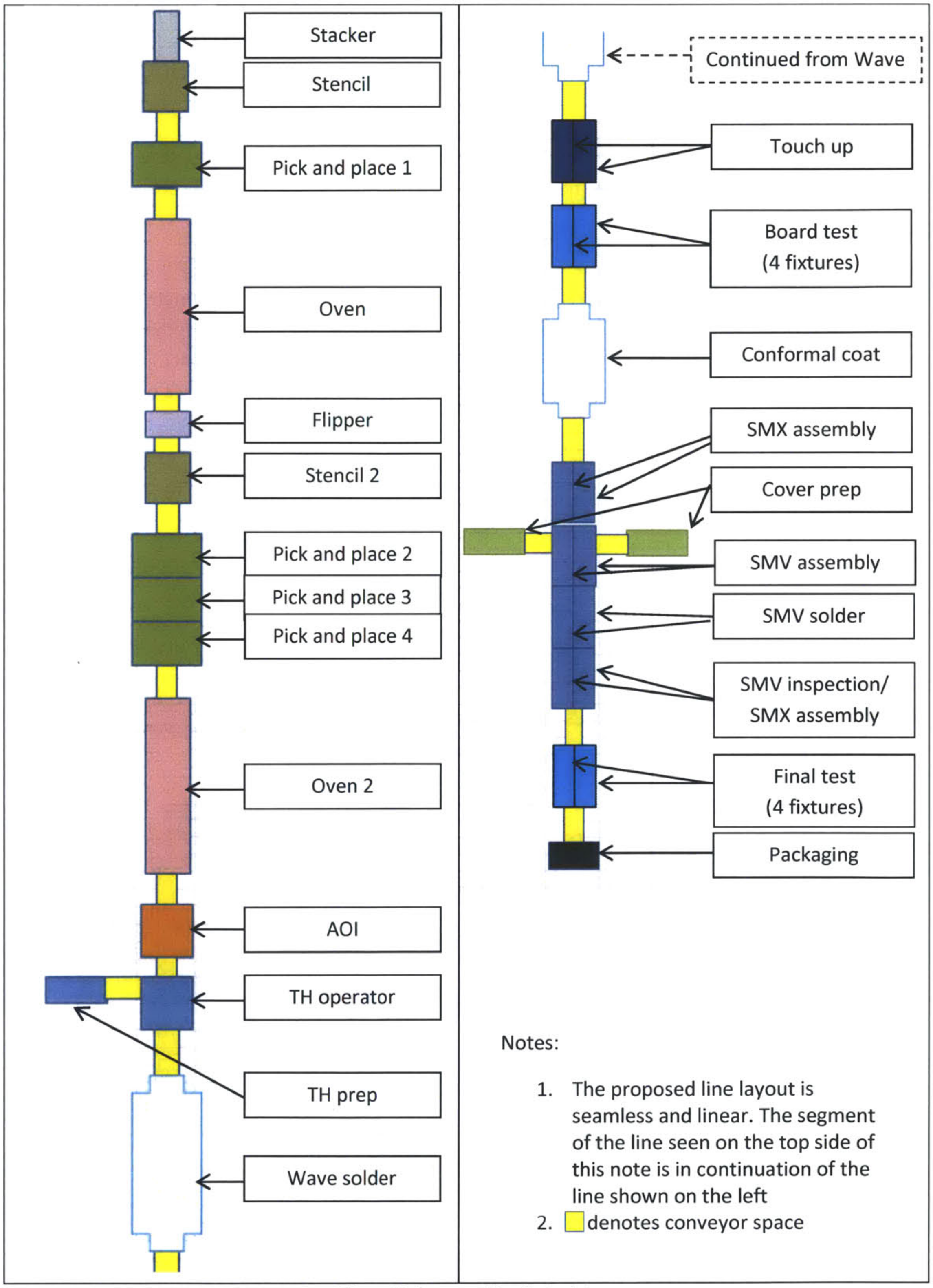


Figure 88: Proposed line footprint

This section was developed from the findings of this thesis and Joseph Falvella's thesis. The primary design of the new line and equipment capacity requirements for the line proposed in Figure 88 were derived from the simulation results presented in Chapter 8. The floor plan for the new line shows equipment sequenced in a linear format with conveyors transferring material from one station to the other. All equipment was drawn to shop floor dimensions in the actual software but Figure 88 is not to scale. The footprint of the line will be used to create the planned factory floor plan. Multiple instances of the proposed line's design will be used to create the planned layout in Section 9.1.3.

9.1.3 Planned factory layout

Figure 89 shows the reconfigured layout of the factory with new lines as well as relocation of existing production lines. The new floor layout is capable of producing 360,000 SMX drives per year and sustaining a part of the existing production capacity. The layout has three new lines highlighted in yellow and existing production equipment is moved to the right half of the factory as seen in Figure 89. A detailed roadmap was created that will transform the factory from Figure 87 to Figure 89. A detailed analysis of the milestone layouts can be found in Joseph's thesis. Special care was taken not to disrupt existing production but to keep making strategic moves to reach the planned.

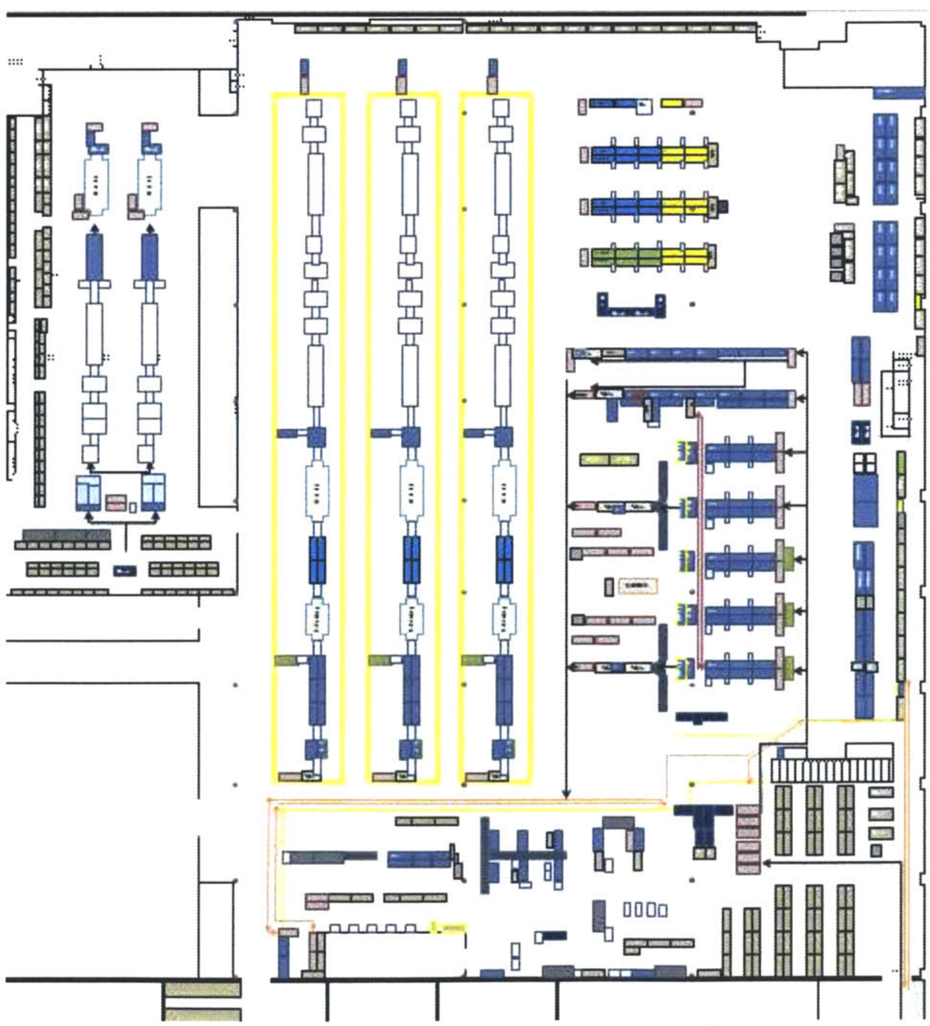


Figure 89: Planned factory floor plan [15]

9.2 Future work

9.2.1 Optimizing material travel paths and updating the simulation model

The final layout of the shop floor presented in Section 9.1 is designed considering the space constraints on the floor. A thorough analysis of this layout based on travel paths and material storage is required. The revised shop floor layout will have an optimized flow path that will ensure minimum material transport time.

The current simulation model is designed to produce four product types and simulates just one line. The equipment proposed in the new line will be shared with other product lines eventually and the impact of the resource sharing needs to be analyzed to accurately predict the factory output. It is recommended that the simulation model be extended to accommodate other product lines so that the overall factory output can be simulated. The frequent updating of the simulation model as changes are made to the equipment on the floor will determine how closely the model represents the actual operations.

9.2.2 Making the simulation model more accessible

Arena has the capability of integrating the simulation model with software like Microsoft Excel which will make the model more accessible to a broader user base. This will ensure sustenance of the project and the model can be used by the manufacturing personnel in designing process plans and as a decision making tool in designing new layouts. The blocks used in the simulation model can be set up to run as individual simulation files, thus providing access to each individual block which would allow the area managers to use the tool for their capacity planning and job scheduling tasks.

Bibliography

[1] http://www.lenzeamericas.com/

[2] http://www.vfd.ca/scf_Series.asp

[3] H. Czarnecki and N. Loyd, "Simulation of Lean Assembly Line for High Volume Manufacturing," vol. Center for. pp. 1–6, 1993.

[4] D. S. Cochran and J. W. Duda, "Simulation and production planning for manufacturing cells," *1998 Winter Simulation Conference*, pp. 973–978, 1998.

[5] N. Ngampak and Busaba Phruksaphanrat, "Cellular Manufacturing Layout Design and Selection : A Case Study of Electronic," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 2, 2011.

[6] http://www.directindustry.com/prod/essemtec-ag/smt-pick-place-machines-55556-456982.html

[7] http://www.sciencedirect.com/science/article/pii/S1568494609002245#fig1

[8] D. McCalib, "Design of an upgradeable electronic PCB testing System," 2013.

[9] Mc Auley John, "Machine Grouping for Efficient Production," no. February. 1972.

[10] J. Miltenburg, "One-piece flow manufacturing on U-shaped production lines : a tutorial," *IIE Transactions*, vol. 33, pp. 303–321, 2001.

[11] J. Black, "Design rules for implementing the Toyota Production System," *International Journal of Production Research*, vol. 45, no. 16, pp. 3639–3664, Aug. 2007.

[12] http://electronicdesign.com/test-amp-measurement/test-matter-life-or-death- automotive-industry

[13] J. P. C. Tong, F. Tsung, and B. P. C. Yen, "A DMAIC approach to printed circuit board quality improvement," *The International Journal of Advanced Manufacturing Technology*, vol. 23, no. 7–8, pp. 523–531, Apr. 2004.

[14] C. Grewal, "An initiative to implement lean manufacturing using value stream mapping in a small company," *International Journal of Manufacturing Technology and Management*, vol. 15, no. 3/4, p. 404, 2008.

[15] J. Falvella, "Continuous Systematic Improvement Form and Functionality Optimization of an Assembly System Design," 2013.

[16] http://sumitshrivastva.blogspot.com/2012/03/industrial-engineering-facilities.html

[17] http://maaw.info/Chapter8.htm

[18] http://www.see.ed.ac.uk/~asherloc/OM/Notes/Layout/Layout.html

[19] A. Kusiak, "The generalized group technology concept," *International Journal of Production Research*, vol. 25, no. 4, pp. 561–569, Apr. 1987.

[20] E. S. Eneyo and G. P. Pannirselvam, "The use of simulation in facility layout design: a practical consulting experience," *1998 Winter Simulation Conference*, pp. 1527–1532, 1998.

[21] D. J. Miller, "Simulation of a semiconductor lin," *Communication of the SCM, ACM*, vol. 33, no. 10, pp. 99–108, 1990.

[22] P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and W. Evans, "Optimization of cycle time and utilization in semiconductor test manufacturing," pp. 727–735, 1999.

[23] D. J. Medeiros, E. F. Watson, J. S. Carson, B. S. Vaidyanathan, D. M. Miller, and Y. H. Park, "Application of discrete event simulation in production scheduling," *Proceedings of the 1998 Winter Simulation Conference*, 1998.

[24] S. B. Gershwin, "Lecture notes, 2.854 Mechanical Engineering, MIT." 2012.

[25] T. A. Spedding and G. Q. Sun, "Application of discrete event simulation to the activity based costing of manufacturing systems," *International Journal of Production Economics*, vol. 58, no. 3, pp. 289–301, Jan. 1999.

[26] D. E. Hardt, "Lecture notes, 2.830 Mechanical Engineering, MIT," 2013.

[27] M. K. Uddin, J. Luis, and M. Lastra, "Assembly Line Balancing and Sequencing," *InTech*, no. section 2, 2011.

[28] M. Burman, "New Results in Flow Line Analysis," MIT, 1995.

[29] S. Gershwin and M. Burman, "A Decomposition Method for Analyzing Inhomogeneous Assembly/Disassembly Systems," *Annals of Operations Research*, vol. 93, pp. 91–116, 2000.

[30] S. B. Gershwin, *Manufacturing systems engineering*, Prentice Hall, 1993.

[31] D. W. Kelton, R. P. Sadowski, and N. B. Swets, *Simulation with Arena citation file*. Mc Graw Hill, 2009.

[32] G. R. Drake, J. S. Smith, and B. A. Peters, "Simulation as a planning and scheduling tool for flexible manufacturing system," *Proceedings of the 1995 Winter Simulation Conference*, 1995.

[33] T. M. Tirpak, "Developing and Deploying Electronics Assembly Line Optimization Tools : A Motorola Case Study," vol. 2, no. 1, pp. 63–78, 2008.

[34] S. F. Qin, R. Harrison, A. A. West, and D. K. Wright, "Development of a novel 3D simulation modelling system for distributed manufacturing," *International Journal of Computers In Industry*, vol. 54, no. 1, pp. 69–81, 2004.

[35] S. Ramakrishnan, P. Tsai, K. Srihari, and C. Foltz, "Using Design of Experiments and Simulation Modeling to Study the Facility Layout for a Server Assembly Process," *Proceedings of the 2008 Industrial Engineering Research Conference*, pp. 601–606, 2008.

Appendix

Section A: Machine breakdown times

1. SMT

MTTF and MTTR calculations for SMT			
Date	**Time**	**Downtime(in minutes)**	**Uptime (in hours)**
23-06-2013	08:55	115	50.4
20-06-2013	08:25	25	16.5
19-06-2013	08:00	30	32.9
17-06-2013	07:00	60	79.5
13-06-2013	18:35	40	12.4
13-06-2013	07:00	15	25.6
12-06-2013	15:40	5	23.7
11-06-2013	22:25	60	18.3
10-06-2013	23:45	20	7.4
10-06-2013	07:10	195	63.2
07-06-2013	19:25	50	13.7
07-06-2013	09:10	80	15.5
06-06-2013	07:40	60	84.2
01-06-2013	07:00	30	50.9
30-05-2013	23:59	30	134.0
22-05-2013	22:10	40	28.9
21-05-2013	10:05	45	25.0
20-05-2013	18:10	45	65.7
16-05-2013	16:00	20	34.0
14-05-2013	16:00	15	5.5
14-05-2013	10:30	10	17.1
13-05-2013	10:40	10	58.8
10-05-2013	18:30	15	2.5
10-05-2013	16:00	15	35.9
08-05-2013	18:00	10	198.2
26-04-2013	12:30	10	55.2
23-04-2013	16:45	290	
		MTTF (in minutes)	**2665.7**

2. Wave

MTTF and MTTR calculations for Wave			
Date	**Time**	**Downtime(in minutes)**	**Uptime (in hours)**
20-12-2012	08:30	25	
26-12-2012	07:30	30	3000
28-12-2012	10:45	45	1215
21-01-2013	07:00	110	12015
25-01-2013	10:30	50	2250
29-01-2013	03:00	30	1590
14-02-2013	07:35	15	8435
20-02-2013	11:00	90	3265
27-02-2013	10:00	90	3510
07-03-2013	13:15	45	4275
11-03-2013	07:00	45	1665
14-03-2013	13:20	35	1910
15-03-2013	07:00	30	130
17-03-2013	07:10	30	1030
19-03-2013	09:15	105	1145
18-04-2013	07:00	120	15165
18-04-2013	09:00	60	120
09-05-2013	07:00	450	10590
14-05-2013	07:00	40	2550
03-06-2013	10:00	81	10380
08-06-2013	07:00	210	2370
13-06-2013	11:00	90	2790
13-06-2013	12:30	180	90
18-06-2013	11:00	90	2460
		MTTF (in minutes)	**3997.8**

3. CC

MTTF and MTTR calculations for CC			
Date	**Time**	**Downtime(in minutes)**	**Uptime (in hours)**
27-12-2012	11:00	20	
02-01-2013	10:57	93	3057
22-01-2013	10:45	15	10188
30-01-2013	07:30	60	3885
31-01-2013	07:00	120	480
05-02-2013	07:30	210	2580
		MTTF (in minutes)	**4038**

Section B: Operation times

1. SMX labor standards summary:

(All times are in minutes)	Quantity	Unit Time	Total
Apply Heat sink compound to Module	2	0.2	0.5
Assembly PCB to Heat sink	1	0.2	0.3
Assembly Screws	6	0.3	2
Install Ground Clamp Screw Assemblies	2	0.4	0.7
Apply Label to Cover	0		
Assembly Cover	1	0.5	0.5
Print Data plate	1	0.3	0.3
Label with Data plate	1	0.2	0.2
Total Assembly			4.4
Total Assembly with PFD			5.1
Test	1	3	3
Assembly EPM	1	0.3	0.3
Assembly Plugs	3	0.3	0.8
Total Test			4
Total Test with PFD			4.7
Packaging			2.5
Total Labor			12.3

2. SMV time study summary:

SMV drive assembly (all times in seconds)									
Process step / Time study no.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Average**
Apply Silicone	23	20	17	19	20	15	17	15	18.3
Apply Ins. Patch	18	12	13	12	10	12	10	10	12.1
Install spacers, collars, Zip	45	40	50	41	44	40	50	42	44.0
Install bottom components	21	22	34	30	14	18	20	22	22.6
place board screws in board	37	30	38	38	31	32	35	30	33.9
tighten screws	28	27	28	30	35	30	40	36	31.8
install bracket	15	12	12	10	10	12	8	10	11.1
clip leads	37	30	40	23	27	32	37	40	33.3
dump leads	9	8	10	10	10	10	8	10	9.4
inspect	7	6	6	6	5	3	4	6	5.4
Pick drive	10	10	11	10	1	5	10	5	10.0
Solder (33 joints)	90	106	126	110	95	78	135	75	95.0
Inserting capacitor (orange)	7	17	8	12	5		4	6	3.0
Cleaning with alcohol	9	15	15	8	5	10	10	3	15.0
Insert tie wrap x2	12	14	14	13	8	15	16	18	18.0
Apply silicon (2 spots)	13	13	11	10	6	11	10	10	8.0
Insert capacitors (2 no.)	13	16	18	12	11	77	5	9	7.0
Soldering caps (4 joints)	28	28	23	24	28	75	28	30	28.0
Clip leads of orange cap	5	4	5	4	2	2	2	3	6.0
Tighten and clip tie wrap	10	14	13	10		5	5	9	7.0
Pick drive	2	5	5	3	4	3	2	6	4.0
Inspection	26	38	48	30	18	33	26	38	87.0
Cover assembly	33	21	12	14	20	26	30	28	16.0
Insert fixture and topple	5	6	8	8	6	5	3	5	6.0
Color machined surf	21	19	28	20	18				
Insert screw and torque	52	72	75	46	54	72	59	68	47.0
Read bar code	13.5	13	3	9	4	8	8	12	7.0
Print	5	5	7	7	5	4	4	5	9.0
Stick label	10	10	10	4	14		7	5	23.0
Scan new bar code	8	8	4	7.5	7.5	7.5	7.5		
Put finish part in bin	3	3.5	3	6	3	5	5	5.5	5.5
Total	**615.5**	**644.5**	**695**	**586.5**	**520.5**	**645.5**	**605.5**	**561.5**	**623.3**

Section C: Simulation building blocks

Arena provides numerous flow chart modules that were used in designing the simulation model. However, the five basic modules that were used the most are described below.

1. *Create/Dispose*

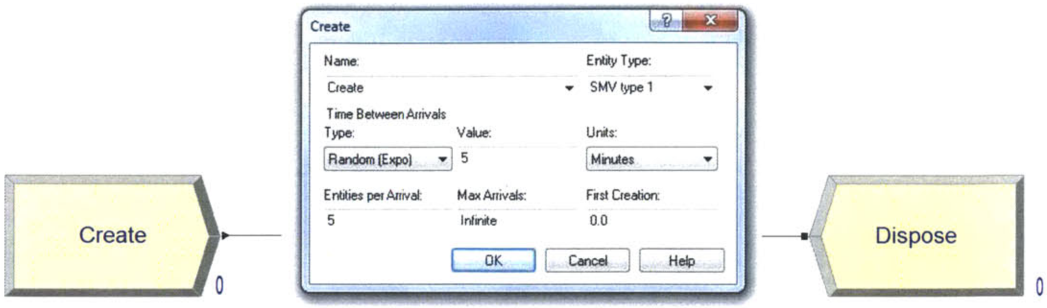


Figure 90: *Create* and *dispose* module and data input window

The *create* module is responsible for *entity* generation and is an entry point for an *entity* in the simulation. The dialog box seen in Figure 90 shows various options available to define *entity* characteristics. The text in the 'Name' field is displayed on the module and the 'Entity Type' is the product family of the generated *entity*.The time between *entity* introductions can be driven by random distrutions like normal, exponential, beta,etc or it can be user defined. The 'First creation' field in Figure 90 is used to define the time of first *entity creation*.

The *dispose* module terminates the travel path of an *entity* and captures *entity* related statistics like time in system, value added/non-value added time, etc.

2. *Assign*

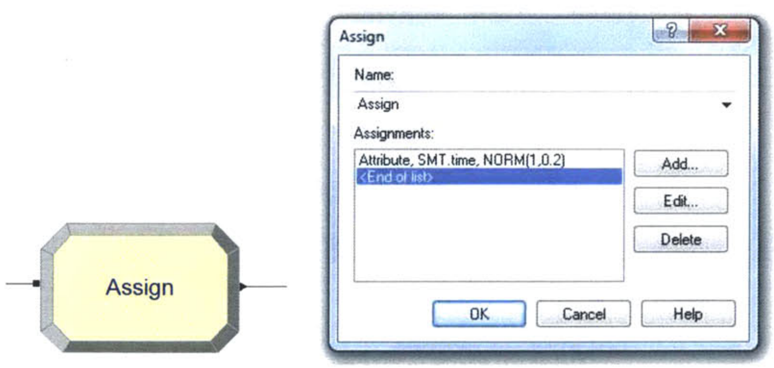


Figure 91: *Assign* module and data input window

The *assign* module is used to designate *attributes* to every *entity* that passes through it. This module was used to assign process times on different machines using *attributes* as shown in Figure 91. Another way to use this module is to keep count of number of *entities* passing through a strategic location on the line.

3. *Process*

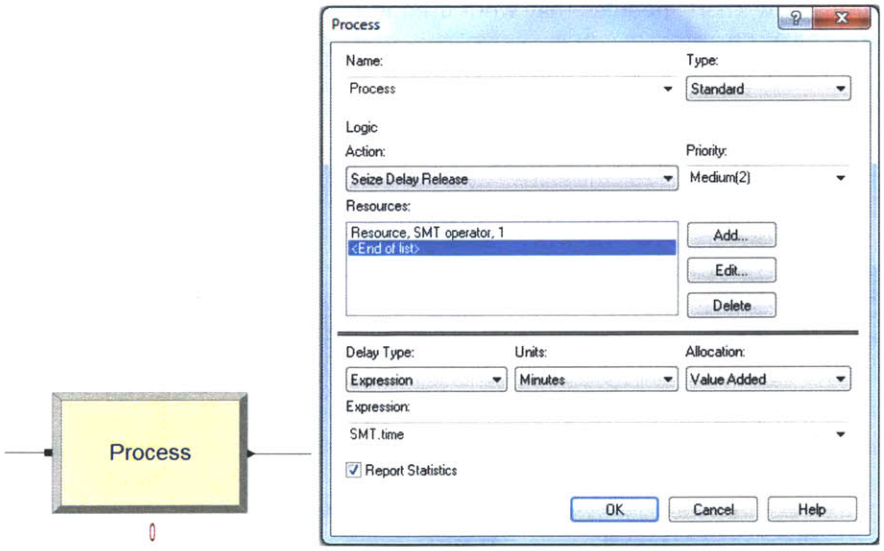
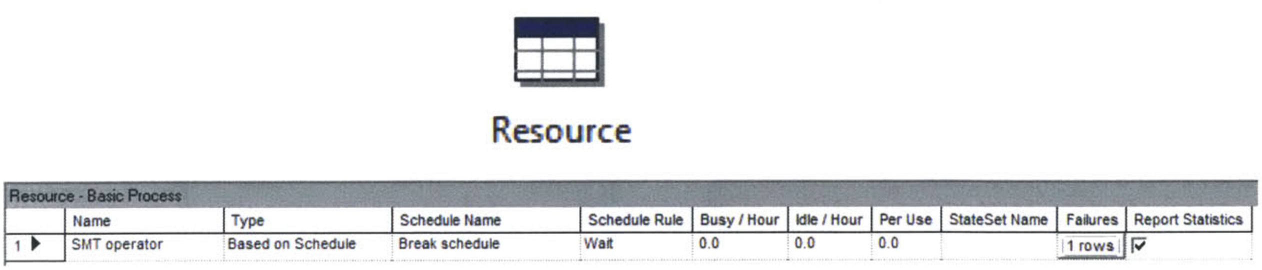


Figure 92: *Process* module and data input window

The *process* module is used to represent operations like CC and wave solder. The module works by *delaying* an *entity's* progression in the simulation by a time equivalent to the operation time at that station. The dialog box in Figure 92 shows details that define a *process* module. The 'action' field is set to 'seize delay release' which means the process cannot be initiated unless the requried *resource* is available. After the *resource* is *seized* the *entity* is held in a *delay* equivalent to the operation time following which it is transferred to the next station and the *resource* is released. The *delay* time can be defined in the *process* module as a number or as an *attribute* from the *assign* module. In Figure 92 the 'delay' field reads expression 'SMT.time' which points to an *attribute* that was assigned at a previous *assign* module.

4. *Resource*



	Name	Type	Schedule Name	Schedule Rule	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	SMT operator	Based on Schedule	Break schedule	Wait	0.0	0.0	0.0		1 rows	☑

Figure 93: *Resource* data module input field

A '*resource*' is a machine or human operator required to perform a process. The *resource* data set maintains records of all *resources* used in the model and is shown in Figure 93. Our system deals with machines like SMT, wave, CC, test fixtures and human operators, all of which are defined in the *resource* data module. *Resource* availability is governed by a 'schedule rule' and a breakdown trend defined in the 'Failure' field. These features help in creating a realistic representation of the machine.

5. *Decide*

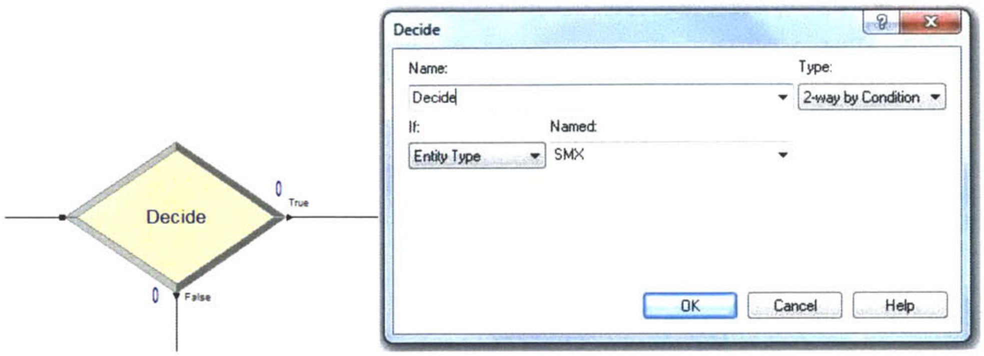


Figure 94: *Decide* module and data input window

The 'decide' module emulates an "if-else" logic and can be used to represent quality test stations like board/final test. The condition is decided by input in the field 'Type' (Figure 94) and it can be based on percentage values or expression driven. Most 'decide' modules used in our model are expression driven as it provides greater flexibility in modeling.