RC22762 (W0304-022) April 3, 2003
Computer Science

IBM Research Report

Inventory Management in High-Technology Value Chains

Markus Ettl, Feng Cheng, Grace Y. Lin,
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

David D. Yao
IEOR Department, 302 Mudd Building
Columbia University, New York, NY 10027

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Inventory Management in High Technology Value Chains

Feng Cheng, Markus Ettl, Grace Lin
IBM Research Division, T.J. Watson Research Center
Yorktown Heights, NY 10598
fcheng,msettl,gracelin@us.ibm.com

David D. Yao[*]
IEOR Department, 302 Mudd Building
Columbia University, New York, NY 10027
yao@ieor.columbia.edu

February 2003

Abstract

This chapter presents some of the analytical models and tools that we have developed to support IBM business units in their effort to manage inventory and improve value chain operations. In IBM's businesses, inventory-driven costs, which include price protection, financing, inventory write-downs (price erosion), and inventory write-offs (obsolescence) are tremendous cost drivers outweighing all others in terms of impact on business performance. The complexity of the end-to-end value chain makes it a serious challenge to determine where to hold safety stock to minimize inventory costs, and provide a committed level of service to the final customer. We describe the successful development and applications of analytical models for finding the optimal placement of safety stocks in multi-echelon value chains that are subject to forecast, lead time, and attach-rate uncertainty. We focus on three types of value chain architecture: the complex configured hardware value chains, the configure-to-order value chains, and the semiconductor value chains.

[*] Research undertaken while an academic visitor at IBM Research Division, T.J. Watson Research Center.

1 Introduction and Overview

In the computer industry we see dramatic reductions in the price of computers, driven by advances in technology and competitive forces pushing towards lower margins. Product life cycles are collapsing to months rather than years, giving companies less time to recover product development costs and increasing the pressure to rapidly and flawlessly commercialize new technology. We are constantly being challenged to improve on productivity throughout the enterprise. We continue reengineering business processes and eliminating steps, and over the past several years our attention has increasingly turned toward managing the value chain.

IBM and its partners and competitors seek operational and financial performance improvement through reduced product development cycles, operational efficiency, and better customer responsiveness. This quest for excellence has become significantly more complex as enterprises no longer compete as stand-alone entities. The success is achieved through value chain optimization and collaboration among all value chain participants, from OEM's, Tier-1's, and lower tier suppliers to distributors, trading partners, and retailers. Deploying common business processes across distinct operating entities allows participants to share decision-making, workflows, and capabilities in pursuit of lower costs and greater efficiency.

A significant challenge, and opportunity, for IBM is that we have one of the most vertically integrated supply chains in the industry. We manufacture most major assemblies in our computers. As a result, IBM's integrated supply chain is even more complex and difficult to manage. Our supply chain is under constant pressure to move towards the assembly of components from a vast array of outside suppliers, and to sell components we make to the marketplace. More and more, we deliver components to internal business units and also sell them to competitors, some who manage their supply chains and assembly operations to deliver higher levels of profitability than IBM derives from the same markets. Clearly, such an environment makes managing the extended supply chain critical to our success.

The business environment in the electronics industry, which is characterized by volatility and velocity, requires tools and applications that can recommend timely supply planning decisions that optimize profits and balance business risk. Standard enterprise applications such as enterprise resource planning (ERP), customer relationship management (CRM), and supply chain management (SCM) systems are effective in managing hundreds of product and service offerings, but they often lack in high-quality decision making.

In this chapter, we describe analytical models and tools that we have developed to support IBM business units in their effort to manage inventory and improve value chain operations. In IBM's businesses, inventory-driven costs, which include price protection, financing, inventory write-downs (price erosion), and inventory write-offs (obsolescence) are tremendous cost drivers outweighing all others in terms of impact on business performance. The complexity of the end-to-end value chain makes it difficult to determine where to hold safety stock to minimize inventory costs, and provide a committed level of service to the final customer. We developed analytical optimization models for finding the optimal placement of safety stocks in multi-echelon value chains that are subject to forecast, lead time, and attach-rate uncertainty. We will describe the successful application of these analytical models by cross-functional teams within IBM, our suppliers, and our customers. We also discuss how the teams have used the models to allocate component inventories, reduce finished goods inventories, manage product variety, and improve forecast accuracy.

The rest of the chapter is organized as follows. We start with an overview in section 2 of different value chain architectures, focusing on complex configured hardware value chains, configure-to- order value chains, and semiconductor value chains, with details of the modeling and applications elaborated in the next three sections. In section 3, we introduce a multi-echelon inventory model for complex configured products, and discuss its application in IBM's hard disk drive supply chain. In section 4, we develop an optimization algorithm for safety stock placement in a configure-to-order supply chain with high-volume, high variety products, and describe its application at IBM's personal computer division. In section 5, we describe an inventory model for a semiconductor supply chain, and present our application experience at a large US semiconductor manufacturer. We conclude with a summary in section 6.

A very brief note on related readings: More details of the three application sections can be found in [4, 5] for section 3.1, [3] for section 4, and [2] for section 5. The two edited volumes, [6, 7], collect many recent research works on various aspects of supply chain management. Background materials in inventory theory can be found in [8].

2 High-Technology Value Chains

In this section,we describes three examples of value chain applications in high-technology industries. For each application, we identify end-to-end management processes which enable their business objectives.

- *Complex Configured Hardware Value Chains.* Standard part number based product offerings with complex bills of materials. Fulfilling customer orders through vendor-managed inventory hubs outside of the customer's manufacturing plant.

- *Configure-to-Order Value Chains.* Customizing products and solutions quickly to customer requirements. Developing configure-to-order capabilities for direct selling through the Internet.

- *Semiconductor Value Chains.* Implementing postponement strategies to defer customer-specific configurations until as late as possible. Determining the right safety stock policies at the inventory postponement points.

2.1 Complex Configured Hardware Value Chains

Complex configured hardware value chains support part number based components and products with complex bills-of-materials. Customer orders are typically submitted through enterprise web sites for large enterprise customers, business-to-business portals for business partners, and public websites for consumers and small and medium business customers. An Advanced Planning System (APS) applications reconcile customer forecasts with existing supply, and send allocations to a fulfillment system. The fulfillment system schedules orders, calculates estimated customer arrival dates, and sends manufacturing orders to a floor control system to manage the assembly of products.

The manufacturing process of hard disk drives (HDDs) is a typical example of a complex configured hardware value chain. IBM's Storage Systems Division, now Hitachi Global Technologies, produces disk drives for the OEM market, as well as supplies internal IBM business units. The supply chain for HDDs is complex due both to the vertically integrated nature of IBM and the customers' requirement for Just-In-Time (JIT) inventory hubs, where IBM inventory is stocked outside of a customer's factory.

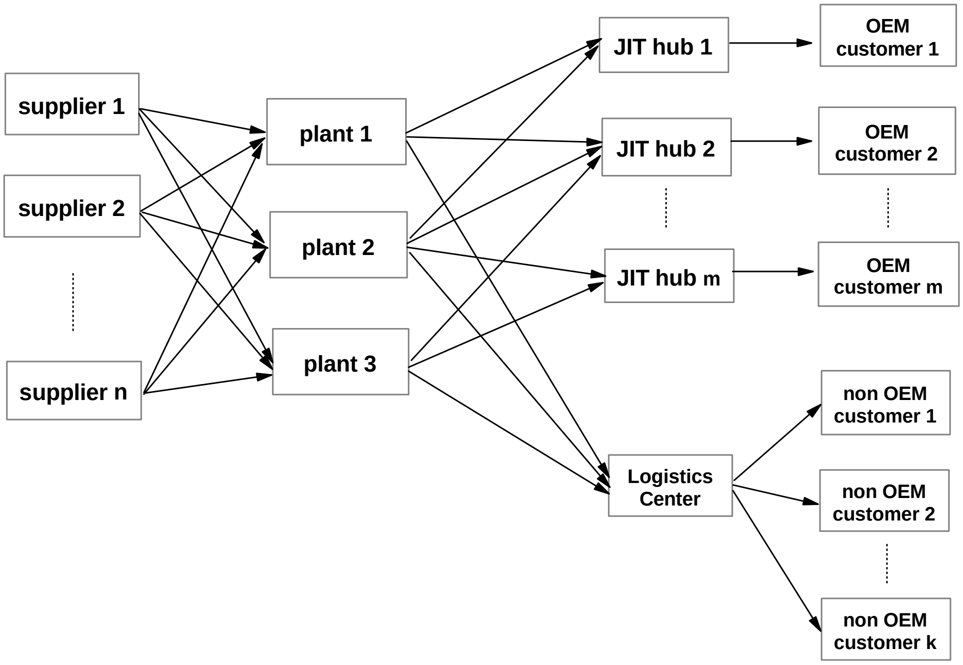

Figure 1: Example of a Complex Configured Hardware Value Chain.

HDDs are sold to OEM customers, distributors, and used in other IBM products such as personal computers and servers. Two of three product segments, mobile and performance HDDs, are manufactured in Thailand and Hungary, and server HDDs are manufactured in Singapore. In addition to the JIT hubs, the distribution network includes Logistics Centers and internal IBM direct shipments as illustrated in Figure 1. When HDD's are in excess supply at one hub but needed at another customer's hub, they can be reconfigured at a Logistics Center and then redirected. While reconfiguration occurs infrequently, it is fairly common to need to reship an HDD for a given customer from one hub to another. HDDs are assembled from non-configured ”vanilla” drives (called ISOs) at the manufacturing plants, and shipped to a customer JIT hub fully configured for that customer. When the JIT hub inventory drops below a reorder point, a pull signal is sent to the plant to configure the ISO's for that hub. The pull strategy requires forecasting to determine safety stock levels that protect against variations in demand.

Because of limited production capacity, the plants typically overproduce (versus demand) early in the quarter to meet peak demands towards the end of the quarter. This demand skew is much more prevalent in Logistics Centers than it is for JIT hubs. Because the configuration process is relatively simple, and because demand for the vanilla drives is easier to predict than HDD demand, it is beneficial to delay customization of an ISO as long as possible.

2.2 Configure-To-Order Value Chains

In computer assembly value chains, products are offered as either fixed configurations or open configurations. Fixed configurations have an individual material code called machine-type/model that is referred to in customer orders. Open configurations, in contrast, can be fully configured by the customer. Here, the customer can navigate from a brand, series, or family via a web-based configurator, which shows the options compatible with the selected machine. The configurator lets the customer choose from a selection of processors, hard disk drives, network access cards, graphics cards, and memory sizes. Only the base unit and standard components (called building blocks) have individual material codes. Whereas fixed configurations are normally made-to-stock, open configurations are assembled-to-order after the customer order is received (configure-to-order, CTO). Figure 2 illustrates a configure-to-order supply chain.

The purpose of forecasting in computer assembly value chains is to produce an accurate component forecast. For fixed configurations, the bills-of-materials of the fixed configurations are simply exploded to component level. For open configurations, component forecasts are derived from a collaborative forecast for a customer segment (or product family), together with attach-rates that define the distribution of components within the customer segment (or product family). As the assembly lead time is much shorter than the component lead time, the component forecast is an important input into procurement decisions. The focus of inventory management in CTO value chains is shifting from configured machines towards building blocks, which are replenished based on the component forecast.

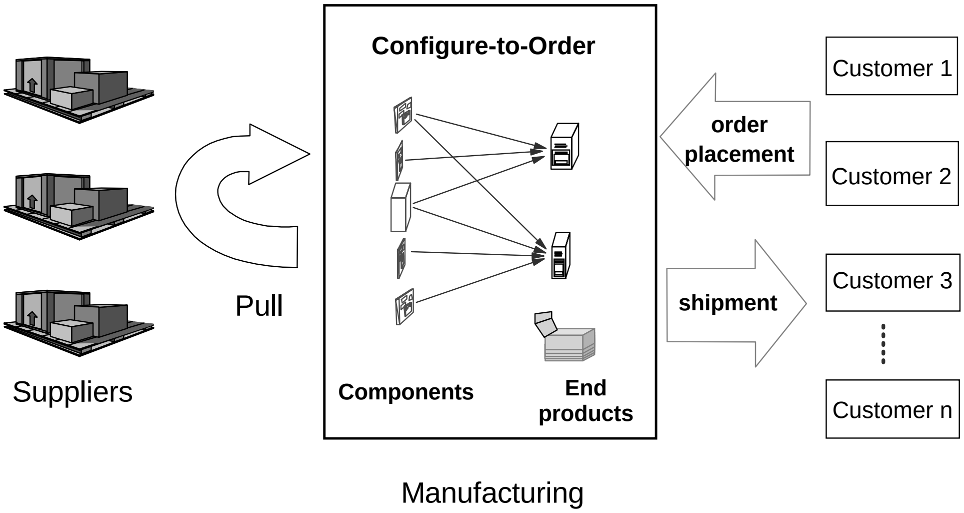

Figure 2: Example of a Configure-to-Order Value Chain.

The configure-to-order paradigm has been widely accepted in the electronics industry. Open configurations offer higher product variety, and hence often results in broader market coverage and increased demand volume. Postponing the final assembly in configure-to-order operations provides flexibility in terms of product variety, and achieves resource pooling in terms of maximizing the usage of component inventory. CTO is an ideal business model for mass customization, and provides quick response time to order fulfillment.

2.3 Semiconductor Value Chains

The semiconductor industry has one of the most complex manufacturing processes and value chains. Equipment necessary for production is extremely capital expensive and difficult to install. The semiconductor industry is at the beginning of a complex network of value chains, which often leads to tremendous demand fluctuations and high uncertainties of demand forecasts. As a result, many semiconductor companies have outsourced key value chain activities to electronic manufacturing service providers (EMS) to focus on core competencies, such as product design, inventory management, order fulfillment, and utilizing production capacity efficiently.

The production process of semiconductors consists of two stages called front-end and back-end that are separated by a die bank. The front-end consists of wafer fabrication and wafer testing, whereas the back-end performs die bonding, assembly, and module testing. The production process is often split over several locations in different geographies, e.g. front-end facilities in SOuth-East Asia, and back-end facilities in Europe and North America. The production lead time of the front-end is eight to twelve weeks. After testing, wafers are sliced into individual chips that are subsequently used to produce different finished goods. The tested chips are stored in a die bank. The back-end operates in a make-to-stock or make-to-order mode. In the assembly stage, chips supplied from the die bank are connected to a platform, bonded, and sealed in plastic. The finished modules are shipped to a customer or stored in finished goods inventory. The lead time of the back-end is four to six weeks.

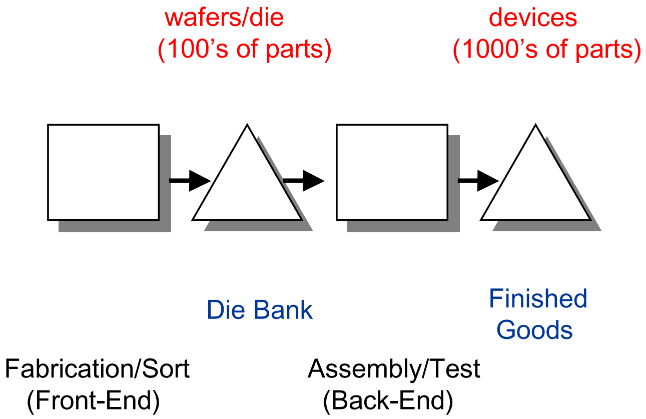

Figure 3: Example of a Semiconductor Value Chain.

The customers of semiconductor manufacturers are usually large resellers or OEM's that can have considerable influence. In this industry, high level of customer serviceability is critical since although an integrated circuit may only be one component of hundreds in a customer's product, a late delivery may shutdown a production line.

3 A Multi-Echelon Inventory Model for Complex Configured Hardware Value Chains

A key driver to achieve supply chain optimization is to manage and reduce uncertainty, and to maintain flexibility so as to adapt to market changes quickly. How much inventory budget is needed to achieve good customer serviceability? How much inventory do we need at which locations to meet required service levels? What is the effect of changes in supplier lead times, or supplier practices? What is the effect on required inventory levels, of changes to where and how we assemble products? IBM's Asset Management Tool (AMT) was designed to answer these questions.

3.1 AMT: The Optimization Engine

The optimization engine performs AMT's main function: quantifing the trade-offs between customer service levels and the inventory in the supply network. The objective is to determine the safety stock for each product at each location in the supply chain to minimize total inventory investment. Below we present an overview of this optimization model, referring the complete technical details to [4].

The supply chain is modeled as a multi-echelon network. Each stocking location in the network is treated as a queuing system that incorporates an inventory control policy: the base-stock control, with the base-stock levels being decision variables. To analyze such a network, we develop an approach based on decomposition. The key idea is to analyze each stocking location in the network individually, and to capture the interactions between different stocking locations through their *actual lead times*.

We model each stocking location by a queue with batch Poisson arrivals, and infinite servers with service times following general distributions, and this model is denoted as $M^Q/G/\infty$ in queueing notation. To do so, we need to first specify the arrival and the service processes. The arrival process at each location is obtained by applying the demand explosion technique in standard MRP (materials requirement planning) to the product structure. The batch Poisson arrival process has three main parameters: the arrival rate, and the mean and the variance of the batch size. Therefore, it allows us to handle many forms of demand data using a three-parameter fit. For instance, demand in a certain period can be characterized by its min, max and the most likely value. The service time is the actual lead time at each stocking location. Figure 4 illustrates how the actual leadtime, $\tilde{L}_i$ is calculated from the nominal leadtime (e.g., production or transportation time), L_i, along with the fill rate, f_j, of location i's supplier j. In particular, when the supplier has a stockout, which happens with probability $1 - f_j$, the actual leadtime at i has an additional delay of τ_j, which is the time required for j to produce the next unit to supply i's order. In our model, the estimation of τ_j is derived from a Markov chain analysis.

With the arrival and service processes in place, we can analyze the queue and derive performance measures such as inventory, backorder, fill rates, and customer service levels. The number of jobs in the $M^Q/G/\infty$ queue i, X_i, is the key quantity in our analysis. The on-hand inventory I_i and the backorder level B_i relate to X_i through the following simply formulas:

$$I_i = [R_i - X_i]^+, \quad \text{and} \quad B_i = [X_i - R_i]^+, \tag{1}$$

where R_i is the base-stock level, and $[x]^+ := \max\{x, 0\}$.

Through the equations in (1), the distributions of both I_i and B_i can be related to the distribution of X_i. To alleviate the computational burden in large-scale applications, we choose to derive the mean and the variance of X_i, and approximate it with a normal distribution. where Z denotes the standard

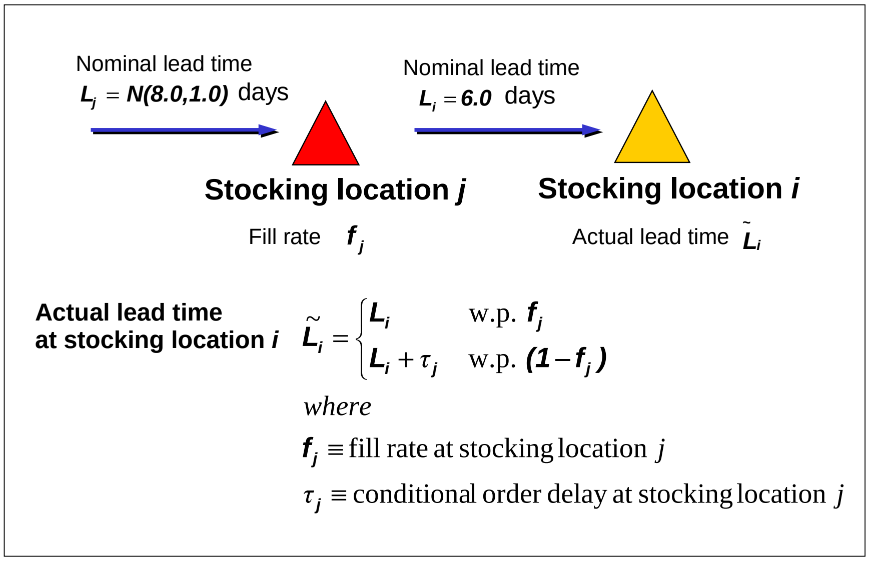
Nominal lead time
L_j = N(8.0,1.0) days
Nominal lead time
$L_i = 6.0$ days
Stocking location j
Stocking location i
Fill rate f_j
Actual lead time $\tilde{L}_i$
Actual lead time at stocking location i
$\tilde{L}_i = \begin{cases} L_i & \text{w.p. } f_j \\ L_i + \tau_j & \text{w.p. } (1 - f_j) \end{cases}$
where
$f_j \equiv$ fill rate at stocking location j
$\tau_j \equiv$ conditional order delay at stocking location j

Figure 4: Actual lead times.

normal variate. Both the mean and the standard deviation of X_i, denoted μ_i and σ_i, depend on the actual lead time (explained above, and used as the service time in the queue model), and can be derived from queueing analysis (refer to [4]). Then, we write:

$$X_i = \mu_i + \sigma_i Z, \tag{2}$$

Similarly, we can express the base-stock level as follows:

$$R_i = \mu_i + k_i \sigma_i, \tag{3}$$

where k_i is the so-called safety factor. This way, we turn the decision variables from the base-stock levels to the safety factors. Also note that the base-stock level consists of two parts: μ_i, the work-in-process, or pipeline inventory; and $k_i\sigma_i$, the safety stock.

The objective of the optimization model is to minimize the total inventory capital: At each stocking location, there are two types of inventory: the finished goods, or on-hand inventory, and the pipeline inventory. The expected pipeline inventory is simply μ_i; the expected on-hand inventory follows from combining (2) and (3) with (1):

$$\mathsf{E}(I_i) = \mathsf{E}[R_i - X_i]^+ = \sigma_i \mathsf{E}[k_i - Z]^+.$$

Let

$$H(x) := \mathsf{E}[x - Z]^+ = \int_{-\infty}^{x} (x - z)\phi(z)dz = x\Phi(x) + \phi(x), \tag{4}$$

where ϕ and Φ denoting, respectively, the density function and the distribution function of Z. Then,

$$\mathsf{E}(I_i) = \sigma_i H(k_i). \tag{5}$$

Hence, the objective of our optimization model is

$$\min_{\mathbf{k}} C(\mathbf{k}) := \sum_{i \in \mathcal{S}} [\hat{c}_i \mu_i + c_i \sigma_i H(k_i)]. \tag{6}$$

Here $\mathcal{S}$ is the set of all stores; c_i is the unit cost of finished goods inventory at store i; and $\hat{c}_i$ is the unit cost of pipeline inventory,

$$\hat{c}_i := \frac{1}{2}(c_i + \sum_{j \in \mathcal{S}_{>i}} c_j u_{ji}),$$

where $\mathcal{S}_{>i}$ is the set of upstream stores of i, and u_{ji} is the usage count, the number of units in store j that is needed to make each unit in store i.

The constraints of the optimization model are the required customer service levels. They are represented as the probability (e.g., 95% or 99%) that customer orders are filled within a given due date. We first derive the required fill rate for each end product so as to meet the required customer service level. This fill rate relates to the actual lead times of all upstream stocking locations, via the BOM structure of the network. Therefore, our model captures the interdependence at different stocking locations, in particular the effect of base-stock levels and fill rates at each stocking location on the service level of the end product.

To allow fast execution of the optimization, we derive analytical gradient estimates in closed forms. Consider stocking location j, or the j-th term in the objective function. First, we derive the partial derivative w.r.t. k_j (the safety factor) for stocking location j. For each immediate downstream stocking location, the partial derivative involves the mean and variance of the number of arrivals over the service time (i.e., the actual leadtime) in the queue model. For stocking locations further downstream, we simply ignore their derivatives, since their actual lead times will be weighted by multiples of no-fill rates, which become negligible as the stocking locations become farther downstream.

This way, we have a constrained non-linear optimization model, with the gradients explicitly derived. A conjugate gradient search procedure is used to generate the optimal solution. As the objective function has a quite rugged surface, we improve upon local minima by following several heuristic procedures. For instance, evaluating the objective function at a reasonably large number of randomly generated points, and selecting the best point to start the gradient search.

3.2 Case Study: IBM's Hard Disk Drive Value Chain

In late 1997, many of IBM's HDD customers had adopted the practice of vendor-managed inventory, which required IBM to establish Just-In-Time (JIT) inventory hubs nearby the customer's plants. The inventory in the JIT hubs was managed and owned by IBM until it was pulled from the hub by the customer.

As more and more JIT hubs were established and an increasing proportion of HDD inventory flowed through the hubs, it became more important to minimize the amount of product in the hubs. However, it was unclear what the impact of delayed customization would be on the supply chain and, in particular, on inventory levels and customer serviceability. It was also unclear what levels of safety stock, and therefore what reorder points were needed in the JIT hubs to fulfill customer service requirements. IBM's Corporate Headquarter formed a cross-functional team whose charge was to assess the impact of delayed customization on the SSD supply chain, and to recommend optimal inventory levels for the JIT hubs.

The primary objective of the team was to use AMT to determine optimal base-stock levels (or reorder points) at the JIT hubs, to implement a configure-to-pull strategy at these hubs, and to improve inventory turnover while meeting customer service requirements. AMT allowed the explicit representation of assembly and transport operations. Bills-of-materials and manufacturing lead times (if outsourced, replenishment lead times) were extracted from SAP and incorporated into the model. Forecast errors were estimated at the product family level to provide demand distribution data for our analysis.

We created end-to-end simulation models for each product family, and validated these models against historical data to assure that the information and materials flow through the HDD value chain was captured accurately. Comparing key output performance measures of our model (i.e., finished goods inventory at JIT hubs and plants) with historical actuals, we found that JIT hub inventory matched at a part number level to within 8% of actual on average. JIT hub inventory matched at a product family level to within just 6% of actual, and plant finished goods inventory matched at a product family level to within 8% of actual. On-time serviceability matched at close to 100% at part number and order level. An example of the validations is shown in Figure 5.

Subsequently, we constructed simulation models for a select set of high-volume HDD products based on historical inventory and demand data. One model represented the current business process, whereas the other represented a ”configure-to-pull” business process with optimized inventory buffers at the JIT hubs. The comparison between the two models showed significant savings in hub inventory, averaging about 69%, can be achieved with configure-to-pull without sacrificing serviceability. The savings were achieved with only a 21% increase in plant inventory of non-configured vanilla drives (ISOs) as shown in Figure 6.

By varying the time needed to customize an ISO in the plant, we then established the relative impact

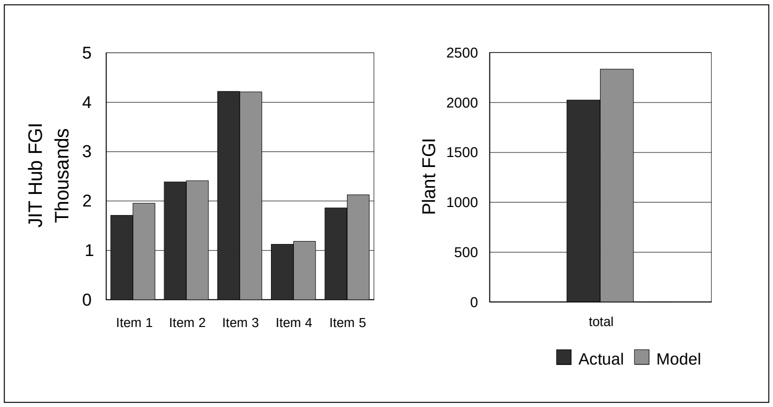

Figure 5: Comparison of FGI between simulation model and historical actuals.

of reducing the manufacturing lead time on inventory. We found that reducing the lead time by one day, on average, leads to 8% savings in total inventory. These results were fairly consistent across product families. Figure 7 depicts this sensitivity.

In summary, the study showed that delayed customization can significantly reduce finished goods inventory at the JIT hubs, and reduce costs for reshipping products without sacrificing serviceability (i.e., without increasing stockouts at the hubs). It also showed that moving towards consumptive pull and shortening the manufacturing and configuration lead times further reduces costs in the form of inventory reduction, which was a key step to solidify management support for eventual implementation of the configure-to-pull process design change.

4 Configure-to-Order for High-Volume High-Variety Products

The focus of the following study is on the inventory-service tradeoff in configure-to-order (CTO) manufacturing in IBM's personal computer value chain. It is part of a larger project that aimed at helping IBM's personal computer division to migrate from fixed configurations to a configure-to-order operation where customer orders are taken from the Internet. We have developed and applied an analytical trade-off model to evaluate three scenarios: assess the cost/benefit of a building-block based manufacturing operation; compare the forecasting of fixed configurations versus customer segment and attach-rate forecasting; and assess the effect of increasing product variety on inventory.

4.1 The Optimization Model

We consider a hybrid model, by which each end product is assembled to order from a set of components, which, in turn, are built to stock. In other words, no finished goods inventory is kept for any end product,

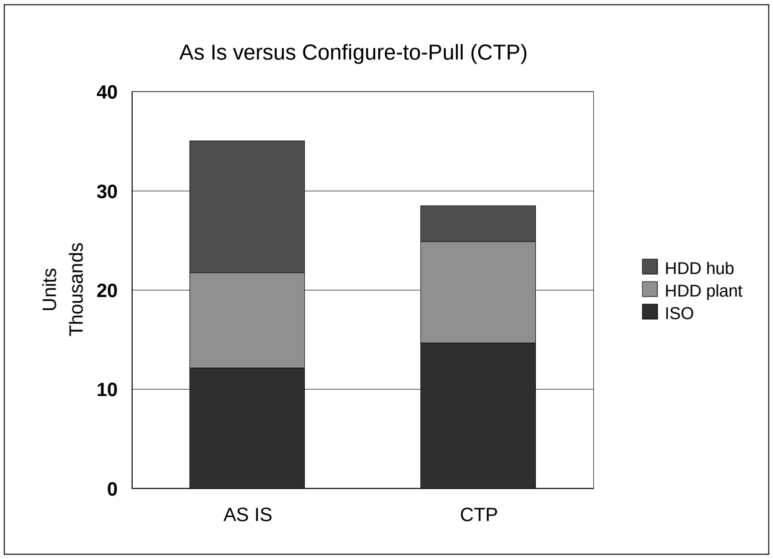

Figure 6: Configure-to-pull results in 69% reduction in Hub FGI with only 21% increase in plant FGI.

whereas each component has its own inventory, replenished from a supplier following a base-stock policy.

Each component inventory is indexed by i, $i \in \mathcal{S}$, where $\mathcal{S}$ denotes the set of all components. Associated with each component is a "store," where the inventory is kept. In the configure-to-order (CTO) environment, there is no pre-specified product "menu"; in principle, every order can require a distinct set of components. Let $\mathcal{M}$ denote the set of product/demand *families* that use the same set of components. For instance, $\mathcal{M} = \{$ low-end machines, high-end machines, servers $\}$; or $\mathcal{M} = \{$ individuals, small business, corporations $\}$.

Let $D_m(t)$ denote the demand associated with product family m in period t. Each order of type m requires a random number of units from component i, denoted as X_{mi}, which takes on non-negative integer values. Denote:

$$\mathcal{S}_m := \mathcal{S} - \{i : X_{mi} \equiv 0\}, \qquad \mathcal{M}_i := \mathcal{M} - \{m : X_{mi} \equiv 0\}.$$

That is, $\mathcal{S}_m$ denotes the set of components used in type m products, whereas $\mathcal{M}_i$ denotes all the product families that use component i. (Here, $X_{mi} \equiv 0$ means $\mathsf{P}(X_{mi} = 0) = 1$.)

The first step in our analysis is to translate the end-product demand into demand for each component. This is done through the bill-of-material structure for the products. Alternatively, component demand can be derived through forecast data on aggregated demand over market segments and the attach-rates of the components. (For example, 90% of products sold to large corporations, and 50% in the small business and consumer segments will use high-end processors, and so forth.)

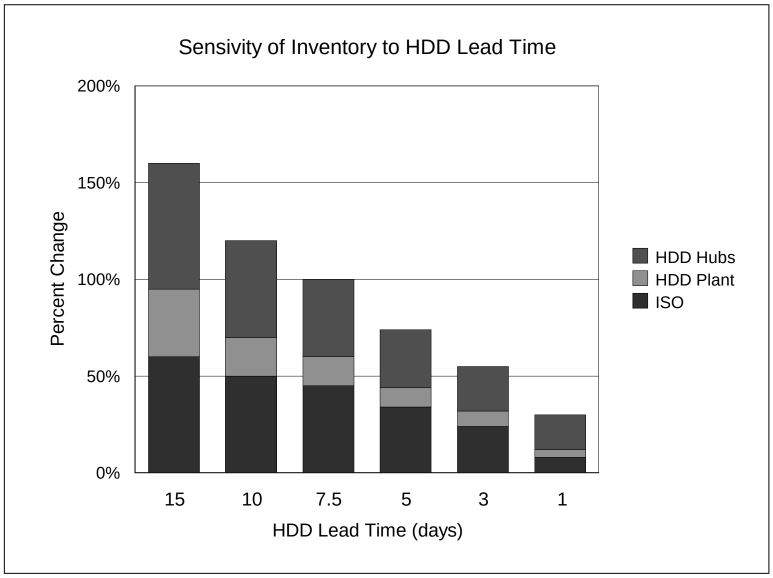

Figure 7: Sensitivity analysis results on reducing manufacturing lead times.

There are two kinds of lead times: those associated with the components (inbound) – the time for the supplier of component i to replenish to store i once an order is placed; and those associated with the end products (outbound) – including the time to process orders, the assembly/reconfiguration time, and the transportation time to deliver the order.

The next step is to compute the mean and the standard deviation of the demand over the in-bound leadtime for each component i, denoted μ_i and σ_i. (The out-bound leadtime is used to offset the time-shift in product orders. Hence, if the out-bound leadtime for a product is one week, then any planning involving orders for this product will have to be shifted a week earlier.)

We can now write the base-stock level for component i as $R_i = \mu_i + k_i\sigma_i$, just like the model in the last section, with k_i denoting the safety factor. With μ_i and σ_i as parameters derived from given data as outlined above, the decision on the base-stock levels is equivalent to the decision on the safety factors.

Our objective is to minimize the expected inventory budget (capital), subject to meeting the service requirement for each product family. The problem can be presented as follows:

$$\begin{aligned} \min \ & \sum_{i \in \mathcal{S}} c_i \sigma_i H(k_i) \\ \text{s.t.} \ & \sum_{i \in \mathcal{S}_m} r_{mi} \bar{\Phi}(k_i) \le \bar{\alpha}_m, \qquad m \in \mathcal{M} \end{aligned}$$

where $r_{mi} := \mathsf{P}(X_{mi} > 0)$ is the probability that product m requires component i; c_i is the unit cost of the on-hand inventory of component i; $\sigma_i H(k_i)$ is the expected safety stock of component i, where the H function is defined in (4); and $\bar{\alpha}_m = 1 - \alpha_m$ with α_m being the required service level for product family m.

4.2 Case Study: IBM's Personal Computer Value Chain

Here we describe a study which was part of a project aimed at the reengineering of IBM's personal computer value chain from a build-to-stock operation to a configure-to-order operation. To carry out the study, we developed two basic models: an "As-Is" model that is a reflection of the present build-to-stock operation with fixed configurations, and a "To-Be" model that is based on open configurations where the component inventory levels were generated by the algorithm described in the previous section. For both models, we aggregated the production-inventory system into two stages, the first stage consisting of the component replenishment process, and the second stage consisting of the assembly and order fulfillment process.

We identified three factors as the focal points of our study:

- *Manufacturing Strategy*. The "As-Is" operation versus the "To-Be" model.
- *Forecast Accuracy*. The accuracy of demand forecast at the end product level versus at the component level.
- *Product Variety*. The effect of mass customization on inventory as a result of direct sales over the Internet.

To study the first factor, we selected a high-volume product family that consisted of 18 finished products that were assembled from 17 components. We used existing bills-of-materials, unit costs, and procurement lead times to develop a detailed simulation model. Demands for each end product were generated statistically based on historical data. The inventoy buffers were set to meet a 95% service level requirement for all end products. We then used the product data and the statistically generated demand streams as inputs into the optimization model to determine the optimal base-stock levels for the component inventory.

Figure 8 shows the comparison between the "As-Is" and the "To-Be" model in the form of overall inventory investment. To protect proprietary information, the vertical axes were normalized with respect to the inventory investment of the "As-Is" model, which is 100. As expected, the inventory investment for end products was eliminated in the "To-Be" model. (The cost shown is due to WIP; the cost due to finished goods is nil.) The "As-Is" model, in contrast, keeps a significant amount of end-product inventory. On the other hand, the amount of component inventory is higher in the "To-Be" model, which is again expected, since the required service level of 95% is common to both models. Overall, the "To-Be" model reduced the overall inventory investment by roughly 30%. Both models used the same demand forecasts for end products.

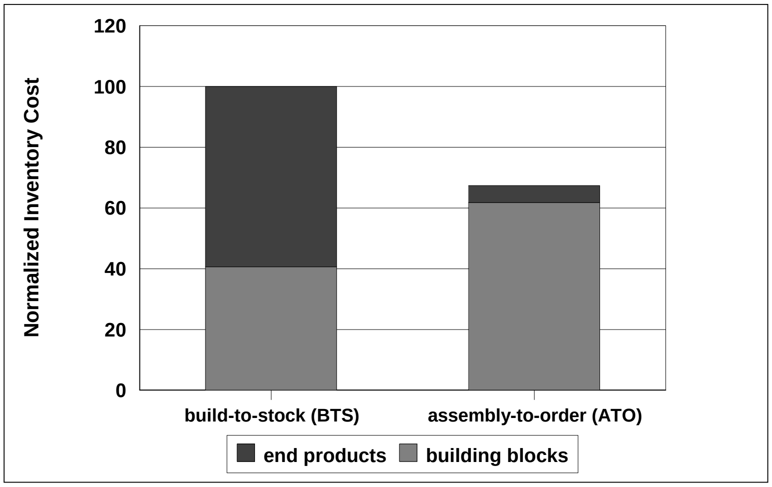

Figure 8: Comparison between Build-to-Stock ("As Is") and Assemble-to-Order ("To-Be").

In our study of the second factor, we evaluated the effect of forecast accuracy through sensitivity analysis. Figure 9 shows the overall inventory investment associated with three different levels of forecast accuracy. The first two columns repeat the comparison shown in the previous figure. The next two columns represent improved forecast errors, at 20% and 10%, achieved by switching to component forecasting in the "To-Be" model.

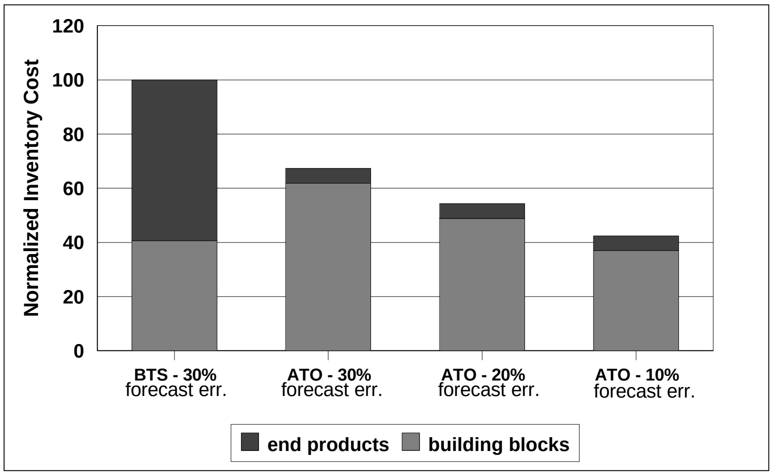

Figure 9: Effect of improving forecast accuracy.

Our study of the third factor aimed at analyzing the impact of higher product variety on inventory, with the motivation to support mass customization. In an Internet-based direct sales environment, the number of customer-configured products can be significantly larger than what was supported in the build-to-stock environment with fixed configurations. Figure 10 shows the inventory investments. The four columns on the left correspond to the current product set (1x), with scenario S1 as the "As-Is" model, and the other three scenarios as "To-Be" models at the current and improved forecast accuracy levels (scenarios S2–S4). The four columns on the right repeat these scenarios with a product set that is ten times larger in variety (10x). Table 1 summarizes all scenarios.

Scenario	Description	1x Cases	10x Cases
S1	"As-Is"	original product set, 30% forecast error, 90% service	ten times larger product set, $30\%\times\sqrt{10}$ forecast error at MTM level
S2	"To-Be"	forecast at MTM level, 30% forecast error, 90% service	ten times larger product set, forecast error as in S2(1x)
S3	"To-Be"	forecast at BB level, 20% forecast error, 90% service	ten times larger product set, forecast error as in S3(1x)
S4	"To-Be"	forecast at BB level, 10% forecast error, 90% service	ten times larger product set, forecast error as in S4(1x)

Table 1: Summary of the scenarios used to study the effect of product variety.

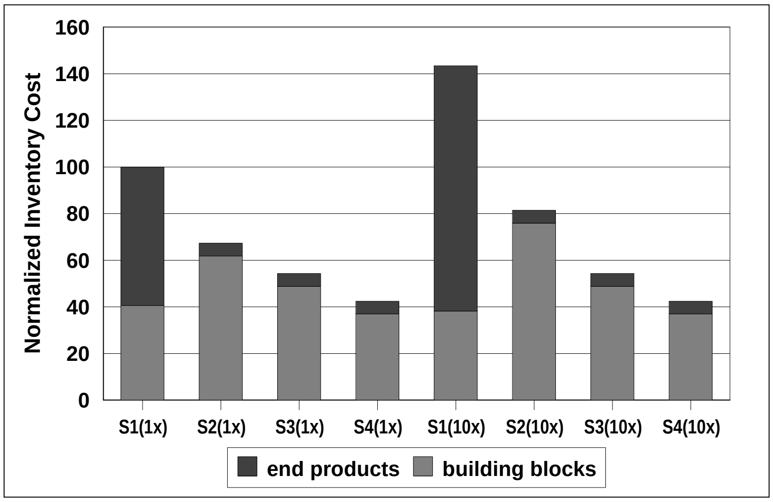

Figure 10: Effect of product variety on inventory.

Observe that as the product variety increases, a significantly higher level of inventory is required in the "As-Is" model. This is because forecast accuracy deteriorates when the end products proliferate (i.e., larger varieties result in smaller demand volumes). On the other hand, in the "To-Be" environment, the increase in inventory investment is very modest. This is because the proliferation of end products will have minimal effect on the forecast accuracy at component level, due to parts commonality. This strongly supported the fact that the building-block model is the right process to support a direct-sales operation.

5 Semiconductor Value Chains

To meet high levels of service with long lead times and uncertain demands, semiconductor manufacturers often hold large inventories despite the risk and expense. Xilinx, a large manufacturer of application specific integrated circuits, uses design strategies like postponement to better control their inventory expenses [1]. Under the postponement strategy, inventory is held at in a generic, non-differentiated form in the die bank and is differentiated when the demand is better known. To take full advantage of the postponement strategy, we worked with Xilinx to develop an optimization model that determines the inventory levels in the intermediate and finished goods stocking points to allow for the best service at the lowest cost.

Xilinx, like many semiconductor manufacturers, contracts out most of the manufacturing. Fabrication is performed at vendors in Taiwan and Japan. The wafers are then shipped to assembly vendors in Korea, Taiwan, and the Philippines, where they are held in die bank inventory until needed. Although separate die banks are maintained at each assembly vendor, transfer of die between vendor locations is done as needed. Thus, the die bank is modeled as a single inventory stocking location. Assembled parts are usually shipped to Xilinx facilities in San Jose or Ireland for testing, where they are held in finished goods inventory.

Since die bank inventory is more generic, it serves as an inventory postponement point. The die bank inventory is managed using a wafer starts planning package. Given a desired inventory target, this package determines the total amount to start at the fabrication contractor using a monthly base-stock policy. Finished goods inventories are managed using a planning system based on assembly starts, with separate systems being run in Ireland and San Jose. In the problem formulation below, we discuss how such parts affect the results and how we include such parts in the actual implementation of the model at Xilinx.

As with most inventory decision problems, choosing the appropriate objective function is difficult and highly dependent on the business environment. The key measure of service at Xilinx was total costed "delinquencies" (costed backorders) across all parts, with each part having a different unit delinquency cost. The individual unit delinquency costs were determined using a percentage of unit revenue based on input from the sales department using the following factors: stage in product life-cycle, impact of product availability on future sales of other products, competitive nature of product, and proportion of the product's demand due to key customers. Separate inventory budgets were to be specified for different echelons (die bank versus finished goods) and for different locations within an echelon (San Jose finished goods and Ireland finished goods).

5.1 The Optimization Model

The inventory system consists of two echelons. The upstream echelon is die bank inventory and the downstream is finished goods (FG) inventory. The FG inventory is distributed at a set of locations, indexed by $i \in \mathcal{M} := \{1, ..., M\}$. Each location i supplies a set of end products, $j \in \mathcal{S}_i := \{1, ..., N_i\}$. A separate inventory is kept for each product to satisfy its own demand stream. Let D_{ij} denote the demand (per time unit) for product j at location i, which is assumed to follow a normal distribution. For simplicity, we shall refer to such product at a given location as type ij product.

Let 0 index the die bank location. There are N_0 types of die, each with its own inventory, indexed by $d \in \mathcal{S}_0 := \{1, ..., N_0\}$. The relationship between the two stages is a one-to-many mapping: each type of die is used to make one or more end products, but each end product uses only a single type of die. Hence, for each $d \in \mathcal{S}_0$, let $\mathcal{S}_d$ denote the set of end products that use type d die. Then, the demand for type d die is: $\sum_{(i,j)\in\mathcal{S}_d} D_{ij}$. (One unit of end product uses one unit of die.)

Let L_d be the production leadtime for each die d. Let L_{ij} be the leadtime to transform die d into type ij product. The replenishment lead time for FG inventory (end products) is this nominal leadtime, L_{ij}, plus a delay time which takes into account the possible stockout of die bank inventory. This *actual* leadtime, denoted $\tilde{L}_{ij}$, has expected value

$$\mathsf{E}(\tilde{L}_{ij}) = \mathsf{E}(L_{ij}) + \tau_d p_d, \tag{7}$$

where p_d is the stockout probability of type d die inventory used to make type ij product, and τ_d is the expected additional delay when this stockout occurs. Both quantities depend on the inventory levels of type d die. The derivation of the term $\tau_d p_d$ in (7) will be deferred to the next section.

Given information on the demand and leadtimes, we can derive the means and the standard deviations of demand over leadtime: μ_d and σ_d for each die bank location d, and $\tilde{\mu}_{ij}$ and $tsig_{ij}$ for each product ij. Refer to the detailed derivation in [4].

For each ij product, let h_{ij} be the unit inventory holding cost (per time unit), and let s_{ij} be the unit backorder cost. For each type d die, let h_d be the unit inventory holding cost. The decision variables are R_{ij}, the base-stock level for each type ij product's FG inventory, and R_d, the base-stock level for each type d die inventory. As in the last section, we relate these to the safety factors k_d and k_{ij} as follows:

$$R_d = \mu_d + \sigma_d k_d, \qquad R_{ij} = \tilde{\mu}_{ij} + \tilde{\sigma}_{ij} k_{ij}; \tag{8}$$

and treat the the safety factors as decision variables. We have the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{M}} \sum_{j\in\mathcal{S}_i} s_{ij}\tilde{\sigma}_{ij} G(k_{ij}) && (9) \\ \text{s.t.} \quad & \sum_{j\in\mathcal{S}_i} h_{ij}\tilde{\sigma}_{ij} H(k_{ij}) \le C_i, \quad i \in \mathcal{M}; && (10) \\ & \sum_{d\in\mathcal{S}_0} h_d \sigma_d H(k_d) \le C_0. && (11) \end{aligned}$$

Here, s_{ij} is the penalty cost for each unit of backordered ij product; h_{ij} and h_d are unit inventory costs; $\tilde{\sigma}_{ij} H(k_{ij})$ and $\sigma_d H(k_{ij})$ are expected safety-stock inventories, where H is the function defined in (4); $\tilde{\sigma}_{ij} G(k_{ij})$ is the expected backorders, where G relates to H as follows:

$$G(x) := \mathsf{E}[Z - x]^+ = x - H(x) = \phi(x) - x\bar{\Phi}(x), \qquad (12)$$

where $\bar{\Phi}(x) := 1 - \Phi(x)$.

To minimize overstocking risk for various products, managers often want to be able to place upper limits on the level of inventory allowed. For example, for a mature product nearer the end of its product life, setting a high inventory target would not be wise. Thus, we add the following constraints on the safety factors to the optimization problem in (9) - (11):

$$k_{ij} \le \bar{k}_{ij}, \quad j \in \mathcal{S}_i, i \in \mathcal{M}; \qquad k_d \le \bar{k}_d, \quad d \in \mathcal{S}_0; \qquad (13)$$

where $\bar{k}_{ij}$ and $\bar{k}_d$ are positive upper limits.

5.2 Case Study: Xilinx Semiconductor Value Chain

One key question we wished to investigate was how the total inventory holding cost budget $C_0 + C_1$ should be split between the die bank and end products in order to minimize the delinquency cost. Figure 11 shows the optimal delinquency cost z^* as a function of $C_0/(C_0 + C_1)$, i.e., the relative amount of inventory holding cost budget allocated to the die bank, for four scenarios. The inventory holding cost for all die types was held constant at $h_d = 1.0$ across scenarios, but the values of h_{ij} are changed. For the four scenarios we used h_{ij} values of 1.0, 1.75, 5.25, and 10.50 and total inventory holding cost budget of 3,000, 4,000, 8,000, and 12,000 dollars.

From Figure 11, we learn that most of the inventory budget should be allocated to end products. For instance, when $h_{ij} = h_d = 1.0$, the optimal operating point is 0.2, suggesting that 20 percent of safety stock should be kept at the die bank, and 80 percent should be kept in finished goods. Table 2 shows the delinquency cost z^* and the optimal split C_0^* and C_1^* of the total inventory cost budget for each of the scenarios under study. In additional to reporting the budgets, the table also shows the total safety stock

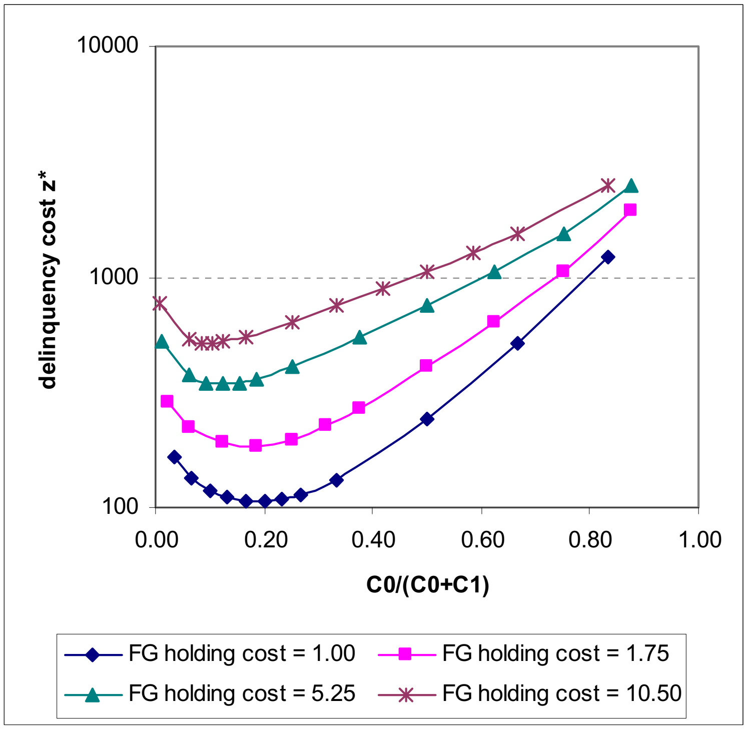

Figure 11: Delinquency cost vs. relative budget allocated to the die bank.

in units allocated to the die bank and finished goods. We observe that the while percent of inventory cost budget allocation to die bank actually decreases as the cost of holding FG stock increases, the number of *units* allocated to the die bank $\mathsf{E}(I_0^*)$ versus the number of units allocated to finished goods $\mathsf{E}(I_1^*)$ increases. For example, when $h_{ij} = 1.0$, about 20 percent of safety stock inventory should be kept in the die bank, whereas when $h_{ij} = 10.50$ the amount in die bank should be increased to roughly 55 percent. When h_{ij} is large, allocating a larger amount of inventory to the die bank is optimal since multiple units of die can be stocked at the same cost as, say, one unit of finished goods.

Table 2: Optimal split of safety stock inventory between die bank and finished goods.

					die (in units)	finished goods (in units)
h_{ij}	$C_0 + C_1$	C_0^*	C_1^*	z^*	$\mathsf{E}(I_0^*)$	$\mathsf{E}(I_1^*)$
1.00	3,000	600	2,400	106	605	2423
1.75	4,000	750	3,250	185	755	1880
5.25	8,000	1,000	7,000	343	1005	1349
10.50	12,000	1,250	10,750	519	1255	1035

Figure 12 shows the percent of safety stock (in units) allocated to die bank as a function of FG holding cost h_{ij}, for three different values of the FG backorder cost s_{ij}. The amount kept in the die bank increases monotonically with h_{ij}. When s_{ij} is high, the relative amount of safety stock that should be kept at the die bank becomes smaller. For example, in the specific case of $h_{ij} = 10.50$, it is optimal to keep 64 percent die stock when $s_{ij} = 0.875$, 56 percent when $s_{ij} = 1.75$, and 48 percent when $s_{ij} = 4.25$.

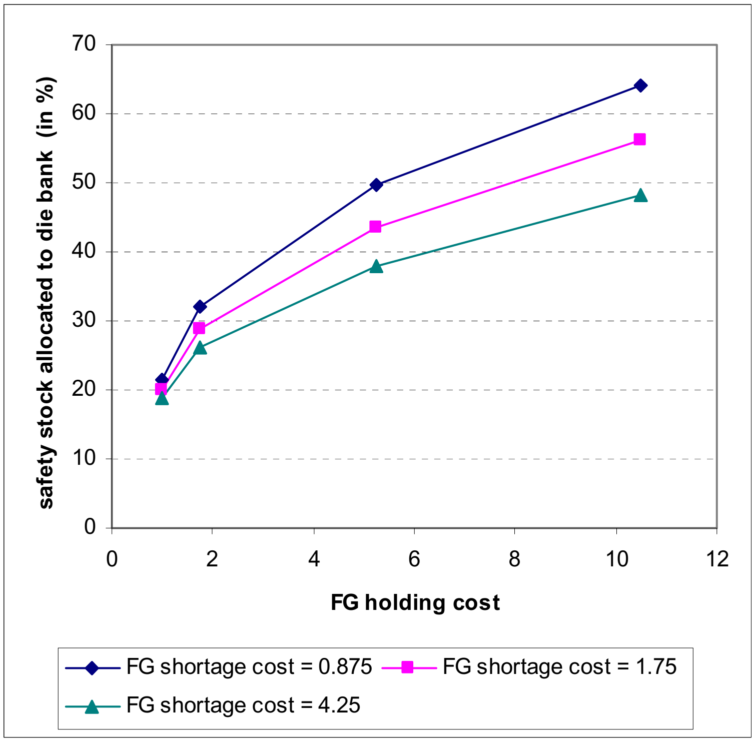

Figure 12: Relative amount of safety stock (in units) allocated to the die bank.

In order to project the business improvements that can be achieved with optimization, we compared the optimized policy to the original policy implemented at Xilinx when we began the development of the model. In the original policy, all finished goods buffers (for parts that were build-to-stock) had the same target days-of-inventory, and all die bank buffers had the same target days-of-inventory. We used a full product set in order to compare the original policy and the optimized policy. The product set consisted of 104 die bank parts, 314 finished goods parts held at the Xilinx facility in Ireland, and 1194 parts held at the facility in San Jose. Comparisons between the two policies were made in two different ways. First, the total inventory holding cost budget was held at the same level for both policies, and the projected improvement in delinquency cost was found. Thus, the inventory holding cost that resulted from the original policy was calculated and used as the holding cost budget constraint for the inventory optimization. The ideal split of this inventory holding cost budget between die bank and finished goods was found by running the inventory optimization under a number of different splits. Under the resulting inventory targets, the total delinquency cost was reduced by 54%. The huge improvement was primarily due to reallocation of the inventory among the various finished goods parts. Inventory targets for stable finished goods parts with low delinquency costs and high holding costs were set lower, while inventory targets for less stable parts with higher delinquency costs were set higher.

Second, the total delinquency cost was held constant for both policies and the projected improvement in inventory holding cost budget was found. Then, the inventory optimization was run with a number of different settings of the inventory holding budget until the budget was found that yielded approximately the same delinquency cost. The optimized policy was found to yield a policy with a 19.9% reduction in overall inventory holding cost (which includes work-in-process) and a 51.2% reduction in inventory holding cost budget when only safety stock was considered.

6 Summary

We described optimization models of value chains intended to analyze the trade-offs between inventory and service levels in high-technology value chains. We focused on three value chain architectures, complex configured hardware value chains, computer assembly value chains, and semiconductor value chains, and discussed case studies where we utilized the analytical models to aid in the determination of operational inventory targets. These models capture the true multi-tier nature of industrial value chains that are subject to non-stationary demands, stochastic lead times, and attach-rate uncertainty. The application examples also illustrate the importance of not only focusing on optimal inventory planning and control, but also exploring alternative value chain designs such as delayed product differentiation and configure-to-order to seek operational and financial performance improvements.

Acknowledgements

The authors would like to thank Brian Eck who was instrumental in conducting the case study for IBM’s hard disk drive value chain. We also thank Gerald Feigin for his invaluable contributions to the development and implementation of the optimization engine described in section 3. Finally, we acknowledge the contributions of Alex Brown, Luis Diaz, and Raja Petrakian at Xilinx who helped tremendously in identifying, refining, and implementing the semiconductor inventory model.

References

[1] BROWN, A., LEE, H., AND PETRAKIAN, R., Xilinx improves its Semiconductor Supply Chain using Product and Process Postponement, *Interfaces*, 30 (4), 2000, 65-80.

[2] BROWN, A.O., ETTL, M., LIN, G., PETRAKIAN, R. AND YAO, D.D., Inventory Allocation at a Semiconductor Company: Modeling and Optimization. In: *Supply Chain Structures: Coordination, Information, and Optimization,* J.S. Song and D.D. Yao (eds.), Kluwer, 2001, Chapter 9, 283-309.

[3] CHENG, F., ETTL, M., LIN, G., AND YAO, D.D., Inventory-Service Optimization in Configure-to-Order Systems. *Manufacturing and Service Operations Management.* **4** (2002), 114-132.

[4] ETTL, M., FEIGIN, G.E., LIN, G.Y., AND YAO, D.D., A Supply Network Model with Base-Stock Control and Service Requirements, *Operations Research*, **48** (2000), 216-232.

[5] LIN, G., ETTL, M., BUCKLEY, S., BAGCHI, S., YAO, D.D., NACCARATO, B.L., ALLAN, R., KIM, K., AND KOENIG, L., Extended-Enterprise Supply-Chain Management at IBM Personal Systems Group and Other Divisions, *Interfaces,* **30** (2000), 7-25.

[6] SONG, J.S. AND YAO, D.D., (eds.) *Supply Chain Structures: Coordination, Information and Optimization*, Kluwer Academic Publishers, Norwell, 2001, 7-44.

[7] TAYUR, S., GANESHAN, R., AND MAGAZINE, M., (eds.) *Quantitative Models for Supply Chain Management*, Kluwer Academic Publishers, Norwell, 1999, 553-584.

[8] ZIPKIN, P., *Foundations of Inventory Management*, Irwin/McGraw-Hill, New York, 2000.