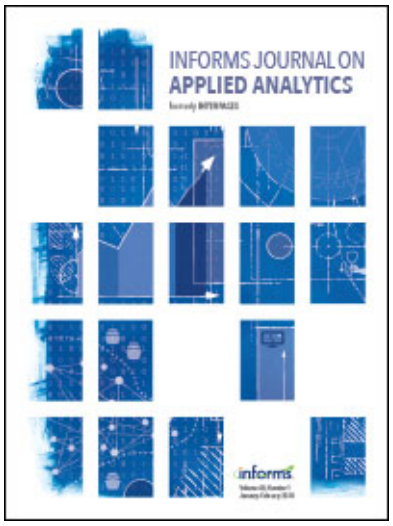


INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Intel Realizes $25 Billion by Applying Advanced Analytics from Product Architecture Design Through Supply Chain Planning

John Heiney, Ryan Lovrien, Nicholas Mason, Irfan Ovacik, Evan Rash, Nandini Sarkar, Harry Travis, Zhenying Zhao, Kalani Ching, Shamin Shirodkar, Karl Kempf

To cite this article:
John Heiney, Ryan Lovrien, Nicholas Mason, Irfan Ovacik, Evan Rash, Nandini Sarkar, Harry Travis, Zhenying Zhao, Kalani Ching, Shamin Shirodkar, Karl Kempf (2021) Intel Realizes $25 Billion by Applying Advanced Analytics from Product Architecture Design Through Supply Chain Planning. INFORMS Journal on Applied Analytics 51(1):9-25. https://doi.org/10.1287/inte.2020.1067

http://pubsonline.informs.org/journal/inte

INFORMS JOURNAL ON APPLIED ANALYTICS
Vol. 51, No. 1, January–February 2021, pp. 9–25
ISSN 0092-2102 (print), ISSN 1526-551X (online)

THE FRANZ EDELMAN AWARD
Achievement in Operations Research

Intel Realizes $25 Billion by Applying Advanced Analytics from Product Architecture Design Through Supply Chain Planning

John Heiney,[a] Ryan Lovrien,[b] Nicholas Mason,[b] Irfan Ovacik,[a] Evan Rash,[b] Nandini Sarkar,[a] Harry Travis,[b] Zhenying Zhao,[a] Kalani Ching,[a] Shamin Shirodkar,[a] Karl Kempf[b]

[a] Supply Chain Decision Solutions, Global Supply Chain, Intel Corporation, Chandler, Arizona 85226; [b] Decision Engineering, Data Platforms Group, Intel Corporation, Chandler, Arizona 85226

Contact: john.p.heiney@intel.com (JH); ryan.c.lovrien@intel.com (RL); nicholas.mason@intel.com (NM); irfan.m.ovacik@intel.com (IO); evan.a.rash@intel.com (ER); nandini.sarkar@intel.com (NS); harry.s.travis@intel.com (HT); zhenying.zhao@intel.com (ZZ); kalani.ching@intel.com (KC); shamin.a.shirodkar@intel.com (SS); karl.g.kempf@intel.com, https://orcid.org/0000-0001-9860-5419 (KK)

Received: October 6, 2020
Accepted: October 6, 2020

https://doi.org/10.1287/inte.2020.1067

Abstract. Due to its scale, the complexity of its products and manufacturing processes, and the capital-intensive nature of the semiconductor business, efficient product architecture design integrated with supply chain planning is critical to Intel's success. In response to an exponential increase in complexities, Intel has used advanced analytics to develop an innovative capability that spans product architecture design through supply chain planning with the dual goals of maximizing revenue and minimizing costs. Our approach integrates the generation and optimization of product design alternatives using genetic algorithms and device physics simulation with large-scale supply chain planning using problem decomposition and mixed-integer programming. This corporate-wide capability is fast and effective, enabling analysis of many more business scenarios in much less time than previous solutions, while providing superior results, including faster response time to customers. Implementation of this capability over the majority of Intel's product portfolio has increased annual revenue by an average of $1.9 billion and reduced annual costs by $1.5 billion, for a total benefit of $25.4 billion since 2009, while also contributing to Intel's sustainability efforts.

Keywords: product design optimization • supply chain planning optimization • design and planning integration • problem decomposition • mixed-integer programming • genetic algorithms • device physics simulation • Edelman Award • semiconductor manufacturing

Intel Corporation

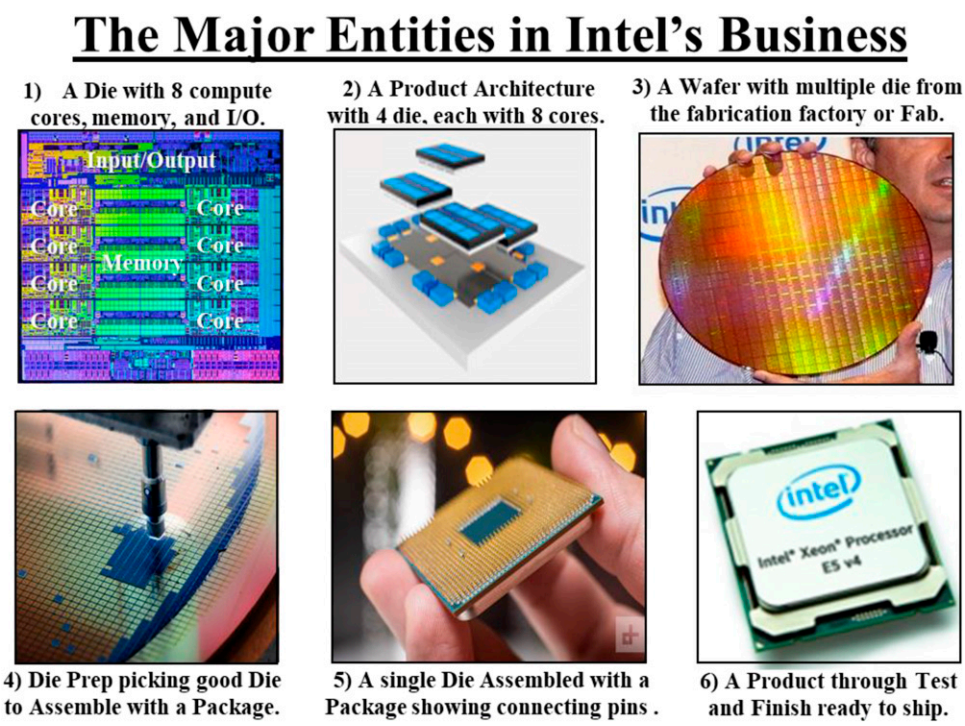
The integrated circuits designed and produced by Intel Corporation have been a driving force in the information revolution for over 50 years. Desktop and laptop computers have contributed to large productivity gains in the office and the factory. Modern supercomputers have revolutionized the scientific process by enabling research to move beyond the constraints of physical experimentation. Perhaps most importantly, servers and networking have transformed many activities of importance to global society. This list includes telecommunications for individuals and businesses, banking and finance, commercial and recreational transportation, manufacturing and services industries, wholesale and retail businesses, medicine, education, and agriculture. Recently, Intel-driven computers have enabled the advent of cloud computing, big data, and the practical resurgence of artificial intelligence.

Also beneficial is how Intel has achieved these technical and economic results. Intel is a world leader in the use of green power and conflict-free minerals and has an ongoing focus on reducing water pollution and greenhouse gas emissions.

An Operational Decision Process

Intel's business is designing, manufacturing, and supplying state-of-the-art semiconductor products. The semiconductor industry uses several technical terms to describe these processes (Figure 1). A "wafer" is a thin circular disk of silicon 200 millimeters (mm) to 300 mm in diameter. Arrays of integrated circuits are fabricated on each wafer during manufacturing. A small piece of the fabricated wafer, which contains an entire integrated circuit, is called a "die" and is comprised of one or more computational cores, memory, and input/output capability. "Die yield" refers to the number of functional die on a wafer at the culmination of the fabrication process determined by "sorting" into functional and nonfunctional categories. Each wafer is sawed into a number of die in "die prep," the nonfunctional die are discarded, and each functional

Figure 1. A Visual Representation of the Entities Involved in Intel's Business



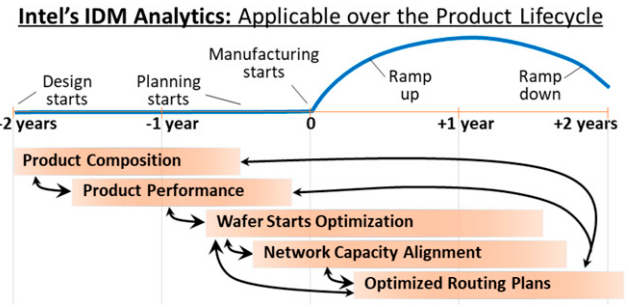
Note. Included are a typical die and product architecture, followed by the basic manufacturing sequence from a wafer containing fabricated die, a wafer with die tested for functionality (sort) and sawed out of the wafer (die prep), a die assembled with a package, and a tested and marked (finish) product.

die is "assembled" with a "package." This is a protective container with external connectors to provide an attachment to printed circuit boards. The die assembled with a package is Intel's "product," and once the combination is tested and labeled, it is ready to ship. Because many of Intel's microprocessor products contain multiple die in one package, each microprocessor product has an "architecture" specifying the number and types of die to include. The performance of a die is determined by the number of computational cores, their execution speed (higher is better), and the power they consume during execution (lower is better). The performance of a microprocessor product is determined by its architecture and is a function of the performance of the die included. Each microprocessor product is available as a set of stock-keeping units (SKU-set) with a range of performance for different applications. A SKU-set supports selling a microprocessor product into multiple markets to maximize revenue. Other important financial considerations include engineering cost and the cost of manufacturing the product. The engineering cost with which we are concerned is associated with developing the architecture, not the detailed design of each die. For manufacturing, the cost is fabricating the die, acquiring the package, and assembling the two, followed by testing. We are especially concerned with the efficiency of using every die on every wafer to supply a saleable product across the architectures of our entire product portfolio.

As one the few remaining integrated device manufacturers (IDMs), Intel holds a strong position in the semiconductor industry. Most semiconductor companies are "fabless," which means they design their own products but do not own manufacturing operations. Other companies in the industry focus only on manufacturing and build products designed by the fabless companies. Intel is one of the few companies that designs and manufactures its products. As a result, at any point in time, Intel is delivering hundreds of different products to its customer base through its large and complex supply chain, while simultaneously designing the next generation of products to meet its customers' needs. This gives Intel an important opportunity to follow an integrated five-step decision process for designing, manufacturing, and supplying its products (Figure 2).

Development of a new product with a new computational core begins with product architecture design, which starts roughly two years ahead of the delivery of products to customers. The goal is to find the best balance between product features to satisfy customer requirements, the cost of engineering a SKU-set with those features, and the cost of manufacturing that SKU-set in sufficient volume to satisfy demand. We decompose the decision processes into two steps.

Figure 2. (Color online) An Integrated Sequence of Modules Is Necessary to Run Intel's Business



Notes. The two product architecture design modules produce a SKU-set composition with optimized performance, which is passed to the three supply chain planning modules. The result is a set of products that are manufactured and shipped to customers.

In the product composition step, utilizing forecasted demand and manufacturing die yields, architects decide on the number and type of die that should be assembled together to create multiple-die products that align to market requirements with minimum engineering and manufacturing cost. The resulting decision is a SKU-set architected around a new computational core.

In the product performance step, the performance of the members of the SKU-set is the focus. The better the performance goal of the individual products in the SKU-set, the higher the revenue possibilities, but also the higher the manufacturing cost. The resulting decision is the specification of the performance goals of the SKU-set based on balancing revenue and cost.

Supply chain planning (SCP) processes overlap with product architecture design by roughly one year ahead of the delivery of products to customers. The goal of SCP is to minimize wafer starts and optimally route them all the way to finished products to meet customer demand, while effectively utilizing our high-investment internal factory network. We decompose the SCP decision processes into three steps.

Wafer starts optimization is the third step in the overall design and planning decision process. It focuses on the wafer fabrication and sort process steps and ensures that capacity is optimally used to support customer demand and minimize costs. Wafer fabrication is the most capital-intensive and constrained process step; therefore, it is optimized first.

In network capacity alignment, the fourth step, we determine the capacity allocations for the rest of the supply chain. This step requires a detailed capacity analysis in the internal factories and a request-response process with the external factories.

In the fifth and last step, optimal routing plans are finalized for the die-prep, assembly, and test and finish steps to complete the planning process. The output of this step not only provides guidance to the rest of the supply chain but is also used across the enterprise for financial projections and customer order confirmation.

The first two steps are sequentially performed over a two-year period. The third, fourth, and fifth steps are run sequentially each month over the two-to-four-year production life of the SKU-set. These five steps are integrated in a number of ways to optimize the overall system under a variety of conditions. The initial integration occurs in the normal operation of the system: the output of the first step is required input to the second step, the output of the second is input to the third, and so on. This sequential process can involve feedback. The execution of step 2 may point to a change needed in step 1, which is then rerun; a reallocation of manufacturing capacity between product lines in step 3 may require re-execution of step 1, restarting the entire sequence. Additional common inputs to each of the five steps include the details of forecasted market demand and forecasted manufacturing yields. These forecasts change over time and are eventually replaced by actual data. Minor changes will reactivate step 5 or steps 4 and 5. Major changes can force restarting the entire sequence beginning with a re-execution of step 1, followed by re-execution of step 2, and so on. Surprise introduction of a new feature by a competitor, or demand for an additional feature by a major customer, might have the same effect and restart the entire sequence from step 1.

Introduction to the Business Problem

Faster, better, and more integrated decision making is increasingly important to Intel's five-step decision process. Over the past decade, our products have become more complex. Initially, we designed single-die products with one processing core. Today we are shipping products consisting of multiple die, each with multiple processing cores. Over the same period, our manufacturing processes and product routings have become much more complex with more manufacturing steps, longer throughput times, and many more capacity-related complications. These complexities have rendered our previous generation of decision tools inadequate to run the business. Regardless of the complexities that we face, our customers expect more frequent product introductions, faster product delivery, and more flexibility than ever before from Intel to meet their needs. Meeting and exceeding these customer expectations requires seamless integration of product architecture design and supply chain planning, which is only possible through more nimble decision processes supported by better, faster decision tools. The advanced analytics that we have developed, integrated, and deployed provide a decision support system that has

realized the required decision environment. In doing so, we have faced four primary sources of complexity: manufacturing, product architecture design, demand, and supply chain planning.

Manufacturing Process Technology Complexity

Many of our microprocessor products contain as many as 10 billion transistors, each of which is made up of approximately 40,000 atoms. Precisely and repeatably positioning individual atoms at the scale needed to manufacture our semiconductors is currently impossible. Hence, volume production of any individual product yields a range of values for each product characteristic of interest, such as functioning computational cores, power consumption, and execution speed (Figure 3).

Some die perform as intended, some have minor deficiencies, and some do not work at all. Any batch of die will have a range of execution speeds, some slower and others faster than intended. Some will consume more power, some less. An ongoing challenge is to accurately forecast the results of this variability prior to actual production for use in the product architecture design process.

Handling manufacturing variability continues to be important in later stages of the product life cycle. Supply chain planning and execution processes account for manufacturing variability by holding right-sized inventory at the appropriate locations and implementing delayed differentiation and postponement strategies.

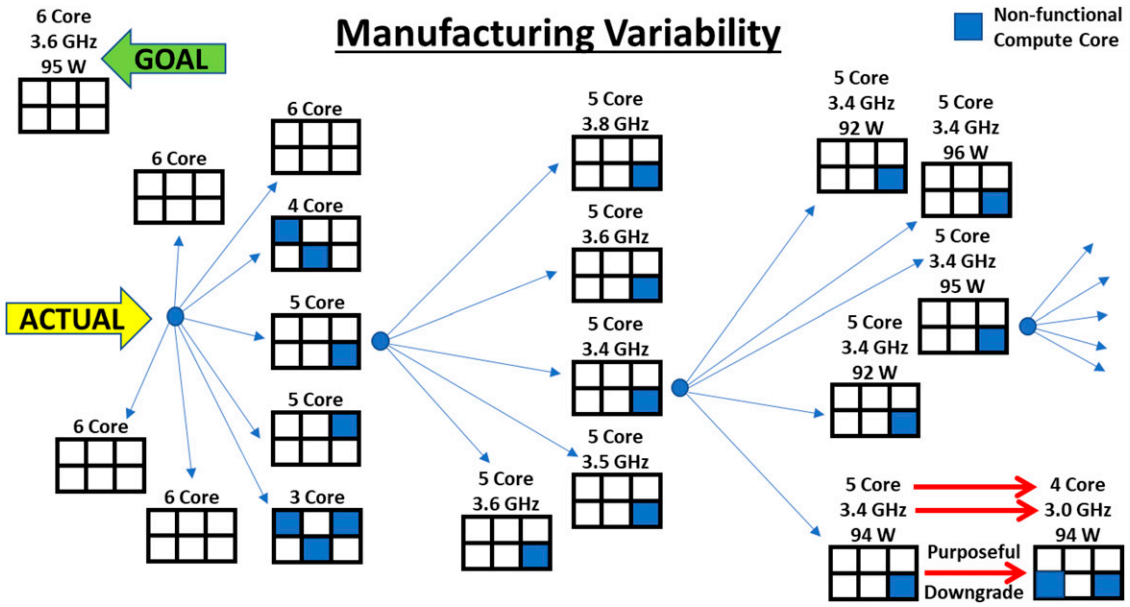
In addition to this inescapable variability, sustaining Moore's law (Moore 1965) by manufacturing exponentially smaller and smaller die on a two-to-three-year cadence has become increasingly difficult from a physics perspective. Production equipment has become sophisticated and exponentially more expensive, and the number of processing steps for more complex transistor technologies has increased greatly. These technological advances have led to impressive increases in product functionality and execution speed, as well as decreased cost per transistor for our customers.

However, the time and cost to develop the next process in the Moore's law sequence has risen dramatically, as has manufacturing throughput time. Long lead times and expensive equipment dramatically increase the risk of building the wrong product in the wrong volume at the wrong time, making integrated decisions in product architecture design and supply chain planning even more critical to Intel's health.

Product Architecture Design Complexity

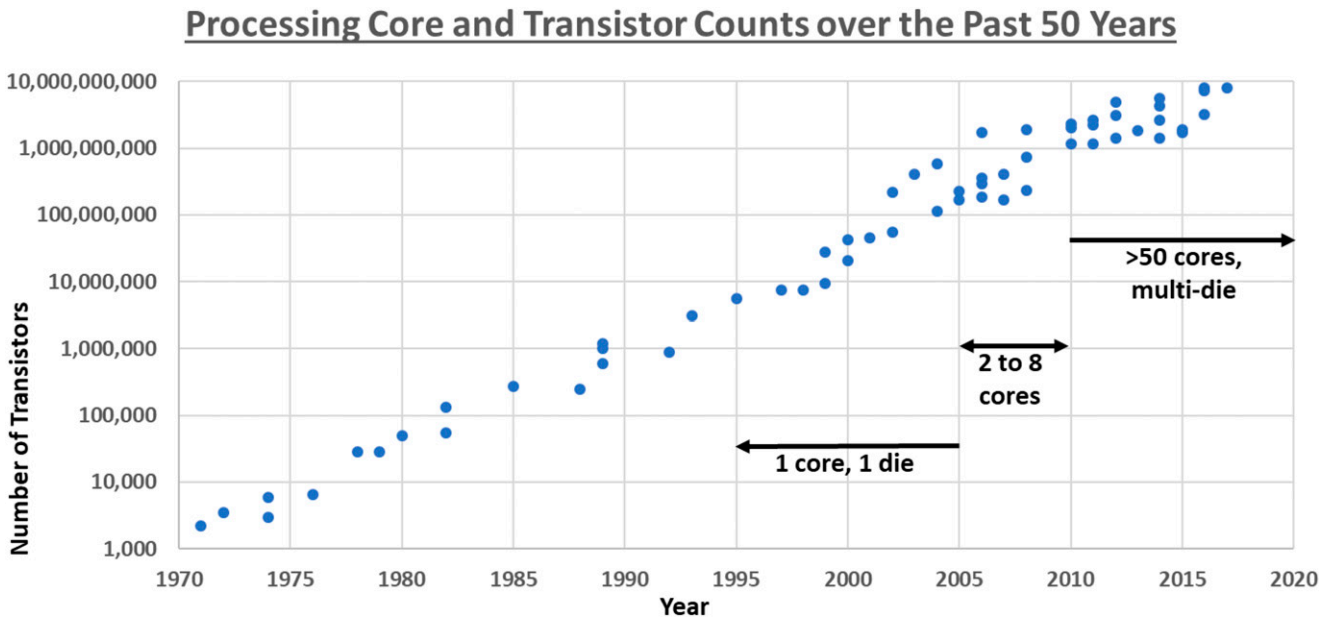
Figure 4 shows the trajectory of increasing product complexity. For roughly 30 years, our microprocessor products contained one silicon die with one computational core, while transistor counts grew from a

Figure 3. (Color online) Example Showing Variability Due to Manufacturing Process Complexity



Notes. Intel's manufacturing GOAL was to make six-core, 3.6-gigahertz (GHz) speed, 95-watt power devices. The ACTUAL results achieved contained many six-core chips, but some with less cores working (left side of diagram). Some devices ran faster or slower than 3.6 GHz (middle of diagram). Some devices consumed more or less power than 95 watts (right side of diagram). Note that we can purposefully "downgrade" devices to satisfy demand, as we show in the lower right of the diagram. We can selectively turn off cores (but not reanimate nonfunctional cores) and turn down speed (but not turn it up).

Figure 4. (Color online) Release Years of Microprocessor Examples by Number of Processing Cores and Total Number of Transistors



Notes. For the first 35 years of developing microprocessors, Intel products had one core on one die. This was followed by a 5-year period consisting of products with two, four, six, and eight cores. In the most recent 10 years, complexity has continued to grow exponentially in both number of cores and total number of transistors.

few thousand to hundreds of millions. The insatiable demand for increasingly more powerful microprocessor products resulted in multiprocessor architectures containing one die with two cores followed by four, six, and eight cores. Total transistor counts grew to exceed one billion. Today we are shipping products with architectures containing multiple die, each with multiple computational cores in packages often including additional input/output, graphics, and memory die (Figure 1). Core counts per product exceed 50, and total transistors exceed 10 billion. Products of this complexity take more time to design, validate, and manufacture.

Demand Complexity

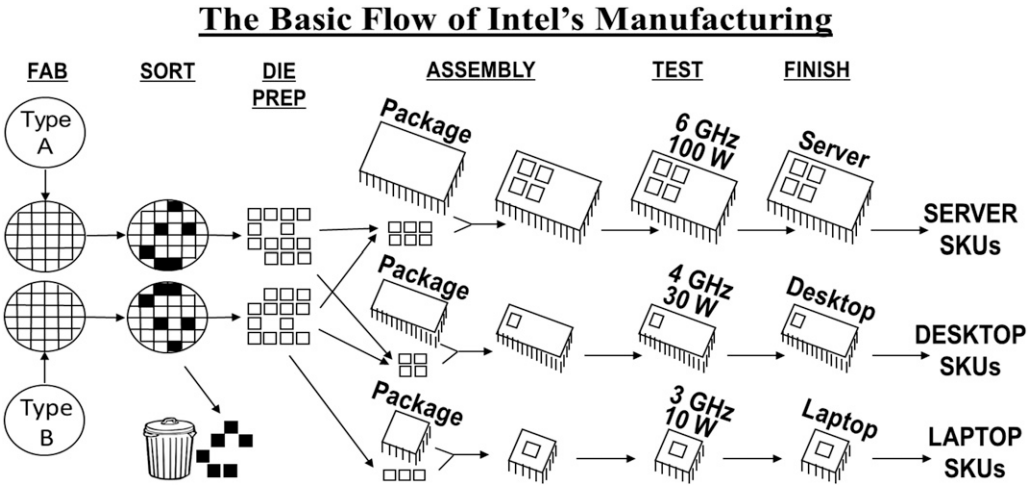
Another vector of complexity stems from our customer base, which requires a wide range of product characteristics. The biannual development of a more powerful computational core stimulates a spectrum of demand. At one extreme, customers who build supercomputers are interested in a product with 64 of the new cores, while other customers build 4-core servers for small businesses. The Intel sales and marketing organization is happy to forecast sales for all core counts in between, forming a diverse SKU-set to appeal to all markets. However, we do not have enough engineers to custom design each individual product in such a set, and it is not economically viable to mass produce 64-core products with 60 cores disabled to serve the 4-core market. Novel approaches to designing product architectures and determining the performance of each product in a SKU-set are needed to address this complex situation.

Supply Chain Planning Complexity

To meet the market demand for powerful multiple-die products using ever-more sophisticated manufacturing technologies and increased variability, Intel must effectively leverage its $100 billion internal factory assets and partner with outside manufacturers and component suppliers to meet cost and quality objectives in a timely manner. These objectives require complex manufacturing routing scenarios, where a product traverses several internal and external factories around the globe before becoming a sellable entity.

As Figure 5 shows, our basic manufacturing flow begins with wafer fabrication; each wafer holds from 15 to 30,000 die depending on wafer diameter and die dimensions. After fabrication, each die on each wafer is tested and sorted into performance categories indicating their speed and power based on test results. Individual die are then picked from the wafer and assembled onto a multiple-pin package, often using multiple die from different wafers for multiple-die products. The assembled products are tested again to measure actual performance and to separate items into categories defined by performance ranges. Finally, the items go through a finish and mark process that results in individual SKUs. To mitigate any risk of

Figure 5. Intel's Basic Semiconductor Manufacturing Flow



Notes. Initially integrated circuits are fabricated (FAB) on wafers. Nonfunctional die are identified in SORT and eliminated in DIE PREP. The remaining functional die are joined to packages in ASSEMBLY and their performance properties are determined in TEST. After being labeled and boxed in FINISH, they are shipped.

business disruption and to maintain flexibility, each manufacturing process step can be executed at two or more factories, adding complexity to the manufacturing flow.

Intel manufactures roughly 400 unique wafers types totaling 2.5 million wafers per year, yielding 4,600 unique SKUs, which account for 600 million units in SKU volume. Adding to this complexity, in an average year, Intel introduces dozens of new products, each more complex than its predecessors. Products can flow through six or more manufacturing stages at more than 200 internal and outsourced factories across 13 countries. As Figure 6 shows, each wafer could undergo any of several thousand possible interdependent manufacturing routing options before it is transformed into one of the SKUs to be shipped to customers.

The supply chain planning challenge is to effectively route and manufacture the right products to meet demand for the designed SKU-set through an extremely cost-intensive factory network. This is accomplished by leveraging the vast number of interdependent routing options, each with its unique characteristics to best utilize all die and the associated factory capacity.

Product Architecture Design

Any solution to the two-step product architecture design process (Figure 2) must consider both product complexity and manufacturing complexity. On one hand, the design process must supply the set of SKUs desired by marketing to satisfy the forecasted demand, while minimizing the engineering costs. On the other hand, the design process must deliver a SKU-set that is manufacturable and meets the requirements for computational cores, execution speeds, and power requirements, while minimizing manufacturing costs. We have developed two modules to address these factors.

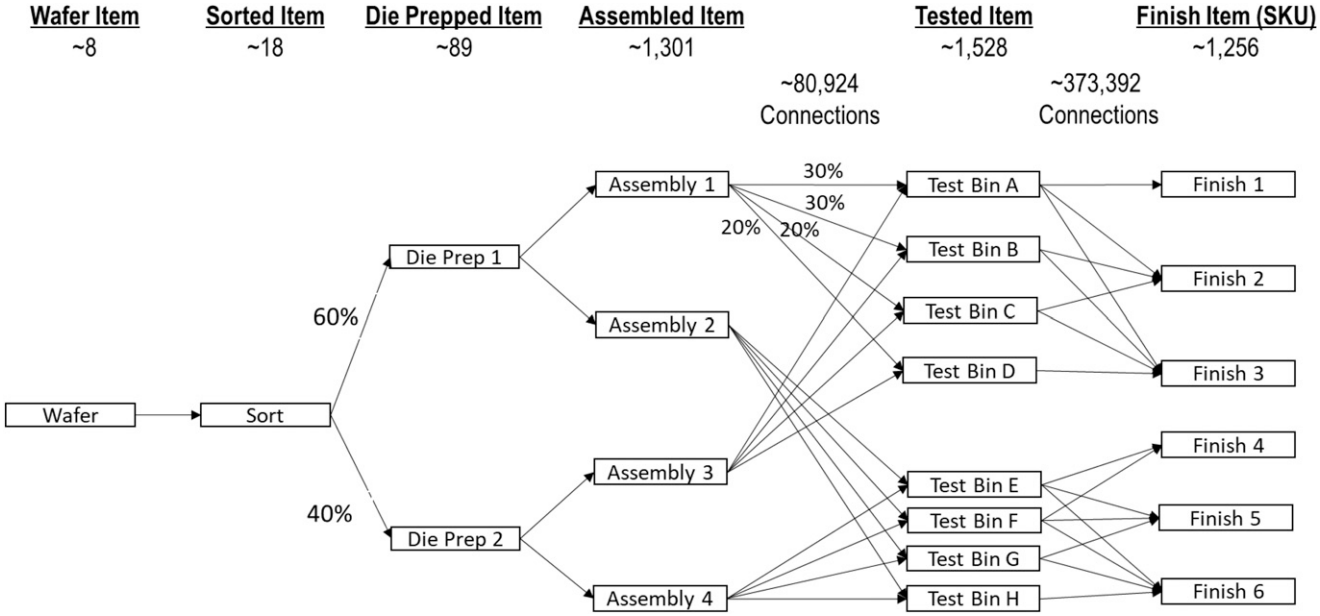
Product Composition: The Problem

Determining the number of design configurations is the first step. As a simple example, consider a marketing request for a SKU-set ranging from 4 cores to 24 cores in increments of 2, making 11 SKUs in all (Figure 7). As noted in the previous section, it is impractical from an engineering cost perspective to custom design each of the 11 SKUs. Furthermore, from a manufacturing cost perspective, it is impractical to satisfy all demand with a 24-core die. For example, satisfying the 4-core demand by turning off 20 cores of a large 24-core device would be an expensive waste of silicon and manufacturing capacity.

We could design a 6-by-4 matrix of cores to satisfy the top 24-core SKU. We could then remove a row from the design for a 5-by-4 matrix, remove another row for a 4-by-4 matrix, and so on to produce a set of designs with 24, 20, 16, 12, 8, and 4 cores. Alternatively, we could remove a column from the design of the 6-by-4 matrix to produce a 6-by-3 matrix for an 18-core SKU, and so on to produce a set of designs with 24, 18, 12, and 6 cores. Or we could remove a row and a column from the design of the 6-by-4 matrix to create a 5-by-3 matrix for a 15-core SKU and remove another row and column to create a 4-by-2 matrix for a set of designs with 24, 15, and 8 cores.

In each case, the variable core yield (on a multicore die, the number of working cores) from the manufacturing process, as well as our ability to turn off cores, would support filling the demand. Consider the design options with 24, 18, 12, and 6 cores. The 24-core design

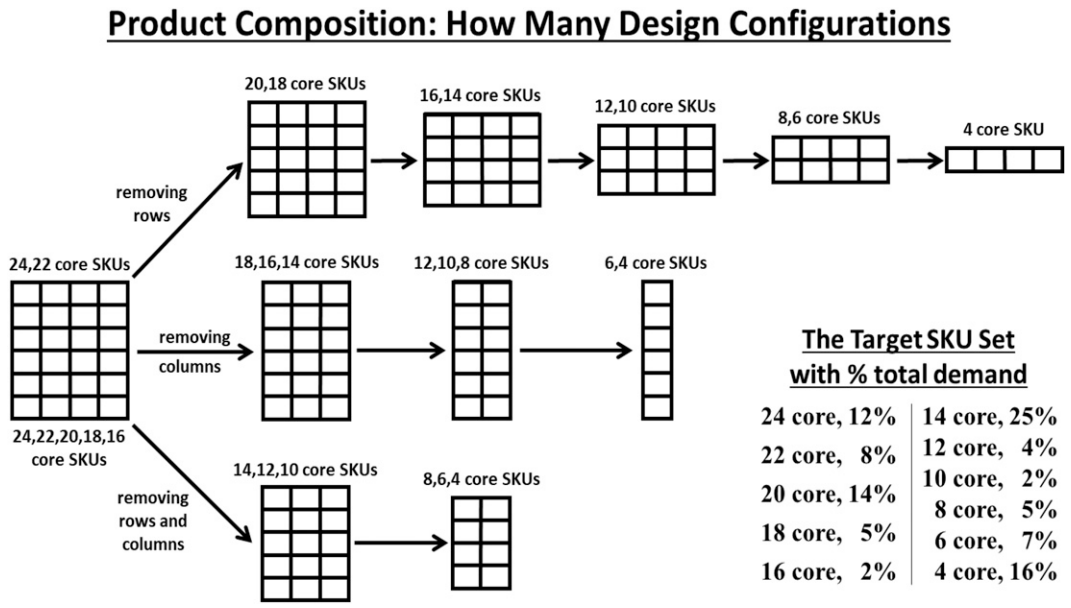
Figure 6. Simple Product Flow Showing Each Path with Unique Characteristics to Fill the SKU-Set Demand



Note. At the top of the diagram are the number of item-location combinations, assembly-to-test routings, and test-to-finish routings for a complex product once the product flow is extended to the factory network; each process step can be performed in two or more factories.

would be used to satisfy demand for 24, 22, and 20 cores, the 18-core to satisfy demand for 18, 16, and 14 cores, and so on. Minimizing engineering cost pushes in the direction of fewer design configurations (24, 15, and 8 in this example). Minimizing manufacturing cost pushes in the direction of more design configurations to conserve expensive fabricated silicon by turning off fewer working cores (24, 20, 16, 12, 8, and 4 in this example). The first module within our product architecture design solution focuses on optimizing this trade-off considering

Figure 7. The Product Composition Module Develops a Target SKU-Set



Notes. The upper approach uses the least silicon and so has the lowest manufacturing cost; however, it also has the most design configurations and therefore has the highest engineering cost. The lower approach has the fewest design configurations and the lowest engineering cost; however, it uses the most silicon and therefore has the highest manufacturing cost. Perhaps the middle option is the best compromise; but it depends on the relative costs of engineering and manufacturing. It also depends heavily on the volume of each SKU in the set required to fill the forecasted demand.

the demand forecast from sales and marketing and the core yield forecast from manufacturing.

As we increase the number of cores on a die, this architectural optimization becomes more difficult. Further complicating the problem, we now design products including multiple die, each of which has multiple cores. High demand for one core count and low demand for another core count in a SKU-set is another complicating factor (Figure 7). Finally, the wider the variability in core yield, the more difficult the optimization, although in some cases this can be addressed by adding a few extra cores to the design (e.g., designing 26 cores to ensure that 24 cores are functional).

Product Composition: The Capability

We developed a mathematical model to determine the optimal number and selection of architectural designs that meet market requirements with minimum engineering and manufacturing costs. To evaluate the effectiveness of each architectural option, we must assess the variation in performance and cost of each SKU on that proposed option. The device physics that determine the performance and cost of each SKU are complex and cannot be expressed in a closed form suitable for direct mathematical optimization. Therefore, we utilize simulations of device physics to generate a set of candidate die for each architecture option, based on the variation in the manufacturing process. We then simulate the device physics for each die and track which SKU's performance and feature requirements are satisfied. The results of these device physics simulations provide inputs to the mathematical optimization that constrain which SKUs can be realized within which architectural options. The inputs to the model are market requirements, manufacturing and engineering costs, the set of plausible architecture options, and the device physics simulated performance of each architecture option.

The model combines device physics simulations to address complex nonlinear interactions between features and a mixed-integer program (MIP) to select an optimal architectural option. The decisions generated specify a combination of architectural options that minimize the total costs, while satisfying market requirements.

By combining the simulation with the MIP, we developed a general-purpose capability that matches every product at Intel with the product architecture that optimizes its engineering and manufacturing costs. This breakthrough capability has enabled Intel to design for optimal manufacturability from the beginning of the product development life cycle and has saved the company hundreds of millions of dollars. Because this technique is product independent, all current and future design projects will use it to optimize their architectures.

Product Composition: An Example

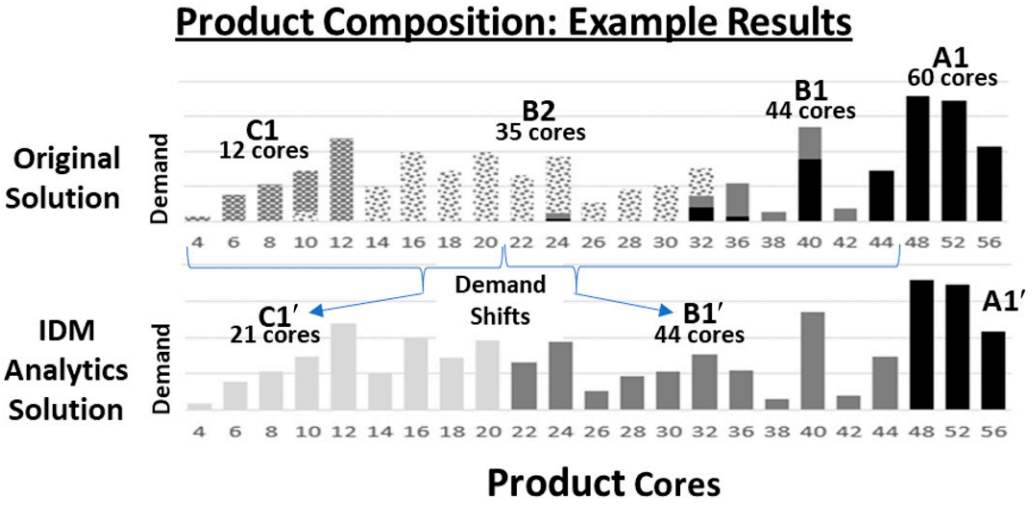
This optimization model has yielded substantial cost savings. In the example in Figure 8, the lifetime manufacturing cost of the product was reduced by $750 million after applying the optimization model. The engineering baseline obtained using the best-known previous design methodology is shown at the top of the figure. This baseline consisted of four designs to meet demand ranging from 4 to 56 computational cores. It included 60-core (A1), 44-core (B1), 35-core (B2), and 12-core (C1) designs. As we show at the bottom of Figure 8, the optimization model achieved significant cost savings by counter-intuitively enlarging the smallest design C1′ from 12 to 21 cores and eliminating the 35-core design B2, thereby lowering engineering costs. This resulted in a large shift of demand in the 14-to-20-core range, moving it from the larger 35-core design B2 to the smaller 21 core design C1′, thereby saving manufacturing costs. The solve also lowered manufacturing costs by assigning the 44-core design B1′ additional medium-core demand so that core failures in this design could be used to satisfy lower-core demand.

Many alternative designs, which we must consider, are clearly possible. Individual rows or columns of cores can be added or removed, or entirely different combinations of die can be used to achieve different core-count levels. Each of these design options has its own implications for engineering and manufacturing costs, product performance, and both die and core yields. The product composition module found the optimal architecture to satisfy market requirements, redefining the medium and low core-count solutions by changing rows and columns of cores. This shifted SKU-set demand to designs with lower overall manufacturing costs and reduced engineering costs. This resulted in a savings of $750 million in costs on this one SKU-set.

Product Performance: The Problem

Once the product architecture is specified, the next step focuses on the trade-off between the performance of the SKU-set and its manufacturing cost. Intel's customers are willing to pay a premium for the higher-performing die but not for lower-performing ones. Although we release enough wafers into the factory to theoretically satisfy all demand, our variable manufacturing process sometimes may not produce enough of the high-performance die. This forces us to release more wafers into the factories to supply the number of high-performing die required, thereby producing more overall die than the total demand requires. This phenomenon of "chasing" demand

Figure 8. (Color online) Illustration of the Product Composition Step Using an Example



Notes. By eliminating the 35-core B2 design from the initial hand-generated design set, the optimizer reduced engineering costs. In addition, the optimizer replaced the initial 12-core C1 design with a new 21-core C1′ design. These changes resulted in (1) a shift of demand for 12-core C1 and 35-core B2 designs onto the new 21-core C1′ design, and (2) a shift of demand for 35-core B2 and 44-core B1 designs onto the new 44-core B1′ design. These changes resulted in manufacturing cost savings of $750 million.

increases manufacturing costs, because an excess of low-performing die for which there is no demand will be produced to obtain enough die to satisfy the high-performance demand. These overproduced low-performing die are referred to as "leftovers." They can be difficult to sell and may accumulate in inventory to eventually be scrapped.

The performance and manufacturing costs of the SKU-set exist in a state of constant tension. Finding an efficient trade-off was a challenging, time-consuming process using the previously best-known design methodology, with various business groups iteratively exploring only a handful of manually generated options. The second module of our solution focuses on product performance to quickly find the optimal trade-off between performance and manufacturing cost. By intelligently exploring thousands of SKU-sets, we generate an efficient frontier of SKU-sets based on the criteria of manufacturing cost and performance. Using the efficient frontier, product teams select the preferred SKU-set for their specific markets and maximize overall profit for Intel.

Product Performance: The Capability

Generating the efficient frontier is a daunting task. Each SKU-set typically consists of 10 to 100 SKUs, each with dozens of speeds that must be configured to 1 of roughly 10 possible values. The performance of an individual SKU is a function of its execution speed and power consumption, and the performance of the SKU-set is the weighted average of the SKU performances. Hence, the size of the solution space for even a moderately sized SKU-set is on the order of 10^{100}. In addition to the sheer size of the solution space, the problem is highly nonlinear. As in the product composition problem, the interactions between performance and material variation are nonlinear and often cannot be described in closed form. Furthermore, determining the manufacturing cost of any candidate SKU-set requires solving a linear program (LP). Thus, an exhaustive search would require solving 10^{100} LPs.

Our module consists of a combination of the device physics simulation of the interrelationships between performance and material variation, a genetic algorithm to search the SKU configurations for candidate SKU-sets, and LPs to estimate the minimum manufacturing cost of each candidate SKU-set.

The key decisions are what frequencies to configure to what values for each SKU in the set to obtain an optimal trade-off between performance, defined as a weighted average of the SKU performance, and manufacturing cost, which is a function of the minimum number of wafers that must be produced to meet those SKU demand requirements. The key inputs to the model are the baseline SKU-set from the product composition module, the performance weights for each SKU and each execution speed, and the device physics simulation, which captures the interrelationships between performance and material variation.

The optimization algorithm is implemented as a hierarchical decomposition with two levels (Rash and Kempf 2012). The outer level is a genetic algorithm that searches for candidate SKU-sets. Each chromosome in the population is a candidate SKU-set whose values are represented as changes from the baseline. Mutation and crossover operations are implemented

by randomly mutating speeds or swapping speeds between parent chromosomes. Targeted improvement heuristics greatly improve the performance of the genetic algorithm by intelligently improving existing solutions. For example, specifically perturbing the performance of select SKUs up or down often incrementally improves the solution by finding higher performance or lower cost.

The inner level of the model is a series of LPs, one for each candidate solution, to determine the minimum manufacturing cost necessary to achieve the specified performance targets. The genetic algorithm has an unusual fitness function because it is a multiobjective optimization. Rather than calculating a single fitness value, the goal of the algorithm is to find the optimal trade-off between performance and manufacturing cost. Thus, the fitness of each candidate is its distance from the current efficient frontier. At the end of each generation, any nondominated solutions on the frontier are preserved and solutions that are far from the frontier are discarded.

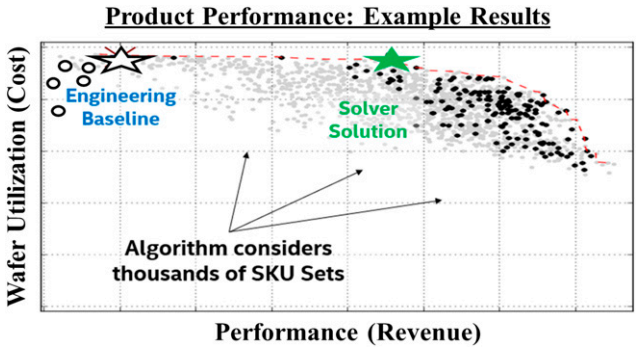
Combining the device physics simulation with the genetic algorithm and inner LP enabled the product performance module to quickly and efficiently find the optimal trade-off between cost and performance. Our approach has been applied to several products to identify and realize significant cost or performance savings. This technique is product independent; therefore, it applies to all current and future product performance projects.

Product Performance: An Example

Figure 9 shows the efficient frontier for an actual Intel product after running the genetic algorithm for 100 generations. Each point represents a full SKU-set that is a candidate solution. The light gray points represent sets from early generations, and the black ones denote those in the final generation. The dashed line highlights the nondominated solutions that comprise the efficient frontier. Our module recommended a solution at the solid star with roughly equivalent cost but significantly higher performance than the baseline solution at the hollow star, which was generated manually using the previous best-known design methodology. The benefit to Intel for this one product was a $235 million uplift in revenue.

Our capability allows us to increase performance, while holding manufacturing cost constant, or to reduce manufacturing cost, while holding performance constant. Senior managers have been receptive to the efficient frontier representation because it gives them a marketing control mechanism that they have not had previously. When entering a new and uncertain market, they can pick a SKU-set that reduces manufacturing cost by sacrificing some performance and sliding left on the efficient frontier. When going into a known but especially competitive market, they can pick a SKU-set that increases performance and absorbs some manufacturing cost by sliding right on the efficient frontier.

Figure 9. (Color online) The Product Performance Module Generates Thousands of SKU-Sets to Build an Efficient Frontier



Notes. Each dot represents a complete SKU-set; black dots are the final SKU-sets generated by the genetic algorithm, and gray dots are earlier generations. Compared with the engineering baseline that only generated a few options, we found a solution that achieved a $235 million uplift in revenue.

Supply Chain Planning

The product architecture design process output serves as an input into the supply chain planning processes. In addition to the product designs and corresponding customer demand signals, supply chain planning requires information as to which set of factories can execute which process stages for each SKU-set and the associated manufacturing parameters such as yield, throughput time, and capacity, as well as constraints such as batching requirements. Outsourced manufacturing locations often require additional constraints based on Intel's contractual agreements with suppliers. The product and manufacturing information is combined with information that reflects the strategies around planning. Some examples are target and minimum inventory levels that need to be maintained to account for uncertainties in demand and manufacturing, and relative product priorities when allocating capacity. Another key input to planning is the current state of the supply chain with respect to manufacturing work in progress. It is critical that all these relevant details are captured as inputs into the decision process to communicate feasible plans to the factory network to ensure seamless execution.

Intel has invested over $100 billion to build and maintain its wafer fabrication facilities. Given the large investment, long lead times, and variability described earlier, it is critical to effectively utilize the capacity in these factories. Building the wrong die, which can end up sitting in inventory rather than meeting near-term demand, translates to wasted

capital and lost sales opportunities. Therefore, the most critical planning decisions in the supply chain are determining what products to start at each of the wafer fabrication facilities to meet the projected demand. Once the wafer start decisions are made, the rest of the planning process involves devising the optimal routing of those wafers across the manufacturing network to minimize costs, while meeting customer demand at the appropriate time. These decisions span the entire network of more than 200 factories across all process stages and products. The wafer start decisions are made months ahead of the delivery of finished goods to the customer due to long manufacturing lead times; however, changes to customer demand and unexpected events in the manufacturing network can happen daily. To respond to these changes, the planning decisions are revisited and reoptimized on an as-needed basis, typically weekly, but sometimes as frequently as daily. The nature and the timing of various decisions lead to a natural decomposition of the overall planning process into three steps: wafer starts optimization, network capacity alignment, and optimized routing plans.

Each of these steps in turn follows the iterative process shown in Figure 10. Every time we start a new planning cycle, our key business decision makers provide a set of inventory strategies and product priorities. However, due to the complexities described earlier, they struggle to predict where we will face constraints and problems in the supply chain, how severe they will be, and how the strategy should change to deal with those problems. In our iterative process, our planners run the optimization engine, analyze the results, and work with their key decision makers to get feedback on the business trade-offs and scenarios that they want to evaluate. They create additional scenarios to explore the range of potential trade-off decisions that could be made, and go through this loop multiple times within the course of the planning cycle before aligning on the scenario that they want to execute and publish to the rest of the supply chain as our operating plan.

Figure 10. (Color online) Flowchart Illustrating Intel's Iterative Approach to Planning

With this approach, the strategic decisions are made by the business operations teams; however, the execution is done by the optimization program. The operations teams can explore many possible business scenarios, while the optimization engine shows them the best possible results they could achieve for each set of strategic decisions.

Wafer Starts Optimization

In the first step of our problem decomposition, we focus on our wafer fabrication and sort facilities, which consume the bulk of our capital investments. We model the full product-flow complexity to understand how wafer start decisions impact the final finished goods supply, but we ignore capacity constraints at all other stages of the supply chain. This model runs simultaneously for all products at Intel with a one-year horizon. It has 13 million decision variables, 30,000 integer variables, and 4.1 million constraints, and our engine can typically solve it in approximately six minutes.

Because these decisions are so critical for Intel's success, we run dozens of scenarios and carefully analyze how these decisions impact our capability to support our customer demand and inventory strategies. The outputs of this step are the capacity allocations and schedules for the wafer fabrication and sort factories.

Network Capacity Alignment

The wafer start decisions and how much supply we will be generating from each wafer start is finalized in the previous step. We can now use that information as a constraint when we generate a capacity request to all our die-prep, assembly, and test and finish factories both within Intel and at our external manufacturing partners. Because the shared-wafer capacity has already been allocated, and capacity at all other stages is assumed to be unconstrained, each product family can be planned independently. This breaks down one large problem into hundreds of smaller, easier-to-solve subproblems. The results from these subproblems are later consolidated into a single signal that is sent to both the internal and the external factory partners.

Our internal and external factory partners analyze the requested schedules and allocate their capacity to meet the demand and product priorities provided. Analyzing and allocating our internal factory capacity is a complex problem, with over 7.9 million decision variables, 10,000 integer variables, and 7.8 million constraints, which can be solved in approximately four minutes. The output of this step is the capacity allocation by product family for each of the die-prep, assembly, and test and finish factories across the internal and external manufacturing network.

Optimized Routing Plans

After executing the previous two steps, we know the supply available for each wafer and the capacity allocations for the rest of the network. We now can run a fully constrained solve to optimize our final factory routing plans and provide detailed instructions to each factory on what it should build in each week in the planning horizon. Once again, each product family is independent, allowing the problem to be decomposed into hundreds of smaller pieces. A typical large product family has about 2.1 million variables, 1.6 million constraints, and a solve time of less than two minutes.

The optimized routing plans are automatically published to our enterprise systems and are used as the plan of record for customer order confirmation, factory execution, and our chief financial officer's financial projections to market analysts.

Although we optimize the entire network on a monthly cadence by executing all three steps of the SCP process, the decomposition allows us to make quick adjustments to the plans and, when needed, with weekly and sometimes even daily frequency and only for subsets of the supply chain. For example, if the demand mix changes for a given product family, then we can simply re-execute the last step in the process for only that product family. If the demand changes cannot be accommodated within the allocated capacity, then we can re-execute the second and third steps with the set of products and factories impacted.

SCP optimizes the global supply chain, providing guidance to the entire supply network. Our SCP solution integrates very closely with the tactical execution planning processes, which focus primarily on the period within the manufacturing lead time. Tactical execution planning provides instructions to the factories in the short term to align factory execution to plans provided by SCP and, when warranted, makes the required changes to SCP plans in response to customer demand changes and unexpected manufacturing events. As already noted, one of the key inputs to SCP is the current state of the supply chain network. The tactical execution planning process provides input to the SCP process in terms of the expected output of the factory network in the short term, ensuring a closed-loop process. This is a key feature of the overall planning environment to ensure that execution and planning always stay in sync and that the output of planning is feasible when it comes to execution.

SCP Modeling

The planning capabilities are built on an integrated framework that includes the optimization and mathematical modeling components, user interfaces for planners to interact with the models, and an artificial intelligence/machine learning (AI/ML) module for explaining optimization results. Included are interfaces to and from the rest of the enterprise and factory systems to collect the data needed to support the planning process and to communicate decisions back to the enterprise and factory network. This suite of capabilities guides SCP decisions of what and how to optimally manufacture to support demand and minimize waste. The goal is to provide detailed plans for the entire supply chain network, while comprehending demand and production strategies.

The SCP model covers a one-year horizon in weekly granularity. We represent SCP as an MIP problem, similar to the classical production planning problems covered in Moench et al. (2017). The model is modified for each step of the planning process to reflect the modeling assumptions. For example, in the wafer starts optimization step, the die-prep, assembly, and test and finish factory network is aggregated to a single factory, assuming infinite capacity for those steps, whereas wafer fabrication and sort factories, where key decisions are made, are modeled explicitly. Another example relates to the last step of optimized routing plans. In that step, the wafer supply is frozen; therefore, the model does not need the wafer fabrication and sort factories. This approach allows us to create simpler models without sacrificing the solution quality.

Advanced Operations Research Algorithms and Complexity

The size and the complexity of the SCP models are challenging, especially considering the business requirements for rapid runs to enable many iterations and scenario analyses. Additionally, the high number of objective trade-offs in one optimization run makes reaching a satisfactory solution even harder. Our advanced analytics models and data-reduction techniques allow us to arrive at optimal solutions on average in less than six minutes for these complex problems. We employ various innovative techniques and use our deep understanding of Intel's business to exploit the structure of the problem to simplify the mathematical models and achieve faster results. The following are a few of the techniques that we have leveraged:

- *Standard and flexible formulation*: Our SCP model leverages the item, bill of material, and route model (Hugos 2007) to represent the supply chain. This translates to a robust and flexible formulation that effectively handles the evolving supply chain network as more complex products and manufacturing processes are introduced into the flows.
- *Model decomposition and effective cuts*: In all the SCP models, numerous production-related rules exist for each process stage from fab to finish (Figure 5). Some of these rules may vary depending on whether manufacturing is planned at an internal factory or an external factory. In other cases, select rules apply

across all factories and process stages. This special structure enables us to partition our large model into manageable subproblems and use decomposition (e.g., for fab/sort and for assembly/test) to reduce solve times (Dantzig and Wolfe 1960, Barahona et al. 2005). To handle the complexity of solving binary and integer variables for minimum lot sizing and batching requirements, we add a number of stronger cuts based on our practical knowledge of our problem to speed up the branch-and-cut process, and reach the optimal integer solution faster (Andersen et al. 2005).

- *Interactive sequential goal programming*: The business rules for planning require that we balance several hundred objectives in a single optimization run. To handle the large number of objectives efficiently, we use a tiered solve approach focusing on a subset of objectives in each tier. We also convert objectives to goals and use weighted goal programming to optimize all unwanted deviations from the desired levels in the goal achievement function within the objective function (Masud and Hwang 1981).
- *Constraint caching and parallel data processing*: To further improve end-to-end solve times, we cache all constraints in memory and only regenerate constraints that have coefficient or right-hand side (RHS) changes from solve to solve. Moreover, we employ parallel data loading for different constraint coefficients and RHS values when we load data from the database into memory.
- *Path-based constraint formulation*: Our solution solves for multiple, sequential stages, as we show in Figure 5. We employ a mix of path-based and stage-based formulations to avoid the introduction of a large number of variables and constraints by modeling each stage. The path-based formulation pregenerates paths from wafer to finish and calculates parameters for each path to most effectively generate the constraints (Fleischer and Skutella 2002).
- *Artificial intelligence (AI) in support of explainability and autonomous planning*: One of the biggest challenges in deploying optimization solutions in practice is to explain the results. To overcome this challenge, we have developed an AI platform, which consists of a complex events processing (CEP) engine that deciphers results from the optimization engine into multiple events (Adi 2007), a knowledge-based-system (KBS) module that helps explain plan changes cycle over cycle (Tiwana 2000), and an ML module that conducts classification, prediction, and harvesting of business rules based on historical plans (Wilkins and desJardins 2001).

The platform accumulates both business process flow knowledge and production rules with advanced ML models working interactively with our optimization model to aid users in areas including (1) tuning the model weights and parameters, (2) checking and imputing bad data, (3) identifying binding and conflicting constraints, (4) understanding outputs such as demand support patterns, and (5) facilitating sensitivity analysis to ensure robustness of supply chain plans.

Outputs from the CEP Engine, and KBS and ML modules feed a decision support (DS) module consisting of a rules engine that leverages predefined rules and harvested rules. This DS module determines whether to accept a plan or to iterate in search of a better plan.

Analytics Framework and Implementation

Delivering end-to-end advanced analytics to support product architecture design and supply chain planning requires a framework that is (1) easy to use, (2) flexible to support changing requirements, and (3) performant and scalable to meet the growing advanced analytics needs of the company. Our analytics framework is designed to support a broad range of product architecture design and supply chain planning capabilities. The framework is multilayered and modular. Multilayered architecture enables different parts of the solution to be implemented with different technologies, while still being part of a single solution framework. Modular architecture allows us to modify existing components or add components independently without impacting the rest of the solution.

The application layer includes a suite of applications, each focused on a specific business need. The applications provide facilities to manage relevant data, schedule analyses, perform what-if analyses, and view results and dashboards. Applications are implemented as a mix of client and web-based applications, depending on the specific business need. An event-driven architecture is used across the applications with real-time notifications of analysis progress. Applications are implemented using a variety of languages and platforms, including C#, Python, and various JavaScript frameworks.

The microservice layer is responsible for invoking analyses as requested by applications and other microservices. New applications are composed by connecting to existing microservices or adding new microservices. Microservices provide an abstraction so that the details of running analyses can evolve over time without impacting applications and are implemented using a variety of languages and platforms, including C# and Python.

The compute layer is where analyses run. Microservices expose the ability to trigger the analyses' functions and review the analyses' outputs. The solution framework uses on-premises private scale-out clusters for data locality and security requirements. Many analyses are currently being migrated to run on local Kubernetes clusters for maximum scalability.

The optimization models and AI/ML models are implemented using a variety of languages and frameworks, depending on the specific business needs. Many of the LP and MIP models discussed in this paper are implemented using CPLEX and Concert Technology with custom C# logic that manages preprocessing, data validation and imputation, decision variable creation, constraint generation, tiered-solve sequencing, CPLEX parameter setting, and algorithm configuration. The custom genetic algorithms are implemented in Python using standard frameworks, such as Numpy, and solve their linear subproblems by invoking CPLEX.

The data layer is where analysis data are stored. SQL Server, MongoDB, and other enterprise services are used in this layer. The compute and microservice layers interact with the data layer to retrieve the inputs needed to run the optimization models. Applications access data through microservices. The details of data storage are hidden from applications so storage can change without impacting applications.

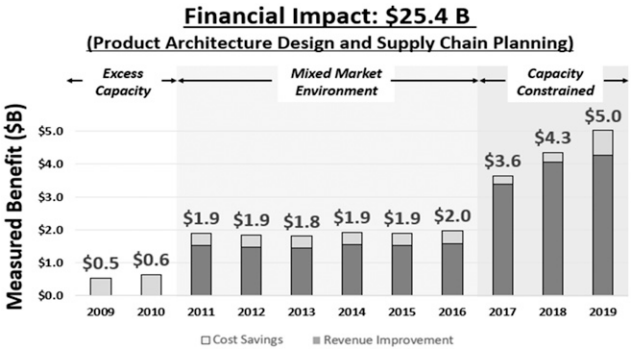
Benefits

Over the course of the efforts described in this paper, Intel has realized financial benefits, organizational benefits, and environmental benefits. All the modules that have been developed and integrated are product independent from the perspectives of design, manufacturing, and supply; therefore, these benefits will continue to be realized into the future.

Financial Benefits: Product Architecture Design

Since the initial effort to expand our project to include product architecture design eight years ago (Rash and Kempf 2012), we have not found any commercially available off-the-shelf (OTS) software that can address any of the steps of our complex problems. As we developed our modules to solve each part, we have had the opportunity to apply them to a wide variety of Intel products. A member of Intel Finance representing the target product division has been assigned to each use case to provide financial data as required during the effort and to help evaluate the benefits at the conclusion of the use case. Over the past four years, successful examples include decreasing the number of design configurations required and lowering engineering cost, optimizing the manufacturing cost trade-off with increased performance, increasing revenue, and improving the utilization of the die from manufactured wafers, thus lowering manufacturing costs. The product design benefits computed by Intel Finance include $1.04 billion in cost savings and $340 million in revenue uplift for a total of $1.38 billion over the past four years (Figure 11). Because the techniques that we have developed are applicable to every product that Intel will design in the future, we consider the valuation so far as the tip of the iceberg.

Figure 11. Monetary Benefits in Terms of Revenue Upside and Cost Savings as a Result of Using Our Product Architecture Design and Supply Chain Planning Solution



Financial Benefits: Supply Chain Planning

To ensure that our solution provides Intel with a clear competitive advantage over commercially available OTS supply chain management software solutions, we periodically compare our solutions with OTS solutions with respect to capabilities and performance and conduct a gap analysis based on our current and future needs. Traditionally, we have found OTS software unable to meet Intel's required criteria with respect to business rules and constraints. Given the flexibility in our solution to customize the models to fit the manufacturing network and capture business requirements, we can provide feasible and executable plans to the supply chain, minimizing postoptimization manipulation and postexecution reconciliation efforts. As a result of our fast-changing business needs, we are required to deliver new capabilities and major enhancements every four to six months, making OTS solutions impractical for our scale and fast-changing environment.

Before putting our supply chain planning solution into production in 2009, we did extensive testing and analysis against the legacy solutions in place at Intel at the time. We compared the optimization-based solution with the legacy, greedy, algorithm-based heuristic solutions, by running the solutions in parallel for a year. During this period, planners would pick a set of representative products with all the corresponding input data and use the heuristic solution to determine the plans for all manufacturing routes to support demand and meet inventory targets. Then, we would use the same set of products and the same inputs and run our supply chain planning modules to generate the optimal plan to achieve the same demand support and inventory targets. Based on our year-long comparison of the computer-generated plans, which included 12 monthly planning cycles, even for very simple products, we observed that planners could only manage to consider a subset of the potential feasible solutions and, as a result, came up with answers that

resulted in 6%–15% more wafer starts for the same level of demand support compared with our optimization-based solution. Today, with more complex multichip and multicore products, the legacy solutions and process would not be able scale to produce feasible results in a timely manner.

Given the observed range of results, we have arrived at a very conservative valuation of 6% reduction in wafer starts for our supply chain planning modules. Analysis by Intel Finance has shown that this translates to a 6% revenue upside in years of high-capacity constraints (6% more revenue for the same wafer starts), a 6% cost savings due to fewer wafers manufactured to satisfy demand in years where capacity is not constrained, and a 3% revenue upside and 3% cost saving in years of medium levels of capacity constraints.

The overall product architecture design and supply chain planning results are summarized in Figure 11, showing an overall benefit of $25.4 billion thus far between 2009 and 2019. This is based on $568 billion revenue for Intel during the same period for the products that are designed and planned by our solution suite. Similar levels of annual savings are also projected in future years, with our capability continuing to drive a competitive edge for Intel.

Organizational Benefits

In addition to the technical challenges, we had to overcome a variety of organizational challenges to achieve our outstanding results, detailed earlier, which position our advanced analytics as the gold standard for the company. Our solutions have set the direction on key product architecture design decisions across multiple products and business units and are vital to SCP decisions performed on a daily, weekly, and monthly basis across Intel's worldwide manufacturing network. We faced many common biases, including "these tools will take away our jobs," "not invented here," and "we have always done it this way." Despite these challenges, our users and our management have transitioned from skeptics, to believers, to champions.

We overcame these biases by always involving the individuals, teams, and organizations who would benefit from or be impacted by our solutions. Our efforts began by observing stakeholders perform the existing "standard" process for solving those problems and watching them report to their management. We built proof-of-concept solutions using data from previous situations and shared the method and results with potential users before sharing them with their management, which led to requests for production solutions. We found this approach to be more effective in terms of the longevity of the solutions, where the users took ownership of the solution early on and became advocates, compared with solutions that are pushed "top-down" from higher management or corporate information technology groups. The solutions that we have deployed include training materials across all levels of the corporation, a framework for our tools to enhance their usability, including data acquisition and solution broadcast, and techniques to explain the solutions. The users became champions because our capability gives them the benefit of efficiency and the satisfaction of generating better solutions. They can now solve their problems in minutes or hours instead of days or weeks, as they did previously. Alternatively, they can evaluate 5 to 10 times more scenarios than ever before to improve solution quality under different business conditions. In most cases, they enjoy both benefits, always free from manually introduced errors.

Our management became champions based on the financial results from sustained revenue increases and cost reduction, as well as the new insights and controls that they have relative to the product architecture design and supply chain planning processes.

Environmental Benefits

Semiconductor manufacturing has a high reliance on water, and Intel is leading the way with respect to reductions in water usage and the wastewater generated during manufacturing. Intel recycles most of the water it uses in manufacturing. The postmanufacturing treatment practiced by Intel produces water that often exceeds local government drinking water standards. Our supply chain planning capability resulted in reducing water usage by two billion gallons of water and preventing over 500 million gallons of wastewater so far over the 10-year time frame of our project. Although water savings are not explicitly modeled in our analytics solutions, these benefits are the direct result of the reduced number of wafers started in the factories, especially in years where capacity is not constrained. Reducing water usage and wastewater is important for the environment, especially in states, such as Arizona and New Mexico, that have Intel factories and where water is a precious commodity. It continues to be a major focus at Intel along with many other socially responsibility efforts, such as leading the industry in the use of green power and conflict-free minerals.

Summary

Considering the ever-increasing complexity of our products, manufacturing processes, and routings, Intel could not efficiently manage the product architecture design and supply chain planning processes that are vital to our success without the integrated solution suite described in this paper. We have achieved the goal of increasing the profitability of our enterprise by over $25 billion and changing our

culture in a way that will sustain the benefits far into the future. Our customers and society benefit from our ability to continue our long tradition of designing and supplying ever-more powerful and capable computing products. We have also contributed to Intel's ongoing efforts to protect and enhance the environment. We hope that our success will encourage others to apply advanced analytics to complex problems and thus provide them and society with substantial benefits.

References

Adi A (2007) Complex event processing. Accessed July 1, 2020, https://www.research.ibm.com/haifa/dept/services/papers/cep_Jan07_nc.pdf.

Andersen K, Cornuéjols G, Li Y (2005) Reduce-and-split cuts: Improving the performance of mixed-integer Gomory cuts. *Management Sci.* 51(11):1720–1732.

Barahona F, Bermon S, Gunluk O, Hood S (2005) Robust capacity planning in semiconductor manufacturing. *Naval Res. Logist.* 52(5):459–468.

Dantzig GB, Wolfe BP (1960) Decomposition principle for linear programs. *Oper. Res.* 8(1):101–111.

Fleischer L, Skutella M (2002) The quickest multicommodity flow problem. Cook WJ, Schulz AS, eds. *Integer Programming and Combinatorial Optimization* (Springer, Berlin), 36–53.

Hugos MH (2007) *Essentials of Supply Chain Management*, 3rd ed. (Wiley, Hoboken, NJ).

Masud A, Hwang C (1981) Interactive sequential goal programming. *J. Oper. Res. Soc.* 32(5):391–400.

Moench L, Uzsoy R, Fowler JW (2017) A survey of semiconductor supply chain models part III: Master planning, production planning and demand fulfillment. *Internat. J. Production Res.* 56(13):4565–4584.

Moore G (1965) Cramming more components onto integrated circuits. *Electronics* 38(8):114–117.

Rash E, Kempf K (2012) Product line design and scheduling at Intel. *Interfaces* 42(5):425–436.

Tiwana A (2000) *The Knowledge Management Toolkit* (Prentice Hall, Englewood Cliffs, NJ).

Wilkins DE, desJardins M (2001) A call for knowledge-based planning. *AI Magazine* 22(1):99–115.

John Heiney is an enterprise architect in the Supply Chain Decision Solutions Group at Intel Corporation. He has played a leadership role in the design and implementation of large-scale transformations in Intel's supply chain planning processes and capabilities. He holds a BS in chemical engineering from Brigham Young University and received an MS in chemical engineering and an MBA from the Massachusetts Institute of Technology as a fellow in the Leaders for Global Operations program.

Ryan Lovrien is a data scientist in the Decision Engineering Group at Intel Corporation. He focuses on building the end-to-end analytics platform that schedules, runs, and manages the analytics used in product feature design, ensuring that it can scale to many users and analyses and can be easily incorporated into business processes. He holds a BS in computer engineering and an MEng in computer science from Cornell University.

Nicholas Mason is a data scientist in the Decision Engineering Group at Intel Corporation. His focus on mathematical rmodeling and optimization problems is centered around product design and manufacturability. He has been involved in formulating optimization models at Intel, bringing expertise from his doctoral studies in production planning, supply chain management, auction design, and coordination mechanisms. He holds a BS, MS, and PhD in industrial engineering and operations research.

Irfan Ovacik is a principal engineer in the Supply Chain Decision Solutions Group at Intel Corporation. His primary focus is building decision support tools in the supply chain space, and specifically on data infrastructure needed to support these tools. Irfan holds a PhD in industrial engineering from Purdue University. His professional interests span all aspects of supply chain management, specifically in the semiconductor manufacturing industry.

Evan Rash is a principal engineer and senior data scientist for the Decision Engineering Group at Intel Corporation. He focuses on applying optimization techniques to Intel's product design and manufacturing decision problems and is technical lead for SKU Stack Analytics initiatives. Evan has published many internal publications and is a winner of the 2011 INFORMS Wagner Prize. He holds a BA in operations research and management science from UC-Berkeley.

Nandini Sarkar is a principal engineer and data scientist for the Supply Chain Decision Solutions Group at Intel Corporation. She focuses on path finding and large-scale enterprise architectures, including big data management, autonomous planning, and data science for supply planning decision systems. Previously she served as an enterprise architect in the Internet of Things Group at Intel. She holds an MS in electrical engineering.

Harry Travis is a data engineer and data science manager for the Decision Engineering Group at Intel Corporation. He leads a team that delivers analytics focused on product design and roadmap decisions. He has been involved in designing and implementing software tools focused on improving strategic and tactical decision making across a wide swath of Intel's business. He holds a BS in computer engineering from West Virginia University.

Zhenying Zhao is a principal engineer in the Supply Chain Decision Solutions Group at Intel Corporation. He is one of Intel's key experts in mathematical optimization and supply chain management arena. Prior to joining Intel, he worked as a research professor at the R. H. Smith School of Business, University of Maryland College Park. He holds a BS, MS, and PhD in production and industrial engineering, and he completed postdoctoral studies in operations research and information systems.

Kalani Ching is the head of business operations for the Supply Chain Decision Solutions Group at Intel Corporation. He leads the strategic planning and operations management for a global team of data scientists, operations research engineers, and software developers delivering Intel's world-class enterprise planning capability and solutions. Kalani holds a BS in materials science and engineering and an MBA from Carnegie Mellon University.

Shamin Shirodkar is the senior director of the Supply Chain Decision Solutions Group at Intel Corporation. He leads the design, development, and production support of Intel's proprietary enterprise supply chain planning solutions.

A recognized expert in supply chain optimization and research, he and his team are instrumental in driving Intel's competitive advantage in the worldwide manufacture and delivery of semiconductor products.

Karl Kempf is a senior fellow and the director of the Decision Engineering Group at Intel Corporation. He leads a team of decision scientists charged with building decision-support processes and tools focused on faster, better decision making across Intel. He has co-edited three books and has more than 175 publications. He has been a research adjunct at Missouri State, Arizona State, North Carolina State, and Stanford Universities. He is a member of the National Academy of Engineering, a fellow of the Institute of Electrical and Electronics Engineers, and an INFORMS fellow.