INTEGRATED PROCESS DESIGN, PLANNING,

AND STOCHASTIC SYSTEM OPTIMIZATION

Approved by:

____

Dr. Richard S. Barr

____

Dr. Jerome L. Elkind

____

Dr. Richard V. Helgason

____

Dr. Eli V. Olinick

____

Dr. Jerrell R. Stracener

INTEGRATED PROCESS DESIGN, PLANNING, AND STOCHASTIC SYSTEM OPTIMIZATION

A Praxis Presented to the Graduate Faculty of the

School of Engineering and Applied Science

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Engineering

with a Major in

Engineering Management

by

Osama Ahmed Elsafadi

(B.S.E.E., University of Southern California, 1992)
(M.S.E.E., University of Southern California, 1993)

TBD, 2009

ACKNOWLEDGEMENTS

TBD

Osama Ahmed Elsafadi B.S.E.E., University of Southern California, 1992
M.S.E.E., University of Southern California, 1993

Integrated Process Design, Planning and Stochastic System Optimization

Advisor: Professor Richard S. Barr

Doctor of Engineering degree conferred TBD, 2009

Praxis completed TBD, 2009

An integrated process design, planning and stochastic-system-optimization model is developed to address dynamic manufacturing, operations, planning, and quality requirements based on customer needs while focusing on reducing cost and increasing profit. The model optimizes the process steps to achieve the output goals at minimum cost in the presence of engineering, business, and quality constraints. A case study ~~in~~ semiconductor manufacturing illustrates the value of this new approach where process decisions take into account various work-in-process quality grades and uncertain yield rates for a new product.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DEDICATION

TBD

Osama A. Elsafadi

Dallas, Texas

TBD, 2009

CHAPTER 1

INTRODUCTION

1.1 Motivation

In today's quality driven-market place, high level of quality ~~is one of the top priorities.~~ It is one of the main factors that determine the survival of products and services. ~~As~~ the quality level is often determined by the process used for product creation and delivery, manufacturing companies invest heavily in systems and methods for optimizing processes to output the desired quality level. Maintaining the desired quality level over time is a challenge as new products, technologies and services have ~~an~~ associated learning curve, and initial implementations of processes may ~~produce output of different~~ levels of quality. When customers' quality requirements are high and their need for volume production is urgent, innovative methods are often employed to achieve the desired outcome.

~~Literatures have many references to production planning and scheduling models. They~~ address both manufacturing and services [23]. Production planning addresses decisions about the size of production lots of different products, the time to produce the lots, the production facility or equipment used, and the production lots sequence. Often, the objective is to minimize the cost while meeting the forecast demand. Production planning also addresses tradeoffs between economic objectives such as the ~~cost~~ contribution to profit and customer satisfaction by quickly responding to customer or demand changes.

~~Systems~~ engineering literatures~~ are~~ rich of material covering system design process and optimization models [4]. Topics ~~usually cover~~ system design and development, test and evaluation, system analysis mathematical models, analytical tools and application of models and system analysis. ~~P~~roduction planning including supply chain management [15] and optimization models [30] is based on a predefined sequence of events. The models work within the predefined parameters to produce optimal results ~~consistent with the~~ objective function. All introduced methodologies assume a pre-determined fixed process (~~process = a~~ given), but the process design description~~s~~ affects the steps and models. System process design can be dynamic with multiple independent flows based on business environments and market conditions. There have been no noticeable literatures on incorporating both the operations planning optimization models and the process design to produce an integrated process design, planning, and system optimization model. What is missing is a way to help in design decisions in a dynamic process. Such a model considers downstream consequences of upstream decisions, and analyzes the process design parameters and the production planning constraints to produce a varying system design to address dynamic manufacturing, operations, planning, and quality requirements based on customer needs while focusing on reducing cost and increasing profit.

In this praxis, an integrated process design, planning and stochastic system optimization model is developed to meet dynamic manufacturing, operations, planning, and quality requirements based on customers' needs while satisfying the desired objective function. The model optimizes the process steps to achieve the output goals at minimum cost in the presence of engineering, business, and quality constraints. The model will help producers satisfy customers' schedule deliveries and meet the required quality targets. A case study in semiconductor manufacturing illustrates the value of this new approach where process decisions take into account various work-in-process quality grades and uncertain yield rates for a new product.

This praxis and model will help practitioners address such questions as: how to decide what quality grade to ship, what to process further, and what to scrap? Does further processing improve the output quality level? How will change in the desired quality level influence the process's throughput? How to accommodate changing customer's requirements? Such questions are particularly relevant in environments dealing with process design planning and system optimization.

1.2 Applications

Applications of the integrated process design, planning and system optimization model are broad and cover a wide spectrum of products and services. Applications include personnel manpower planning, commercial airplanes loading cycle time, and manufacturing process and quality optimizations. Human resources practitioners can use the model to decide how much to invest in training personnel to obtain the desired caliber workforce. Model may be used in manufacturing production planning to determine how much reliability testing is required before a product meets quality requirements and outlines the optimal flows that is consistent with the desired objective. The model is able to provide answers to many questions such as:

- How much processing is needed for a desired outcome?
- Will more processing improve the output quality level and at what cost?
- Is additional processing to obtain desired quality level beneficial?
- Would relaxing the output quality level result in increasing of supply and by how much?
- How much to make and inventory each period of various works-in-progress (WIPs) to achieve the finished goods delivery schedule?

In addition, the model has dynamic objective functions that can be customized based on business needs and the desired outcome. Here are examples of objective functions:

- Reduce finished goods inventory and WIPs while meeting the demand.
- Reduce cost due to limited available budget.
- Allow for increased inventory of finished good in early periods to help meet increasing demand in later periods.

1.3 Contributions

This research provides a novel methodology to incorporate operations planning and optimization with system process design. There has been no noticeable research covering the subject. The integration of these two disciplines enables the practitioners to make calculated operational decisions with the objective of improving the effectiveness and efficiency of the process. This research constructs, defines and documents a pseudo-process that has been utilized in many industries in an ad-hoc fashion to address operational issues. This methodology can be applied in a recurring manufacturing and services setting. This research also offers an in-depth testing of the approach across a variety of scenarios and assumptions often encountered in practice.

This praxis presents a case study to illustrate the effectiveness of the model. In it, a manufacturing process is designed to deal with varying quality levels while meeting the target delivery schedule. A deterministic and a stochastic mathematical formulation are presented to illustrate the use of the model. Key parameters are selected for use in a design of experiment analysis to explore how to deal with different requirements while maintaining profitability.

CHAPTER 2

PROCESS DESIGN OVERVIEW

2.1 Process Definition

A process is a sequence of organized activities that will results in a predicted outcome. A process model addresses how inputs are transformed into outputs by some organized activities or tasks. Process Design (PD) involves the following activities:

- Designing the steps of the process
- Making decisions about what steps to use
- Posing and selecting choices

A particular process can be recognized as a system or subsystem in itself. A system is defined as an orderly integration of different elements to achieve a defined purpose [4]. A process can be considered as a single or multiple system elements. Similar to a system, a process is:

- Simplistic or complex combination of resources that can be characterized as Tangible such as physical, human and monetary, or intangible such as data.

- Hierachical such as higher-level elements are influenced by lower-level elements.
- Divisible into smaller processes or subsystems, depending on its complexity, to allow for straightforward approach in requirements allocation, and functional and system analysis.
- Driven by a purpose and an objective function.

Similarly, PD is a subset or a whole of a system design and uses similar analytical tools for evaluation, optimization and documentation [4]. In this praxis, an optimization model is employed along with stochastic analysis. Statistical concepts are used in development the design-of-experiments to assess the statistical significance of design factors.

PD is a formal integration of executable tasks into a family of activities to produce a predicable outcome. The design portion pertains to designing and sequencing the process steps for optimal effectiveness and efficiency. Choices are posed and selected based on criterion to achieve the desired outcome. Planning, strategic and operational constraints and assumptions are introduced to achieve the required output. Stochastic system optimization methodology can be incorporated into the PD to account for environment instability, markets, and customers unpredictably ever changing requirements.

A process can be documented using diagrams or descriptions. There are many process diagramming techniques in use. IDEF modeling methods are among the widely used modeling methods. IDEF stands for Integrated Definition for Function Modeling [11].

The Air Force program for Integrated Computer-Aided Manufacturing (ICAM) developed the first ICAM definition or IDEF in 1981 [6]. It is a method for process description capture, which was developed under the Information Integration for Concurrent Engineering (IICE) project. There are several IDEF methods that are in use. The following is a list of the IDEF family of methods [14]:

1.	IDEF0	Function Modeling (purpose: description)
2.	IDEF1	Information Modeling (purpose: description)
3.	IDEF1X	Data Modeling (purpose: design)
4.	IDEF2	Simulation Model Design (purpose: design)
5.	IDEF3	Process Description Capture (purpose: description)
6.	IDEF4	Object-Oriented Design (purpose: design)
7.	IDEF5	Ontology Description Capture (purpose: description)

Other IDEF methods that have not been widely used:

8.	IDEF6	Design Rationale Capture
9.	IDEF7	Information System Auditing
10.	IDEF8	User Interface Modeling
11.	IDEF9	Scenario-Driven IS Design
12.	IDEF10	Implementation Architecture Modeling
13.	IDEF11	Information Artifact Modeling
14.	IDEF12	Organization Modeling

15. IDEF13	Three Schema Mapping Design
16. IDEF14	Network Design

Figure 2.1 below shows the IDEF0 diagram for a simple selling process [6]. The process contains 3 basic functions: order processing, dispatch of goods and invoicing. Customer order starts the order processing. Sales department and inventory data are involved in the process to confirm the availability of the ordered quantity and certify the order for dispatching. Dispatch of goods process involves the warehouse department and control of dispatched goods to insure that the proper goods are sent to the customer via the requested method of delivery. Invoicing process involves the invoicing department to track the invoice and receipt of payment for the goods delivered. Note that the IDEF0 diagram does not detail the individual processes involved in tasks' executions.

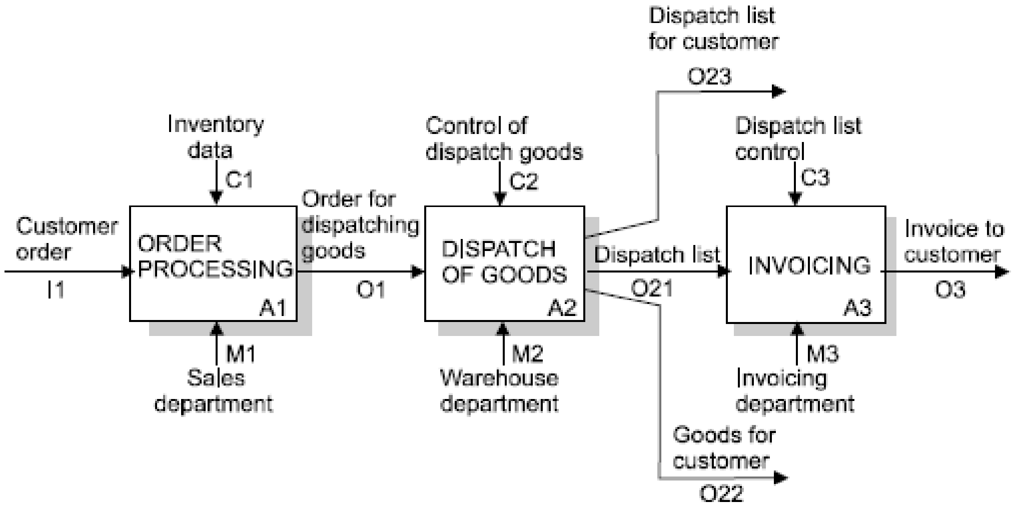

Figure 2.1 IDEF0 Diagram

Process Oriented Analysis (POA) [20] tool is also evaluated for use in this praxis. POA offers, among other features, both static (time-independent) and dynamic (time-dependent) modeling. Static models are useful in analyzing flows relevant to finance and environment. Dynamic modeling allows for time-dependent production analysis and simple programming and documentation.

Another process diagramming techniques in use is data flow diagrams and N^2 charts. Function Flow Block Diagrams (FFBDs) were originally used in conjunction with the N^2 charts to represent functional decomposition in systems engineering. FFBD provides a hierarchical decomposition of the system's functions along with a control structure specifying the order to which each function is executed. The N^2 charts method was created in the 1960s to describe data or items as inputs or outputs of a function in the functional architecture [11]. The higher level functions are displayed in a diagonal format with N rows and N columns. Similar to IDEF0 and data flow diagrams, the N^2 elements provide the same description of hierarchical decomposition of the system's functions. The N^2 charts method provides value in the information where there is no interaction between functions.

2.2 IDEF3 Process

IDEF3 is a scenario-driven process flow modeling technique. The objective of IDEF3 is to assist in documenting and analyzing the processes of a proposed or an existing system.

IDEF3 is selected to describe the process as it supports both object-centered and process-centered knowledge acquisition strategies to help describe real-world processes and events in a way that most consistent with human expressions that describes the execution of the process from multiple viewpoints [19]. ProSim software version 7.0 from Knowledge Based Systems at www.kbsi.com is used to illustrate the model. The software symbol descriptions are listed below. Refer to figure 2.2 for symbols used for IDEF3 process description [19].

Processes. Processes are also referred to as units of behavior (UOB). UOBs are used to describe consistent behavior situations and patterns, events, decisions and functions that take place in a system. Scenarios and decompositions consist of processes with common pattern of activities that are tied together with logical and temporal constraints.

Decompositions. Decomposition diagrams provide more detailed view of a process in a scenario and contain links, junctions and referents. Decomposition is indicated by a shadow that appears behind the process box.

Junctions. Junctions are useful indicators for specifying the process logic including timing and the sequence order of multiple process paths. The following fans and logic types are used:

- Fans:

  - *Fan-out* junctions are used to show divergence of a process flow into multiple parallel or alternative processes.
  - *Fan-in* junctions are used to show convergence of a process flow of multiple or alternative processes into a single flow.
- Logic Types:
  - *Asynchronous AND* junction is used to indicate all processes linked to the junction will occur before the flow continues. Simultaneous occurrence is not required.
  - *Synchronous AND* junction is used to indicate all processes linked to the junction will occur before the flow continues. Simultaneous occurrence is required.
  - *Asynchronous OR* junction is used to indicate one or more of the processes linked to the junction will occur before the flow continues. Simultaneous occurrence is not required.
  - *Synchronous OR* junction is used to indicate one or more of the processes linked to the junction will occur before the flow continues. Simultaneous occurrence is required.
  - *XOR* junction is used to indicate only one of the processes linked to the junction will occur before the flow continues.

Links. Links are used to indicate relationships between processes. Links have three types: precedence links, relational links and object flow links.

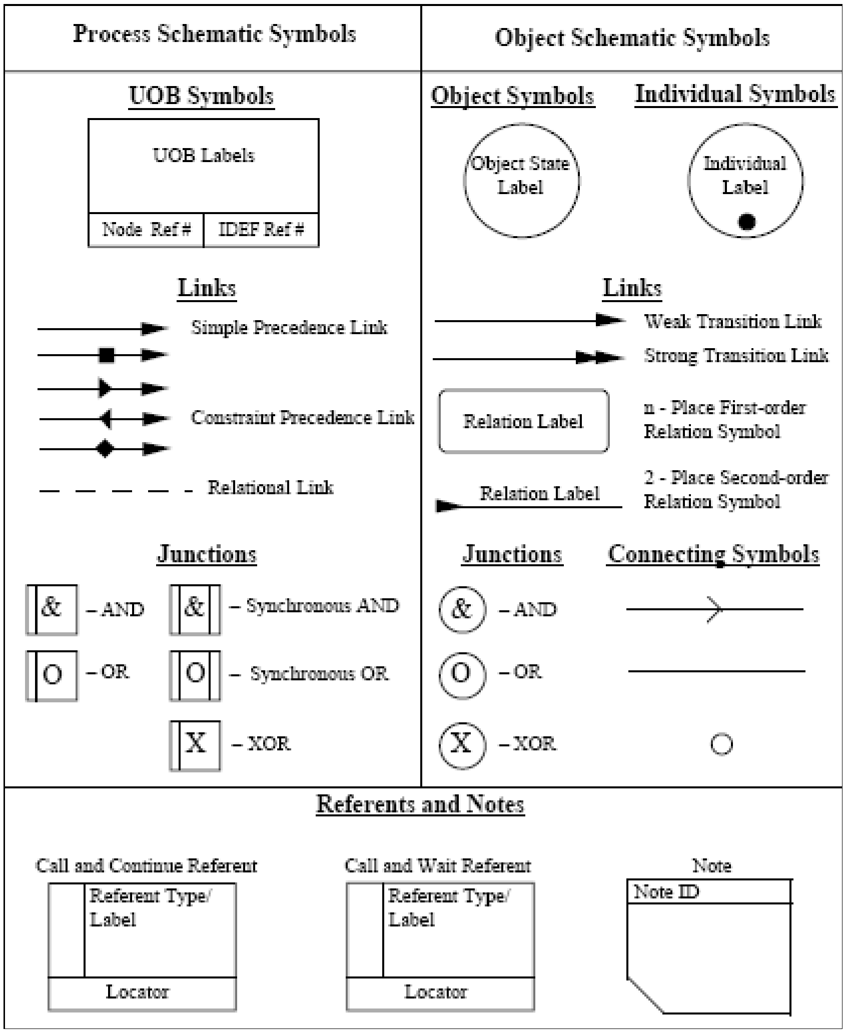

Figure 2.2 Symbols Used for IDEF3 Process Description Schematics

Referent. Referents are used to provide more details about processes, junctions and links. There are four types of referent:

- *Informational* referents are used to provide more information about elements attached to them.
- *Goto* referents are used to redirect the process flow to the next process step.

- *Call-and-wait* referents are used to temporarily suspend the process flow until the element has been called complete.
- *Call-and-continue* referents are used to indicate now response is needed for the process flow to continue.

2.2.1 Process Description

Two of the main strengths of IDEF3 are the simplicity of its notation demonstrated by the UOB [6] and the hierarchal function in the precedence capture and the causality relationship between tasks. The use of process decomposition is an effective way to demonstrate real-life activities as they are used in the model. "The goal of an IDEF3 model is to provide a structured method for expressing the domain experts' knowledge about how a particular system or organization works." [6]. Figure 2.3 below details the order processing operation shown in figure 2.1. It shows the process decomposition of the Order Processing operation.

Among all the IDEF family of methods, the POA tool, N^2 charts and FFBDs, IDEF3 was selected for use in this praxis as it provides the best activity-level description that can be used for input/output characterizations such as quality, performance, cost and schedule, in additional to the simplicity of its notation and the hierarchal function in the precedence capture and the causality relationship between tasks.

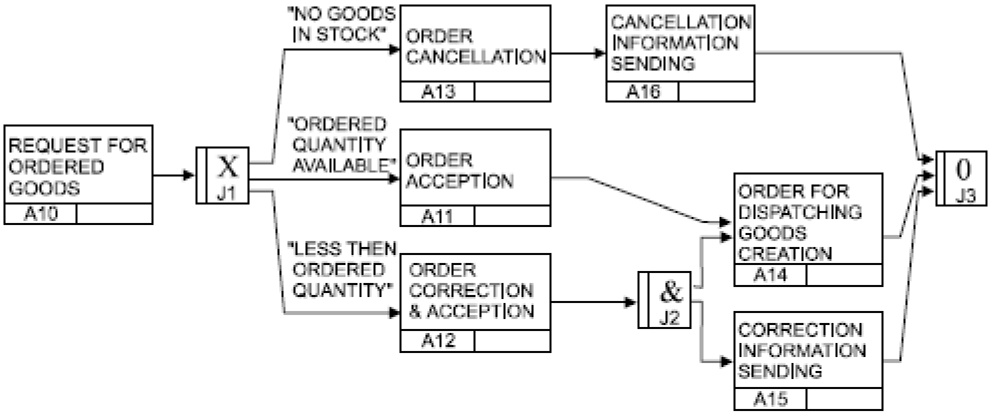
"NO GOODS IN STOCK"
ORDER CANCELLATION
A13
CANCELLATION INFORMATION SENDING
A16
REQUEST FOR ORDERED GOODS
A10
X
J1
"ORDERED QUANTITY AVAILABLE"
ORDER ACCEPTION
A11
ORDER FOR DISPATCHING GOODS CREATION
A14
0
J3
"LESS THEN ORDERED QUANTITY"
ORDER CORRECTION & ACCEPTION
A12
&
J2
CORRECTION INFORMATION SENDING
A15

Figure 2.3 IDEF3 Diagram

CHAPTER 3

MATHEMATICAL OPTIMIZATION MODELS FOR PROCESS DESIGN

Given below is an integrated production planning and process design model for discrete-flow repetitive manufacturing systems such is found in process industries. Straight-line flow is assumed and inventories are possible for all WIP types and finished goods. This is a new method to support process design by allowing not only the traditional optimization approach, but adding a novel way to help with process design based on quality, capacity and demand requirement. This model goes beyond traditional methods by integrating the process design piece with the optimization model for a more comprehensive tool to approach problem solving.

The models presented in chapter 3 and chapter 5 can be considered an extension of the MRP II (material requirements planning) model [30]. The model includes a cost objective along with multiple routings, late delivery penalty and capacity optimization. In addition, there are product quality requirements and process alternatives for enhancing or verifying quality levels at each stage. Target shipping levels are given, in units, corresponding to customer demands or master production schedule plans. Both the production plan and process design are optimized according to the organization's objective function, shown in several forms.

This section presents the deterministic mathematical formulation for the straight line generic process model shown in figure 5.

3.1 Notations and Conventions

The following notations, symbols and conventions are used:

- Sets or sequences are denoted as italicized upper-case letters, such as *S*. The sets' cardinality is denoted as |*S*|.
- Members of a set are denoted as italicized lower-case letter such as *b*.
- Variables are denoted as non-italicized upper-case letter such as D.
- Scalars are denoted as non-italicized lower-case letters such as c.
- Sets of real numbers and integers are denoted as *R* and *Z*, respectively.

The following symbols are used in the mathematical formulation using the above notation:

Sets:

T Set of time periods: $T = \{1,2,3,....,\mathrm{n}_1\}$, where n_1 is the number of periods

G Set of material quality grades: G = $\{0,1,2,..,\mathrm{n}_2\}$, where n_2 is the number of material grades and 0 = ungraded material

M Set of operations: M = {1,2,.., n_3}, where n_3 is the number of operations

Q Set of quality enhancements: Q = {0,1,2,.., n_4}, where n_4 is the number of quality enhancement and 0 = undetermined quality enhancement

Variables:

X_{mt}^{gq} units of material of grade $g \in G$ using quality enhancement criterion $q \in Q$ entering operation $m \in M$ in period $t \in T$ (see discussion below)

V_{mt}^{gh} units in inventory of material created by operation $m \in M$ of grade $g \in G$ using quality enhancement criterion $q \in Q$ at the start of period $t \in T$

O_t units of shipped products above customer demand requirements in period $t \in T$

U_t units of shipped products below customer demand requirements in period $t \in T$

Γ^{gq} 0/1 = don't/do use quality enhancement criterion $q \in Q$ for grade $g \in G$

Scalars:

y_m^{gq} yield (including average conversion of units) of operation $m \in M$ for material of quality enhancement $q \in Q$ of grade $g \in G$

a^q total variable manufactuing cost per unit of finished product using quality enhancement criterion $q \in Q$

d_t customer demand requirements in period $t \in T$, in units

α material-starts capacity limit, $\alpha \in Z$

ψ unit selling price, $\psi \in Z$

f maximum acceptable DPPM

$\hbar^{gq}$ Defective Part Per Million (DPPM) expected after 100,000 hours of operation, $\hbar \in Z$, $g \in G$, $q \in Q$

φ upper limit on all X_{mt}^{gh} values

α processing capacity per time period $t \in T$, $\alpha \in Z$

3.2 Mathematical Formulation

A straight line production flow optimization model is developed using the above notation. This multi-period model provides alternative methods of enhancing/ensuring quality of product at each production stage based on the production level and finished goods targets. Each element of the model is expressed mathematically followed by a detailed discussion. The objective function is:

$$\text{Optimize} \qquad Z = f\,(X, O, U, V) \tag{3.1}$$

Objective function. In this formulation, the objective function in (3.1) is to optimize the defining process variables subject to production planning and process design constraints listed below. The key variables, X and V, represent different materials – work-in-process (WIP) and finished goods – at each stage in the process of converting material into shippable finished product. Variables O and U represent units of over-shipped and under-shipped products, respectively, with respect to customer demand requirements. The objective function can be formulated to meet the business' objective. It can be formed to minimize cost and maximize profit to maximize output while minimizing cost, or to maximize output while reducing work-in-process inventory.

Constraints (3.2) to (3.8) represent a generalized network model for the material flow, processing and inventory.

$$\mathrm{y}_{m}^{gq}\mathrm{X}_{m,t-c_m}^{gq} + \mathrm{V}_{m,t-1}^{gq} = \mathrm{X}_{mt}^{gq} + \mathrm{V}_{mt}^{gq} \qquad \forall g \in G, t \in T, q \in Q, m \in M \tag{3.2}$$

Material Flow. Constraint (3.2) is a flow balance equation at a given point in time. It defines the flow of material in each processing step plus inventory as equal to the flow in the previous step multiplied by the process yield plus its inventory. The flow in the previous step is in relation to the cycle time of the specific operation. The input of a specific operation at time *t* may have started one period or several periods prior to time *t*, depending on the associated cycle time. The constraint takes into account the yield and cycle time of each operation. In addition, the constraints cover all material quality grades and sets of quality enhancements.

$$\sum_{q \in Q-\{0\}} \mathrm{X}_{n_3 t}^{gq} = \mathrm{y}_{n_3}^{g}\mathrm{X}_{n_3 t}^{00} \qquad \forall g \in G-\{0\}, t \in T \tag{3.3}$$

Yield Multiplier. Constraint (3.3) defines the process yield multiplier. It represents the loss in the process and the conversion of material to units except for g = 0, which represents scrap material, and q = 0, which represents material with undetermined quality status.

$$\mathrm{X}_{mt}^{gq} \leq \varphi\Gamma^{gq} \qquad \forall g \in G,\; q \in Q,\; t \in T \tag{3.4}$$

Quality Enhancement selection. Constraint (3.4) defines an upper limit φ on all units of material X_{mt}^{gq} and employs an integer variable Γ^{gq} to determine whether or not a particular quality enhancement step is to be used. The upper limit may be real or artificial for the purpose of determining the appropriate quality enhancement step.

$$\sum_{Q \in q} \Gamma^{gq} = 1 \qquad \forall g \in G \tag{3.5}$$

Grade QE step selection. Constraint (3.5) requires one quality enhancement step per grade. This constraint may be changed based on the needs and flexibility of the users' environment.

$$\hbar^{gq}\Gamma^{gq} \leq \mathrm{f} \qquad \forall g \in G,\; q \in Q \tag{3.6}$$

Select QE criteria for each grade. Constraint (3.6) uses the level of quality acceptable by the customer. It ensures that the expected DPPM $\hbar^{gq}$ of the selected quality enhancement criterion is within the maximum DPPM limit f acceptable by the customer for a given grade.

$$\mathrm{X}_{1t}^{00} \leq \alpha \qquad \forall t \in T \tag{3.7}$$

Material-starts capacity limit. Material-starts capacity limitation is defined in constraint (3.7) for operation m=1 of ungraded material with undetermined quality enhancement in a given time period.

$$\sum_{q \in Q} \sum_{g \in G} \mathrm{X}_{n_3 t}^{gq} = \mathrm{d}_t + \mathrm{O}_t - \mathrm{O}_{t-1} - \mathrm{U}_t + \mathrm{U}_{t-1} \quad \forall t \in T \tag{3.8}$$

Units shipments. Constraint (3.8) defines the finished goods shipped to be equal to the customer demand plus the cumulative over-ship (surplus) and minus the cumulative under-ship (backorder). This formulation allows the model to add appropriate weights to over-shipping and/or under-shipping to encourage behavior that meets the users' end objectives.

To summarize, below is a consolidated list of the mathematical formulation equations (3.1) to (3.8):

Optimize $\quad Z = f(X, O, U, V)$

subject to:

$$\mathrm{y}_{m}^{gq}\mathrm{X}_{m,t-c_m}^{gq}+\mathrm{V}_{m,t-1}^{gq}=\mathrm{X}_{mt}^{gq}+\mathrm{V}_{mt}^{gq} \qquad \forall g\in G, t\in T, q\in Q, m\in M$$

$$\sum_{q\in Q-\{0\}}\mathrm{X}_{n_3t}^{gq}=\mathrm{y}_{n_3}^{g}\mathrm{X}_{n_3t}^{00} \qquad \forall g\in G-\{0\}, t\in T$$

$$\mathrm{X}_{mt}^{gq}\le\varphi\Gamma^{gq} \qquad \forall g\in G,\ q\in Q,\ t\in T$$

$$\sum_{Q\in q}\Gamma^{gq}=1 \qquad \forall g\in G$$

$$\hbar^{gq}\Gamma^{gq}\le \mathrm{f} \qquad \forall g\in G,\ q\in Q$$

$$\mathrm{X}_{1t}^{00}\le\alpha \qquad \forall t\in T$$

$$\sum_{q\in Q}\sum_{g\in G}\mathrm{X}_{n_3t}^{gq}=\mathrm{d}_t+\mathrm{O}_t-\mathrm{O}_{t-1}-\mathrm{U}_t+\mathrm{U}_{t-1} \qquad \forall t\in T$$

3.3 IDEF3 Representation

IDEF3 presentation of the model is captured in the figures 3.1, 3.2 and 3.3. The diagrams represent a straight-line production flow. Figure 3.1 is the top level representation of the model and shows the main processes and decision points. Figure 3.2 is one-level decomposition of the shaded "Sub-Standard Process 1." It shows the different grades split and their flow. Figure 3.3 shows the decompositions of "Grade N" and the quality enhancements decision flow. As shown in the diagram, the precedence capture evident in the hierarchal function and the causality relationship between tasks demonstrated in the use of process decomposition makes IDEF3 modeling method an effective way to display real-life activities as they are used in the model.

3.4 Conclusions

As shown in the mathematical model and the IDEF3 model representation, a system-level model that captures both process design decisions and schedule considerations would be particularly valuable. The model allows the flexibility to customize the model per the users' dynamic environment. It integrates process design and planning with system optimization to produce results aligned with users' constraints and expectations. These components are usually found in separate models or activities. The next chapter will outline a case study with expanded use of this model along with mathematical formulations in chapter 5.

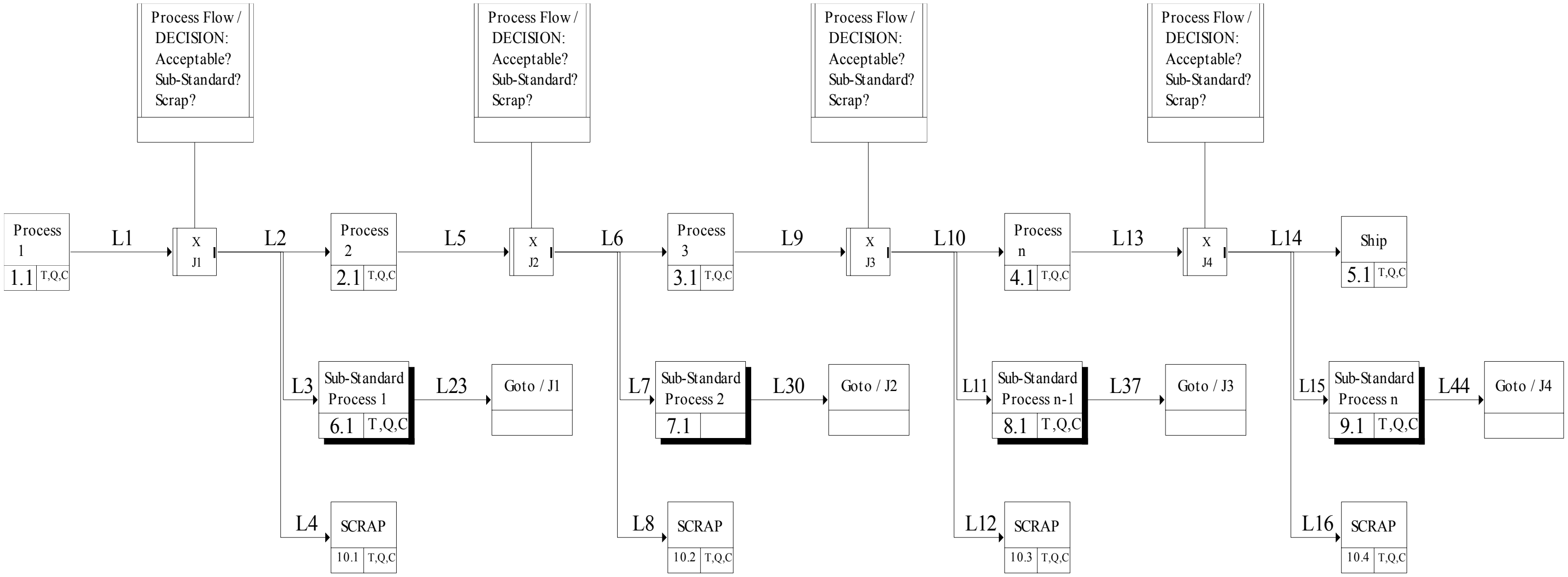

Figure 3.1 ProSim Model – Level 1

Lxx

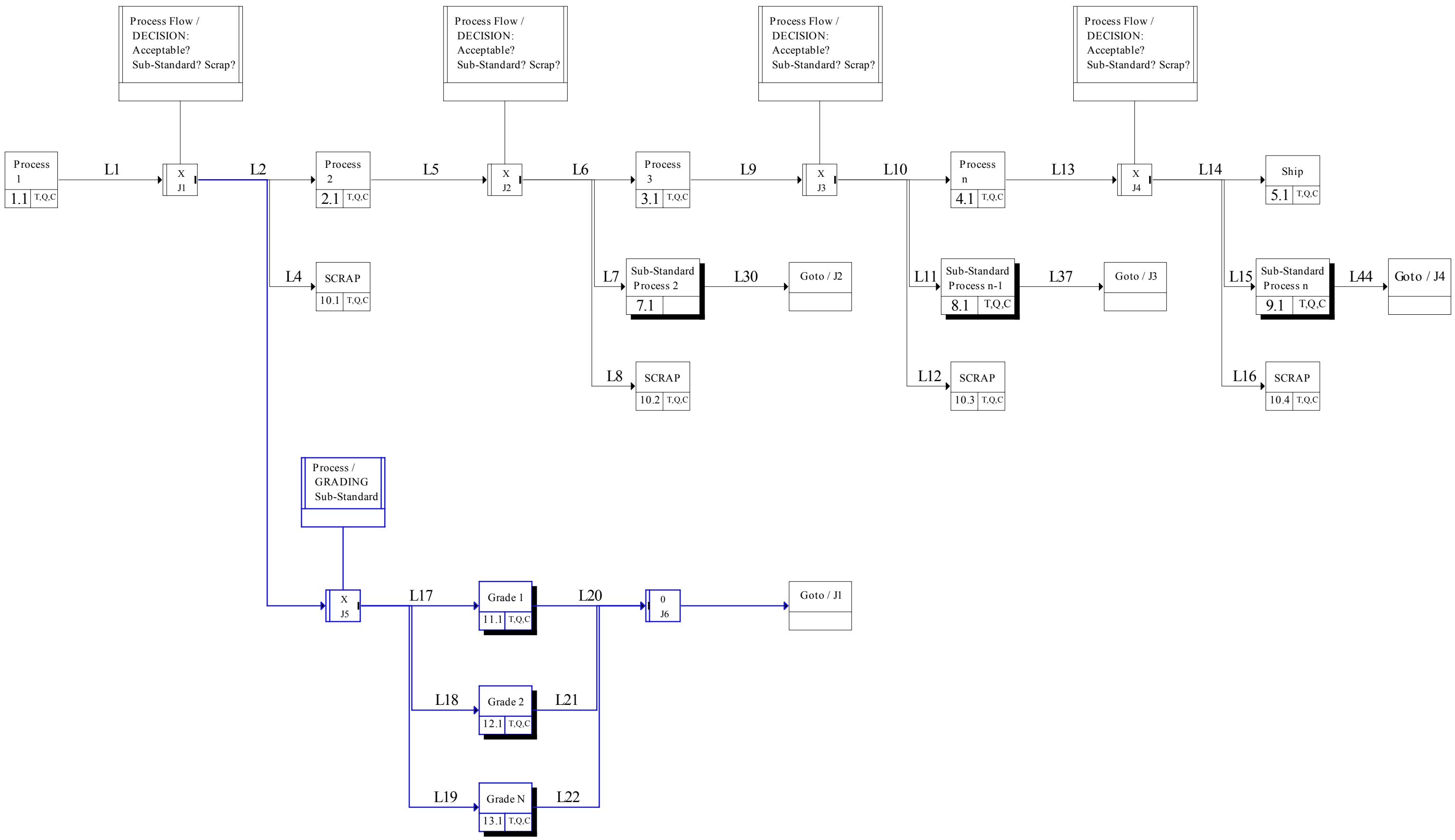

Figure 3.2 ProSim Model – Level 2

Lxx

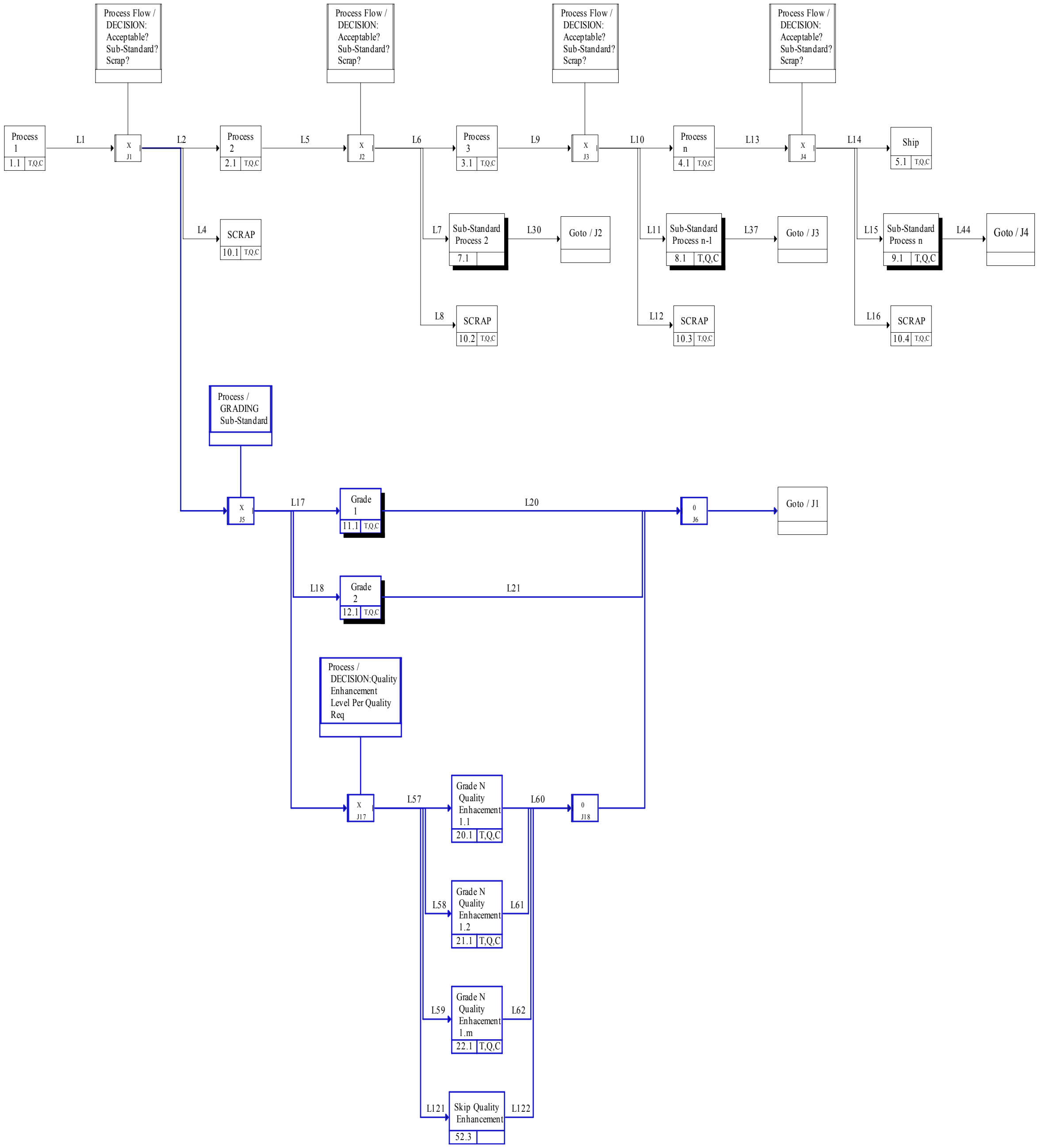

Figure 3.3 ProSim Model – Level 3

CHAPTER 4

CASE STUDY: QUALITY GRADING IN SEMICONDUCTOR INDUSTRY

Christmas season is the most lucrative shopping season of the year. Companies, Manufacturers, builders, and suppliers of most products and services place significant weight on the Christmas season in their annual plans. Missing high revenue opportunities generated in the Christmas season may have a devastating impact on the company's financial future. Company Y has to deal with the possibility of going through the Christmas season with less than optimal product portfolio on the shelves due to production quality issues. Company X, the supplier of company Y, is encountering a quality issue in the production line that is limiting the supply. Both companies have one top priority in mind: products must be ready for the best shopping season of the year while maintaining a high level of quality consistent with both companies' long standing reputation and commitment to quality.

This chapter presents a case study based on real events in the semiconductor industry. The events are driven by the urgent need for company Y to have its products ready for shoppers on time for the Christmas shopping season, and for company X to truly illustrate its commitment to customer satisfactions in the most creative ways by enabling one of its most strategic customers to meet its financial goals. The case study is using an integrated process design, planning and system optimization to address dynamic manufacturing, operations, planning, and quality requirements based on customer needs while focusing on reducing cost and increasing profit. The case study illustrates the value of this new approach where process decisions take into account various WIPs quality grades and uncertain yield rates for a new product.

4.1 Semiconductor Manufacturing Overview

Figure 4.1 [28] shows a high level view of the semiconductor chip-making process. The process starts with circuit design layout transformed into photo masks that are used in the fabrication process. Photo masks control the projection of lights on bare silicon wafers to allow building circuit components with extremely small dimensions. A wafer fabrication begins with transforming sands, containing high level of the silicon element, into long ingot of solid silicon that are then sliced into disks of semiconductor material called wafers. The wafers are then used a base to build identical integrated circuits. After several stages in the fabrication process, arrays of dies of square or rectangular shapes containing identical integrated circuits are created on the wafer. Two or more integrated circuit designs can be built on the same wafer. The integrated circuitry is made of thousands of electronic components of very small dimensions. A die is a collection of semiconductor components or parts that have not yet been packaged. Once packaged, the resulting product is called an “IC” for Integrated Circuit chip. The finished wafer goes to through a wafer multi-probe testing which is a set of hardware that connects the die pads on each die on the wafer to a tester to be tested using a specific test program to confirm compliance with expected functionality and desired quality requirements. Wafer probing is often used as part of wafer acceptance. Wafer probe testing is also often called wafer multi-probe testing. The packaging process begins after the wafer probe testing. The wafer is then sliced per the geometry of the individual die pattern to allow the dies to be picked up and placed into an IC package, wire bonded to the pins and encapsulated into a device form. A semiconductor device is essentially a die that has been assembled and put into a package IC form. Device is also an active or passive device used in an electronic system..

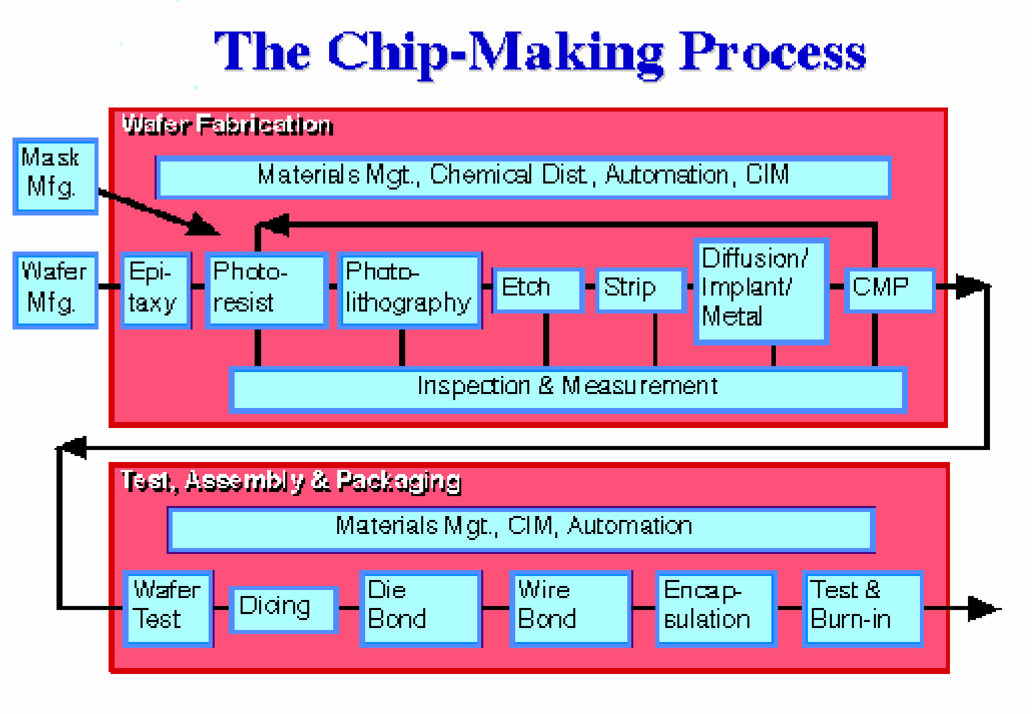

Figure 4.1 The Chip-Making Process

CIM is Computer Integrated Manufacturing
CMP is chemical mechanical polishing

Current semiconductor technology allows for more than 800 transistors to be packed onto an area as small as the diameter of a single human hair. With such fine dimensions, a tiny particle such as dust particle can damage the chip. As a result, wafers are fabricated in a clean room environment, and inspections and measurements are performed many times during the wafer fabrication process.

The case study takes place in an electronic semiconductor manufacturing environment where company X identifies a process abnormality in the wafer fabrication manufacturing line causing defects that compromised the Gate Oxide Integrity (GOI). Gate oxide is a thin layer of pure dielectric between n- and p-type materials in a Complementary Metal-Oxide Semiconductor (CMOS) transistor technology. An oxide serves as an isolation dielectric layer between conductive layers. The manufacturing of the oxide layer is a very sensitive, highly monitored and controlled process as any defect may result in rupturing the thin layer, permanently damaging the transistor and rendering the device non-functional. Therefore, GOI testing is a common practice in the semiconductor industry. In-line, end-of-line, and wafer probe testing are commonly performed to ensure the integrity of the gate oxide. The in-line tests are usually performed on the wafers during the fabrication process for defect detection. Results highlight any manufacturing issues early to help with faster recovery and to avoid further processing of defective material. End-of-line test is a test microelectronics vendor uses immediately after manufacturing the parts (wafers), to detect and reject parts (wafers) with defects.

4.2 Quality Grade

A quality grade is a qualitative category within a classification hierarchy that reflects the condition of the subject element. It is usually reflection of product characteristics of concern to an end user.

Quality grading has been used in some fashion since the early days of humanity. It has been used by fabric merchants to advertise their material. Food vendors use quality grades to differentiate their goods. Today, we see quality grading in most aspects of our lives. From buying tires to booking a vacation, quality grades differentiate and promote goods and services to the respective customers.

Quality grading can sometimes be a subjective art in the eye of the beholder as the case in grading art works such as movies reviews. It can also be a precise science as the case in grading gasoline products based on the octane rating. In other cases, quality grading is a combination of subjectivity and precise science as the case in grading hardwood plywood where the appearance of the product defines the grade as well as the number of natural characteristics and repairs permitted. The grading system of plywood is detailed in a

Voluntary Standard published by the Hardwood Plywood and Veneer Association (HPVA).

Rice is graded primarily on the kernel form and grain shape, and there are several sub-categories under each grade. The United States Department of Agriculture (USDA) classifies potatoes based on size, shape, injury and freshness (freezing, blackheart, loose sprouts, dirt or foreign material, etc.). Tuna is graded based on the intended market and the following characteristics: meat color, oil content, presence of burnt meat, body shape, size, freshness, external appearance, and presence of disease or parasites.

There are many publications on quality grading as it pertains to a specific product. However, little has been published on how to set quality grades in general terms and how quality grades impact process design planning and system optimization. The methodology of how to set quality grades is absent from quality literature.

Quality grading is based on defects identification and classification. Variable data or attributes data can be used for defect identification and classification. Variable data are continuous quantitative measures such as voltage, current, temperature, weight, etc. For example, a customer may ask that products be graded using a statistical distribution. Thus, grade A for a voltage offset is one standard deviation from the mean in both directions and all remaining material within the specification limits would be classified as grade B.

Attributes data are based on pass/fail criteria or number of defect, etc. For example, in the semiconductor industry, an outlier control plan can specify that a silicon wafer is scrapped if 5 or more dies fail the Gate Oxide Integrity (GOI) test. If the number of failures is between zero and two, the wafer would be classified as grade A material or, for three to five failures, classified as grade B.

The pass/fail criteria tend to be used subjectively in most quality grading applications and are primarily based on the consequences. A customer that is in dire need for products maybe more willing to accept products that are marginally failing the pass/fail criteria or even relax the limit to allow more products to pass.

4.3 How It Started

Company Y has been successful in designing and manufacturing state of the art high volume products for the electronic market place. In order to differential itself from the competition, and to allow for the addition of creative features and improvements, company Y has contracted company X to design and manufacture IC chips with specific design criteria for the intended electronic system. Since the design is highly customized and supplied exclusively to company Y, there is an inherent interdependency between company X and company Y. Company X is investing significant resources for the research and development of a custom IC chip for the exclusive use of company Y, while company Y has significant market share in the intended application and has committed its future to the successful design and manufacturing of this new IC chip that will enable it to grow financially and gain market share. The success of company Y will translate into direct unit sales increase of the custom IC chip manufactured by company X. It is an ideal marriage recipe where the success of one party is directly dependent on the success of the other party. As a result, the spirit of cooperation, collaboration and team work is clearly manifested in this relationship.

Nearly two years and hundreds of thousands of dollars later, the custom IC chip has passed all quality requirements, has been confirmed to meet the required specifications, and is now ready for volume production. The production ramp plan shows a very steep increase in the order quantities. A near perfect execution in manufacturing and delivery of these custom IC chips is paramount to the success of both companies. All engines are now running full steam ahead per the production plan with no significant issues to report.

This indeed pleasant journey has to come to a grinding halt as company X identifies a major quality excursion in the manufacturing line that would impact the quality of the custom IC chips. The defect is detected by the in-line tests. High levels of defects are detected in the gate oxide manufacturing process resulting in compromising the gate oxide integrity (GOI). GOI defects will cause the custom IC chips to fail early in the lifetime of the intended application.

4.4 Notification, Containment and Recovery

Soon after identifying the quality incident in the manufacturing line, company X realizes the boarder ramification of the issue. Initial assessment cannot conclusively establish the root-cause of the defects, and therefore, corrective actions are still to be determined. In keeping the spirit of partnership, company X chooses to inform company Y of this quality excursion and its impending consequences. Company X is well aware of the criticality of the issue and is not surprised by the serious reaction of company Y as a result. The issue has moved from a quality production issue to an issue that is threatening the survival of company Y.

A crisis team quickly forms of high caliber experts from both companies. The team's main objectives are:

1. Assess the impact of the quality issues on the number of good units that can be used for the intended application.
2. Identify and implement containment actions to limit the exposure to the GOI issue.
3. Salvage any and all IC chips from the defect affected material that can be used for the production of the intended electronic system while maintaining a minimum quality level until the root-cause is identified and corrective actions are implemented.

While failure to keep the company Y's production line running has severe financial consequences, compromising the quality and the reliability of the final product is not an option. Such a challenging assignment gives way to very creative methodologies to be implemented in the pursuit of achieving the desired results.

To assess the impact of the quality issue and start containment actions, the team needs to address the following questions:

1. How much of the wafer lots, that are currently running in the production line, are affected by the gate oxide defect?
2. How much of the new wafer lots will be impacted by this quality excursion?
3. What is the severity of the defects?
4. How will it affect the quality and reliability of the intended application?
5. Has the root-cause been identified? If not, when is it expected to be identified and corrective actions implemented?
6. When will the affected wafer lots complete the fabrication process and be ready for end-of-line tests?
7. What is the metric to implement for screening the wafers based on end-of-line and multi-probe test results?

To aid the containment activities, additional GOI tests are quickly implemented at the multi-probe test operation that is performed on finished wafers. Results from both end-of-line tests and multi-probe tests on finished wafers provide an assessment of the impact of the defects and how many wafers can be assembled and used in the intended application. To further assess the situation, Company X implements the following containment actions:

1. The wafer lots in the manufacturing line are assessed on whether they are expected to produce products that meet the customers' quality requirements. Criteria are assigned using the in-line test results mainly based on whether the material is processed through the equipment suspected of producing the most defects. The categories are: good, better, bad, and ugly, reflecting the likelihood of producing top-quality chips.
2. The criteria are further refined based on the results of end-of-line wafer acceptance test, which is a set of tests performed on integrated circuit wafers designed to distinguish between wafers that will yield a high percentage of good dies and those that will not.
3. The wafer lots are then split into sub lots and are assigned a grade based on the results of the wafer multi-probe testing and the projected DPPM level listed in table 1 per the burn-in (BI) hours completed per table 4.1. The quality grades are:

    a. Grade P – Prime
    b. Grade W
    c. Grade X
    d. Grade Y
    e. Grade R

		Hours of Operations			
Grade	**BI Hrs**	**24**	**1000**	**10000**	**100000**
P	0	450	490	660	705
P	5	3	3	4	5
P	24	1	1	1	1
W	0	1,150	1,525	1,700	1,830
W	5	8	10	11	12
W	24	2	2	2	3
X	0	3,000	63,000	120,000	166,500
X	5	20	420	800	1,110
X	24	4	90	171	238
Y	0	5,000	101,250	188,750	262,500
Y	5	40	810	1,510	2,100
Y	24	7	145	270	375
R	0	14,000	525,000	1,484,000	2,408,000
R	5	93	3,500	9,893	16,053
R	24	20	750	2,120	3,440

Table 4.1 Grades, BI Hrs and Hours of Operations Matrix

4. The graded lots are processed through assembly and test following the flow outlined in figure 4.2.

5. The overall expected test yield per operation is listed in table 4.2.

	Test Yield in Percentage				
Operation\Grade	**P**	**W**	**X**	**Y**	**R**
MPYLD	0.81	0.78	0.63	0.63	0.61
FT1YLD	0.99	0.99	0.99	0.99	0.98
BIYLD.0hrs	1.00	1.00	1.00	1.00	1.00
BIYLD.5hrs	0.99	0.99	0.99	0.99	0.99
BIYLD.24hrs	0.98	0.98	0.98	0.98	0.98
FT2YLD.0hrs	1.00	1.00	1.00	1.00	1.00
FT2YLD.5hrs	0.99	0.99	0.99	0.99	0.99
FT2YLD.24hrs	0.998	0.998	0.998	0.998	0.998

Table 4.2 Test Yield per Test Operation

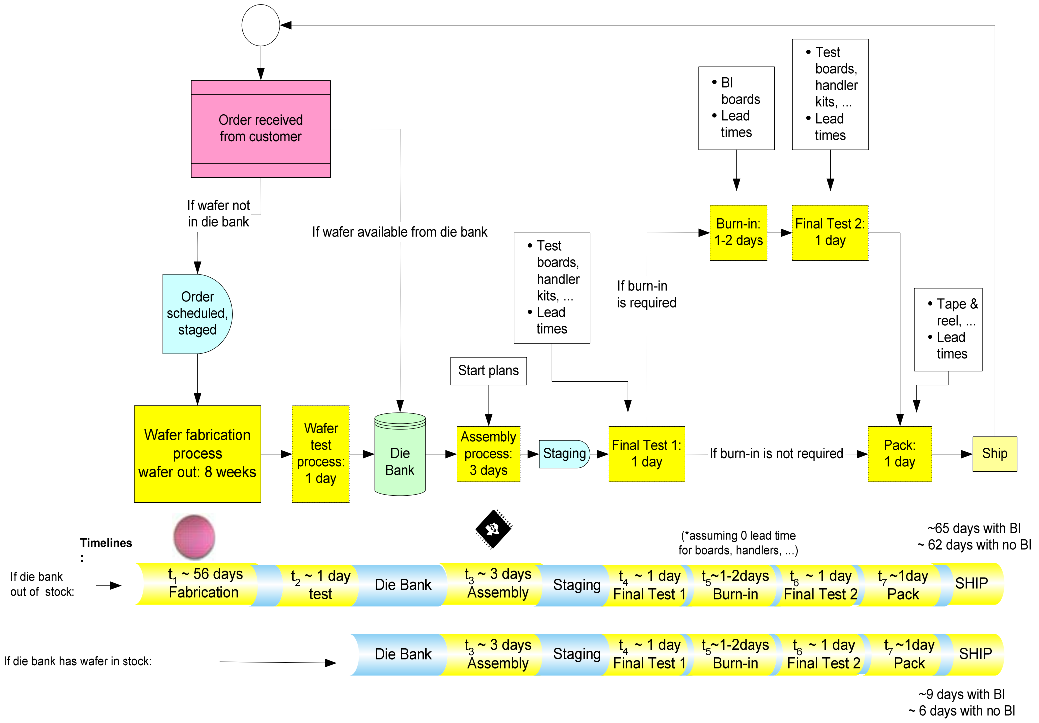
Order received from customer
If wafer not in die bank
If wafer available from die bank
Order scheduled, staged
Wafer fabrication process wafer out: 8 weeks
Wafer test process: 1 day
Die Bank
Start plans
Assembly process: 3 days
Staging
• Test boards, handler kits, ... • Lead times
Final Test 1: 1 day
If burn-in is required
If burn-in is not required
• BI boards • Lead times
Burn-in: 1-2 days
• Test boards, handler kits, ... • Lead times
Final Test 2: 1 day
• Tape & reel, ... • Lead times
Pack: 1 day
Ship
Timelines :
If die bank out of stock:
t_1 ~ 56 days Fabrication
t_2 ~ 1 day test
Die Bank
t_3 ~ 3 days Assembly
Staging
t_4 ~ 1 day Final Test 1
t_5~1-2days Burn-in
t_6 ~ 1 day Final Test 2
t_7~1day Pack
SHIP
(*assuming 0 lead time for boards, handlers, ...)
~65 days with BI
~ 62 days with no BI
If die bank has wafer in stock:
Die Bank
t_3 ~ 3 days Assembly
Staging
t_4 ~ 1 day Final Test 1
t_5~1-2days Burn-in
t_6 ~ 1 day Final Test 2
t_7~1day Pack
SHIP
~9 days with BI
~ 6 days with no BI

Figure 4.2 Assembly and Test Process Flow

Figure 4.2 shows the production flow starting with order received to shipping finished goods. The flow goes through the process of transforming silicon wafers into integrated circuits encapsulated in the selected package to ship to the customer. The flow starts with order received. The order is checked against die quantity available in die bank. If the order quantity exceeds that dies available in die bank, the order is executed and staged to start manufacturing in the wafer fabrication facility. The wafers are then tested for confirm conformance with functional and quality requirements before being transferred to die bank which is a staging area for the assembly manufacturing process. The assembly process begins based on the required demand. The wafers are sawn and placed in the designated packages. Dies are bonded to the leadframe and encapsulated with a mold compound. The encapsulated packages are then marked with lot date code for traceability and moved to the final test are. Final test flow includes two test insertions (operations) with the burn-in reliability stress in between. The packaged units that pass all testing with no failures are then pack and shipped to the customer.

Next chapter presents the mathematical formulation of this case study. The objective is to keep company's Y production line from going down while maintaining a certain level of quality consistent with the company's long history of quality products.

CHAPTER 5

CASE STUDY MATHEMATICAL FORMULATION

5.1 Deterministic Formulation

This section presents the deterministic mathematical formulation for the case study presented in chapter 4 using CPLEX linear programming. This multi-period model is based on the formulation presented in section 3.2. The stochastic mathematical formulation is detailed in section 5.2.

Using the generic model presented in chapter 3 as a base, the formulation presented below integrates optimization methods to meet customers' demand in the presence of manufacturing challenges in terms of cost, quality and capacity of multiple processes with intra-dependency as the allocated capacity at a particular process influences the capacity allocation of another process.

5.1.1 Notation and Conventions

The following notations, symbols and conventions are used:

- Sets or sequences are denoted as italicized upper-case letters, such as *S*. The sets' cardinality is denoted as $|S|$.
- Members of a set are denoted as italicized lower-case letter such as *b*.

- Variables are denoted as non-italicized upper-case letter such as D.
- Scalars are denoted as non-italicized lower-case letters such as c.
- Sets of real numbers and integers are denoted as *R* and *Z*, respectively.

The following symbols are used in the mathematical formulation using the above notation:

Sets:

T Set of time periods: $T = \{1,2,3,....,\mathrm{n}\}$, where n is the number of periods

G Set of material quality grades: $G = \{0,1,...,5\}$, where 0 = ungraded, 1 = grade P, 2 = grade W, 3 = grade X, 4 = grade Y, and 5 = grade R.

M Set of operations: $M = \{1,2,...,9\}$, where 1 = wafer fabrication, 2 = wafer multi-probe testing, 3 = grading step, 4 = plastic units assembly, 5 = final-test 1, 6 = burn-in, 7 = final-test 2, 8 = pack, and 9 = finished goods ship.

H Set of burn-in hours: $H = \{0, 1, 2,3\}$, where 0 = undetermined number of burn-in hours, 1 = 0 burn-in hour, 2 = 5 burn-in hours, and 3 = 24 burn-in hours.

B Set of boards: $B = \{1, 2,3\}$, where 1 = multi-probe-test boards, 2 = final-test boards, and 3 = burn-in boards.

Variables:

X_{mt}^{gh} units of material of grade $g \in G$ using burn-in criterion $h \in H$ entering operation $m \in M$ in period $t \in T$ (see discussion below)

V_{mt}^{gh} units in inventory of material created by operation $m \in M$ of grade $g \in G$ using burn - in criterion $h \in H$ at the start of period $t \in T$

O_t units of shipped products above customer demand requirements in period $t \in T$

U_t units of shipped products below customer demand requirements in period $t \in T$

E_t^b boards ordered at the start of time period $t \in T, b \in B$

D_t^b boards received at the start of time period $t \in T, b \in B$

F_t^b boards available at the start of time period $t \in T, b \in B$

Γ^{gh} 0/1 = don't/do use burn-in criterion $h \in H$ for grade g $\in G$

Scalars:

y_m^{gh} yield (including average conversion of units) of operation $m \in M$ for material of burn - in hours $h \in H$ of grade $g \in G$

a^h total variable manufactuing cost per unit of finished product using burn - in criterion $h \in H$

d_t customer demand requirements in period $t \in T$, in units

c_m^h cycle time for operation $m \in M, \quad h \in H$ in days

k^b leadtime in days to receive ordered boards, $b \in B$

j^b number of boards used per tester/oven, $b \in B$

q_m multiplier = number of seconds in one day * efficiency factor / test time in seconds, m ={2,5,7}, b $\in R$ (note: $b_5 = b_7$)

e number of dice per wafer, e $\in Z$

s number of sockets per burn-in board, $s \in Z$

α wafer fabrication daily capacity in number of wafers, $\alpha \in Z$

θ assembly operation daily capacity in units, $\theta \in Z$

ψ unit selling price, $\psi \in Z$

η^{b} cost of one board, $\eta \in Z, b \in B$

f maximum acceptable DPPM

$\hbar^{gh}$ DPPM expected after 100,000 hours of operation, $\hbar \in Z, h \in H, g \in G$

φ upper limit on all X_{mt}^{gh} values

β^{b} maximum number of testers/ovens available for use, $\beta \in Z, b \in B$

5.1.2 Integrated Process Design, Planning, and System Optimization Model

An integrated process design, planning and system optimization model is developed using the above notation. This model is a generalized network model for the material flow, processing and inventory and is an extension of the model presented in chapter 3. The model is customized for the case study presented in chapter 4. Each element of the model is expressed mathematically followed by a detailed discussion. The goal is to mathematically formulate the case-study's variables and constraints to effectively implement and test the model using a mathematical programming and optimization high-level modeling system presented in chapter 6. The objective function is:

$$\text{Minimize } \mathrm{P}_1\left(\sum_{t\in T}^{n} \mathrm{U}_t\right) - \mathrm{P}_2\left(\sum_{h\in H}\sum_{g\in G}\sum_{t\in T}^{n} \mathrm{X}_{9t}^{gh}\left(\psi - \mathrm{a}^h\right) - \sum_{t\in T}^{n}\sum_{b\in B} \mathrm{E}_t^b \eta^b\right) \tag{5.1}$$

Objective function. Constraint (5.1) represents the objective function that is defined as a combination of two preemptive priorities with the first one, P_1, being the primary priority. The primary objective P_1 is to minimize the number of units of shipped products below customer demand requirements U_t at any particular time period. This makes on-time shipments the top priority. The second priority, P_2, is to maximize profit. Profit is calculated by multiplying the total number of units shipped X_{9t}^{gh} at operation *m*=9 (finished goods ship) by the unit price ψ and then subtracting the manufacturing variable cost a^h and ordered BI board and test hardware costs $\mathrm{E}_t^b \eta^b$. Priorities are selected to minimize disruption to the customer's production line as a result of the quality excursion as a top priority while maximizing profit as second priority. The model's flexibility allows changing priorities according to the business' needs.

$$\mathrm{y}_m^{gh}\mathrm{X}_{m,t-c_m}^{gh} + \mathrm{V}_{m,t-1}^{gh} = \mathrm{X}_{mt}^{gh} + \mathrm{V}_{mt}^{gh} \qquad \forall g \in G, t \in T, h \in H, m \in M - \{3\} \tag{5.2}$$

Material Flow. Similar to equation (3.2), constraint (5.2) represents a flow balance equation at a given point in time. It defines the flow of material in each processing step plus inventory as equal to the flow in the previous step multiplied by the process yield plus its inventory. The flow in the previous step is in relation to the cycle time of the specific operation. The key variables, X and V, represent different materials – wafers, WIPs and finished goods – at each stage in the process of converting wafers into shippable finished product.

$$\sum_{h\in H-\{0\}} \mathrm{X}_{3t}^{gh} = \mathrm{y}_3^g \mathrm{X}_{3t}^{00} \qquad \forall g \in G - \{0\},\, t \in T \tag{5.3}$$

Yield Multiplier. The process yield multiplier in constraint (5.3) represents the loss in the process during the conversion of material from wafers to finished goods, where applicable. Step *m*=3 is a grading process step that splits all ungraded material into its component grades based on a defined criteria. The yield is accounted for in the yield grading process. Yield calculations do not apply to ungraded material *g*={0} and material with undetermined number of BI hours *h*={0}.

$$\mathrm{X}_{3t}^{gh} \le \varphi \Gamma^{gh} \qquad \forall g \in G,\ h \in H,\ t \in T \tag{5.4}$$

BI criteria selection. Constraint (5.4) defines an upper limit φ on all material at grading process step *m*=3 and employs an integer variable Γ^{gh} to determine whether or not a particular burn-in criterion is selected. The upper limit may be real or artificial for the purpose of selecting the burn-in criteria.

$$\sum_{H\in h} \Gamma^{gh} = 1 \qquad \forall g \in G \tag{5.5}$$

Grade g BI criteria selection. Constraint (5.4) requires one burn-in criteria selection on one criterion per grade. This constraint may be modified based on the needs and flexibility of the users' environment.

$$\hbar^{gh}\Gamma^{gh} \leq \mathrm{f} \quad \forall g \in G,\ h \in H \tag{5.6}$$

Select BI criteria for each grade. Constraint (5.6) uses the level of quality acceptable by the customer. It ensures that the expected DPPM $\hbar^{gh}$ of the selected burn-in criterion is within the maximum DPPM f limit acceptable by the customer for a given grade.

$$\mathrm{X}_{mt}^{gh} = 0 \text{ for } m = \{1,2,3\},\ \forall\, g \in G - \{0\}, h \in H - \{0\},\ t \in T \tag{5.7}$$

$$\mathrm{X}_{mt}^{00} = 0 \text{ for } m = \{4,....,9\},\ \forall\, t \in T \tag{5.8}$$

$$\mathrm{V}_{mt}^{gh} = 0 \text{ for } m = \{1,2,3\},\ \forall\, g \in G - \{0\}, h \in H - \{0\},\ t \in T \tag{5.9}$$

$$\mathrm{V}_{mt}^{00} = 0 \text{ for } m = \{4,....,9\},\ \forall\, t \in T \tag{5.10}$$

Grading Process. Constraints (5.7) to (5.10) define the X and V variables before and after the grading process step *m* = 3. All materials before completing the grading process step *m* = 3 are defaulted to the ungraded grade *g* = 0 and the undetermined burn-in criterion *h* = 0. Both, the ungraded grade *g* = 0 and the undetermined burn-in criterion h=0, are not allowed after the grading process step *m* = 3. Consequently, variables X and V are null for processes m = {1,2,3} for all grades *G* except the ungraded material *g* = 0 and for all burn-in criteria *H* except for the undetermined criterion *h* = 0 as defined in constraints (5.7) and (5.9). Variables X and V are also null after the grading process spec *m* = 3 for the ungraded grade *g* = 0 and undetermined burn-in criterion h = 0 as defined in constraints (5.8) and (5.10).

$$\mathrm{X}_{1t}^{00} \le \alpha \qquad \forall t \in T \tag{5.11}$$

Wafer fabrication capacity limit. Constraint (5.11) specifies the capacity limitation for the wafer fabrication with an upper limit α to wafer fabrication daily capacity in units.

$$\sum_{t\in T} \mathrm{D}_t^b \le \mathrm{j}^b\, \beta^b \quad \forall\, b \in B \tag{5.12}$$

Testers/Ovens available. The relationship between the number of boards used and the number of the available tester/ovens is defined in constraint (5.12). It limits the total number of multi-probe, final-test boards and burn-in boards that are acquired to be less or equal to the number of available testers/ovens for each operation multiplied by the number of boards used per tester/oven j^{b}.

$$\mathrm{D}_t^b = \mathrm{E}_{t-\mathrm{k}^b}^b \quad \forall t \in T < \mathrm{r}, b \in B \tag{5.13}$$

Boards lead-time. Constraint (5.13) defines the lead time for each board type. It specifies the boards received D_t^b at time *t* equal the boards ordered $\mathrm{E}_{t-\mathrm{k}^b}^{b}$ at time *t* - k^b.

$$\mathrm{F}_t^b = \mathrm{F}_{t-1}^b + \mathrm{D}_t^b \quad \forall b \in B, t \in T \tag{5.14}$$

Boards availability. Constraint (5.14) defines the boards available F_t^b at time t as the sum of boards received D_t^b at same period *t* plus boards available from the previous period F_{t-1}^b.

$$\sum_{\bar{\lambda}=1}^{c_2^0} \sum_{g \in G, h \in H} \sum \mathrm{X}_{2,t-\bar{\lambda}+1}^{00} \leq \mathrm{F}_t^1 \mathrm{q}_2/\mathrm{e} \qquad \forall t \in T, \forall h \in H, \forall g \in G \tag{5.15}$$

Wafer multi-probe capacity. Constraint (5.15) specifies the wafer multi-probe test capacity based on the number of available test boards F_t^1 at time *t*, available time, test time per dice, number of dice per wafer, and an efficiency factor. The efficiency factor e accounts for the setup time and test time spent testing rejected dies as testing usually stops at first failure. The capacity calculation accounts for the cycle time to complete the multi-probe test per wafer to discount testers and boards that are currently in-use from capacity consideration. Only ungraded material *g* = 0 and material with undetermined number burn-in hours are available at multi-probe test operation *m* = 2.

$$\sum_{g \in G} (\mathrm{X}_{3t}^{gh} + \mathrm{X}_{3,t-c_2^h+1}^{gh}) \leq \theta \qquad \forall t \in T \tag{5.16}$$

Assembly capacity. The upper limit θ for the assembly capacity $m = 3$ is defined in constraint (5.16) while accounting for the assembly cycle time to discount the assembly capacity that is in-use.

$$\sum_{\lambda=1}^{c_5^h} \sum_{g \in G} \sum_{h \in H} \mathrm{X}_{5,t-\lambda+1}^{gh} + \sum_{\lambda=1}^{c_5^h} \sum_{g \in G} \sum_{h \in H} \mathrm{X}_{7,t-\lambda+1}^{gh} \leq \mathrm{F}_t^2 \mathrm{q}_5 \qquad \forall t \in T, \forall h \in H, \forall g \in G \tag{5.17}$$

Final-test test capacity. Constraint (5.17) specifies the final-test test capacity based on the number of available test boards F_t^2 at time *t*, available time, test time per dice, number of dice per wafer, and an efficiency factor. The efficiency factor e accounts for the setup time and test time spent testing rejected devices as testing usually stops at first failure. The capacity calculation accounts for the final-test cycle time while discounting testers and boards that are in-use for both final-test1 and final-test2 operations from capacity consideration.

$$\sum_{\lambda=1}^{c_6^h} \sum_{g \in G} \sum_{h \in H} \mathrm{X}_{6,t-\lambda+1}^{gh} \leq \mathrm{F}_t^3 \mathrm{s} \quad \forall t \in T, \forall h \in H, \forall g \in G \tag{5.18}$$

Burn-in capacity. The burn-in operation capacity per time period is defined in constraint (5.18) while accounting for the burn-in cycle time to discount the burn-in boards that are currently in-use from the capacity consideration. The number of units that can go into the burn-in operation in time *t* is limited by the number of available boards F_t^3, number of sockets per board s and the duration of the burn-in operation c_6^h.

$$\sum_{h\in H}\sum_{g\in G} X_{9t}^{gh} = d_t + O_t - O_{t-1} - U_t + U_{t-1} \quad \forall t \in T$$

(5.19)

Finished goods shipments. Constraint (5.19) defines the finished goods shipped to be equal to the customer demand d_t plus over-ship (surplus) material at current O_t period and previous period O_{t-1} minus under-ship (backorder) material at current period U_t and previous period U_{t-1}. Ideally in a 100% efficient manufacturing and ample capacity, the number of units shipped would be exactly equal to the customer demand and both the over-ship and under-ship would go to zero. The over-ship and under-ship variables are used in the objective function according to the business' requirement.

To summarize, below is a consolidated list of the case-study deterministic mathematical formulation equations (5.1) to (5.19):

$$\text{Minimize } P_1\left(\sum_{t\in T}^{n} U_t\right) - P_2\left(\sum_{h\in H}\sum_{g\in G}\sum_{t\in T}^{n} X_{9t}^{gh}\left(\psi - a^h\right) - \sum_{t\in T}^{n}\sum_{b\in B} D_t^b \eta^b\right)$$

subject to:

$$\mathrm{y}_{m}^{gh}\mathrm{X}_{m,t-c_m}^{gh} + \mathrm{V}_{m,t-1}^{gh} = \mathrm{X}_{mt}^{gh} + \mathrm{V}_{mt}^{gh} \qquad \forall g \in G, t \in T, h \in H, m \in M - \{3\}$$

$$\sum_{h \in H-\{0\}} \mathrm{X}_{3t}^{gh} = \mathrm{y}_{3}^{g}\mathrm{X}_{3t}^{00} \quad \forall g \in G - \{0\}, t \in T$$

$$\mathrm{X}_{3t}^{gh} \le \varphi\Gamma^{gh} \quad \forall g \in G,\ h \in H,\ t \in T$$

$$\sum_{H \in h} \Gamma^{gh} = 1 \quad \forall g \in G$$

$$\hbar^{gh}\Gamma^{gh} \le \mathrm{f} \quad \forall g \in G,\ h \in H$$

$$\mathrm{X}_{mt}^{gh} = 0 \text{ for } m = \{1,2,3\},\ \forall\, g \in G - \{0\}, h \in H - \{0\},\ t \in T$$

$$\mathrm{X}_{mt}^{00} = 0 \text{ for } m = \{4,....,9\},\ \forall\, t \in T$$

$$\mathrm{V}_{mt}^{gh} = 0 \text{ for } m = \{1,2,3\},\ \forall\, g \in G - \{0\}, h \in H - \{0\},\ t \in T$$

$$\mathrm{V}_{mt}^{00} = 0 \text{ for } m = \{4,....,9\},\ \forall\, t \in T$$

$$\mathrm{X}_{1t}^{00} \le \alpha \qquad \forall t \in T$$

$$\sum_{t \in T} \mathrm{D}_{t}^{b} \le \mathrm{j}^{b}\ \beta^{b} \quad \forall\, b \in B$$

$$\mathrm{D}_{t}^{b} = \mathrm{E}_{t-\mathrm{k}^{b}}^{b} \quad \forall t \in T < \mathrm{r}, b \in B$$

$$\mathrm{F}_{t}^{b} = \mathrm{F}_{t-1}^{b} + \mathrm{D}_{t}^{b} \quad \forall b \in B, t \in T$$

$$\sum_{\lambdabar=1}^{c_2^0} \sum_{g \in G, h \in H} \mathrm{X}_{2,t-\lambdabar+1}^{00} \le \mathrm{F}_{t}^{1}\mathrm{q}_{2}/\mathrm{e} \qquad \forall t \in T, \forall h \in H, \forall g \in G$$

$$\sum_{g \in G} (\mathrm{X}_{3t}^{gh} + \mathrm{X}_{3,t-c_2^h+1}^{gh}) \le \theta \qquad \forall t \in T$$

$$\sum_{\lambdabar=1}^{c_5^h} \sum_{g \in G} \sum_{h \in H} \mathrm{X}_{5,t-\lambdabar+1}^{gh} + \sum_{\lambdabar=1}^{c_5^h} \sum_{g \in G} \sum_{h \in H} \mathrm{X}_{7,t-\lambdabar+1}^{gh} \le \mathrm{F}_{t}^{2}\mathrm{q}_{5} \qquad \forall t \in T, \forall h \in H, \forall g \in G$$

$$\sum_{\lambda=1}^{c_6^h}\sum_{g\in G}\sum_{h\in H} \mathrm{X}^{gh}_{6,t-\lambda+1} \le \mathrm{F}^3_t \mathrm{s} \quad \forall t\in T, \forall h\in H, \forall g\in G$$

$$\sum_{h\in H}\sum_{g\in G} \mathrm{X}^{gh}_{9t} = \mathrm{d}_t + \mathrm{O}_t - \mathrm{O}_{t-1} - \mathrm{U}_t + \mathrm{U}_{t-1} \quad \forall t\in T$$

5.2 Stochastic Formulation

This section presents the stochastic mathematical formulation for the case study in chapter 4 using the CPLEX linear programming model. This stochastic model is an extension of the deterministic model presented in section 5.1 which is based on the generic model presented in chapter 3. This model uses multistage stochastic programs with recourse to account for continuous results for sequence of decisions [3]. Linearity is assumed with fixed recourse and a finite horizon.

The stochastic formulation is of a great significance as it models real-life situations and uncertainty. It designed to help users account for uncertain environment such as quality output and manage uncertain expectation such as customer demand.

5.2.1 Notation and Conventions

The following notations, symbols and conventions are used:

- Sets or sequences are denoted as italicized upper-case letters, such as *S*. The sets' cardinality is denoted as |*S*|.
- Members of a set are denoted as italicized lower-case letter such as *b*.
- Variables are denoted as non-italicized upper-case letter such as D.
- Scalars are denoted as non-italicized lower-case letters such as c.
- Sets of real numbers and integers are denoted as *R* and *Z*, respectively.

The following symbols are used in the mathematical formulation using the above notation:

<u>Sets:</u>

T Set of time periods: $T = \{1,2,3,....,\mathrm{n}\}$, where n is the number of periods

G Set of material quality grades: $G = \{0,1,...,5\}$, where 0 = ungraded, 1 = grade P, 2 = grade W, 3 = grade X, 4 = grade Y, and 5 = grade R.

M Set of operations: $M = \{1,2,...,9\}$, where 1 = wafer fabrication, 2 = wafer multi-probe testing, 3 = grading, 4 = plastic units assembly, 5 = final-test 1, 6 = burn-in, 7 = final-test 2, 8 = pack, and 9 = finished-goods ship.

H Set of possible burn-in hours: $H = \{0, 1, 2, 3\}$, where 0 = undetermined number of burn-in hours, 1 = 0 burn-in hour, 2 = 5 burn-in hours, and 3 = 24 burn-in hours.

R Set of root-cause-resolution scenarios: $R = \{0, 1, 2\}$, where 0 = no root-cause is identified, 1 = partial root-cause is identified and corrective actions implemented, and 2 = full root-cause is identified and corrective actions implemented.

B Set of boards: $B = \{1, 2, 3\}$, where 1 = multi-probe-test boards, 2 = final-test boards, and 3 = burn-in boards.

Variables:

X_{mt}^{ghr} units of material of grade $g \in G$ using burn - in criterion $h \in H$ entering operation $m \in M$ in period $t \in T$ under root - cause - resolution scenario $r \in R$

V_{mt}^{ghr} units in inventory of material created by operation $m \in M$ of grade $g \in G$ using burn - in criterion $h \in H$ at the start of period $t \in T$ under root - cause - resolution scenario $r \in R$

O_{t}^{r} units of shipped products above customer demand requirements in period $t \in T$ under root - cause - resolution scenario $r \in R$

U_{t}^{r} units of shipped products below customer demand requirements in period $t \in T$ under root - cause - resolution scenario $r \in R$

E_{t}^{br} boards ordered at the start of time period $t \in T, b \in B, r \in R$

D_{t}^{br} boards received at the start of time period $t \in T, b \in B, r \in R$

F_{t}^{br} boards available at the start of time period $t \in T, b \in B, r \in R$

Γ^{ghr} 0/1 = don't/do use burn-in criterion $h \in H$ for grade g $\in G$ under root-cause-resolution scenario $r \in R$

Scalars:

p^{r} probability of full - partial - none root - cause - resolution $r \in R$

ω days until root - cause - identification and resolution

y_m^{gh} yield (including average conversion of units) of operation $m \in M$ for material of burn - in hours $h \in H$ of grade $g \in G$

a^h total variable manufactuing cost per unit of finished product using burn - in criterion $h \in H$

d_t customer demand requirements in period $t \in T$, in units

c_m^h cycle time for operation $m \in M, \quad h \in H$ in days

k^b leadtime in days to receive ordered boards, $b \in B$

j^b number of boards used per tester/oven, $b \in B$

q_m multiplier = number of seconds in one day * efficiency factor / test time in

seconds, $m \quad =\{2,5,7\}$, $b \in R$ (note: $b_5 = b_7$)

e number of dies per wafer, $e \in Z$

s number of sockets per burn-in board, $s \in Z$

α wafer fabrication daily capacity in number of wafers, $\alpha \in Z$

θ assembly operation daily capacity in units, $\theta \in Z$

ψ unit selling price, $\psi \in Z$

η^b cost of one board, $\eta \in Z, b \in B$

f maximum acceptable DPPM

$\hbar^{gh}$ DPPM expected after 100,000 hours of operation, $\hbar \in Z, h \in H, g \in G$

φ upper limit on all X_{mt}^{ghr} values

β^{br} maximum number of testers/ovens available for use, $\beta \in Z, b \in B, r \in R$

5.2.2 Integrated Process Design, Planning, and Stochastic System Optimization Model

An integrated process design, planning and stochastic system optimization model is developed using the above notation. This model is a generalized network model for the material flow, processing and inventory and is an extension of the deterministic model presented in section 5.1. The stochastic model is customized per the uncertainty defined in the case study presented in chapter 4. Each element of the model is expressed mathematically followed by a detailed discussion. The description for the deterministic formulation in section 5.1 generally applies for the stochastic formulation. Some elements in the following description are repeated for the purpose of completeness. The main difference in this formulation is the addition of the set of root-cause-resolution scenarios "R" and the probability of full-partial-none root-cause resolution p^r to account to the stochastic dimension of the formulation.

The goal is to mathematically formulate the case-study's variables and constraints to effectively implement and test the model using a mathematical programming and optimization high-level modeling system presented in chapter 6. The objective function is:

$$\text{Minimize } \mathrm{P}_1\left(\sum_{r\in R}\sum_{t\in T}^{n}\mathrm{U}_t^r \mathrm{p}^r\right) - \mathrm{P}_2\left(\sum_{h\in H}\sum_{g\in G}\sum_{t\in T}^{n}\sum_{r\in R}\mathrm{X}_{9t}^{ghr}\left(\psi-\mathrm{a}^h\right)\mathrm{p}^r - \sum_{t\in T}^{\omega\text{-}1}\sum_{b\in B}\sum_{r\in R}\mathrm{E}_t^{br}\eta^b - \sum_{t=\omega}^{t}\sum_{b\in B}\sum_{r\in R}\mathrm{E}_t^{br}\eta^b\mathrm{p}^r\right) \tag{5.20}$$

Objective function. Constraint (5.20) represents the objective function that is defined as a combination of two preemptive priorities with the first one, P_1, being the primary priority. The primary objective P_1 is to minimize the cumulative expected sum of units of shipped products below customer demand at time *t*, which is the number of units of shipped products below customer demand requirements at time *t* U_t^r multiplied by the probability of full-partial-none root-cause resolution p^r. This makes on-time shipments the top priority. The second priority, P_2, is to maximize effective profit. The effective profit calculation is divided into three terms. First term calculates the expected revenues by multiplying the probability of full-partial-none root-cause resolution p^r and the total number of units shipped X_{9t}^{ghr} at operation $m = 9$ (finished-goods ship) by the unit price ψ and then subtracting the manufacturing variable cost a^h. Second term calculates the cost of ordering boards $\mathrm{E}_t^{br}\eta^b$ before root-cause identification and resolution noted by ω. Third term is the cost of BI board and test hardware $\mathrm{E}_t^{br}\eta^b$ multiplied by the probability of full-partial-none root-cause resolution p^r for boards ordered one the day or after root-cause identification and resolution noted by ω. As in the deterministic model, priorities are selected to minimize disruption to the customer's production line as a result of the quality excursion as a top priority while maximizing profit as second priority. The model follows a deterministic path before root-cause identification and resolution. Once the root-cause is identified and a resolution is implemented, the probability of the full-partial-none resolution is factored into the formulation. The model's flexibility allows changing priorities according to the business' needs.

$$\mathrm{y}_{m}^{ghr}\mathrm{X}_{m,t-c_m}^{ghr}+\mathrm{V}_{m,t-1}^{ghr}=\mathrm{X}_{mt}^{ghr}+\mathrm{V}_{mt}^{ghr} \quad \forall g\in G, t\in T, h\in H, r\in R, m\in M-\{3\} \tag{5.21}$$

Material Flow. Similar to constraint (5.2), constraint (5.21) represents a flow balance equation at a given point in time. It defines the flow of material in each processing step plus inventory as equal to the flow in the previous step multiplied by the process yield plus its inventory. The flow in the previous step is in relation to the cycle time of the specific operation. The key variables, X and V, represent different materials – wafers, WIPs and finished goods – at each stage in the process of converting wafers into shippable finished product.

$$\mathrm{X}_{1,t}^{000}=\mathrm{X}_{1,t}^{001} \quad \forall\, t\in T<\mathrm{r} \tag{5.22}$$

$$\mathrm{X}_{1,t}^{000}=\mathrm{X}_{1,t}^{002} \quad \forall\, t\in T<\mathrm{r} \tag{5.23}$$

Wafer starts before root cause identification. Constraints (5.22) and (5.23) use the same logic used in objective function (5.20) to follow a deterministic path before root-cause identification and resolution. They specify that the number of wafers started for all root-cause resolution scenarios must be equal before root-cause identification and resolution r.

$$\sum_{h \in H-\{0\}} \mathrm{X}_{3t}^{ghr} = \mathrm{y}_3^g \mathrm{X}_{3t}^{00} \qquad \forall r \in R,\ g \in G-\{0\},\ t \in T \tag{5.24}$$

Yield Multiplier. The process yield multiplier in constraint (5.24) represents the loss in the process during the conversion of material from wafers to finished goods where applicable. Step *m*=3 is a grading process step that splits all ungraded material into its component grades based on a defined criteria. The yield is accounted for in the yield grading process. Yield calculations do not apply to ungraded material *g*={0} and material with undetermined number of BI hours *h*={0}.

$$\mathrm{X}_{3t}^{ghr} \le \varphi\, \Gamma^{ghr} \qquad \forall g \in G,\ h \in H,\ r \in R, t \in T \tag{5.25}$$

BI criteria selection. Constraint (5.25) defines an upper limit φ on all material at grading process step *m*=3 and employs an integer variable Γ^{gh} to determine whether or not a particular burn-in criterion is selected. The upper limit may be real or artificial for the purpose of selecting the burn-in criteria.

$$\sum_{H \in h} \Gamma^{ghr} = 1 \qquad \forall g \in G, r \in R \tag{5.26}$$

Grade g BI criteria selection. Constraint (5.26) requires one burn-in criteria selection on one criterion per grade. This constraint may be modified based on the needs and flexibility of the users' environment.

$$\hbar^{gh}\Gamma^{ghr} \leq \mathrm{f} \quad \forall g \in G,\ h \in H, r \in R \tag{5.27}$$

Select BI criteria for each grade. Constraint (5.6) uses the level of quality acceptable by the customer. It ensures that the expected DPPM $\hbar^{gh}$ of the selected burn-in criterion is within the maximum DPPM f limit acceptable by the customer for a given grade.

$$\mathrm{X}_{mt}^{ghr} = 0 \text{ for } m = \{1,2,3\},\ \forall\, g \in G - \{0\}, h \in H - \{0\},\ r \in R, t \in T \tag{5.28}$$

$$\mathrm{X}_{mt}^{00r} = 0 \text{ for } m = \{4,....,9\},\ \forall\, t \in T, r \in R \tag{5.29}$$

$$\mathrm{V}_{mt}^{ghr} = 0 \text{ for } m = \{1,2,3\},\ \forall\, g \in G - \{0\}, h \in H - \{0\}, r \in R, t \in T \tag{5.30}$$

$$\mathrm{V}_{mt}^{00r} = 0 \text{ for } m = \{4,....,9\},\ \ \forall\, t \in T, r \in R \tag{5.31}$$

Grading Process. Constraints (5.28) to (5.31) define the X and V variables before and after the grading process step *m* = 3. All materials before completing the grading process step *m* = 3 are defaulted to the ungraded grade *g* = 0 and the undetermined burn-in criterion *h* = 0. Both, the ungraded grade *g* = 0 and the undetermined burn-in criterion h=0, are not allowed after the grading process step *m* = 3. Consequently, variables X and V are null for processes m = {1,2,3} for all grades *G* except the ungraded material *g* = 0 and for all burn-in criteria *H* except for the undetermined criterion *h* = 0 as defined in constraints (5.28) and (5.30). Variables X and V are also null after the grading process spec *m* = 3 for the ungraded grade *g* = 0 and undetermined burn-in criterion h = 0 as defined in constraints (5.29) and (5.31).

$$\mathrm{X}_{1t}^{00r} \le \alpha \qquad \forall\, t \in T,\ r \in R \tag{5.32}$$

Wafer fabrication capacity limit. Constraint (5.32) specifies the capacity limitation for the wafer fabrication with an upper limit α to wafer fabrication daily capacity in units.

$$\sum_{t\in T} \mathrm{D}_t^{br} \le \mathrm{j}^b\, \beta^{br} \quad \forall\, b \in B, r \in R \tag{5.33}$$

Testers/Ovens available. The relationship between the number of boards used and the number of the available tester/ovens is defined in constraint (5.33). It limits the total number of multi-probe, final-test boards and burn-in boards that are acquired to be less or equal to the number of available testers/ovens for each operation multiplied by the number of boards used per tester/oven j^{b}.

$$\mathrm{D}_t^{br} = \mathrm{E}_{t-\mathrm{k}^b}^{br} \quad \forall t \in T < \mathrm{r}, b \in B, r \in R \tag{5.34}$$

Boards lead-time. Constraint (5.34) defines the lead time for each board type. It specifies the boards received D_t^{br} at time *t* equal the boards ordered $\mathrm{E}_{t-\mathrm{k}^b}^{br}$ at time *t* - k^b.

$$\mathrm{E}_t^{b0} = \mathrm{E}_t^{b1} \quad \forall b \in B, t \in T < \mathrm{r} \tag{5.35}$$

$$\mathrm{E}_t^{b0} = \mathrm{E}_t^{b2} \quad \forall b \in B, t \in T < \mathrm{r} \tag{5.36}$$

Boards ordered before root cause identification. Constraints (5.35) and (5.36) use the same logic used in objective function (5.20) to follow a deterministic path before root-cause identification and resolution. They specify that the number of boards ordered for all root-cause resolution scenarios must be equal before root-cause identification and resolution r.

$$\mathrm{F}_t^{br} = \mathrm{F}_{t-1}^{br} + \mathrm{D}_t^{br} \quad \forall b \in B, t \in T, r \in R \tag{5.37}$$

Boards availability. Constraint (5.37) defines the boards available F_t^{br} at time t as the sum of boards received D_t^{br} at same period *t* plus boards available from the previous period F_{t-1}^{br}.

$$\sum_{\lambda=1}^{c_2^0} \sum_{g \in G} \sum_{h \in H} \sum_{r \in R} \mathrm{X}_{2,t-\lambda+1}^{00r} \leq \mathrm{F}_t^{1r} \mathrm{q}_2/\mathrm{e} \qquad \forall t \in T, h \in H, g \in G, r \in R \tag{5.38}$$

Wafer multi-probe capacity. Constraint (5.38) specifies the wafer multi-probe test capacity based on the number of available test boards F_t^{1r} at time *t*, available time, test time per dice, number of dice per wafer, and an efficiency factor. The efficiency factor e accounts for the setup time and test time spent testing rejected dies as testing usually stops at first failure. The capacity calculation accounts for the cycle time to complete the multi-probe test per wafer to discount testers and boards that are currently in-use from capacity consideration. Only ungraded material $g = 0$ and material with undetermined number burn-in hours are available at multi-probe test operation $m = 2$.

$$\sum_{g\in G}(\mathrm{X}_{3t}^{ghr} + \mathrm{X}_{3,t-c_2^h+1}^{ghr}) \leq \theta \qquad \forall t \in T, r \in R \tag{5.39}$$

Assembly capacity. The upper limit θ for the assembly capacity $m = 3$ is defined in constraint (5.39) while accounting for the assembly cycle time to discount the assembly capacity that is in-use.

$$\sum_{\lambda=1}^{c_5^h}\sum_{g\in G}\sum_{h\in H}\mathrm{X}_{5,t-\lambda+1}^{ghr} + \sum_{\lambda=1}^{c_5^h}\sum_{g\in G}\sum_{h\in H}\mathrm{X}_{7,t-\lambda+1}^{ghr} \leq \mathrm{F}_t^{2r}\mathrm{q}_5 \quad \forall t \in T, h \in H, g \in G,\ r \in R \tag{5.40}$$

Final-test test capacity. Constraint (5.40) specifies the final-test test capacity based on the number of available test boards F_t^{2r} at time *t*, available time, test time per dice, number of dice per wafer, and an efficiency factor. The efficiency factor accounts for the setup time and test time spent testing rejected devices as testing usually stops at first failure. The capacity calculation accounts for the final-test cycle time while discounting testers and boards that are in-use for both final-test1 and final-test2 operations from capacity consideration.

$$\sum_{\lambda=1}^{c_6^h} \sum_{g \in G} \sum_{h \in H} \mathrm{X}_{6,t-\lambda+1}^{ghr} \le \mathrm{F}_t^{3r}\mathrm{s} \quad \forall t \in T, h \in H, g \in G, r \in R \tag{5.41}$$

Burn-in capacity. The burn-in operation capacity per time period is defined in constraint (5.41) while accounting for the burn-in cycle time to discount the burn-in boards that are currently in-use from the capacity consideration. The number of units that can go into the burn-in operation in time *t* is limited by the number of available boards F_t^{3r}, number of sockets per board s and the duration of the burn-in operation c_6^h.

$$\sum_{h \in H} \sum_{g \in G} \mathrm{X}_{9t}^{ghr} = \mathrm{d}_t + \mathrm{O}_t^r - \mathrm{O}_{t-1}^r - \mathrm{U}_t^r + \mathrm{U}_{t-1}^r \quad \forall t \in T, r \in R \tag{5.42}$$

Finished-goods shipments. Constraint (5.42) defines the finished-goods shipped to be equal to the customer demand d_t plus over-ship (surplus) material at current O_t^r period and previous period O_{t-1}^r minus under-ship (backorder) material at current period U_t^r and previous period U_{t-1}^r. Ideally in a 100% efficient manufacturing and ample capacity, the number of units shipped would be exactly equal to the customer demand and both the over-ship and under-ship would go to zero. The over-ship and under-ship variables are used in the objective function according to the business' requirement.

To summarize, below is a consolidated list of the case study stochastic mathematical formulation equations (5.20) to (5.42):

$$\text{Minimize } \mathrm{P}_1\left(\sum_{r\in R}\sum_{t\in T}^{n}\mathrm{U}_t^r\mathrm{p}^r\right) -$$
$$\mathrm{P}_2\left(\sum_{h\in H}\sum_{g\in G}\sum_{t\in T}^{n}\sum_{r\in R}\mathrm{X}_{9t}^{ghr}\left(\psi-\mathrm{a}^h\right)\mathrm{p}^r-\sum_{t\in T}^{\omega\text{-}1}\sum_{b\in B}\sum_{r\in R}\mathrm{E}_t^{br}\eta^b-\sum_{t=\omega}^{t}\sum_{b\in B}\sum_{r\in R}\mathrm{E}_t^{br}\eta^b\mathrm{p}^r\right)$$

subject to:

$$\mathrm{y}_m^{ghr}\mathrm{X}_{m,t-c_m}^{ghr}+\mathrm{V}_{m,t-1}^{ghr}=\mathrm{X}_{mt}^{ghr}+\mathrm{V}_{mt}^{ghr}\quad \forall g\in G, t\in T, h\in H, r\in R, m\in M-\{3\}$$

$$\mathrm{X}_{1,t}^{000}=\mathrm{X}_{1,t}^{001}\qquad \forall\, t\in T<\mathrm{r}$$

$$\mathrm{X}_{1,t}^{000}=\mathrm{X}_{1,t}^{002}\qquad \forall\, t\in T<\mathrm{r}$$

$$\sum_{h\in H-\{0\}}\mathrm{X}_{3t}^{ghr}=\mathrm{y}_3^g\mathrm{X}_{3t}^{00}\qquad \forall r\in R,\ g\in G-\{0\},\ t\in T$$

$$\mathrm{X}_{3t}^{ghr}\le\varphi\,\Gamma^{ghr}\qquad \forall g\in G,\ \ h\in H,\ \ r\in R, t\in T$$

$$\sum_{H\in h}\Gamma^{ghr}=1\qquad \forall g\in G, r\in R$$

$$\hbar^{gh}\Gamma^{ghr}\le\mathrm{f}\qquad \forall g\in G,\ \ h\in H, r\in R$$

$$\mathrm{X}_{mt}^{ghr}=0\ \text{ for }\ m=\{1,2,3\},\ \forall\, g\in G-\{0\}, h\in H-\{0\},\ \ r\in R, t\in\ T$$

$$\mathrm{X}_{mt}^{00r}=0\ \text{ for }\ m=\{4,....,9\},\ \forall\, t\in\ T, r\in R$$

$$\mathrm{V}_{mt}^{ghr}=0\ \text{ for }\ m=\{1,2,3\},\ \forall\, g\in G-\{0\}, h\in H-\{0\}, r\in R, t\in\ T$$

$$\mathrm{V}_{mt}^{00r}=0\ \text{ for }\ m=\{4,....,9\},\ \ \forall\, t\in\ T, r\in R$$

$$\mathrm{X}_{1t}^{00\,r} \leq \alpha \qquad \forall\, t \in T\,,\, r \in R$$

$$\sum_{t\in T} \mathrm{D}_t^{br} \leq \mathrm{j}^b\, \beta^{br} \quad \forall\, b \in B, r \in R$$

$$\mathrm{D}_t^{br} = \mathrm{E}_{t-\mathrm{k}^b}^{br} \quad \forall t \in T < \mathrm{r}, b \in B, r \in R$$

$$\mathrm{E}_t^{b0} = \mathrm{E}_t^{b1} \quad \forall b \in B, t \in T < \mathrm{r}$$

$$\mathrm{E}_t^{b0} = \mathrm{E}_t^{b2} \quad \forall b \in B, t \in T < \mathrm{r}$$

$$\mathrm{F}_t^{br} = \mathrm{F}_{t-1}^{br} + \mathrm{D}_t^{br} \quad \forall b \in B, t \in T, r \in R$$

$$\sum_{\lambda=1}^{c_2^0} \sum_{g\in G} \sum_{h\in H} \sum_{r\in R} \mathrm{X}_{2,t-\lambda+1}^{00r} \leq \mathrm{F}_t^{1r}\mathrm{q}_2/\mathrm{e} \qquad \forall t \in T, h \in H, g \in G, r \in R$$

$$\sum_{g\in G} (\mathrm{X}_{3t}^{ghr} + \mathrm{X}_{3,t-c_2^h+1}^{ghr}) \leq \theta \qquad \forall t \in T, r \in R$$

$$\sum_{\lambda=1}^{c_5^h} \sum_{g\in G} \sum_{h\in H} \mathrm{X}_{5,t-\lambda+1}^{ghr} + \sum_{\lambda=1}^{c_5^h} \sum_{g\in G} \sum_{h\in H} \mathrm{X}_{7,t-\lambda+1}^{ghr} \leq \mathrm{F}_t^{2r}\mathrm{q}_5 \qquad \forall t \in T, h \in H, g \in G,\ r \in R$$

$$\sum_{\lambda=1}^{c_6^h} \sum_{g\in G} \sum_{h\in H} \mathrm{X}_{6,t-\lambda+1}^{ghr} \leq \mathrm{F}_t^{3r}\mathrm{s} \quad \forall t \in T, h \in H, g \in G, r \in R$$

$$\sum_{h\in H} \sum_{g\in G} \mathrm{X}_{9t}^{ghr} = \mathrm{d}_t + \mathrm{O}_t^r - \mathrm{O}_{t-1}^r - \mathrm{U}_t^r + \mathrm{U}_{t-1}^r \quad \forall t \in T, r \in R$$

CHAPTER 6

CASE STUDY IMPLEMENTATION AND COMPUTATIONAL TESTING

6.1 Implementation Description

A General Algebraic Modeling System (GAMS) is selected to implement the mathematical formulation described in chapter 5 that models the case study presented in chapter 4. GAMS is a mathematical programming and optimization high-level modeling system that consists of a language compiler and a number of integrated high-performance solvers. GAMS is designed for complex and large scale modeling applications that allows to build large adaptable models that can be customized to new situations.

CPLEX optimizer is selected to solve the modeling problem. A mixed integer programming (MIP) procedure and the barrier linear programming algorithm method with a relative optimality tolerance of 0.10 are used to optimize results. The barrier method is an alternative to the simplex method for solving linear programs. The barrier method or the internal-point is mainly employing primal-dual logarithmic barrier algorithm that generates a sequence of strictly positive primal and dual solutions. The barrier algorithm is used because of its advantage for large complex sparse problems.

The integrality restrictions on several variables are relaxed to allow for efficient execution of the model. The integrality restrictions are enforced on the following variables to avoid ordering partial wafers and boards that is not consistent with the manufacturing environment.

FABin(T,R)	Wafers STARTED in the fab in time period T
BIB_ordered(T,R)	Qty of Burn-in boards ordered
MPHW_ordered(T,R)	Qty of Multi-Probe boards ordered
FTHW_ordered(T,R)	Qty of Final-Test boards ordered;

The GAMS model is developed using the production flow outlined in figure 4.2. Figure 6.1 outlines the interaction between process operations consistent with the material flow defined in constraints 5.2 and 5.21. It defines the flow of material in each processing step plus inventory as equal to the flow in the previous step plus its inventory. The burn-in and the final-test2 operations are valid for selected grades based on the expected DPPM level required by the customer.

Figure 6.2 show a top-level view of the production flow assuming grades X, Y and R will be required to go through the burn-in and final-test2 operations. The flow starts with wafer fabrication going through the multi-probe test operation leading to the grading splits and assembly process. The model allows adding or altering steps in the production flow as necessary.

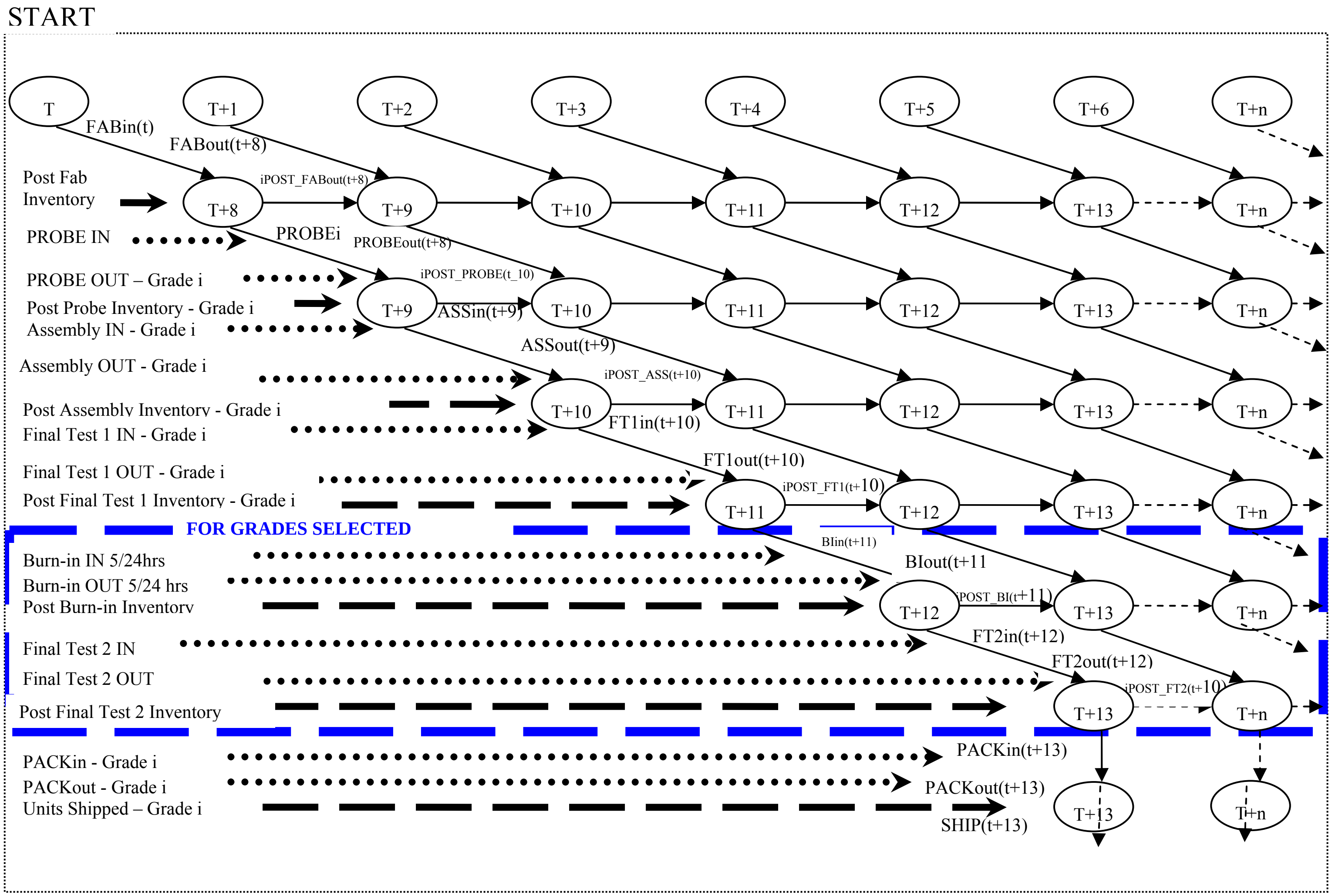

Figure 6.1 Production Flow Diagram

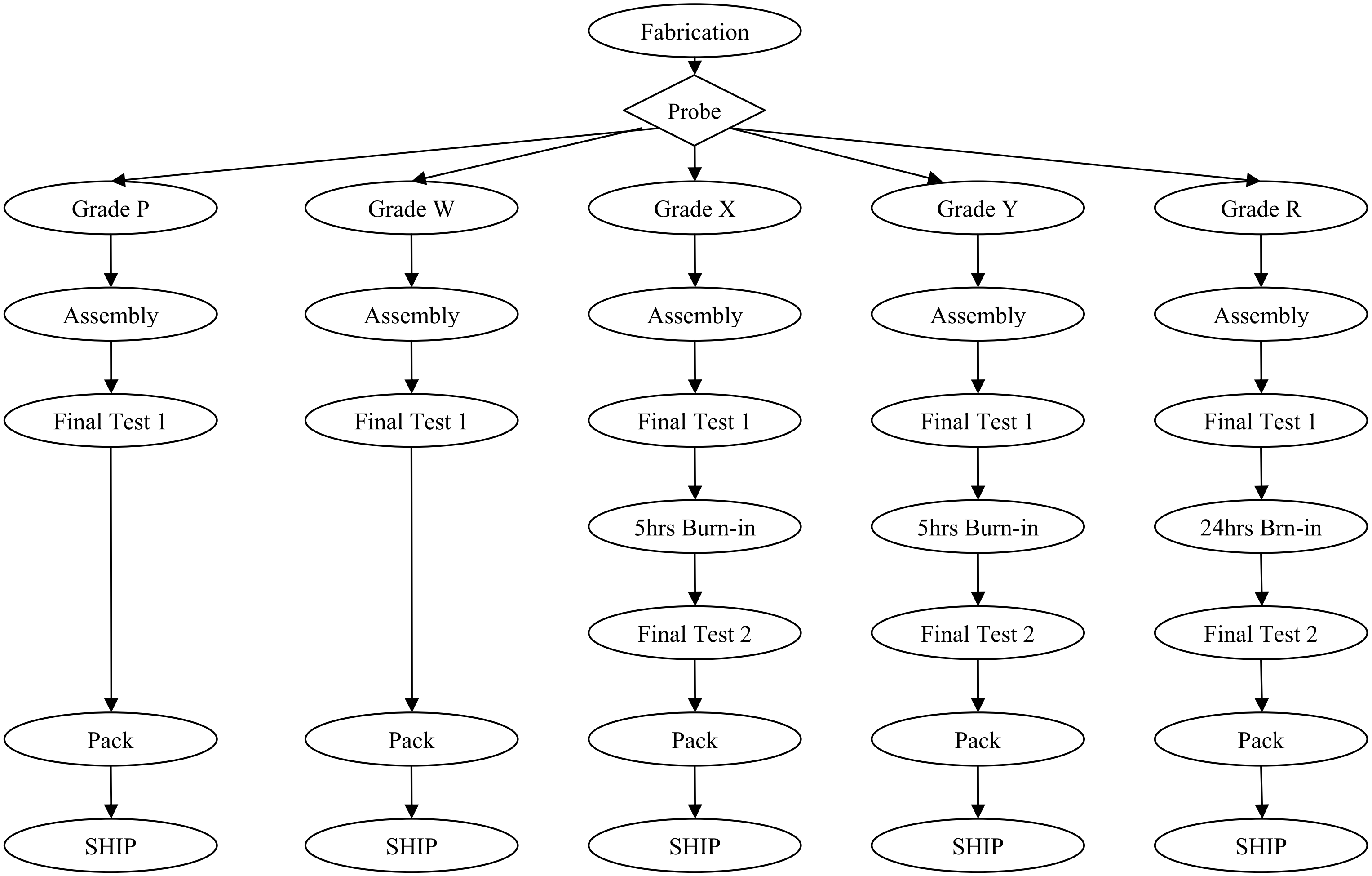

Figure 6.2 Top Level View

The parameters listed in table 6.1 are used in the developing the model.

Scalar	**Description**	**Value**
Trend	Customer Demand Trend in # of Devices	4000
Stddev	Customer Demand Standard Deviation in # of Device	300
RTDAYS	Days Unit Root-cause Identify	30
DIES	Number of Dies Per Wafers	1500
SOCKETS	Total sockets per burn-in board	50
Capacity		
ASSCAP	Assembly Capacity in Units	250000
Lead Time		
BIBLT	Burn-in boards order lead time in days	45
MPHWLT	Multi-probe hardware order lead time days	28
FTHWLT	Final-test hardware order lead time days	28
HW Cost		
MPHWCOST	Multi Probe Hardware Cost in US Dollars	4000
FTHWCOST	Final-test Hardware Cost in US Dollars	3000
BIBCOST	Burn-in Board Cost in US Dollars	2500
Test Time		
MPTT	Multi-Probe Program Test Time in Seconds	2
FTTT	Final-test Program Test Time in Seconds	2.25
Yield		
PY	Fab Process Yield	0.95
MPEFF	Multi-probe Test Percent Efficiency (uptime)	0.9
AY	Assembly Yield	0.99
FTEFF	Final-test Percent Efficiency (uptime)	0.9
PackYield	Packing Yield	0.8
Cycle Time		
FABCT	Fab Cycle Time in Days	56
MPCT	Multi Probe Test Cycle Time for one 24-wafer lot in Days	1
ACT	Assembly Cycle Time in day	3
PACCT	Packing Cycle Time in days	1

Table 6.1 Parameters List

In addition to the inputs parameters listed in table 6.1, the following capacity and cycle time assumptions are defined as constraints:

1. FAB Capacity, which determines the number of wafers started (FABIN) – also used in the design of experiment analysis.
2. Customer demand baseline – also used in the design of experiment analysis.
3. Maximum total burn-in boards ordered – 300 boards.
4. Maximum number of Multi-probe testers available – 10 testers.
5. Maximum number of Final-test testers available – 10 testers.
6. Final-test 1 cycle time is 1 day.
7. Test time is for one unit testing at a time.

Grade split percentage are defined per the root-cause-identification scenario and in-line test data. The grades split are listed in table 6.2.

	Root-cause Identification Scenario		
Grade\Scenario	**None**	**Partial**	**Full**
P	0.53	0.80	0.98
W	0.11	0.12	0.00
X	0.11	0.05	0.00
Y	0.06	0.00	0.00
R	0.06	0.00	0.00

Table 6.2 Grades Split

Root-cause-identification scenarios (NONE, Partial, Full) are based on the expectation of identifying the root-cause of the quality excursion and implementing corrective actions within the next 30 days. NONE indicates that the root-cause is not expected to be identified within the next 30 days. Partial indicates that only a partial resolution of the quality issue is expected within the next 30 days. Full indicates that full resolution and corrective actions implementation is expected within the next 30 days.

The GAMS implementation is based on the stochastic mathematical formulation by using the root-cause-resolution scenarios as one of the design factors. The root-cause identification scenarios are future predictions of the resolution of the quality excursion. Stochastic formulation is addressed by assigning probabilities to the root-cause identification scenarios. These probabilities are used to assess the expected impact of the quality excursion and help in decision making based on today's data.. Stochastic mathematical formulation is detailed in chapter 5.2. The probabilities are selected by the following function.

```
Parameter PROB(R);
If (RootCauseProbability = 1,
    PROB('None') = 0.1;
    PROB('Partial') = 0.3;
    PROB('Full') = 0.6;
else
    PROB('None') = 0.6;
    PROB('Partial') = 0.1;
    PROB('Full') = 0.3;);
```

The implementation can be easily changed from stochastic to deterministic by removing the uncertainty in the root-cause-resolution scenarios. This can be accomplished by setting the probabilities of the "Partial" root-cause-resolution scenario to 0 and the scenarios "None" and "Full" to mutually exclusive probabilities zero or one.

6.2 Cost and Pricing

The manufacturing cost utilized is based on estimated variable cost for each manufacturing operation. The cost per operation is listed in table 6.3.

Operation	Cost
Fab (per wafer)	$0.55
Multi-probe (per wafer)	$0.05
Assembly (per device)	$0.04
Final-test1 (per device)	$0.04
Burn-in (per hour)	$0.02
Final-test2 (per device)	$0.04
Packing (per device)	$0.001
Shipping (per device)	$0.001

Table 6.3 Cost Per Operation

In addition to the operations' cost, a grade cost multiplier is defined to encourage shipping higher prime grades over lower grades. The grade multiplier costs are listed in table 6.4. The unit price is set to $1.60 that is a based on the fair market value at the time.

Grade	Grade Multiplier Cost
P	$1.00
W	$1.01
X	$1.02
Y	$1.03
R	$1.04

Table 6.4 Grade Cost Multiplier

6.3 Computational Testing

Extensive computational testing and evaluation is performed to prove model's assumptions and validate results. The GAMS model execution includes 64 runs covering all factor level combinations (flc) defined in a design of experiments to test significant corners as outlined in section 6.4. The "normal" function shown in (6.1) is used to select random customer demand based on normal distribution using trend = 4000 and standard deviation = 300.

```
PARAMETER       CUSTDMD(T)      Customer Demands Per Time Period T;
 CUSTDMD(T) $ (mod(ord(t),7) = 0) = Floor (normal(baseline +
                                (trend * ord(T)), stddev));                    (6.1)
```

The “seed” value is varied to generate different customer demands. GAMS pseudo random number generator is used to select different customer demands by choosing a different “seed”. One of the following ‘seed’ values is selected at a time.

```
option seed = 2784;
option seed = 3141;
option seed = 3263;
option seed = 4135;
```

6.4 Design of Experiment

The objective of the design of experiment (DOE) is to determine the factors that affect the expected profit. The design factors selected for this study are demand (D), fab capacity (C), root-cause-identification probability (P), and the DPPM quality level (Q). As shown in table 6.5, the projected daily customer demand is a based on 3 parameters: baseline, trend and standard deviation. The model calculates the demand using a normal distribution formula. For the purpose of this analysis, only the baseline is varied between 50,000 units (low) and 150,000 units (high). The fab daily capacity levels are 72 wafers (low) and 120 wafers (high). The root-cause identification probability within 30 days contains 3 elements: NONE (no root-cause identified), Partial (partial root-cause identification) and Full (full root-cause identification). The low probability factor level has 60% for NONE, 10% for Partial and 30% for full root-cause identification. The high probability factor level has 10% for NONE, 30% for Partial and 60% for full root-cause identification. The forth factor is the acceptable projected DPPM quality level at 100,000 of operation with two levels: 1500 for low and 4000 for high. The design factors are summarized in table 6.5.

Factor	Low	High

Projected Daily Customer Demand (D)	Baseline = 50,000 units
	Trend = 4,000 units
	Standard Deviation = 300 units
		Baseline = 150,000 units
		Trend = 4,000 units
		Standard Deviation = 300 units

Daily Capacity (C	Fab = 72 wafers	Fab = 120 wafers

Root-cause Identification Probability (P)	NONE = 0.6
	Partial = 0.1
	Full = 0.3
		NONE = 0.1
		Partial = 0.3
		Full = 0.6

Acceptable Projected DPPM Quality Level at 100,000 of Operation (Q)	1,500	4,000

Table 6.5 DOE Design Factors

To study the effects of the factors, a complete 2^4 balanced factorial experiments with 4 repeat is constructed. There are 16 factor level combinations in the experiment, enumerated in table 6.6. 4 repeats were performed based on randomized demand baseline to make a total of 64 runs. The demand baseline is randomized using a normal distribution function and specify a different seed value as described in section 6.3. All of the runs and seed values have been enumerated and randomized, as listed in table 6.7. The runs were randomized using EXCEL random number generator function to eliminate any sources of bias in the experiment. However, due to the nature of running the CPLEX model, each run yields the same expected profit despite of the runs' random order.

Factor Level Combinations				
flc	Demand (D)	Capacity C	Root Cause Identification Probability (P)	DPPM (Q)
1	Low	Low	Low	Low
2	Low	Low	Low	High
3	Low	Low	High	Low
4	Low	Low	High	High
5	Low	High	Low	Low
6	Low	High	Low	High
7	Low	High	High	Low
8	Low	High	High	High
9	High	Low	Low	Low
10	High	Low	Low	High
11	High	Low	High	Low
12	High	Low	High	High
13	High	High	Low	Low
14	High	High	Low	High
15	High	High	High	Low
16	High	High	High	High

Table 6.6 Factor Level Combinations

Randomized Table of Runs				
Run	Group 1 Demand Seed=3141	Group 2 Demand Seed=3263	Group 3 Demand Seed=2784	Group 4 Demand Seed=4135
1	6	5	4	16
2	13	12	13	14
3	2	6	5	1
4	5	13	13	11
5	6	8	12	1
6	3	14	2	14
7	7	15	3	13
8	15	14	6	12
9	16	15	4	3
10	14	14	11	13
11	16	14	13	16
12	14	5	3	7
13	2	2	9	5
14	16	14	3	15
15	9	6	5	5
16	15	13	6	6

Table 6.7 Randomized Table of Runs

6.5 Results and Analysis

For data collection, the GAMS model is executed 64 times to run all the factor level combinations with 4 repeats. The expect profit is documented in table 6.8. The results are entered into a SAS program. Source file is available in Appendix B. The SAS output file is shown in Appendix C. The output file includes the ANOVA procedure for all specified factor interactions and the tukey's analysis.

.

Factor Level Combinations - Maximizing Expected Profit								
flc	Demand (D)	Capacity C	Root Cause Identification Probability (P)	DPPM (Q)	Seed=2784 Demand Low=4,559,953 Hi=5,559,953	Seed=3141 Demand Low=4,559,125 Hi=5,559,125	Seed=3263 Demand Low=4,559,581 Hi=5,559,581	Seed=4135 Demand Low=4,559,554 Hi=5,559,554
1	Low	Low	Low	Low	$1,114,835.05	$1,114,805.95	$1,114,804.32	$1,114,811.60
2	Low	Low	Low	High	$1,268,269.28	$1,312,464.55	$1,308,245.80	$1,312,464.55
3	Low	Low	High	Low	$1,702,390.06	$1,702,377.45	$1,702,372.56	$1,698,738.41
4	Low	Low	High	High	$1,792,450.07	$1,753,019.27	$1,796,143.27	$1,787,705.78
5	Low	High	Low	Low	$3,773,818.06	$3,792,644.98	$4,085,554.09	$3,746,536.49
6	Low	High	Low	High	$4,408,674.93	$4,408,674.93	$4,408,674.93	$4,408,674.93
7	Low	High	High	Low	$4,325,741.57	$4,325,586.60	$4,324,728.83	$4,315,829.92
8	Low	High	High	High	$4,413,078.16	$4,413,078.16	$4,413,078.16	$4,413,078.16
9	High	Low	Low	Low	$1,106,120.07	$1,106,058.98	$1,106,496.07	$1,106,442.68
10	High	Low	Low	High	$1,305,109.82	$1,305,151.23	$1,305,758.71	$1,305,190.04
11	High	Low	High	Low	$1,681,193.11	$1,680,967.27	$2,098,063.64	$1,682,465.23
12	High	Low	High	High	$1,779,652.98	$1,778,835.81	$1,780,961.75	$1,779,813.42
13	High	High	Low	Low	$4,003,766.76	$4,003,768.59	$4,003,750.81	$4,003,768.59
14	High	High	Low	High	$4,408,674.93	$4,408,674.93	$4,408,674.93	$4,402,382.70
15	High	High	High	Low	$4,311,639.40	$4,311,639.40	$4,311,639.40	$4,311,639.40
16	High	High	High	High	$4,394,201.46	$4,413,078.16	$4,394,201.46	$4,375,430.33

Table 6.8 Expected Profit

Based on the ANOVA analysis in table 6.9, demand has no significant effect on the expected profit. However, there is evidence that the daily fab capacity has the most statistical significant effect followed by the root-cause probability and DPPM level.

The high statistical significance of the capacity design factor is consistent with the case study premise. Capacity is the number one driving factor for the case study innovative grading scheme to allow company X to ship products despite the less than needed output of desired quality products. Has there been enough capacity to produce all required products by company Y, company X would have simply increased the wafer lots start to allow for increased output to compensate for the higher yield loss due to the quality excursion.

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	15	1.28E+14	8.50E+12	1954.75	<.0001
Error	48	2.08788E+11	4349754551		
Corrected Total	63	1.28E+14			
Source	**DF**	**Anova SS**	**Mean Square**	**F Value**	**Pr > F**
DEMAND	1	10046583875	10046583875	2.31	0.1351
CAPACITY	1	1.24E+14	1.24E+14	28415.3	<.0001
DEMAND*CAPACITY	1	490044057.2	490044057.2	0.11	0.7386
ROOTPROB	1	2.24E+12	2.24E+12	515.64	<.0001
DEMAND*ROOTPROB	1	2374242688	2374242688	0.55	0.4636
CAPACITY*ROOTPROB	1	5.25474E+11	5.25474E+11	120.81	<.0001
DEMAND*CAPACI*ROOTPR	1	17958321290	17958321290	4.13	0.0477
DPPM	1	6.35722E+11	6.35722E+11	146.15	<.0001
DEMAND*DPPM	1	13977911118	13977911118	3.21	0.0793
CAPACITY*DPPM	1	1.14074E+11	1.14074E+11	26.23	<.0001
DEMAND*CAPACITY*DPPM	1	2023929254	2023929254	0.47	0.4984
ROOTPROB*DPPM	1	3.02085E+11	3.02085E+11	69.45	<.0001
DEMAND*ROOTPROB*DPPM	1	575709096.7	575709096.7	0.13	0.7176
CAPACIT*ROOTPRO*DPPM	1	57627283430	57627283430	13.25	0.0007
DEMA*CAPA*ROOTP*DPPM	1	15422267844	15422267844	3.55	0.0658

Table 6.9 ANOVA Table

The root-cause probability is a distant-second most significant design factor after capacity. With capacity being the most significant design factor, it is reasonable to expect that identifying the root-cause-resolution, resulting in restoring the supply-chain to its normal operation, to have an impact on the expect profit.

The DPPM factor is not statistically significant. This is an indication that although company Y is in great need for products, quality remains a priority. The DPPM factor would have been a lot more statistically significant, had company Y been willing to compromise quality and ship material with a high expected DPPM.

ANOVA analysis shows that the demand is not statistically significant factor. Although, this may have come against expectation, it does indicate that the demand of company Y is reasonable and would have been easily satisfied if it is not for the quality excursion that compromised the production output plan.

The turkey HSD test is also performed. Tukey's test results are shown in Appendix C. Results indicate that the factors interaction between capacity and all other design factors produce means that are significantly different. While the interaction of all other factors, excluding capacity, with each other produce means that are not significantly different. This result is consistent with the capacity factor being the most statistically significant factor impacting the expected profit.

CHAPTER 7

SUMMARY AND CONCLUSIONS

7.1 Contributions

Integrated process design, planning and stochastic system optimization model provides a novel methodology for integrating process design and planning with system optimization to allow the users to make process optimization decisions while considering down stream consequences. The model is able to provide guidance in complex and dynamic environments about how much material to start processing in order to achieve the desired objectives. The model's strength is that while it assists in making the usual system optimization decisions, it also designs the process along the way looking for the optimal solution given a set of requirements and constraints. In addition, the model's flexibility illustrated in the dynamic objective function allows for easy customization for the intended user's situation

The case study illustrates the model's effectiveness in determining the optimal planning schedule and production flow to minimize the number of units shipped below customer's demand and maximizing profit. The statistical analysis results validates the model and confirms that capacity is the most statistically significant design factor. This result is consistent with the issue that case study is attempting to resolve. Also the model confirms company Y continue to maintain a high quality level despite the pressure to increase throughput which is consistent with statistical results showing the DPPM design factor not having a significant affect on the expect profit.

The quality grading technique presented in chapter 4 provides guidance on how to define quality grades. Quality grades implementation illustrates the importance of quality such that companies continue to enforce quality requirements in the most dire circumstances.

7.2 Future Research Topics

The model developed in this praxis and the computational testing can serve as the basis for future research. This model is most effective for straight line production flow. The mathematical formulation works well in manufacturing environment having straight line flow similar to semiconductor manufacturing. The A broader model can be developed for multi-line production flow such as the automotive manufacturing of different vehicles' makes and models and the clothing manufacturing of different garments with different fabric and sizes.

The praxis introduced and documented a quality grading methodology used in the case study in chapter 4. Future research can include developing a quality grading mathematical formulation and model and incorporate it with process design, planning and systems optimization model.

APPENDIX A

GAMS CODE

Listed below is the complete GAMS code used for the model's implementation. The code can be copied into a GAMS software and run by selecting the CPLEX solver and MIP procedure. The default factor level combination is 'low' for all factors. The levels are easily varied in the "DOE Design Factors (Low=0, Hi=1)" section. Stochastic implementation is assumed. As discussed in section 6.1, deterministic implementation can be achieved by removing the uncertainty in the root-cause-resolution scenarios. This can be accomplished by setting the probabilities of the "Partial" root-cause-resolution scenario to 0 and the scenarios "None" and "Full" to mutually exclusive probabilities zero or one.

CODE STARTS BELOW THE LINE

```
$OFFSYMLIST OFFSYMXREF OFFUELLIST
option limrow=3, limcol=3, sysout=off;
option iterlim = 9999999;
option optcr = 0.1;
* SEEDS TO RESET  PSEUDO RANDOM NUMBER GENERATOR FOR
CUSTOMER DEMAND
option seed = 2784;
*option seed = 3141;
*option seed = 3263;
*option seed = 4135;
```

```
Sets
I        Grades based on Defect Severity  /P,W,X,Y,R/
Y         Test Yield                    /MPYLD, FT1YLD, BIYLD, FT2YLD/
D         Device Hours of Operation       /24,1000,10000,100000/
R         Root Cause Resolution           /None, Partial, Full/
T         Time Periods in Days            /D1*D160/
C         Cycle Time in Days              /FT1CT, BICT, FT2CT/
S        Device Components Cost          /FabCost, ProbeCost, SCRAP, AssCost,
                                   FT1Cost, BICostPerHR, FT2Cost,
                                       PACKCost, SHIPCost/
BIHRS     Burn-in Hours                  /0hrs, 5hrs, 24hrs/;

alias (t, tt);

SCALARS
*         ============DOE Design Factors (Low=0, Hi=1)
CustomerDemand       Weekly Customer Demand                         /0/
Capacity             FAB Capacity                            /0/
RootCauseProbability Probability of Full-Partial-None Root Cause Resolution  /0/
DPPMQualityTarget    DPPM Quality target at a specific hours of operations   /0/
;

* ======Projected Daily Customer Demand (D) - DOE Design Factor (Low,
Hi)=======
Parameter Baseline;
If (CustomerDemand = 1,
      Baseline = 150000;
else
      Baseline = 50000;
);

* =======Daily Capacity (C) - DOE Design Factor (Low, Hi)=======
Parameter FABCAP;
If (Capacity = 1,
      FABCAP = 120;
else
      FABCAP = 72;
);

*=======Probability of Full-Partial-None Root Cause Resolution (P)===========
*=======DOE Design Factor (Low, Hi)=======
Parameter PROB(R);
If (RootCauseProbability = 1,
        PROB('None') = 0.1;
        PROB('Partial') = 0.3;
        PROB('Full') = 0.6;
else
        PROB('None') = 0.6;
        PROB('Partial') = 0.1;
        PROB('Full') = 0.3;
);

*=======Quality target at a specific hours of operations (Q)=======
*=======DOE Design Factor (Low, Hi)=======
Parameter Qlty_Target(D);
If (DPPMQualityTarget = 1,
        Qlty_Target('100000') = 4000;
else
        Qlty_Target('100000') = 1500;
);

SCALARS
* Root Cause Identification and Resolution
RTDAYS       Days until Root Cause Identification and Resolution     /30/
DEFCT        Days in FAB until Operation inducing the Defects        /15/
* Capacity
Trend        Customer Demand Trend                                   /4000/
Stddev       Customer Demand Standard Deviation                      /300/
ASSCAP       Assembly Capacity in Units per Time Period t            /250000/
* Lead Time
MPHWLT       Multiprobe hardware order lead time in days             /28/
FTHWLT       Final Test hardware order lead time days                /28/
BIBLT        Burn-in boards order lead time in days                  /45/
* Hardware Cost
MPHWCOST     Multi Probe Hardware Cost in US Dollars                 /4000/
FTHWCOST     Final Test Hardware Cost in US Dollars                  /3000/
BIBCOST      Burn-in Board Cost in US Dollars                        /2500/
SOCKETS      Units per burn-in board                                 /50/
* Test Time
MPTT         Multi-Probe Program Test Time in Seconds                /2.00/
FTTT         Final Test Program Test Time in Seconds                 /2.25/
SECWK        Seconds per 7 day week                                  /604800/
SECDY        Seconds per day                                         /86400/
* Yield & Efficiency Factors
DIES         Number of Dies Per Wafers                               /1500/
PY           Fab Process Yield                                       /0.95/
AY           Assembly Yield                                          /0.99/
```

```
PackYield      Packing Yield                               /0.999/
MPEFF          Multi Probe Test Percent Efficiency (uptime)          /0.90/
FTEFF          Final Test Percent Efficiency (uptime)              /0.90/
* Cycle Time
FABCT          Fab Cycle Time in Days                            /56/
MPCT           Multi Probe Test Cycle Time for one 24-wafer lot in Days  /1/
ACT            Assembly Cycle Time in day                        /3/
PACCT          Packing Cycle Time in days                         /1/;


TABLE     GRDSPLT(I,R)     Grades Split Percentage per Grade I and Scenario R
       None    Partial    Full
P       0.53    0.80       0.98
W        0.11    0.12       0.00
X       0.11    0.05       0.00
Y       0.06    0.00       0.00
R       0.06    0.00       0.00;

PARAMETER      CUSTDMD(T)      Customer Demands Per Time Period T;
       CUSTDMD(T) $ (mod(ord(t),7) = 0) = Floor (normal(baseline +
                                  (trend * ord(T)), stddev));

TABLE   CycleTime(C,I)  Cycle Time in Days per Grade I
              P      W      X      Y      R
FT1CT          1      1      1      1      1;

TABLE   CycleTime_BI(C,BIHRS,I)  Cycle Time in Days
              P      W      X      Y      R
BICT.0hrs      0      0      0      0      0
BICT.5hrs      1      1      1      1      2
BICT.24hrs     2      2      2      2      2
FT2CT.0hrs     0      0      0      0      0
FT2CT.5hrs     1      1      1      1      1
FT2CT.24hrs    1      1      1      1      1;

Parameter   BIHOURS(BIHRS)   BI Cycle Time in Hours (used for cost calculations)
 /0hrs 0, 5hrs 5, 24hrs 24/;

Parameter       TotalCT         Total Cycle Time;
       TotalCT = FABCT + MPCT + ACT +
                   smin(i,CycleTime('FT1CT',i)) +
                   smin((BIHRS,i),CycleTime_BI('BICT',BIHRS,i)) +
                   smin(i,CycleTime('FT2CT',i)) + PACCT;
```

```
TABLE   YIELD(Y,I)     Test Yield Percent for Multi-Probe and Final-Test1
              P       W       X       Y       R
MPYLD             0.81    0.78    0.63    0.63    0.61
FT1YLD            0.99    0.99    0.99    0.99    0.98;

TABLE   YIELD_BI(Y,BIHRS,I)   Test Yield Percent for Final Test 2
              P       W       X       Y       R
BIYLD.0hrs         1.00    1.00    1.00    1.00    1.00
BIYLD.5hrs         0.99    0.99    0.99    0.99    0.99
BIYLD.24hrs         0.98    0.98    0.98    0.98    0.98
FT2YLD.0hrs         1.00    1.00    1.00    1.00    1.00
FT2YLD.5hrs         0.99    0.99    0.99    0.99    0.99
FT2YLD.24hrs        0.998   0.998   0.998   0.998   0.998;

TABLE   DPPM(I,BIHRS,D) Output DPPM for Each Grade & BIHRS per Hrs of
Operations
            24      1000    10000   100000
P.0hrs       450     490     660     705
P.5hrs       3       3       4       5
P.24hrs       1       1       1       1
W.0hrs        1150    1525    1700    1830
W.5hrs        8       10      11      12
W.24hrs       2       2       2       3
X.0hrs       3000    63000   120000  166500
X.5hrs       20      420     800     1110
X.24hrs       4       90      171     238
Y.0hrs       5000    101250  188750  262500
Y.5hrs       40      810     1510    2100
Y.24hrs       7       145     270     375
R.0hrs       14000   525000  1484000 2408000
R.5hrs       93      3500    9893    16053
R.24hrs       20      750     2120    3440;

Parameter  UnitPrice(I)       Unit Selling Price per Grade in US Dollars
/P 1.6, W 1.6, X 1.6, Y 1.6, R 1.6/;

Parameter  GradeMultiplier(I)       Grade Multiplier to encourage prime grades
/P 1.0, W 1.01, X 1.02, Y 1.03, R 1.04/;

Parameter  DeviceCost(S)               Device Cost Per Operation
/FabCost 0.55, ProbeCost 0.05, AssCost 0.4, FT1Cost 0.04,
       BICostPerHR 0.02, FT2Cost 0.04, PACKCost 0.001, SHIPCost 0.001/;

Parameter  ScrapCost(I)          Scrap Cost for each grade I
```

```
/P 0.0, W 0.0, X 0.0, Y 0.0, R 0.0/;

Parameter      TDC(BIHRS)       Total Device Cost;
               TDC(BIHRS) = DeviceCost('FabCost') + DeviceCost('ProbeCost')
                      + DeviceCost('AssCost')+ DeviceCost('FT1Cost')
                      + DeviceCost('BICostPerHR') * BIHOURS(BIHRS)
                      + DeviceCost('FT2Cost')$(BIHOURS(BIHRS) gt 0)
                      + DeviceCost('PACKCost')
                      + DeviceCost('SHIPCost');

Parameter       POST_Fab   Cumulative Cost at Post Fab Inventory;
               POST_Fab = DeviceCost('FabCost');

Parameter      POST_Probe   Cumulative Cost at Post Multi-Probe Test Inventory;
              POST_Probe = DeviceCost('FabCost') + DeviceCost('ProbeCost');

Parameter       POST_Ass   Cumulative Cost at Post Assembly Inventory;
               POST_Ass = DeviceCost('FabCost') + DeviceCost('ProbeCost')
                      + DeviceCost('AssCost');

Parameter       POST_FT1   Cumulative Cost at Post BI Final Test1 Inventory;
               POST_FT1 = DeviceCost('FabCost') + DeviceCost('ProbeCost')
                      + DeviceCost('AssCost')+ DeviceCost('FT1Cost');

Parameter  POST_BI(I,BIHRS)  Cumulative Cost at Post BI Inventory;
        POST_BI(I,BIHRS) = DeviceCost('FabCost') + DeviceCost('ProbeCost')
                      + DeviceCost('AssCost')+ DeviceCost('FT1Cost')
                      + DeviceCost('BICostPerHR') * BIHOURS(BIHRS);

Parameter  POST_FT2(I,BIHRS) Cumulative Cost at Post BI Final Test2 Inventory;
        POST_FT2(I,BIHRS) = DeviceCost('FabCost') + DeviceCost('ProbeCost')
                      + DeviceCost('AssCost')+ DeviceCost('FT1Cost')
                      + DeviceCost('BICostPerHR') * BIHOURS(BIHRS)
                      + DeviceCost('FT2Cost')$(BIHOURS(BIHRS) gt 0);

Parameter  POST_PACK(I,BIHRS) Cumulative Cost at Post PACK Inventory;
        POST_PACK(I,BIHRS) = DeviceCost('FabCost') + DeviceCost('ProbeCost')
                      + DeviceCost('AssCost')+ DeviceCost('FT1Cost')
                      + DeviceCost('BICostPerHR') * BIHOURS(BIHRS)
                      + DeviceCost('FT2Cost')$(BIHOURS(BIHRS) gt 0)
                      + DeviceCost('PACKCost');

INTEGER VARIABLES
    FABin(T,R)              Wafers STARTED in the fab in time period T
```

```
      BIB_ordered(T,R)          Qty of Burn-in boards ordered
      MPHW_ordered(T,R)         Qty of Multi-Probe boards ordered
      FTHW_ordered(T,R)         Qty of Final-Test boards ordered;

POSITIVE VARIABLES
* Demand
      Demand(T)                 Customer Demands per time period T
      TotalDemand               Total Customer Demand over all time periods
* Fab Operation
      FABout(T,R)               Wafers completed fab processing in time period T per
scenario R
      iPOST_FAB(T,R)            Wafers at post fab INVENTORY awaiting probe in time
period T per scenario R
* Multi-Probe Hardware and Tester
      MPHW_received(T,R)        Qty of Multi-Probe boards received in time period T per
scenario R
      MPHW_Available(T,R)       Multi-probe boards available at the start of time T per
scenario R
      MPTSTR(R)                 Testers required for multiprobe operation per scenario R
      MPHWCostB4RTDAYS(T)       MP Hardware Cost for boards ordered before
RTDAYS in time period T
      TotalMPHWCostB4RTDAYS     Total MP Hardware Cost for boards ordered
before RTDAYS
      eMPHWCostAfterRTDAYS(T)   Expected MP Hardware Cost for boards ordered
After RTDAYS per time period T
      eTotalMPHWCostAfterRTDAYS  Expected MP Test Hardware Cost for boards
ordered After RTDAYS
* Multi-Probe Operation
      PROBECAP(T,R)             Multi-Probe Testers Capacity in time period T per
scenario R
      PROBEin(T,R)              Wafers input to probe test in time period T per scenario R
      PROBEout(T,I,R)           Dies completed probe test per grade I in time period T per
scenario R
      iPOST_PROBE(T,I,R)        Dies at post probe test INVENTORY awaiting
assembly start per grade I in time period T & scenario R
* Assembly Operation
      ASSin(T,I,R)              Dies started per grade I in time period T per scenario R
      ASSout(T,I,R)             Units completed assembly per grade I in time period T per
scenario R
      iPOST_ASS(T,I,R)          Units at post assembly INVENTORY awaiting final test1
per grade I in time period T per scenario R
* Final-Test Hardware and Testers
      FTHW_received(T,R)        Qty of FT boards received in time period T per scenario
R
```

FTHW_Available(T,R)   Final-test boards available at the start of time t per scenario R

FTTSTR(R)   Testers required for final test operation per scenario R

FTHWCostB4RTDAYS(T)   Final Test Hardware Cost for boards ordered before RTDAYS in time period T

TotalFTHWCostB4RTDAYS   Total Final Test Hardware Cost for boards ordered before RTDAYS

eFTHWCostAfterRTDAYS(T)   Expected Final Test Hardware Cost for boards ordered After RTDAYS in time period T

eTotalFTHWCostAfterRTDAYS Expected Total Final Test Hardware Cost for boards ordered After RTDAYS

* Final-Test1 Operations

FTCAP(T,R)   Final-test 1 and 2 testers capacity in time period T per scenario R

FT1in(T,I,R)   Units input to final-test1 per grade I in time period T per scenario R

FT1out(T,I,R)   Units completed final-test1 per grade I in time period T per scenario R

iPOST_FT1(T,I,R)   Units at post final test 1 INVENTORY awaiting burn-in per grade I in time period T per scenario R

* Burn-in Boards

BIB_received(t,R)   Qty of BI boards received in time period T per scenario R

BIB_Available(t,R)   Final-test boards available at the start of time T per scenario R

BIBoard(R)   Burn-in Boards required for BI operation per scenario R

BIBCostB4RTDAYS(t)   Burn-in Boards Cost for boards ordered before RTDAYS in time period T

TotalBIBCostB4RTDAYS   Total Burn-in Boards Cost for boards ordered before RTDAYS

eBIBCostAfterRTDAYS(t)   Expected Burn-in Boards Cost for boards ordered After RTDAYS in time period T

eTotalBIBCostAfterRTDAYS   Expected Total Burn-in Boards Cost for boards ordered After RTDAYS

* Burn-in Operation

BICAP(T,R)   Burn-in boards capacity per time period T per scenario R

BIin(T,I,BIHRS,R)   Units input to burn-in per grade I in time period T per scenario R and BIHRS

BIout(T,I,BIHRS,R)   Units completed burn-in per grade I in time period T per scenario R and BIHRS

iPOST_BI(T,I,BIHRS,R)   Units at post burn-in INVENTORY awaiting final test 2 per grade I in time period T per scenario R and BIHRS

* Final-Test2 Operations

FT2in(T,I,BIHRS,R)   Units input to final-test2 per grade I in time period T per scenario R

```
     FT2out(T,I,BIHRS,R)       Units completed final-test2 per grade I in time period T
per scenario R
     iPOST_FT2(T,I,BIHRS,R)    Units at post final-test2 INVENTORY awaiting pack
per grade I in time period T per scenario R
* Pack Operation
     PACKin(T,I,BIHRS,R)       Total units input to pack per grade I in time period T
per scenario R and BIHRS
     PACKinBI(T,I,BIHRS,R)     Units input to pack per grade I in time period T per
scenario R and BIHRS
     PACKout(T,I,BIHRS,R)      Units completed pack per grade I in time period T per
scenario R and BIHRS
     iPOST_PACK(T,I,BIHRS,R)   Units at post pack INVENTORY awaiting SHIP per
grade I in time period T per scenario R and BIHRS
* Ship
     SHIP(T,I,BIHRS,R)         Units SHIP per grade I in time period T per scenario R
and BIHRS
     overship(T,R)             Units shipped above target in period T per scenario R
     undership(T,R)            Units shipped below target in period T per scenario R
* Finished Goods
     FinishedGoods(I,BIHRS,R)          Finshed Goods per grade I and scenario R and
BIHRS
     FinishedGoodsScenario(R)          Finshed Goods per Scenario
     FinishedGoodsCost(I,BIHRS,R)      Finished Goods Cost per grade I and scenario R
and BIHRS
     TotalFinishedGoodsCost            Total Finished Goods Cost for all Grade
     ExpectedTotalFinishedGoodsCost    Expected Total Finished Goods Cost for all
Grade
     TotalFinishedGoodsCostScenario(R)  Total Finished Goods Cost for all Grade per
scenario R
     eTotalFinishedGoodsCostScenario(R) Total Expected Cost for all Grade
     TotalFinishedGoodsCostScenarGM(R)  Finshed Goods Cost per Scenario with
Grade Multiplier
     eTotalFinishedGoodsCostGM         Expected Finished Goods Total cost of all
shipments with grade multiplier
* SCRAP
     SCRAP(T,I,R)              Scrap units per grade I in time period T per scenario R
     TOTALSCRAP(I,R)           Total SCRAP per grade I and scenario R
     TOTALSCRAPScenario(R)     Total SCRAP per scenario R
     eTOTALSCRAPScenario(R)    Expected Total SCRAP per scenario R
     eTOTALSCRAP               Expected Total SCRAP
* DPPM
     outDPPM(I,BIHRS,R)             Total DPPM Cost per grade I and BIHRS and
scenario R
     TotaloutDPPMScenario(R)        Total DPPM Cost for all Grades per scenario R
     TotaloutDPPM                   Total DPPM Cost for all Grades
     ExpectedTotaloutDPPMScenario(R) Total DPPM Cost for all Grades per scenario R
     ExpectedTotaloutDPPM            Total DPPM Cost for all Grades
* Process Cost
     ProcessCost(I,BIHRS,R)         Process Cost for per grade I and BIHRS and scenario
R
     TotalProcessCostScenario(R)    Total Process Cost per scenario R
     TotalProcessCost               Total Process Cost
     eTotalProcessCostScenario(R)   Total Expected Process Cost per scenario R
     eTotalProcessCost              Total Expected Process Cost
* Inventory Cost
     InventoryCost(I,BIHRS,R)       Inventory Cost for per grade I and BIHRS and
scenario R
     TotalInventoryCostScenario(R)  Total Inventory Cost for per scenario R
     eTotalInventoryCostScenario(R) Total Expected Inventory Cost for per scenario R
     eTotalInventoryCost            Total Expected Inventory Cost
* Revenue
     Revenue(I,BIHRS,R)          Revenue per grade I and BIHRS and scenario R
     ScenarioRevenue(R)          Revenue per scenario R
     TotalRevenue                Total Revenue for all grades
     ExpectedScenarioRevenue(R) Revenue per flow per scenario R
     ExpectedTotalRevenue        Total Revenue for all grades
;

Binary Variable
     BI_time_pick(I,BIHRS,R)   Only one BIHRS per grade I;

FREE VARIABLE
     Z1                      Undership defined as priority 1
     Z2                      Effective Profit defined as priority 2
     TotalProfitScenario(R)    Total Profit per Scenario
     eTotalProfit              Total Expected Profit
     Effective_Profit          Expected Effective Profit including BIBs and Test HW Cost;

EQUATIONS
* Demand
     Customer_Demand(T)       Customer Demand per time period T
     Total_Demand             Total Customer Demand
* Fab Operation
     FABinput1(T,R)      Fab wafers input per time period T & scenario R for "NONE"
scenario = "Partial" scenario
     FABinput2(T,R)      Fab wafers input per time period T & scenario R for "NONE"
scenario = "Full" scenario
     FABoutput(T,R)      Fab wafers output per time period T & scenario R
```

```
    POSTFAB(T,R)        Post fab wafers inventory per time period T & scenario R
* Multi-Probe Hardware and Tester
    MPTSTRS(T,R)        Calculates Multi-probe testers required per time period T &
scenario R
    MPHW_orders(T,R)    Calculate Multi-probe Hardware Ordered within leadtime
    MPHW_orders1(T,R)   Multi-probe Hardware Ordered per time period T &
scenario R for "NONE" scenario = "Partial" scenario
    MPHW_orders2(T,R)   Multi-probe Hardware Ordered per time period T &
scenario R for "NONE" scenario = "Full" scenario
    MPHW_MPTSTR(R)      Ensures that Multi-probe hardware available equals
testers available per scenario R
    MPHWCapacity(T,R)   Multi-probe Hardware Capacity at time t per scenario R
    MPHW_Cost_B4_RTDAYS(T)      Multi-probe Hardware Cost for boards ordered
before RTDAYS per time period T
    Total_MPHW_Cost_B4_RTDAYS    Total Multi-probe Hardware Cost for boards
ordered before RTDAYS
    eMPHW_Cost_After_RTDAYS(T)   Expected Multi-probe Hardware Cost for
boards ordered After RTDAYS per time period T
    eTotal_MPHW_Cost_After_RTDAYS Expected Total Multi-probe Hardware Cost
for boards ordered After RTDAYS
* Multi-Probe Operation
    PRBCAPACITY(T,R)    Multi-probe Capacity per time period T & scenario R
    PROBEoutput(T,I,R)  Graded dies (units before package assembly) output after
multi-probe testing per time period T & scenario R
    POSTPRB(T,I,R)      Post multi-probe test graded die inventory awaiting assembly
per time period T and senarion R
* Assembly Operation
    ASSCAPACITY(T,R)    Assembly capacity per time period T and scenario R
    ASSoutput(T,I,R)    Units output after assembly per grade and time periond T and
scenario R
    POSTASS(T,I,R)      Post Assembly Units Inventory awaiting final-test 1 per grade
and time periond T and scenario R
* Final-Test Hardware and Testers
    FTTSTRS(T,R)                Calculate final test capacity per time periond T and
scenario R
    FTHW_orders(T,R)            Calculate FT Hardware Deployment within leadtime
per time periond T and scenario R
    FTHW_orders1(T,R)           Final-test Hardware Ordered per time period T &
scenario R for "NONE" scenario = "Partial" scenario
    FTHW_orders2(T,R)           Final-test Hardware Ordered per time period T &
scenario R for "NONE" scenario = "Full" scenario
    FTHW_FTTSTR(R)              Ensures that FT hardware available equals testers
available per scenario R
    FTHWCapacity(T,R)           Final-test Hardware Capacity at time T per scenario R
    FTHW_Cost_B4_RTDAYS(t)       Final-test Hardware Cost for boards ordered
before RTDAYS per time period T
    Total_FTHW_Cost_B4_RTDAYS    Total Final-test Hardware Cost for boards
ordered before RTDAYS
    eFTHW_Cost_After_RTDAYS(t)   Expected Final Test Hardware Cost for boards
ordered After RTDAYS per time period T
    eTotal_FTHW_Cost_After_RTDAYS Expected Total Final Test Hardware Cost for
boards ordered After RTDAYS
* Final-Test1 Operations
    FTCAPACITY(T,R)              Final-test 1 and 2 capacity per time period T &
scenario R
    FT1output(T,I,R)             Units output after final-test 1 per time period T & grade I
and scenario R
    POSTFT1(T,I,R)               Post Final Test 1 Inventory per time period T & grade I
and scenario R
* Burn-in Boards
    BIBs(T,R)                    Calculate Burn-in Capacity based on available boards in
time period T per scenario R
    BIB_orders(T,R)              Calculate BI Boards Deployment within leadtime per time
period T & scenario R
    BIB_orders1(T,R)             BI Boards Ordered per time period T & scenario R for
"NONE" scenario = "Partial" scenario
    BIB_orders2(T,R)             BI Boards Ordered per time period T & scenario R for
"NONE" scenario = "Full" scenario
    BIHR_BIBoard(R)              Burn-in Boards required for BI operation per scenario R
    BIBCAPACITY(T,R)             Burn-in boards capacity per time period T and
scenario R
    BIB_Cost_B4_RTDAYS(T)        Burn-in Boards Cost for boards ordered before
RTDAYS per time period T
    Total_BIB_Cost_B4_RTDAYS     Total Burn-in Boards Cost for boards ordered
before RTDAYS
    eBIB_Cost_After_RTDAYS(T)    Expected Burn-in Boards Cost for boards ordered
After RTDAYS per time period T
    eTotal_BIB_Cost_After_RTDAYS  Expected Total Burn-in Boards Cost for boards
ordered After RTDAYS
* Burn-in Operation
    BICAPACITY(T,R)              Burn-in capacity based on utilitization per time period
T and scenario R
    BI_Duration(I,R)             Burn-in Duration to selection one BIHRS per grade in
scenario R
    BI_Selector(TT,I,BIHRS,R)    Ensures that one BIHRS is selected per grade I
    BIoutput(T,I,BIHRS,R)        Units output after burn-in in time period T per grade I
& BIHRS & scenario R
```

POSTBI(T,I,BIHRS,R) Post Burn-in Inventory awaiting final-test 2 in time period T per grade I & BIHRS & scenario R

* Final-Test2 Operations

FT2output(T,I,BIHRS,R) Units output after final-test 2 per time period T & grade I & BIHRS & scenario R

POSTFT2(t,I,BIHRS,R) Post final-Test 2 Inventory awaiting pack per time period T & grade I & BIHRS & scenario R

* Pack Operation

PACKoutput(T,I,BIHRS,R) Units output after pack in time period T per grade I & BIHRS & scenario R

* Ship

SHP(T,I,BIHRS,R) Shipped Quantity in time period T per grade I & BIHRS & scenario R

SHIPGOAL(T,R) Shipping goal for each time period with overship and undership per scenario R

* Finished Goods

Finished_Goods(I,BIHRS,R) Finshed Goods per grade & BIHRS and scenario R

Finished_Goods_Scenario(R) Finshed Goods per scenario R

Total_Finished_Goods_Cost_S_GM(R) Finshed Goods Cost per Scenario with Grade Multiplier

eTotal_Finished_Goods_Cost_GM Expected Total cost of finished goods with Grade Multiplier

Finished_Goods_Cost(I,BIHRS,R) Total Cost of all shipments for each grade I per BIHRS & scenario R

Total_Finished_Goods_Cost Total Cost of all shipments

eTotal_Finished_Goods_Cost Expected Total Cost of all shipments

Total_Finished_Goods_Cost_Scen(R) Total finished goods cost per scenario

eTotal_Finished_Goods_Cost_Sce(R) Total expected finished goods cost per scenario

* SCRAP

TOTAL_SCRAP(I,R) Total SCRAP per grade I and scenario R

TOTAL_SCRAP_Scenario(R) Total SCRAP per scenario R

eTOTAL_SCRAP_Scenario(R) Expected Total SCRAP per scenario R

eTOTAL_SCRAP Expected Total SCRAP

* DPPM

out_DPPM(i,BIHRS,R) Total DPPM Count per grade and BIHRS and scenario R

Total_outDPPM_Scenario(R) Total DPPM Count per scenario R

Total_outDPPM Total DPPM Count

Expected_Total_outDPPM_Scena(R) Total expected DPPM Count per scenario R

Expected_Total_outDPPM Total expected DPPM Count

* Process Cost

Process_Cost(i,BIHRS,R) Process Cost per grade and BIHRS & scenario R

```
      Total_Process_Cost_Scenario(R)  Total Process Cost per scenario R
      Total_Process_Cost            Total Process Cost
      eTotal_Process_Cost_Scenario(R) Total expected Process Cost per scenario R
      eTotal_Process_Cost           Total expected Process Cost
* Inventory Cost
      Inventory_Cost(i,BIHRS,R)        Inventory Cost for per grade & BIHRS and
scenario R
      Total_Inventory_Cost_Scenario(R)  Total Inventory Cost for per scenario R
      eTotal_Inventory_Cost_Scenario(R) Total expected Inventory Cost for per scenario
R
      eTotal_Inventory_Cost           Total expected Inventory Cost
* Revenue
      Revenues(i,BIHRS,R)         Revenue per grade and BIHRS and scenario R
      Scenario_Revenue(R)         Revenue per scenario R
      Total_Revenue               Total Revenue for all grades
      Expected_Scenario_Revenue(R) Revenue per scenario R
      Expected_Total_Revenue      Total Revenue
      Total_Profit_Scenario(R)    Total Profit per scenario R
      eTotal_Profit               Total Expected Profit
      calccost                    Calculate expected effective profit
      Priority1                   Undership defined as priority 1
      Priority2                   Expected Effective Profit defined as priority 2;

*********************************FORMULAS************************
***********
* Demand
Customer_Demand(t)..               Demand(t) =e= CUSTDMD(t);
Total_Demand..                    TotalDemand =e= sum(t, CUSTDMD(t));

* Fab Operation
FABinput1(t,R)$(ord(t) lt RTDAYS)..    FABin(t,'NONE') =e= FABin (t,'Partial');
FABinput2(t,R)$(ord(t) lt RTDAYS)..    FABin(t,'NONE') =e= FABin (t,'Full');
FABoutput(t,R)$(ord(t) gt FABCT)..     FABout(t,R) =e= FABin(t-FABCT,R)*PY;
POSTFAB(t,R)$(ord(t) ge FABCT)..       FABout(t,R) + iPOST_FAB(t-1,R) =e=
                              PROBEin(t,R) + iPOST_FAB(t,R);

* Multi-Probe Hardware and Tester
MPTSTRS(t,R)..                          PROBECAP(t,R) =e=
                                     (MPHW_Available(t,R)*
                                     (SECDY*MPEFF)/(MPTT*DIES));
MPHW_orders(t,R)..                        MPHW_received(t,R) =e=
                                     MPHW_ordered(t-MPHWLT,R);
MPHW_orders1(t,R)$(ord(t) lt RTDAYS)..          MPHW_ordered(t,'NONE') =e=
                                     MPHW_ordered (t,'Partial');
```

```
MPHW_orders2(t,R)$(ord(t) lt RTDAYS)..          MPHW_ordered(t,'NONE') =e=
                                       MPHW_ordered (t,'Full');
MPHW_MPTSTR(R)..                               MPTSTR(R) =e=
                                       sum(t,MPHW_received(t,R));
MPHWCapacity(t,R)..                         MPHW_Available(t,R)=e=
                                       MPHW_Available(t-1,R) +
                                       MPHW_Received(t,R);
MPHW_Cost_B4_RTDAYS(t)$(ord(t) lt RTDAYS)..     MPHWCostB4RTDAYS(t) =e=
                                       sum(R,MPHW_ordered(t,R)*MPHWCOST);
Total_MPHW_Cost_B4_RTDAYS..                     TotalMPHWCostB4RTDAYS =e=
                                       sum(t,MPHWCostB4RTDAYS(t));
eMPHW_Cost_After_RTDAYS(t)$(ord(t) ge RTDAYS)..
eMPHWCostAfterRTDAYS(t) =e=

sum(R,MPHW_ordered(t,R)*PROB(R)*MPHWCOST);
eTotal_MPHW_Cost_After_RTDAYS..                 eTotalMPHWCostAfterRTDAYS
=e=
                                       sum(t,eMPHWCostAfterRTDAYS(t));

* Multi-Probe Operation
PRBCAPACITY(t,R)$(ord(t) gt FABCT)..            SUM(tt$(ord(tt) le ord(t) and ord(tt)
                                       ge (ord(t)-MPCT+1)),PROBEin(t,R))
                                       =l= PROBECAP(t,R);
PROBEoutput(t,i,R)..                         PROBEout(t,i,R) =e= PROBEin(t-MPCT,R)*
                                       YIELD('MPYLD',I)*DIES*
                                       ((GRDSPLT(i,R)$(ord(t) ge
                                       (FABCT + RTDAYS - DEFCT)) +
                                       GRDSPLT(i,"None")$(ord(t) lt
                                       (FABCT + RTDAYS - DEFCT)))) ;
POSTPRB(t,i,R)$(ord(t) gt FABCT)..             PROBEout(t,i,R) + iPOST_PROBE(t-
1,i,R) =e=
                                       ASSin(t,i,R) + iPOST_PROBE(t,i,R) +
                                       SCRAP(t,i,R);

* Assembly Operation
ASSCAPACITY(tt,R)$(ord(tt) gt FABCT)..          SUM((i,t)$(ord(t) le ord(tt) and
                                       ord(t) ge (ord(tt)-ACT+1)),
                                       ASSin(t,i,R)) =l= ASSCAP;;
ASSoutput(t,i,R)$(ord(t) gt FABCT)..           ASSout(t,i,R) =e= ASSin(t-ACT,i,R)*AY;
POSTASS(t,i,R)$(ord(t) gt FABCT)..             ASSout(t,i,R) + iPOST_ASS(t-1,i,R) =e=
                                       FT1in(t,i,R) + iPOST_ASS(t,i,R);

* Final-Test Hardware and Testers
```

```
FTTSTRS(t,R)..                          FTCAP(t,R) =e=
((FTHW_Available(t,R)*SECDY*FTEFF)/FTTT);
FTHW_orders(t,R)..                       FTHW_received(t,R) =e= FTHW_ordered(t-
FTHWLT,R);
FTHW_orders1(t,R)$(ord(t) lt RTDAYS)..          FTHW_ordered(t,'NONE') =e=
FTHW_ordered (t,'Partial');
FTHW_orders2(t,R)$(ord(t) lt RTDAYS)..          FTHW_ordered(t,'NONE') =e=
FTHW_ordered (t,'Full');
FTHW_FTTSTR(R)..                          FTTSTR(R) =e= sum(t,FTHW_received(t,R));
FTHWCapacity(t,R)..                       FTHW_Available(t,R) =e= FTHW_Available(t-
1,R) +
                                       FTHW_Received(t,R);
FTHW_Cost_B4_RTDAYS(t)$(ord(t) lt RTDAYS)..     FTHWCostB4RTDAYS(t) =e=
                                       sum(R,FTHW_ordered(t,R)*FTHWCOST);
Total_FTHW_Cost_B4_RTDAYS..                    TotalFTHWCostB4RTDAYS =e=
                                       sum(t,FTHWCostB4RTDAYS(t));
eFTHW_Cost_After_RTDAYS(t)$(ord(t) ge RTDAYS)..  eFTHWCostAfterRTDAYS(t)
=e=

sum(R,FTHW_ordered(t,R)*PROB(R)*FTHWCOST);
eTotal_FTHW_Cost_After_RTDAYS..                 eTotalFTHWCostAfterRTDAYS =e=
                                       sum(t,eFTHWCostAfterRTDAYS(t));

* Final-Test1 Operations
FTCAPACITY(t,R)$(ord(t) gt FABCT)..             SUM((tt,i)$(ord(tt) le ord(t) and ord(tt)
                                       ge (ord(t)-CycleTime('FT1CT',i)+1)),FT1in(t,i,R)) +
                                       SUM((tt,i,BIHRS)$(ord(tt) le ord(t) and ord(tt)
                                       ge (ord(t)-CycleTime_BI('FT2CT',BIHRS,i)+1)),
                                       FT2in(t,i,BIHRS,R)) =l= FTCAP(t,R);

FT1output(t,i,R)$(ord(t) gt FABCT)..            FT1out(t,i,R) =e=
                                       FT1in(t-CycleTime('FT1CT',i),i,R)
                                       *YIELD('FT1YLD',I);
POSTFT1(t,i,R)$(ord(t) gt FABCT)..              FT1out(t,i,R) + iPOST_FT1(t-1,i,R) =e=
                                       sum(BIHRS,BIin(t,i,BIHRS,R)) + iPOST_FT1(t,i,R);

* Burn-in Boards
BIBs(t,R)..                            BICAP(t,R) =e= (BIB_Available(t,R)*SOCKETS);
BIB_orders(t,R)..                        BIB_received(t,R) =e= BIB_ordered(t-BIBLT,R);
BIB_orders1(t,R)$(ord(t) lt RTDAYS)..           BIB_ordered(t,'NONE') =e= BIB_ordered
(t,'Partial');
BIB_orders2(t,R)$(ord(t) lt RTDAYS)..           BIB_ordered(t,'NONE') =e= BIB_ordered
(t,'Full');
BIHR_BIBoard(R)..                         BIBoard(R) =e= sum(t,BIB_received(t,R));
```

```
BIBCapacity(t,R)..                          BIB_Available(t,R) =e= BIB_Available(t-1,R) +
                                        BIB_Received(t,R);
BIB_Cost_B4_RTDAYS(t)$(ord(t) lt RTDAYS)..      BIBCostB4RTDAYS(t) =e=
sum(R,BIB_ordered(t,R)*BIBCOST);
Total_BIB_Cost_B4_RTDAYS..                  TotalBIBCostB4RTDAYS =e=
sum(t,BIBCostB4RTDAYS(t));
eBIB_Cost_After_RTDAYS(t)$(ord(t) ge RTDAYS)..   eBIBCostAfterRTDAYS(t) =e=
sum(R,BIB_ordered(t,R)*PROB(R)*BIBCOST);
eTotal_BIB_Cost_After_RTDAYS..              eTotalBIBCostAfterRTDAYS =e=
sum(t,eBIBCostAfterRTDAYS(t));

* Burn-in Operation
BICAPACITY(t,R)$(ord(t) gt FABCT)..             SUM((tt,i,BIHRS)$(ord(tt) le ord(t) and
ord(tt)
                                        ge (ord(t)-CycleTime_BI('BICT',BIHRS,i)+1)),
                                        BIin(t,i,BIHRS,R)) =l= BICAP(t,R);
BI_Duration(i,R)..                          SUM(BIHRS, BI_time_pick(I,BIHRS,R)) =l= 1;
BI_Selector(tt,i,BIHRS,R)$(ord(tt) gt FABCT)..  BIin(tt,i,BIHRS,R) =l=
sum(t,CUSTDMD(T)) * BI_time_pick(I,BIHRS,R);
BIoutput(t,i,BIHRS,R)$((ord(t) gt FABCT))..     BIout(t,i,BIHRS,R) =e= BIin(t-
CycleTime_BI('BICT',BIHRS,i),
                                        i,BIHRS,R) * YIELD_BI('BIYLD',BIHRS,I);
POSTBI(t,i,BIHRS,R)$((ord(t) gt FABCT))..       BIout(t,i,BIHRS,R) + iPOST_BI(t-
1,i,BIHRS,R) =e=
                                        iPOST_BI(t,i,BIHRS,R) + FT2in(t,i,BIHRS,R);
* Final-Test2 Operations
FT2output(t,i,BIHRS,R)$(ord(t) gt FABCT)..      FT2out(t,i,BIHRS,R) =e=
                                        FT2in(t-CycleTime_BI('FT2CT',BIHRS,i),
                                        i,BIHRS,R)*YIELD_BI('FT2YLD',BIHRS,I);
POSTFT2(t,i,BIHRS,R)$(ord(t) gt FABCT)..        FT2out(t,i,BIHRS,R) + iPOST_FT2(t-
1,i,BIHRS,R) =e=
                                        PACKin(t,i,BIHRS,R) + iPOST_FT2(t,i,BIHRS,R);

* Pack Operation
PACKoutput(t,i,BIHRS,R)$(ord(t) gt FABCT)..     PACKout(t,i,BIHRS,R) =e=
                                        PACKin(t-PACCT,i,BIHRS,R) * PackYield;

* Ship
SHP(t,i,BIHRS,R)$(ord(t) gt FABCT)..    PACKout(t,i,BIHRS,R) + iPOST_PACK(t-
1,i,BIHRS,R) =e=
                                    SHIP(t,i,BIHRS,R)+ iPOST_PACK(t,i,BIHRS,R);
SHIPGOAL(t,R)$(ord(t) gt FABCT)..       SUM((i,BIHRS), SHIP(t,i,BIHRS,R))=e=
                                    CUSTDMD(t) + overship(t,R) - overship(t-1,R)-
                                    undership(t,R)  + undership(t-1,R);
```

```
* Finished Goods
Finished_Goods(i,BIHRS,R)..            FinishedGoods(i,BIHRS,R) =e=
                                  sum(t,ship(t,i,BIHRS,R));
Finished_Goods_Scenario(R)..           FinishedGoodsScenario(R) =e=
                                  sum((t,i,BIHRS),ship(t,i,BIHRS,R));
Total_Finished_Goods_Cost_S_GM(R)..      TotalFinishedGoodsCostScenarGM(R) =e=
                                  sum((BIHRS,i),FinishedGoodsCost(i,BIHRS,R)*
                                  GradeMultiplier(i));
eTotal_Finished_Goods_Cost_GM..         eTotalFinishedGoodsCostGM =e=
                                  sum(R,TotalFinishedGoodsCostScenarGM(R)*PROB(R));
Finished_Goods_Cost(i,BIHRS,R)..        FinishedGoodsCost(i,BIHRS,R) =e=
                                  sum(t,ship(t,i,BIHRS,R)) * TDC(BIHRS);
Total_Finished_Goods_Cost..            TotalFinishedGoodsCost =e=
                                  sum((BIHRS,i,R),FinishedGoodsCost(i,BIHRS,R));
eTotal_Finished_Goods_Cost..           ExpectedTotalFinishedGoodsCost =e=
                                  sum(R,eTotalFinishedGoodsCostScenario(R));
Total_Finished_Goods_Cost_Scen(R)..      TotalFinishedGoodsCostScenario(R) =e=
                                  sum((BIHRS,i),FinishedGoodsCost(i,BIHRS,R));
eTotal_Finished_Goods_Cost_Sce(R)..      eTotalFinishedGoodsCostScenario(R) =e=
                                  TotalFinishedGoodsCostScenario(R) * PROB(R);

* SCRAP
TOTAL_SCRAP(i,R)..                   TOTALSCRAP(i,R) =e=
                                  sum(t,SCRAP(t,i,R));
TOTAL_SCRAP_Scenario(R)..              TOTALScrapScenario(R) =e=
                                  sum((t,i),SCRAP(t,i,R));
eTOTAL_SCRAP_Scenario(R)..             eTOTALScrapScenario(R) =e=
                                  TOTALScrapScenario(R) * PROB(R);
eTOTAL_SCRAP..                      eTOTALScrap =e=
                                  sum(R,eTOTALScrapScenario(R));

* DPPM
out_DPPM(i,BIHRS,R)..                outDPPM(i,BIHRS,R) =e=
                                  sum((t),ship(t,i,BIHRS,R)) *
DPPM(I,BIHRS,'100000')*0.000001;
Total_outDPPM_Scenario(R)..            TotaloutDPPMScenario(R) =e=
sum((BIHRS,i),outDPPM(I,BIHRS,R));
Total_outDPPM..                     TotaloutDPPM =e=
sum((BIHRS,i,R),outDPPM(I,BIHRS,R));
Expected_Total_outDPPM_Scena(R)..       ExpectedTotaloutDPPMScenario(R) =e=
TotaloutDPPMScenario(R) * PROB(R);
Expected_Total_outDPPM..               ExpectedTotaloutDPPM =e=
sum(R,ExpectedTotaloutDPPMScenario(R));
```

```
* Process Cost
Process_Cost(i,BIHRS,R)..             ProcessCost(i,BIHRS,R) =e= SUM(t,FABin(t,R) *
DeviceCost('FabCost'))
                               + SUM(t,PROBEin(t,R) * DeviceCost('ProbeCost'))
                               + SUM(t,ASSin(t,i,R) * DeviceCost('AssCost'))
                               + SUM(t,FT1in(t,i,R) * DeviceCost('FT1Cost'))
                               + SUM(t,BIin(t,i,BIHRS,R) * DeviceCost('BICostPerHR')
* BIHOURS(BIHRS))
                               + SUM(t,FT2in(t,i,BIHRS,R) *
DeviceCost('FT2Cost')$(BIHOURS(BIHRS) gt 0))
                               + SUM(t,PACKin(t,i,BIHRS,R) *
DeviceCost('PACKCost'))
                               + SUM(t,SHIP(t,i,BIHRS,R) * DeviceCost('SHIPCost'));
Total_Process_Cost_Scenario(R)..        TotalProcessCostScenario(R) =e=
                               SUM((i,BIHRS),ProcessCost(i,BIHRS,R));
Total_Process_Cost..                TotalProcessCost =e=
                               SUM(R,TotalProcessCostScenario(R));
eTotal_Process_Cost_Scenario(R)..       eTotalProcessCostScenario(R) =e=
                               TotalProcessCostScenario(R) * Prob(R);
eTotal_Process_Cost..               eTotalProcessCost =e=
                               SUM(R,eTotalProcessCostScenario(R));

* Inventory Cost
Inventory_Cost(i,BIHRS,R)..           InventoryCost(i,BIHRS,R) =e=
                               sum(t,iPOST_FAB(t,R)      * POST_FAB)+
                               sum(t,iPOST_PROBE(t,i,R) * POST_PROBE)+
                               sum(t,SCRAP(t,i,R) * SCRAPCOST(i))+
                               sum(t,iPOST_ASS(t,i,R)   * POST_ASS)+
                               sum(t,iPOST_FT1(t,i,R)   * POST_FT1)+
                               sum(t,iPOST_BI(t,i,BIHRS,R)    * POST_BI(I,BIHRS))+
                               sum(t,iPOST_FT2(t,i,BIHRS,R)   *
POST_FT2(I,BIHRS))+
                               sum(t,iPOST_PACK(t,i,BIHRS,R)  *
POST_PACK(I,BIHRS));
Total_Inventory_Cost_Scenario(R)..      TotalInventoryCostScenario(R) =e=
                               SUM((i,BIHRS),InventoryCost(i,BIHRS,R));
eTotal_Inventory_Cost_Scenario(R)..     eTotalInventoryCostScenario(R) =e=
                               TotalInventoryCostScenario(R) * PROB(R);
eTotal_Inventory_Cost..             eTotalInventoryCost =e=
                               SUM(R,eTotalInventoryCostScenario(R));

* Revenue
```

```
Revenues(i,BIHRS,R)..                  Revenue(i,BIHRS,R) =e=
sum((t),ship(t,i,BIHRS,R)*UnitPrice(i));
Scenario_Revenue(R)..                  ScenarioRevenue(R) =e=
sum((i,BIHRS),Revenue(i,BIHRS,R));
Total_Revenue..                        TotalRevenue =e=
sum((i,BIHRS,R),Revenue(i,BIHRS,R));
Expected_Scenario_Revenue(R)..         ExpectedScenarioRevenue(R) =e=
ScenarioRevenue(R) * PROB(R);
Expected_Total_Revenue..               ExpectedTotalRevenue =e=
sum(R,ExpectedScenarioRevenue(R));
Total_Profit_Scenario(R)..             TotalProfitScenario(R) =e=
                                    ScenarioRevenue(R)
                                    - TotalFinishedGoodsCostScenarGM(R)
                                    - TOTALScrapScenario(R);
eTotal_Profit..                      eTotalProfit =e=
                                    sum(R,TotalProfitScenario(R)*PROB(R))
                                    - TotalBIBCostB4RTDAYS
                                    - eTotalBIBCostAfterRTDAYS
                                    - TotalMPHWCostB4RTDAYS
                                    - eTotalMPHWCostAfterRTDAYS
                                    - TotalFTHWCostB4RTDAYS
                                    - eTotalFTHWCostAfterRTDAYS;
calccost..                         Effective_Profit =e= ExpectedTotalRevenue
                                    - TotalBIBCostB4RTDAYS
                                    - eTotalBIBCostAfterRTDAYS
                                    - TotalMPHWCostB4RTDAYS
                                    - eTotalMPHWCostAfterRTDAYS
                                    - TotalFTHWCostB4RTDAYS
                                    - eTotalFTHWCostAfterRTDAYS
                                    - eTotalFinishedGoodsCostGM
*                                     - eTotalProcessCost
*                                     - eTotalInventoryCost
;
Priority1..          Z1 =e= sum((R,t), undership(t,R) * PROB(R));
*Priority1..           Z1 =e= sum(R, undership('D160',R) * PROB(R));
Priority2..          Z2 =e= Effective_Profit;

FABin.UP(t,R) = FABCAP;
MPTSTR.up(R) = 10;
FTTSTR.up(R) = 10;
BIBoard.up(R) = 300;

CUSTDMD(T)$(ord(t) lt 70) =0;
CUSTDMD(T)$(ord(t) gt 133) =0;
```

```
FABin.up(t,R)$(ord(t) gt (card(t)-TotalCT)) =0;
FABout.up(T,R)$(ord(t) le FABCT)    =0;
iPOST_FAB.up(T,R)$(ord(t) le FABCT) =0;

PROBEin.up(T,R)$(ord(t) le FABCT)    =0;
PROBEout.up(T,I,R)$(ord(t) le FABCT) =0;
iPOST_PROBE.up(T,I,R)$(ord(t) le FABCT) =0;

SCRAP.up(t,i,R)$(ord(t) le FABCT) =0;

ASSin.up(t,i,R)$ord(i) = ASSCAP;
ASSin.up(T,I,R)$(ord(t) le FABCT)    =0;
ASSout.up(T,I,R)$(ord(t) le FABCT)  =0;

iPOST_ASS.up(T,I,R)$(ord(t) le FABCT) =0;

FT1in.up(T,I,R)$(ord(t) le FABCT)    =0;
FT1out.up(T,I,R)$(ord(t) le FABCT)   =0;
iPOST_FT1.up(T,I,R)$(ord(t) le FABCT) =0;

BI_time_pick.up(I,BIHRS,R)$(DPPM(I,BIHRS,'100000') gt Qlty_Target('100000')) =0;

$ontext

$offtext

BIin.up(T,I,BIHRS,R)$(ord(t) le FABCT)     =0;
BIout.up(T,I,BIHRS,R)$(ord(t) le FABCT)    =0;
iPOST_BI.up(T,I,BIHRS,R)$(ord(t) le FABCT) =0;

FT2in.up(T,I,BIHRS,R)$(ord(t) le FABCT)    =0;
FT2out.up(T,I,BIHRS,R)$(ord(t) le FABCT)   =0;
iPOST_FT2.up(T,I,BIHRS,R)$(ord(t) le FABCT) =0;

PACKin.up(T,I,BIHRS,R)$(ord(t) le FABCT)   =0;
PACKout.up(T,I,BIHRS,R)$(ord(t) le FABCT)  =0;
iPOST_PACK.up(T,I,BIHRS,R)$(ord(t) le FABCT) =0;
overship.up(t,R)$(ord(t) le TotalCT) = 0;
undership.up(t,R)$(ord(t) le TotalCT) =0;
ship.up(t,i,BIHRS,R)$(ord(t) le FABCT)   =0;


MODEL TESTPLAN /ALL/;
```

```
TESTPLAN.OPTFILE = 1;

    Solve TESTPLAN minimizing Z1 using MIP;
      Z1.up = Z1.l;
      display demand.l;
      display fabin.l, fabout.l, iPOST_fab.l;
      display TotalCT;
      display probein.l, probeout.l, iPOST_PROBE.l;
      display SCRAP.l;
      display assin.l, assout.l, iPOST_ASS.l;
      display ft1in.l, ft1out.l, iPOST_ft1.l;
      display BIin.l, BIout.l, iPOST_BI.l;
      display ft2in.l, ft2out.l, iPOST_ft2.l;
      display packin.l, packout.l, iPOST_PACK.l, ship.l;
      display FinishedGoods.l,FinishedGoodsScenario.l;
      display TOTALSCRAP.l,
TOTALSCRAPScenario.l,eTOTALSCRAPScenario.l,eTOTALSCRAP.l;
      display
FinishedGoodsCost.l,TotalFinishedGoodsCostScenario.l,TotalFinishedGoodsCost.l;
      display eTotalFinishedGoodsCostScenario.l,ExpectedTotalFinishedGoodsCost.l;
      display TotalFinishedGoodsCostScenarGM.l, eTotalFinishedGoodsCostGM.l;
      display outDPPM.l,TotaloutDPPMScenario.l,TotaloutDPPM.l;
      display ExpectedTotaloutDPPMScenario.l,ExpectedTotaloutDPPM.l;
      display
ProcessCost.l,TotalProcessCostScenario.l,eTotalProcessCostScenario.l,eTotalProcessCos
t.l;
      display
InventoryCost.l,TotalInventoryCostScenario.l,eTotalInventoryCostScenario.l,eTotalInven
toryCost.l;
      display
MPTSTR.l,MPHW_ordered.l,MPHW_received.l,MPHW_Available.l,PROBECAP.l;
      display TotalMPHWCostB4RTDAYS.l, eTotalMPHWCostAfterRTDAYS.l;
      display FTTSTR.l,FTHW_ordered.l,FTHW_received.l,
FTHW_Available.l,FTCAP.l;
      display TotalFTHWCostB4RTDAYS.l, eTotalFTHWCostAfterRTDAYS.l;
      display BIBoard.l,BIB_ordered.l,BIB_received.l,BIB_Available.l, BICAP.l;
      display TotalBIBCostB4RTDAYS.l, eTotalBIBCostAfterRTDAYS.l;
      display Revenue.l,ScenarioRevenue.l,TotalRevenue.l;
      display ExpectedScenarioRevenue.l,ExpectedTotalRevenue.l;
      display TotalProfitScenario.l, eTotalProfit.l;
      display overship.l, undership.l, Effective_Profit.l;
      display z1.l, z2.l;

file DOE_Output_Undership   /DOE_Output_Undership.txt/;
```

```
DOE_Output_Undership.pw = 1000;
DOE_Output_Undership.ap = 1;
DOE_Output_Undership.nj = 2;
*Columns header - Run once at the beginning
put DOE_Output_Undership  'Date', @10, 'Time', @20, 'Seed', @26,
'Customer_Demand', @43, 'FAB_Capacity', @57, 'Root_Cause_Probability', @81,
'DPPM_Quality_Target', @102, 'Undership', @116, 'Effective_Profit', @135,
'Total_Demand', @149, 'MP_Testers(NONE)', @167, 'MP_Testers(Partial)', @190,
'MP_Testers(Full)', @210, 'FT_Testers(NONE)', @228, 'FT_Testers(Partial)', @251,
'FT_Testers(Full)', @269, 'BIBoard(NONE)', @286, 'BIBoard(Partial)', @306,
'BIBoard(Full)', @323, 'FinishedGoods(NONE)', @344, 'FinishedGoods(Partial)', @368,
'FinishedGoods(Full)', @391, 'eTOTALSCRAP', @404, 'TotalFinishedGoodsCost',
@428, 'ExpectedTotalFinishedGoodsCost', @460, 'eTotalFinishedGoodsCostGM',
@487, 'TotaloutDPPM', @505, 'ExpectedTotaloutDPPM', @530, 'TotalProcessCost',
@551, 'eTotalProcessCost', @573, 'eTotalInventoryCost', @597, 'TotalRevenue', @613,
'ExpectedTotalRevenue', @660, 'eTotalProfit'/;
put DOE_Output_Undership  system.date, @10, system.time, @20, '2784', @26,
CustomerDemand, @43, Capacity, @57, RootCauseProbability, @81,
DPPMQualityTarget, @102, z1.l, @116, z2.l, @135, TotalDemand.l, @149,
MPTSTR.l('NONE'), @167, MPTSTR.l('Partial'), @190, MPTSTR.l('Full'), @210,
FTTSTR.l('NONE'), @228, FTTSTR.l('Partial'), @251, FTTSTR.l('Full'), @269,
BIBoard.l('NONE'), @286, BIBoard.l('Partial'), @306, BIBoard.l('Full'), @323,
FinishedGoodsScenario.L('NONE'), @344, FinishedGoodsScenario.L('Partial'), @368,
FinishedGoodsScenario.L('Full'),@391, eTOTALSCRAP.l, @404,
TotalFinishedGoodsCost.l, @428, ExpectedTotalFinishedGoodsCost.l, @460,
eTotalFinishedGoodsCostGM.l, @487, TotaloutDPPM.l, @505,
ExpectedTotaloutDPPM.l, @530, TotalProcessCost.l, @551, eTotalProcessCost.l, @573,
eTotalInventoryCost.l, @597, TotalRevenue.l, @613, ExpectedTotalRevenue.l, @660,
eTotalProfit.l/;
putclose DOE_Output_Undership;

    Solve TESTPLAN maximizing Z2 using MIP;
      display demand.l;
      display fabin.l, fabout.l, iPOST_fab.l;
      display TotalCT;
      display probein.l, probeout.l, iPOST_PROBE.l;
      display SCRAP.l;
      display assin.l, assout.l, iPOST_ASS.l;
      display ft1in.l, ft1out.l, iPOST_ft1.l;
      display BIin.l, BIout.l, iPOST_BI.l;
      display ft2in.l, ft2out.l, iPOST_ft2.l;
      display packin.l, packout.l, iPOST_PACK.l, ship.l;
      display FinishedGoods.l,FinishedGoodsScenario.l;
```

```
        display TOTALSCRAP.l,
TOTALSCRAPScenario.l,eTOTALSCRAPScenario.l,eTOTALSCRAP.l;
        display
FinishedGoodsCost.l,TotalFinishedGoodsCostScenario.l,TotalFinishedGoodsCost.l;
        display eTotalFinishedGoodsCostScenario.l,ExpectedTotalFinishedGoodsCost.l;
        display outDPPM.l,TotaloutDPPMScenario.l,TotaloutDPPM.l;
        display ExpectedTotaloutDPPMScenario.l,ExpectedTotaloutDPPM.l;
        display
ProcessCost.l,TotalProcessCostScenario.l,eTotalProcessCostScenario.l,eTotalProcessCos
t.l;
        display
InventoryCost.l,TotalInventoryCostScenario.l,eTotalInventoryCostScenario.l,eTotalInven
toryCost.l;
        display MPTSTR.l,MPHW_ordered.l,MPHW_received.l,
MPHW_Available.l,PROBECAP.l;
        display TotalMPHWCostB4RTDAYS.l, eTotalMPHWCostAfterRTDAYS.l;
        display FTTSTR.l,FTHW_ordered.l,FTHW_received.l,
FTHW_Available.l,FTCAP.l;
        display TotalFTHWCostB4RTDAYS.l, eTotalFTHWCostAfterRTDAYS.l;
        display BIBoard.l,BIB_ordered.l,BIB_received.l,BIB_Available.l, BICAP.l;
        display TotalBIBCostB4RTDAYS.l, eTotalBIBCostAfterRTDAYS.l;
        display Revenue.l,ScenarioRevenue.l,TotalRevenue.l;
        display ExpectedScenarioRevenue.l,ExpectedTotalRevenue.l;
        display TotalProfitScenario.l, eTotalProfit.l;
        display overship.l, undership.l, Effective_Profit.l;
        display z1.l, z2.l;

file DOE_Output_Profit   /DOE_Output_Profit.txt/;
DOE_Output_Profit.pw = 1000;
DOE_Output_Profit.ap = 1;
DOE_Output_Profit.nj = 2;
*Columns header - Run once at the beginning
put DOE_Output_Profit  'Date', @10, 'Time', @20, 'Seed', @26, 'Customer_Demand',
@43, 'FAB_Capacity', @57, 'Root_Cause_Probability', @81, 'DPPM_Quality_Target',
@102, 'Undership', @116, 'Effective_Profit', @135, 'Total_Demand', @149,
'MP_Testers(NONE)', @167, 'MP_Testers(Partial)', @190, 'MP_Testers(Full)', @210,
'FT_Testers(NONE)', @228, 'FT_Testers(Partial)', @251, 'FT_Testers(Full)', @269,
'BIBoard(NONE)', @286, 'BIBoard(Partial)', @306, 'BIBoard(Full)', @323,
'FinishedGoods(NONE)', @344, 'FinishedGoods(Partial)', @368, 'FinishedGoods(Full)',
@391, 'eTOTALSCRAP', @404, 'TotalFinishedGoodsCost', @428,
'ExpectedTotalFinishedGoodsCost', @460, 'eTotalFinishedGoodsCostGM', @487,
'TotaloutDPPM', @505, 'ExpectedTotaloutDPPM', @530, 'TotalProcessCost', @551,
'eTotalProcessCost', @573, 'eTotalInventoryCost', @597, 'TotalRevenue', @613,
'ExpectedTotalRevenue'/;
```

```
put DOE_Output_Profit  system.date, @10, system.time, @20, '2784', @26,
CustomerDemand, @43, Capacity, @57, RootCauseProbability, @81,
DPPMQualityTarget, @102, z1.l, @116, z2.l, @135, TotalDemand.l, @149,
MPTSTR.l('NONE'), @167, MPTSTR.l('Partial'), @190, MPTSTR.l('Full'), @210,
FTTSTR.l('NONE'), @228, FTTSTR.l('Partial'), @251, FTTSTR.l('Full'), @269,
BIBoard.l('NONE'), @286, BIBoard.l('Partial'), @306, BIBoard.l('Full'), @323,
FinishedGoodsScenario.L('NONE'), @344, FinishedGoodsScenario.L('Partial'), @368,
FinishedGoodsScenario.L('Full'),@391, eTOTALSCRAP.l, @404,
TotalFinishedGoodsCost.l, @428, ExpectedTotalFinishedGoodsCost.l, @460,
eTotalFinishedGoodsCostGM.l, @487, TotaloutDPPM.l, @505,
ExpectedTotaloutDPPM.l, @530, TotalProcessCost.l, @551, eTotalProcessCost.l, @573,
eTotalInventoryCost.l, @597, TotalRevenue.l, @613, ExpectedTotalRevenue.l/;
putclose DOE_Output_Profit;
```

APPENDIX B

GAMS Model Results

Listed below are the a portion of output results of the GAMS model presented in Appendix A. Listing the entire output file will require several hundred pagers, thus only the results are for the FABin variable over time period T are listed.

```
----    791 VARIABLE FABin.L  Wafers STARTED in the fab in time period T

           None     Partial       Full

D1        72.000      72.000      72.000
D2        72.000      72.000      72.000
D3        72.000      72.000      72.000
D4        72.000      72.000      72.000
D5        72.000      72.000      72.000
D6        72.000      72.000      72.000
D7        72.000      72.000      72.000
D8        72.000      72.000      72.000
D9        72.000      72.000      72.000
D10       72.000      72.000      72.000
D11       72.000      72.000      72.000
D12       72.000      72.000      72.000
D13       72.000      72.000      72.000
D14       72.000      72.000      72.000
D15       72.000      72.000      72.000
D16       72.000      72.000      72.000
D17       72.000      72.000      72.000
D18       72.000      72.000      72.000
D19       72.000      72.000      72.000
D20       72.000      72.000      72.000
D21       72.000      72.000      72.000
D22       72.000      72.000      72.000
D23       72.000      72.000      72.000
D24       72.000      72.000      72.000
D25       72.000      72.000      72.000
D26       72.000      72.000      72.000
```

D27	72.000	72.000	72.000
D28	72.000	72.000	72.000
D29	72.000	72.000	72.000
D30	72.000	72.000	72.000
D31	72.000	72.000	72.000
D32	72.000	72.000	72.000
D33	72.000	72.000	72.000
D34	72.000	72.000	72.000
D35	72.000	72.000	72.000
D36	72.000	72.000	72.000
D37	72.000	72.000	72.000
D38	72.000	72.000	72.000
D39	72.000	72.000	72.000
D40	72.000	72.000	72.000
D41	72.000	72.000	72.000
D42	72.000	72.000	72.000
D43	72.000	72.000	72.000
D44	72.000	72.000	72.000
D45	72.000	72.000	72.000
D46	72.000	72.000	72.000
D47	72.000	72.000	72.000
D48	72.000	72.000	72.000
D49	72.000	72.000	72.000
D50	72.000	72.000	72.000
D51	72.000	72.000	72.000
D52	72.000	72.000	72.000
D53	72.000	72.000	72.000
D54	72.000	72.000	72.000
D55	72.000	72.000	72.000
D56	72.000	72.000	72.000
D57	72.000	72.000	72.000
D58	72.000	72.000	72.000
D59	72.000	72.000	72.000
D60	72.000	72.000	72.000
D61	72.000	72.000	72.000
D62	72.000	72.000	72.000
D63	72.000	72.000	72.000
D64	72.000	72.000	72.000
D65	72.000	72.000	72.000
D66	72.000	72.000	72.000
D67	72.000	72.000	72.000
D68	72.000	72.000	72.000
D69	72.000	72.000	72.000
D70	72.000	72.000	72.000
D71	72.000	72.000	72.000
D72	72.000	72.000	72.000
D73	72.000	72.000	72.000
D74	72.000	72.000	72.000
D75	72.000	72.000	72.000
D76	72.000	72.000	72.000
D77	72.000	72.000	72.000
D78	72.000	72.000	72.000
D79	72.000	72.000	72.000

D80	72.000	72.000	72.000
D81	72.000	72.000	72.000
D82	72.000	72.000	72.000
D83	72.000	72.000	72.000
D84	72.000	72.000	72.000
D85	72.000	72.000	72.000
D86	72.000	72.000	72.000
D87	72.000	72.000	72.000
D88	72.000	72.000	72.000
D89	72.000	72.000	72.000
D90	72.000	72.000	72.000
D91	72.000	72.000	72.000
D92	72.000	72.000	72.000
D93	72.000	72.000	72.000
D94	72.000	72.000	72.000
D95	72.000	72.000	72.000
D96	72.000	72.000	72.000
D97	72.000	72.000	72.000
D98	72.000	72.000	72.000

APPENDIX C

SAS Source File

The source file used to run the SAS program is listed below for reference. The data presented in section 6.5 is used.

```
DATA GAMS_MODEL;
        INPUT PROB $ SEED DEMAND CAPACITY ROOTPROB DPPM PROFIT;
        CARDS;

1 2784 50000 72 0.3 1500 1114835.05
2 2784 50000 72 0.3 4000 1268269.28
3 2784 50000 72 0.6 1500 1702390.06
4 2784 50000 72 0.6 4000 1792450.07
5 2784 50000 120 0.3 1500 3773818.06
6 2784 50000 120 0.3 4000 4408674.93
7 2784 50000 120 0.6 1500 4325741.57
8 2784 50000 120 0.6 4000 4413078.16
9 2784 150000 72 0.3 1500 1106120.07
10 2784 150000 72 0.3 4000 1305109.82
11 2784 150000 72 0.6 1500 1681193.11
12 2784 150000 72 0.6 4000 1779652.98
13 2784 150000 120 0.3 1500 4003766.76
14 2784 150000 120 0.3 4000 4408674.93
15 2784 150000 120 0.6 1500 4311639.4
16 2784 150000 120 0.6 4000 4394201.46
17 3141 50000 72 0.3 1500 1114805.95
18 3141 50000 72 0.3 4000 1312464.55
19 3141 50000 72 0.6 1500 1702377.45
20 3141 50000 72 0.6 4000 1753019.27
21 3141 50000 120 0.3 1500 3792644.98
22 3141 50000 120 0.3 4000 4408674.93
23 3141 50000 120 0.6 1500 4325586.6
24 3141 50000 120 0.6 4000 4413078.16
25 3141 150000 72 0.3 1500 1106058.98
26 3141 150000 72 0.3 4000 1305151.23
```

```
27 3141 150000 72 0.6 1500 1680967.27
28 3141 150000 72 0.6 4000 1778835.81
29 3141 150000 120 0.3 1500 4003768.59
30 3141 150000 120 0.3 4000 4408674.93
31 3141 150000 120 0.6 1500 4311639.4
32 3141 150000 120 0.6 4000 4413078.16
33 3263 50000 72 0.3 1500 1114804.32
34 3263 50000 72 0.3 4000 1308245.8
35 3263 50000 72 0.6 1500 1702372.56
36 3263 50000 72 0.6 4000 1796143.27
37 3263 50000 120 0.3 1500 4085554.09
38 3263 50000 120 0.3 4000 4408674.93
39 3263 50000 120 0.6 1500 4324728.83
40 3263 50000 120 0.6 4000 4413078.16
41 3263 150000 72 0.3 1500 1106496.07
42 3263 150000 72 0.3 4000 1305758.71
43 3263 150000 72 0.6 1500 2098063.64
44 3263 150000 72 0.6 4000 1780961.75
45 3263 150000 120 0.3 1500 4003750.81
46 3263 150000 120 0.3 4000 4408674.93
47 3263 150000 120 0.6 1500 4311639.4
48 3263 150000 120 0.6 4000 4394201.46
49 4135 50000 72 0.3 1500 1114811.6
50 4135 50000 72 0.3 4000 1312464.55
51 4135 50000 72 0.6 1500 1698738.41
52 4135 50000 72 0.6 4000 1787705.78
53 4135 50000 120 0.3 1500 3746536.49
54 4135 50000 120 0.3 4000 4408674.93
55 4135 50000 120 0.6 1500 4315829.92
56 4135 50000 120 0.6 4000 4413078.16
57 4135 150000 72 0.3 1500 1106442.68
58 4135 150000 72 0.3 4000 1305190.04
59 4135 150000 72 0.6 1500 1682465.23
60 4135 150000 72 0.6 4000 1779813.42
61 4135 150000 120 0.3 1500 4003768.59
62 4135 150000 120 0.3 4000 4402382.7
63 4135 150000 120 0.6 1500 4311639.4
64 4135 150000 120 0.6 4000 4375430.33
 ;
```

```
PROC ANOVA;
 CLASS DEMAND;
 MODEL PROFIT=DEMAND;
 MEANS DEMAND /TUKEY;

PROC ANOVA;
 CLASS CAPACITY;
 MODEL PROFIT=CAPACITY;
 MEANS CAPACITY /TUKEY;

PROC ANOVA;
 CLASS ROOTPROB;
 MODEL PROFIT=ROOTPROB;
 MEANS ROOTPROB /TUKEY;

PROC ANOVA;
 CLASS DPPM;
 MODEL PROFIT=DPPM;
 MEANS DPPM /TUKEY;

PROC ANOVA;
 CLASS DEMAND CAPACITY;
 MODEL PROFIT=DEMAND|CAPACITY;
 MEANS CAPACITY /TUKEY;

PROC ANOVA;
 CLASS DEMAND ROOTPROB;
 MODEL PROFIT=DEMAND|CAPACITY;
 MEANS ROOTPROB /TUKEY;

PROC ANOVA;
 CLASS DEMAND DPPM;
 MODEL PROFIT=DEMAND|DPPM;
 MEANS DPPM /TUKEY;

PROC ANOVA;
 CLASS CAPACITY ROOTPROB;
 MODEL PROFIT=CAPACITY|ROOTPROB;
 MEANS ROOTPROB /TUKEY;

PROC ANOVA;
 CLASS CAPACITY DPPM;
 MODEL PROFIT=CAPACITY|DPPM;
 MEANS DPPM /TUKEY;

PROC ANOVA;
 CLASS ROOTPROB DPPM;
 MODEL PROFIT=ROOTPROB|DPPM;
 MEANS DPPM /TUKEY;

PROC ANOVA;
 CLASS DEMAND CAPACITY ROOTPROB;
 MODEL PROFIT=DEMAND|CAPACITY|ROOTPROB;
 MEANS ROOTPROB /TUKEY;

PROC ANOVA;
 CLASS DEMAND CAPACITY DPPM;
 MODEL PROFIT=DEMAND|CAPACITY|DPPM;
 MEANS DPPM /TUKEY;

PROC ANOVA;
 CLASS DEMAND ROOTPROB DPPM;
 MODEL PROFIT=DEMAND|ROOTPROB|DPPM;
 MEANS DPPM /TUKEY;

PROC ANOVA;
 CLASS CAPACITY ROOTPROB DPPM;
 MODEL PROFIT=CAPACITY|ROOTPROB|DPPM;
 MEANS DPPM /TUKEY;

PROC ANOVA;
 CLASS DEMAND CAPACITY ROOTPROB DPPM;
 MODEL PROFIT=DEMAND|CAPACITY|ROOTPROB|DPPM;
 MEANS DPPM /TUKEY;
```

APPENDIX D

ANOVA Procedure and Tukey's Analysis

The source file list in Appendix B is entered into the SAS program. The output file listed below includes the ANOVA procedure and tukey's analysis for all specified factor interactions. Font size is reduced to 10 to allow for more manageable view of the results.

```
The SAS System                    17:02 Sunday, March 9, 2008   1

                         The ANOVA Procedure

                       Class Level Information

                    Class        Levels    Values

                    DEMAND            2    50000 150000


                   Number of Observations Read         64
                   Number of Observations Used         64
```

```
                          The SAS System                 17:02 Sunday, March 9, 2008   2

                         The ANOVA Procedure

Dependent Variable: PROFIT

                                     Sum of
          Source              DF        Squares     Mean Square    F Value    Pr > F

          Model                1     10046583875     10046583875       0.00    0.9446

          Error               62    1.2773888E14    2.0603045E12

          Corrected Total     63    1.2774893E14


                   R-Square     Coeff Var      Root MSE    PROFIT Mean

                   0.000079      49.94009       1435376        2874196


          Source              DF       Anova SS     Mean Square    F Value    Pr > F

          DEMAND               1     10046583875     10046583875       0.00    0.9446
```

```
                         The SAS System                 17:02 Sunday, March 9, 2008   3

                        The ANOVA Procedure

               Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                   Alpha                              0.05
                   Error Degrees of Freedom             62
                   Error Mean Square               2.06E12
                   Critical Value of Studentized Range  2.82706
                   Minimum Significant Difference       717341


                Means with the same letter are not significantly different.


                Tukey Grouping         Mean      N    DEMAND

                     A        2886725     32    150000
                     A
                     A        2861667     32    50000
```

```
                    The SAS System                    17:02 Sunday, March 9, 2008   4

                  The ANOVA Procedure

                Class Level Information

             Class          Levels    Values

             CAPACITY            2    72 120


         Number of Observations Read          64
         Number of Observations Used          64
```

```
                         The ANOVA Procedure

Dependent Variable: PROFIT

                                    Sum of
        Source                DF       Squares     Mean Square    F Value    Pr > F

        Model                  1    1.2359938E14    1.2359938E14    1846.75    <.0001

        Error                 62    4.1495481E12     66928195004

        Corrected Total            63    1.2774893E14


             R-Square     Coeff Var     Root MSE    PROFIT Mean

             0.967518      9.000945      258704.8       2874196


        Source                DF       Anova SS     Mean Square    F Value    Pr > F

        CAPACITY                 1    1.2359938E14    1.2359938E14    1846.75    <.0001
```

```
                         The SAS System              17:02 Sunday, March 9, 2008   6

                        The ANOVA Procedure

              Tukey's Studentized Range (HSD) Test for PROFIT

 NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                 Alpha                              0.05
                 Error Degrees of Freedom             62
                 Error Mean Square              6.693E10
                 Critical Value of Studentized Range  2.82706
                 Minimum Significant Difference      129290


            Means with the same letter are not significantly different.


            Tukey Grouping        Mean      N    CAPACITY

                   A       4263887     32    120

                   B       1484506     32    72
```

```
                The SAS System                    17:02 Sunday, March 9, 2008   7

              The ANOVA Procedure

            Class Level Information

         Class        Levels    Values

         ROOTPROB          2    0.3 0.6


      Number of Observations Read         64
      Number of Observations Used         64
```

```
                         The SAS System                17:02 Sunday, March 9, 2008   8

                        The ANOVA Procedure

Dependent Variable: PROFIT

                                   Sum of
          Source            DF        Squares     Mean Square    F Value    Pr > F

          Model              1    2.2429085E12    2.2429085E12      1.11    0.2966

          Error             62    1.2550602E14    2.0242906E12

          Corrected Total   63    1.2774893E14


                 R-Square     Coeff Var      Root MSE    PROFIT Mean

                 0.017557      49.50169       1422776        2874196


          Source            DF       Anova SS     Mean Square    F Value    Pr > F

          ROOTPROB           1    2.2429085E12    2.2429085E12      1.11    0.2966
```

```
                         The SAS System                17:02 Sunday, March 9, 2008   9

                        The ANOVA Procedure

              Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                   Alpha                            0.05
                   Error Degrees of Freedom            62
                   Error Mean Square             2.024E12
                   Critical Value of Studentized Range  2.82706
                   Minimum Significant Difference       711044


            Means with the same letter are not significantly different.


            Tukey Grouping          Mean      N    ROOTPROB

                     A       3061401     32    0.6
                     A
                     A       2686992     32    0.3
```

```
                The SAS System                    17:02 Sunday, March 9, 2008  10

              The ANOVA Procedure

            Class Level Information

         Class          Levels    Values

         DPPM                2    1500 4000


       Number of Observations Read          64
       Number of Observations Used          64
```

```
                        The SAS System                17:02 Sunday, March 9, 2008  11

                       The ANOVA Procedure

Dependent Variable: PROFIT

                                      Sum of
          Source              DF         Squares     Mean Square    F Value    Pr > F

          Model                1    635721624195    635721624195       0.31    0.5796

          Error               62     1.271132E14     2.050213E12

          Corrected Total     63    1.2774893E14


                   R-Square     Coeff Var      Root MSE    PROFIT Mean

                   0.004976      49.81763       1431856        2874196


          Source              DF        Anova SS     Mean Square    F Value    Pr > F

          DPPM                 1    635721624195    635721624195       0.31    0.5796
```

```
                    The SAS System                17:02 Sunday, March 9, 2008  12

                  The ANOVA Procedure

          Tukey's Studentized Range (HSD) Test for PROFIT

  NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


            Alpha                          0.05
            Error Degrees of Freedom           62
            Error Mean Square             2.05E12
            Critical Value of Studentized Range  2.82706
            Minimum Significant Difference       715582


          Means with the same letter are not significantly different.


          Tukey Grouping         Mean     N    DPPM

                 A       2973861     32    4000
                 A
                 A       2774531     32    1500
```

```
               The SAS System                    17:02 Sunday, March 9, 2008  13

             The ANOVA Procedure

           Class Level Information

       Class        Levels    Values

       DEMAND            2    50000 150000

       CAPACITY          2    72 120


      Number of Observations Read          64
      Number of Observations Used          64
```

```
                        The SAS System                  17:02 Sunday, March 9, 2008  14

                       The ANOVA Procedure

Dependent Variable: PROFIT

                                     Sum of
            Source               DF         Squares     Mean Square    F Value    Pr > F

            Model                 3    1.2360991E14    4.1203305E13     597.29    <.0001

            Error                60    4.1390115E12     68983524372

            Corrected Total      63    1.2774893E14


                   R-Square     Coeff Var      Root MSE    PROFIT Mean

                   0.967600      9.138108      262647.1        2874196


            Source               DF        Anova SS     Mean Square    F Value    Pr > F

            DEMAND                1     10046583875     10046583875       0.15    0.7041
            CAPACITY              1    1.2359938E14    1.2359938E14    1791.72    <.0001
            DEMAND*CAPACITY       1    490044057.19    490044057.19       0.01    0.9331
```

```
                    The SAS System                  17:02 Sunday, March 9, 2008  15

                  The ANOVA Procedure

        Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


              Alpha                           0.05
              Error Degrees of Freedom              60
              Error Mean Square             6.898E10
              Critical Value of Studentized Range  2.82893
              Minimum Significant Difference       131347


        Means with the same letter are not significantly different.


        Tukey Grouping        Mean     N    CAPACITY

                A       4263887     32    120

                B       1484506     32    72
              The SAS System                   17:02 Sunday, March 9, 2008  16

             The ANOVA Procedure

           Class Level Information

    Class        Levels    Values

    DEMAND            2    50000 150000

    DPPM              2    1500 4000


   Number of Observations Read          64
   Number of Observations Used          64
```

```
                          The ANOVA Procedure

Dependent Variable: PROFIT

                                  Sum of
           Source           DF       Squares    Mean Square   F Value   Pr > F

           Model             3   659746119188   219915373063      0.10   0.9575

           Error            60   1.2708918E14    2.118153E12

           Corrected Total       63   1.2774893E14


                  R-Square    Coeff Var     Root MSE    PROFIT Mean

                  0.005164     50.63633      1455388        2874196


           Source           DF      Anova SS    Mean Square   F Value   Pr > F

           DEMAND                1    10046583875    10046583875      0.00   0.9453
           DPPM             1   635721624195   635721624195      0.30   0.5858
           DEMAND*DPPM               1    13977911118    13977911118      0.01   0.9355
```

```
                    The ANOVA Procedure

        Tukey's Studentized Range (HSD) Test for PROFIT

 NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


            Alpha                              0.05
            Error Degrees of Freedom             60
            Error Mean Square              2.118E12
            Critical Value of Studentized Range  2.82893
            Minimum Significant Difference     727824


       Means with the same letter are not significantly different.


       Tukey Grouping        Mean      N    DPPM

                A      2973861     32    4000
                A
                A      2774531     32    1500
```

```
           The SAS System                    17:02 Sunday, March 9, 2008  19

         The ANOVA Procedure

       Class Level Information

   Class         Levels    Values

   CAPACITY           2    72 120

   ROOTPROB           2    0.3 0.6


Number of Observations Read         64
Number of Observations Used         64
```

```
                         The SAS System                 17:02 Sunday, March 9, 2008  20

                        The ANOVA Procedure

Dependent Variable: PROFIT

                                        Sum of
        Source                 DF        Squares     Mean Square    F Value    Pr > F

        Model                   3    1.2636776E14    4.2122587E13    1829.87    <.0001

        Error                  60    1.3811653E12     23019422057

        Corrected Total        63    1.2774893E14


              R-Square     Coeff Var      Root MSE    PROFIT Mean

              0.989188      5.278746      151721.5        2874196


        Source                 DF        Anova SS     Mean Square    F Value    Pr > F

        CAPACITY                1    1.2359938E14    1.2359938E14    5369.35    <.0001
        ROOTPROB                1    2.2429085E12    2.2429085E12      97.44    <.0001
        CAPACITY*ROOTPROB       1    525474281493    525474281493      22.83    <.0001
```

```
                              The ANOVA Procedure

                  Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                      Alpha                            0.05
                      Error Degrees of Freedom               60
                      Error Means Square                2.302E10
                      Critical Value of Studentized Range  2.82893
                      Minimum Significant Difference        75874


               Means with the same letter are not significantly different.


               Tukey Grouping         Mean     N    ROOTPROB

                          A       3061401     32    0.6

                          B       2686992     32    0.3
```

```
           The SAS System                 17:02 Sunday, March 9, 2008  22

          The ANOVA Procedure

        Class Level Information

   Class        Levels    Values

   CAPACITY          2    72 120

   DPPM              2    1500 4000


 Number of Observations Read          64
 Number of Observations Used          64
```

```
                          The SAS System                    17:02 Sunday, March 9, 2008  23

                         The ANOVA Procedure

Dependent Variable: PROFIT

                                        Sum of
           Source                DF        Squares     Mean Square    F Value    Pr > F

           Model                  3    1.2434917E14    4.1449724E13     731.52    <.0001

           Error                 60    3.3997527E12     56662545219

           Corrected Total       63    1.2774893E14


                   R-Square     Coeff Var      Root MSE    PROFIT Mean

                   0.973387      8.281931      238039.0        2874196


           Source                DF       Anova SS     Mean Square    F Value    Pr > F

           CAPACITY               1    1.2359938E14    1.2359938E14    2181.32    <.0001
           DPPM                   1    635721624195    635721624195      11.22    0.0014
           CAPACITY*DPPM          1    114073752878    114073752878       2.01    0.1611
```

```
                         The SAS System                 17:02 Sunday, March 9, 2008  24

                       The ANOVA Procedure

              Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                 Alpha                          0.05
                 Error Degrees of Freedom              60
                 Error Mean Square              5.666E10
                 Critical Value of Studentized Range  2.82893
                 Minimum Significant Difference       119041


            Means with the same letter are not significantly different.


            Tukey Grouping         Mean      N    DPPM

                      A      2973861     32    4000

                      B      2774531     32    1500
```

```
               The SAS System                    17:02 Sunday, March 9, 2008  25

                The ANOVA Procedure

               Class Level Information

           Class         Levels    Values

           ROOTPROB          2    0.3 0.6

           DPPM              2    1500 4000


         Number of Observations Read          64
         Number of Observations Used          64
```

```
                              The SAS System                   17:02 Sunday, March 9, 2008  26

                             The ANOVA Procedure

Dependent Variable: PROFIT

                                      Sum of
            Source                DF         Squares     Mean Square    F Value    Pr > F

            Model                  3    3.1807155E12    1.0602385E12       0.51    0.6764

            Error                 60    1.2456821E14    2.0761368E12

            Corrected Total       63    1.2774893E14


                    R-Square     Coeff Var      Root MSE    PROFIT Mean

                    0.024898      50.13160       1440881        2874196


            Source                DF        Anova SS     Mean Square    F Value    Pr > F

            ROOTPROB               1    2.2429085E12    2.2429085E12       1.08    0.3028
            DPPM                   1    635721624195    635721624195       0.31    0.5821
            ROOTPROB*DPPM          1    302085435259    302085435259       0.15    0.7042
```

```
                    The SAS System                17:02 Sunday, March 9, 2008  27

                    The ANOVA Procedure

              Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                    Alpha                          0.05
                    Error Degrees of Freedom             60
                    Error Mean Square              2.076E12
                    Critical Value of Studentized Range  2.82893
                    Minimum Significant Difference        720569


              Means with the same letter are not significantly different.


              Tukey Grouping         Mean      N    DPPM

                       A      2973861     32    4000
                       A
                       A      2774531     32    1500
```

```
        The ANOVA Procedure

       Class Level Information

  Class        Levels    Values

  DEMAND            2    50000 150000

  CAPACITY          2    72 120

  ROOTPROB          2    0.3 0.6


 Number of Observations Read         64
 Number of Observations Used         64
```

```
                          The ANOVA Procedure

Dependent Variable: PROFIT

                                     Sum of
              Source             DF        Squares     Mean Square    F Value    Pr > F

              Model               7    1.2639863E14    1.8056947E13     748.86    <.0001

              Error              56    1.3502961E12     24112430920

              Corrected Total          63    1.2774893E14


                     R-Square     Coeff Var     Root MSE    PROFIT Mean

                     0.989430      5.402616      155281.8       2874196


              Source             DF       Anova SS     Mean Square    F Value    Pr > F

              DEMAND                 1     10046583875     10046583875       0.42    0.5212
              CAPACITY               1    1.2359938E14    1.2359938E14    5125.96    <.0001
              DEMAND*CAPACITY             1    490044057.19    490044057.19       0.02    0.8871
              ROOTPROB               1    2.2429085E12    2.2429085E12      93.02    <.0001
              DEMAND*ROOTPROB             1    2374242688.3    2374242688.3       0.10    0.7548
              CAPACITY*ROOTPROB           1    525474281493    525474281493      21.79    <.0001
              DEMAND*CAPACI*ROOTPR        1     17958321290     17958321290       0.74    0.3918
```

```
                    The ANOVA Procedure

          Tukey's Studentized Range (HSD) Test for PROFIT

 NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


              Alpha                              0.05
              Error Degrees of Freedom             56
              Error Mean Square              2.411E10
              Critical Value of Studentized Range  2.83308
              Minimum Significant Difference       77769


        Means with the same letter are not significantly different.


        Tukey Grouping        Mean      N    ROOTPROB

                 A        3061401     32    0.6

                 B        2686992     32    0.3
```

```
           The SAS System                        17:02 Sunday, March 9, 2008  31

         The ANOVA Procedure

       Class Level Information

  Class         Levels    Values

  DEMAND             2    50000 150000

  CAPACITY           2    72 120

  DPPM               2    1500 4000


 Number of Observations Read          64
 Number of Observations Used          64
```

```
                         The SAS System                  17:02 Sunday, March 9, 2008  32

                        The ANOVA Procedure

Dependent Variable: PROFIT

                                      Sum of
        Source                 DF        Squares     Mean Square    F Value    Pr > F

        Model                   7    1.2437571E14    1.7767959E13     294.97    <.0001

        Error                  56    3.3732142E12     60235968658

        Corrected Total        63    1.2774893E14


                R-Square     Coeff Var      Root MSE    PROFIT Mean

                0.973595      8.539089      245430.2        2874196


        Source                 DF        Anova SS     Mean Square    F Value    Pr > F

        DEMAND                  1     10046583875     10046583875       0.17    0.6845
        CAPACITY                1    1.2359938E14    1.2359938E14    2051.92    <.0001
        DEMAND*CAPACITY         1    490044057.19    490044057.19       0.01    0.9285
        DPPM                    1    635721624195    635721624195      10.55    0.0020
        DEMAND*DPPM             1     13977911118     13977911118       0.23    0.6319
        CAPACITY*DPPM           1    114073752878    114073752878       1.89    0.1743
        DEMAND*CAPACITY*DPPM    1    2023929254.3    2023929254.3       0.03    0.8552
```

```
                         The ANOVA Procedure

           Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                    Alpha                          0.05
                    Error Degrees of Freedom              56
                    Error Mean Square              6.024E10
                    Critical Value of Studentized Range  2.83308
                    Minimum Significant Difference       122917


             Means with the same letter are not significantly different.


             Tukey Grouping         Mean      N    DPPM

                       A       2973861     32    4000

                       B       2774531     32    1500
```

```
        The ANOVA Procedure

      Class Level Information

 Class        Levels    Values

 DEMAND            2    50000 150000

 ROOTPROB          2    0.3 0.6

 DPPM              2    1500 4000


Number of Observations Read        64
Number of Observations Used        64
```

```
                              The SAS System                17:02 Sunday, March 9, 2008  35

                             The ANOVA Procedure

Dependent Variable: PROFIT

                                        Sum of
          Source                DF       Squares     Mean Square    F Value    Pr > F

          Model                  7     3.20769E12    458241427362      0.21    0.9828

          Error                 56    1.2454124E14    2.2239506E12

          Corrected Total           63    1.2774893E14


                R-Square     Coeff Var      Root MSE    PROFIT Mean

                0.025109      51.88552       1491292       2874196


          Source                DF       Anova SS     Mean Square    F Value    Pr > F

          DEMAND                   1     10046583875     10046583875      0.00    0.9467
          ROOTPROB                  1    2.2429085E12    2.2429085E12      1.01    0.3196
          DEMAND*ROOTPROB           1    2374242688.3    2374242688.3      0.00    0.9741
          DPPM                   1    635721624195    635721624195      0.29    0.5950
          DEMAND*DPPM               1     13977911118     13977911118      0.01    0.9371
          ROOTPROB*DPPM             1    302085435259    302085435259      0.14    0.7138
          DEMAND*ROOTPROB*DPPM        1    575709096.72    575709096.72      0.00    0.9872
```

```
                    The SAS System                 17:02 Sunday, March 9, 2008  36

                      The ANOVA Procedure

            Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                    Alpha                         0.05
                    Error Degrees of Freedom              56
                    Error Mean Square               2.224E12
                    Critical Value of Studentized Range  2.83308
                    Minimum Significant Difference       746874


              Means with the same letter are not significantly different.


               Tukey Grouping         Mean      N    DPPM

                    A       2973861     32    4000
                    A
                    A       2774531     32    1500
            The SAS System                    17:02 Sunday, March 9, 2008  37

           The ANOVA Procedure

          Class Level Information

   Class        Levels    Values

   CAPACITY          2    72 120

   ROOTPROB          2    0.3 0.6

   DPPM              2    1500 4000


 Number of Observations Read          64
 Number of Observations Used          64
```

```
                         The SAS System                    17:02 Sunday, March 9, 2008  38

                        The ANOVA Procedure

Dependent Variable: PROFIT

                                         Sum of
             Source                  DF         Squares     Mean Square    F Value    Pr > F

             Model                    7    1.2747727E14    1.8211038E13    3754.06    <.0001

             Error                   56    271657227678    4851021922.8

             Corrected Total         63    1.2774893E14


                    R-Square     Coeff Var      Root MSE    PROFIT Mean

                    0.997874      2.423261      69649.28        2874196


             Source                  DF        Anova SS     Mean Square    F Value    Pr > F

             CAPACITY                 1    1.2359938E14    1.2359938E14    25479.0    <.0001
             ROOTPROB                 1    2.2429085E12    2.2429085E12     462.36    <.0001
             CAPACITY*ROOTPROB        1    525474281493    525474281493     108.32    <.0001
             DPPM                     1    635721624195    635721624195     131.05    <.0001
             CAPACITY*DPPM            1    114073752878    114073752878      23.52    <.0001
             ROOTPROB*DPPM            1    302085435259    302085435259      62.27    <.0001
             CAPACIT*ROOTPRO*DPPM     1     57627283430     57627283430      11.88    0.0011
```

```
                        The SAS System                 17:02 Sunday, March 9, 2008  39

                       The ANOVA Procedure

              Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                    Alpha                          0.05
                    Error Degrees of Freedom              56
                    Error Mean Square              4.851E9
                    Critical Value of Studentized Range  2.83308
                    Minimum Significant Difference        34882


             Means with the same letter are not significantly different.


             Tukey Grouping         Mean      N    DPPM

                      A      2973861     32    4000

                      B      2774531     32    1500
```

```
          The SAS System                    17:02 Sunday, March 9, 2008  40

         The ANOVA Procedure

       Class Level Information

Class        Levels    Values

DEMAND            2    50000 150000

CAPACITY          2    72 120

ROOTPROB          2    0.3 0.6

DPPM              2    1500 4000


Number of Observations Read         64
Number of Observations Used         64
```

```
                         The SAS System              17:02 Sunday, March 9, 2008  41

                        The ANOVA Procedure

Dependent Variable: PROFIT

                                     Sum of
          Source                 DF         Squares     Mean Square    F Value    Pr > F

          Model                  15    1.2754014E14    8.5026758E12    1954.75    <.0001

          Error                  48    208788218455    4349754551.1

          Corrected Total        63    1.2774893E14


                   R-Square     Coeff Var      Root MSE    PROFIT Mean

                   0.998366      2.294647      65952.67        2874196


          Source                 DF        Anova SS     Mean Square    F Value    Pr > F

          DEMAND                  1     10046583875     10046583875       2.31    0.1351
          CAPACITY                1    1.2359938E14    1.2359938E14    28415.3    <.0001
          DEMAND*CAPACITY         1    490044057.19    490044057.19       0.11    0.7386
          ROOTPROB                1    2.2429085E12    2.2429085E12     515.64    <.0001
          DEMAND*ROOTPROB         1    2374242688.3    2374242688.3       0.55    0.4636
          CAPACITY*ROOTPROB       1    525474281493    525474281493     120.81    <.0001
          DEMAND*CAPACI*ROOTPR    1     17958321290     17958321290       4.13    0.0477
          DPPM                    1    635721624195    635721624195     146.15    <.0001
          DEMAND*DPPM             1     13977911118     13977911118       3.21    0.0793
          CAPACITY*DPPM           1    114073752878    114073752878      26.23    <.0001
          DEMAND*CAPACITY*DPPM    1    2023929254.3    2023929254.3       0.47    0.4984
          ROOTPROB*DPPM           1    302085435259    302085435259      69.45    <.0001
          DEMAND*ROOTPROB*DPPM    1    575709096.72    575709096.72       0.13    0.7176
          CAPACIT*ROOTPRO*DPPM    1     57627283430     57627283430      13.25    0.0007
          DEMA*CAPA*ROOTP*DPPM    1     15422267844     15422267844       3.55    0.0658
```

```
                         The SAS System              17:02 Sunday, March 9, 2008  42

                        The ANOVA Procedure

               Tukey's Studentized Range (HSD) Test for PROFIT

   NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error
rate than REGWQ.


                    Alpha                          0.05
                    Error Degrees of Freedom              48
                    Error Mean Square              4.3498E9
                    Critical Value of Studentized Range  2.84352
                    Minimum Significant Difference        33152


           Means with the same letter are not significantly different.


           Tukey Grouping         Mean      N    DPPM

                   A       2973861     32    4000

                   B       2774531     32    1500
```

BIBLIOGRAPHY

[1] APTE, A., AND BARR, R. S. An Interval-Pivoting Algorithm for the Uniform-Bound Interval-Flow Transportation Problem. *Technical Report 04-EMIS-01*, 2004.

[2] BAI, D., CARPENTER, T., AND MULVEY, J. Making a Case for Robust Optimization Models. *Institute for Operations Research and the Management Sciences* 43:7 1997.

[3] BIRGE, J. R., AND LOUVEAUX, F. *Introduction to Stochastic Programming*, Springer Series in Operation Research (1997), 128-135.

[4] BLANCHARD, B. S. *System Engineering Management*, John Wiley & Sons (2008), 2-6, 221-241.

[5] BLANCHARD, B. S., AND FABRYCKY, W. J. *System Engineering and Analysis*, Pearson Prentice Hall (2006), 54-159.

[6] BOSILJ-VUKSIC, V, GIAGLIS, G. M., AND HLUPIC, V. IDEF Diagrams and Petri Nets for Business Process Modeling. *International Conference on Enterprise Information Systems*, 2000.

[7] BOUKOUVALAS, C., DE NATALE, F., DE TONI, G., KITTLER, J., MARIK, R., MIRMEHDI, M., PETROU, M., LE ROY, P., SALGARI, R., AND VERNAZZA, G. An integrated system for quality inspection of tiles. *International Conference on Quality Control by Artificial Vision* (1997), 49-54.

[8] BROOKE, A., KENDRICK, D., MEERAUS, A., AND RAMAN, R. *GAMS: A User's Guide*, GAMS Development Corporation, 1998.

[9] BROWNING, T. R. The Many Views of a Process: Toward a Process Architecture Framework for Product Development Processes. *Systems Engineering* 12:1 (2009), 69-90.

[10] BROWNING, T. R., FRICKE, E., AND NEGELE, H. Key Concepts in Modeling Product Development Processes. *Systems Engineering, The Journal of The International Council on Systems Engineering* 9:2 (2006).

[11] BUEDE, D. M. *The Engineering Design of Systems, Models and Methods*, John Wiley and Sons (2006), 59-74, 331-358.

[12] CODY, R. P., AND SMITH, J. K. *Applied Statistics and the SAS Programming Language*, Pearson Prentice Hall, 2006.

[13] DAVIS, T. L., AND FARLEY, J. H. Size distribution of southern bluefin tuna (Thunnus maccoyii) by depth on their spawning ground. *Fishery Bulletin 99* (2001), 381-386.

[14] DEMIRAG, O., JOHNSON, A., NAZZAL, D., AND WAN, Y. Integrated Definition (IDEF) Modeling Techniques. Georgia Institute of Technology, www2.isye.gatech.edu/~lfm/8851/IDEF_V4.ppt, 2008.

[15] HÜBNER, R. *Strategic Supply Chain Management in Process Industries*," Springer, 594 Lecture Notes in Economics and Mathematical Systems, 2007.

[16] KAUPPINEN, H. A two stage defect recognition method for parquet slab grading. *International Conference on Pattern Recognition*, 2000.

[17] LAMBERT, J. H., JENNINGS, R. K., AND JOSHI, N. N. Integration of Risk Identification with Business Process Models. *Systems Engineering, The Journal of The International Council on Systems Engineering*, 9:3 (2006).

[18] MASON, R. L., GUNST, R. F., AND HESS, J. L. *Statistical Design and Analysis of Experiments*, John Wiley and Sons, 1989.

[19] MAYER, R. J., MENZEL, C. P., PAINTER, M. K., DEWITTE, P. S., BLINN, T., AND PERAKATH, B. *Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report*, Knowledge Based Systems, Incorporated, 1995.

[20] MEYER, U. B., CREUX, S. E., AND MARIN, A. K. W. *Process Oriented Analysis*, CRC Taylor and Francis Group (2007), 6-21.

[21] MULVEY, J. M., AND SHETTY, B. IDEF Financial planning via multi-stage stochastic optimization. *Computer & Operations Research 31* (2004) 1-20.

[22] MULVEY, J. M., VANDERBEI, R. J., AND ZENIOS, S. A. Robust Optimization of Large-Scale Systems. *Operations Research* 43:2 (1995), 264-281.

[23] PINEDO, M. L. *Planning and Scheduling in Manufacturing and Services*, Springer Series in Operation Research, 2005.

[24] POCHET, Y., AND WOLSEY, L. A. *Production Planning by Mixed Integer Programming*, Springer Series in Operation Research and Financial Engineering, 2006.

[25] ROSENTHAL, R. E. *A GAMS Tutorial*, GAMS Development Corporation, 2006.

[26] SCHEFFLER, M., FRANZON, P. D., AND TRÖSTER, G. A “Defect Level Versus Cost” System Tradeoff for Electronics Manufacturing, *IEEE Transactions on Electronics Packaging Manufacturing* 27:1 (2004).

[27] SEN, S., AND HIGLE, J. L. An Introductory Tutorial on Stochastic Linear Programming Models. *Institute for Operations Research and the Management Sciences, Interfaces 29* (1999), 33-61.

[28] TEXAS INSTRUMENTS, INC. Semiconductor Manufacturing: How a Chip is Made. Texas Instruments – RFAB, http://www.ti.com/corp/docs/rennerroadfab/chip.shtml, 2009.

[29] UNARY, D., AND GOSSELIN, B. A Quality Grading Approach for “JONAGOLD” Apples. *IEEE Benelux Signal Processing Symposium*, 2004.

[30] VOß, S., AND WOODRUFF, D. L. *Introduction to Computational Optimization Models for Production Planning in a Supply Chain*, Springer (2006), 59-79, 96-105.

[31] WALLACE, S. W., AND ZIEMBA, W. T. *Applications of Stochastic Programming*, Mathematical Programming Society, 2005.

[32] ZAKARIAN, A., AND KUSIAK, A. Analysis of Process Models. *IEEE Transactions on Electronics Package Manufacturing* 23:2 (2000).