**Understanding Semiconductors – A Technical Guide for Non-Technical People**

**Forward**

The SCI[[1]](#footnote-2) is one of the biggest and most valuable industries in the world. It drives advancement in other enormous industries like auto manufacturing, financial services, and medicine. Product differentiation is largely dependent on SC technology in markets like smartphones, industrial robots, and pacemakers.

**Introduction**

In 1965, Gordon Moore, founder of Intel, posited ***Moore’s Law***, which predicted that the number of transistors a chip could hold and the resulting computing power would double every two years. Since the first hand-sized transistor developed (William Shockley, John Bardeen, Walter Brattain) at Bell Labs in 1947 to the 3nm transistors in development today, this doubling has happened every 18 months on average. This is the cornerstone of the “Digital Revolution” which has led to a myriad of technologies we depend on today and to the creation of countless industries and innovations. In the US alone, the SCI, which is the third largest exporter after oil and aircraft, employs 250K people with an additional 1M indirect jobs.

**Chapter 1 Semiconductor Basics**

Human advancement is characterized by our ability to harness and control forces. SCs have been the key to harnessing the power of electrical energy. The basics of electricity and conductivity is key to understanding how this works.

**Electricity and Conductivity**

***Electricity*** describes the relationship between charge and current. Electric charge is a fundamental property of matter stemming from protons and electrons that are held together by a balance between two forces – ***electromagnetic force*** (opposite charges attract, similar charges repel; protons attract orbiting electrons; lends mobility to electrons between atoms) and ***strong force*** (holds neutrons and protons together). In some ***elements***, electrons stay close to the atom’s nucleus (***insulators***) while in others, electrons are constantly bouncing around to other nearby atoms (***conductors***).

In a neutral state, electrons move from atom to atom randomly but in aggregate the collective charge of an object remains neutral. ***Electric current***, measured in ***amperes*** (the number of electrons that flow past a given point in one second), results when electrons collectively flow in the same direction across a wire at nearly the speed of light (individual electrons themselves travel only a few millimeters a second). Current transfers energy from the atoms at the front of a conductor to the atoms at back analogous to how a physical moving object carrying **kinetic energy** transmits its kinetic energy to other objects upon contact.

By creating a **charge differential** (**electric potential,** **voltage, electromotive force, potential difference**) between two regions, we can initiate a chain reaction of electron movement by connecting the regions via a conductive material. A **circuit** is any closed loop between a source of voltage (e.g. a **battery**), a conductive wire, and other electrical elements. A circuit can also be formed by placing two regions in direct contact. The potential difference describes the amount of **work** required to displace a positive charge in an electric field (positive regions are at higher potential than negative regions). Electrons flow from low to high potential while “conventional” current flows from high to low potential. The greater the charge difference between the regions, the greater the voltage. ***Electrons cannot flow without voltage***.

Rather than depending on potential difference in nature, **batteries** work by separating charge and therefore creating a potential difference, in the form of chemical energy, between its cathode (+) and anode (-). Current will not stop flowing until all of the surplus negative charges at the anode have flowed to the cathode thus depleting the battery. The battery’s stored chemical energy is converted into electric energy by completing the circuit.

Current and voltage tell us how electricity works in principle, but **power**, measured in **watts** (amount of work done, in Joules, when one amp is pushed through one volt), is the key to understanding how electricity can be used to do productive things. Power describes the work done when electric current is converted into some other form of energy such as motion, light, heat, or sound.

This relationship is called **Joule’s Law**, named after the **James Prescott Joule** who discovered it in 1840. The challenge with electricity and power is to figure out how to optimally harness current to do something useful. This is where conductive materials come into play. Conductivity measures of how easily current passes through a material and thus manipulating conductivity allowing current to flow in some cases and restricting it in others. Materials fall into three main types:

**Conductors**, like copper, have high conductivity and low electrical resistance measured in **Ohms**.

**Insulators**, like plastics, have low conductivity and high electrical resistance.

**Semiconductors** can act like conductors and insulators and our ability to control exactly when SCs conduct and when SCs insulate is the key to the electronics revolution.

**Silicon – The Crucial Semiconductor**

There is a large variety of SC materials – each with varying levels of conductivity – such as Germanium and Gallium Arsenide. Silicon is the most prevalent SC material because it has numerous advantages including beneficial mechanical and thermal properties, cheap, and abundant – comprising 30% of the Earth’s crust, Silicon is the second most abundant element on Earth and is found in sand, rocks, clays, and soils.

**A Quick Semiconductor History**

In 1947, scientists used the semi-conductive properties of silicon to build simple transistors which served as switches for electric current. By arranging transistors in intricate patterns, they realized they could selectively guide current along a path of their choosing. For a decade after its invention, transistor manufacturing was slow, cumbersome, and costly. Transistors were manufactured as individual units that were manually connected to each other using “**flywire**”. These circuits could fill up entire rooms. This all changed in 1959 – the beginning of the semiconductor revolution – with two breakthroughs:

1. The invention of the **integrated** **circuit**, by Jack Kilby at Texas Instruments and Robert Noyce at Fairchild Semiconductor, allowed hardware designers to fit a bunch of transistors together on a single chip.
2. The invention of the **planar manufacturing process** by Jean Hoerni at Fairchild Semiconductor that allowed chip companies to fabricate a multitude of components at the same time on the same substrate.

These two breakthroughs continue to serve as the foundation for the design and manufacturing-based value chain upon which the SCI is built on to this very day.

**Semiconductor Value Chain – Our Roadmap**

To understand the SC value chain, it’s helpful to put yourself in the operations of a SC company. Starting from a product concept, the SCVC can be broken down into six main components:

1. **Customers need and market demand** – assessing the need for a system or product. Remember that customers may not always tell you what they need. “*If I had asked people what they wanted, they would have said faster horses*” – Henry Ford.
2. **Chip design** – a firm must design a chip that will fit that product using **front-** and **back-end** **design**. FE design involves gathering system requirements, developing a detailed schematic that represents the design concept, and testing and verifying it before moving to BE which involves creating a **netlist**, a detailed list of instructions that is converted into a physical layout which can be tested and validated. After testing and validation, the physical layout can be sent to a **semiconductor fabrication plant**, or **fab**, to be manufactured.
3. **Fabrication and manufacturing** – numerous **integrated circuits** (**IC**, also known as **dies**) are printed onto a sheet of silicon called a wafer using **photolithography**.
4. **Packaging and assembly** – after dies have been cut apart, they are individually packaged during the IC packaging process called **assembly**. These **package-die assemblies** are tested one last time before being sent to end systems or product companies.
5. **System integration** – after receiving the assemblies, system/product companies solder the assemblies onto larger circuit boards or substrates with other components or ICs and integrate everything into a consumer-ready product.
6. **Product delivery** – the final product is shipped to the end customers.

SC companies generally concentrate of Steps 2 – 5. “Fabless” SC companies focus on Step 2. Each step of the value chain is essential to get a chip from concept to customer and at every stage in the SCVC, companies fight for profits and market share by trying to balance the challenges of these stages more effectively than their competitors. Since the 1940s, the SCVC has remained relatively stable. However, SCVC business strategies, driven by innovative companies competing to provide the best performing chips and highest quality products, have been very dynamic.

**Performance, Power, Area, and Cost (PPAC)**

For companies focused on Steps 2 – 5, the goal is to achieve the **highest performance (clock frequency, speed)** using the **lowest** **power (watts, longer battery life, heat constraints)** and the **smallest area (nm, less power, space constraints)** possible at the **lowest cost (design costs, manufacturability, and yield risk)** and in the **shortest time** possible. Balanced PPAC is essential for achieving an optimal solution. Design teams must keep the problem they are trying to solve in mind while creating the circuit needed for end-use application. The ideal chip is high performance, low power, and small but cost and time require making tough tradeoffs.

Time to delivery can be critical – it may be worth lower performance if you can cut the design cycle time short in order to be first to market. PPAC and time are critical for managing trade-offs in context to the product or application. For example, requiring high performance and small area will constrain power efficiency.

**Who Uses Semiconductors?**

The Semiconductor Industry Association (SIA), a key SCI trade group, defines six end-use applications segments:

1. **Consumer** (12%) – TV, video, audio, household appliances, other consumer goods like cameras, games, smart watches, fitness monitors, alarm clocks, etc.
2. **Automotive** (11%) – in-vehicle entertainment and information systems, power train, controls, etc.
3. **Computing** (32%) – PCs, office equipment and peripherals, handheld computing devices, servers, etc.
4. **Industrial** (12%) – power supplies, IoT devices, manufacturing test, control, measuring equipment, etc.
5. **Communications** (31%) – cell phone and wireless handsets, networking and remote access devices, base stations, broadcasting equipment, etc.
6. **Government** (1%) – military and aerospace electronics.

While Communications and Computing accounted for the greatest proportion of SCI sales (66%), Automotive, Industrial, and Instruments are expected to increase revenue share as cars become more electrified and industrial operations become more automated. Each of these segments has different PPAC requirements and drivers based on their unique purpose, and this impacts each stage of the SCVC.

**Chapter 2 Circuit Building Blocks [25 – 39]**

**Discrete Components – The Building Blocks of Circuits**

Electronic devices are built using some combination of **discrete components**:

1. **Transistor** – function like electronic switches. By stringing many transistors together, transistor switches can form patterns that can represent and manipulate information. This is the basis for the **binary computer language** used in **digital electronics** and serves as the core foundation **for modern computing**.
2. **Resistor** – impedes the flow of electricity through a circuit thus controlling voltage and current.
3. **Capacitor** – stores electrical energy.
4. **Inductor** – uses magnetic fields to control the flow of electricity. Found in power supplies that convert a battery or **Alternating Current** (**AC**) wall power supply into low voltage **Direct Current** (**DC**). Capacitors and inductors regulate and stabilize voltages.
5. **Diode** – like transistors except that they the direction of electron flow thus acting as a one-way gate or valve for electricity.

These discrete components vary wide in shape and size and are used in different combinations to manipulate the flow of electricity. They can either be discrete (manufactured separately, non-integrated) or integrated (manufactured on the same substrate).

**Integrated circuits** – instead of manufacturing discrete components separately and connecting them afterward, patterns can be etched together on the same chip, or die, using specialized manufacturing technologies like **photolithography**. ICs can be fabricated by the tens of thousands on a single wafer. This creates many benefits including: saving power (smaller system, less power to drive current across shorter distances), increasing speed (greater density), and reducing area. All of this lowers manufacturing unit costs by reducing the required material per process run.

The first functional digital computer (**ENIAC**), invested **by J. Presper Eckert** and **John Mauchly** at University of Pennsylvania in 1946, was basically a giant room full of vacuum tubes taking up almost 1,800 square feet and weighing nearly 50 tons. ENIAC-on-a-Chip, 1996, measures 7.44mm x 5.29mm by comparison and illustrates the progress made by the SCI in shrinking computer sizes over the course of several decades.

**Transistor Structure**

Modern transistors called MOSFET (metal-oxide field effect devices) consist of a source, gate, and drain. These features are analogous to the emitter, base, and collector in earlier bipolar junction transistors (BJT). The SGD blocks are commonly made of silicon, which by itself is not useful. The magic of semiconductors happens after doping silicon with impurities that either have an excess (N-type) or a deficit (P-type) of electrons. Transistors come in two configurations NPN (NMOS) and PNP (PMOS). The important thing to note is that the gate is a different charge from the source and the drain.

**How Transistors Work**

Without power or voltage, a transistor is stagnant and thus no electrons flow through the system. However, when a positive voltage is applied to the gate (for NPN), the excess source and drain electrons are drawn to this voltage, while the positive gate charges are pushed away. This opens a **channel** through which the electrons can flow, allowing current to pass through.

**FinFET vs. MOSFET Transistors**

There are two primary transistor types – bipolar junction transistors (BJT) and field effect transistors (FET). BJTs are primarily used for a limited set of applications like power management and signal amplification for wireless and audio devices. FETs are used in most modern computing devices and the most popular configuration within the FET family is the MOSFET (metal oxide semiconductor field effect transistor). The MOSFET was developed at Bell Labs in the 1970s and has served as the bedrock for microelectronic design and manufacturing for decades.

In the MOSFET a metal oxide separates the gate from the channel. An electric field, when applied to the gate, creates a channel between the source and drain through which electrons flow. Just know that there are structural differences that make MOSFETs different from other types of transistors.

As SC technology evolves, engineers have devised new and creative ways to make them more efficient. FinFETs have helped mitigate performance challenges as transistors reach their physical limitations. By raising the source and drain to surround the gate on three sides, FinFETs allow more efficient control of current through the transistor and also consume less power and reduce current leakage. The tradeoff is that FinFET are more difficult to manufacture than MOSFET.

There are other new developments on the horizon – Gate All Around (GAA) and Nanosheet transistors – that will enable greater control and significant performance advantages.

**CMOS**

Making high-performance ICs at scale is challenging and expensive especially as transistors shrink to ever smaller geometries. Most chips today use advance CMOS (complementary metal-oxide semiconductor) technology. CMOS may be used to refer to the circuitry itself, but can also refer to the design methodology and processes that are used to manufacture ICs. The “complementary” part of CMOS just mean that both p-channel and n-channel transistors are used on the same substrate.

CMOS has long been the dominant IC design and fabrication technology and enjoys a competitive advantage in power consumption, area requirements, and cost over more specialized alternatives like bipolar semiconductor manufacturing.

Each successive generation of CMOS technology has accomplished these advantages by shrinking transistors and other components through a process called **geometric scaling** in which the key metric is the gate length (source to drain distance). For example, 7nm technology refers to the gate length of 7nm or equivalent 7nm between source and drain. Geometric scaling is at the core of Moore’s Law.

Geometric scaling has driven functional scaling which measures meaningful, real-world performance improvements. Because geometric scaling has been so impactful, engineers have not had to squeeze as much efficiency out of their designs at each process node. In recent years, however, the pace of geometric scaling has slowed as the physical (unwanted effects such as quantum tunneling) and practical boundaries (transistor size is inversely proportional to manufacturing difficulty and cost, EUV) of transistors size is being reached. Each successive generation of SC manufacturing technology is called a **technology node** or **process node** driven by a mix of improved equipment, new materials, and process improvements.

Heading into the future functional scaling will start playing a more significant role since it aims to increase performance by maximizing utilization at existing transistor sizes by way of application-specific design, tighter system integration, and developing new packaging and interconnect technologies.

**How Transistors Are Used**

Transistors used together can form the building blocks of computer engineering – **logic gates** – which are simple circuits that use Boolean logic to enable simple computation (AND, OR, NOT). Using logic gates as a lowest common denominator or functional unit, hardware engineers can build complex systems that perform important base functions like arithmetic operations. ICs are really just collections of functional components all fit together on a single piece of silicon.

**Chapter 3 Building a System [41 – 56]**

Transistors are rather unremarkable on their own. It is only strung together, with trillions in investment and the herculean efforts of millions of humanity’s brightest minds focused on building ever more complex designs that transistors can do what they do today.

The most advanced IC’s in production require as many as two trillion transistors on a single chip. To accomplish this feat, hardware designers must group and organize electronic systems across ever higher levels of abstraction. Furthermore, we must understand how different abstraction levels fit together.

**Different Levels of Electronics – How the System Fits Together**

The ground level of an electronic system consists of a combination of (1) separate discrete components soldered directly to a printed circuit board (PCB [other components like larger capacitors and inductors that make up a system’s power circuitry lend themselves to integration at the package or PCB level]) or (2) functional components integrated onto a single die (groups of transistors).

Electronic systems can be viewed as a hierarchy:

1. **Discrete component** level - transistors and other discrete components comprise the basic building blocks that form the foundation on which all higher-level systems are built.
2. **Integrated circuit** level – entire chips are developed and designed using a combination of smaller discrete or functional components. Designs can be extraordinarily complex with billions of transistors (e.g. CPUs) to specialized circuits (e.g. memory)
3. Semiconductor **packaging** level – individual, and sometimes multiple, ICs are wrapped in a protective enclosure to ensure they do not experience interference from neighboring components. Multiple components may be grouped into something referred to as a **module**, which describes a bundle of smaller circuits and components that work together in a unit to perform a task.
4. **Printed circuit board** level – smaller components at “lower” levels (usually seen as black components) are soldered onto a board, where they are connected to one another to form a larger system. A PCB mechanically supports and serves as a foundation by which electronic components are connected to one another using conductive tracks, pads, and other features etched onto its surface.
5. **System** level – everything is tied together to create a fully functional system or product (laptop, phone, PC, television, etc…). It is important to note that the term “system” can be used at different levels to describe a fully functional and discrete structure as it relates to the task it is designed to do at that level.

**Integrated Circuit Design Flow**

The SC design flow can be broken down into six major steps. Understanding these steps end-to-end can serve as an anchor on which to build a deeper understanding of each of its constituent parts.

1. **System level architecture** – system architect develops an idea of what chip their team is trying to design based on input from a business and marketing team that understands the market and/or customer need. The architect needs to decide specifically what the chip will do, what technologies, materials, and components will be used to build the chip, and how the team will evaluate whether a chip is a success. Architects continuously monitor the progress throughout the design process and guide their team as they move through each step of the way. Additionally, they have to interface with business and marketing throughout the process.
2. **Front-end design** – after hammering out higher level details and before construction begins, engineers create detailed models of the system. High-level model 🡪 more specific models 🡪 detailed schematics. This is the stage where **logic design engineers** fill in the details in a process often referred to as **logic design** or **RTL design** (**register transfer level**) that uses **hardware description languages** – such as VHDL or Verilog/System Verilog RTL – to describe what they want to the circuit to do.

HDL’s are specialized computer “programming” languages used to describe the physical structure of ICs and electronic systems. Physical design tools translate these languages into the individual gates and transistors for a given function. At the end of this state, designers should have completed a virtual version of a chip that should, in theory, serve its purpose flawlessly in the real world.

1. **Design verification** – to ensure a successful project, you must verify that the output from front-end design will translate into a fully functional subsystem or **system on chip** (**SoC,** are ICs that include an entire system on a single substrate, e.g. processor, memory, and GPUs all on a single substrate). Verifying a complex design is a difficult task and can take up over half of the total design time. Verification is critical because chip manufacturing is so expensive.

One approach, **functional verification**, which uses SystemVerilog HDL code to simulate the exact behavior of the circuit in any possible condition. **Universal verification methodology** is a relatively new way to verify numerous possible conditions. UVM even has the capability to collect getting statistics on which portions of the design have been verified, and which have not.

Any flaws in design (golden model) should result in an incorrect output from a given testbench input which can be identified and corrected through debugging (after debugging the golden model becomes the actual implementation to be sent to manufacturing).

An alternative, or supplemental, verification methodology, **emulation** uses FPGA (field programmable gate array) to program a design and observe a circuit in the real world. To do this, an **FPGA emulator** can be “programmed” with the newly designed audio processor (for example) so you can listen directly through your headphones or speaker. FPGA is not used to build something ready for market, but rather a testable prototype as close to the real deal as possible. Emulation is clunky and expensive, but the results are more tangible.

1. **Physical design** – the chip is physically built at this stage. How does the output of the front-end and verification process using RTL code turn into wires and transistors? Advanced **EDA (electronic design automation) tools**,which is incredibly complex and can sometimes take as long as the front-end design phase, bring what your front-end design team has built into the real world. Physical design is broken up into five sub-steps:
   1. **High Level Synthesis (HLS)** – marks the end of front-end design and the beginning of back-end design. Physical design engineers now convert RTL code into transistors and wires using **synthesis**. For a billion-gate design, synthesis (Synopsys, SNPS, is a leading developer of synthesis tools) is incredibly process. By converting from one language to another; physical, or “back-end,” design engineers can manipulate the front-end design team’s work and complete the next stage of the design process.
   2. **Design Netlist** – the netlist is the product of HLS and comprises a list of electronic components in a circuit and all the nodes (circuit nodes, individual elements that an electrical signal can be sent to) to which they are attached.
   3. **Floorplanning** – physical design engineers decide where everything should be located. Floorplanning ensures that those blocks are placed in such a way that PPAC is optimized given the limited space available.
   4. **Place-and-Route** – engineers decide exactly where to put all the electronic components and circuitry in the placement step during which a placement tool is used to allocate specific locations for each and every logic gate. Routing follows placement where CAD tools integrate all the wiring required to connect the placed components.
   5. **Clock-tree Synthesis (CTS) –** engineers make sure that the electrical signals that delivers information around the circuit “clocks” evenly, or as intended, throughout the chip.

Clock frequency, or clock rate, a common measure of process speed and performance, assesses how quickly a signal travels through an IC, which thus determines how quickly it can execute instructions. **Synchronous design** uses a common “clock” across all circuits. All parts of the circuit processing instructions at the proper clock edge – or capture edge – is critical to proper system function otherwise system failure or severe degradation in performance can result.

A common simulation method to compute and verify the expected timing of digital circuits is called **static timing analysis (STA)** that ensures that all logic paths are properly timed relative to each other so there is never a change of a timing error.

1. **Validation (physical verification or back-end verification)** – the circuits is ready for manufacture and a GDS design file is generated that includes all the necessary information required by the SC factory or fab. Prior to actual shipping of the GDS file, validation engineers need to double check that the chip is manufacturable by using **electronic design automation (EDA)** software tools like **design rule checkers (DRC)** to verify that the chip complies with all the rules (chip size, wire proximity, component proximity, etc…) of the chosen foundry.
2. **GDS II generation (GDS II)** – is a standardized format that is used to send a finished design to the fab at the end of the IC design cycle (**RTL-to-GDS**, also called **tapeout**). This stage is reached after the culmination of millions of dollars of engineering ingenuity.

Manufacturing is neither simple nor cheap with modern processes taking over 12-16 weeks from GDS to finished wafer.

**EDA Tools**

EDA, also known as e-CAD, includes tools that are used throughout the design process to help hardware engineers and chip designers as they build electronic systems. EDA tools automate and streamline everything from functional verification to high level synthesis. EDA tools have reduced the cost and difficulty of an arduous and complicated design cycle.

EDA companies (three biggest are Cadence, Synopsys, and Mentor Graphics) helped develop and spread VLSI HDLs like Verilog and VHDL, which automated the design flow and enabled further technology development. The EDA tool market was worth about $11B in 2020 and is expected to grow to over $21B by 2026 (Mordor Intelligence, 2021).

A good design on its own is no more useful than a drawing of your dream home – we’re halfway there, but we are still a long way away from walking in the front door.

**Chapter 4 Semiconductor Manufacturing [57 – 80]**

**Manufacturing Overview**

SC manufacturing involves the steps from the point that the GDS II file is received. Once received, manufacturing starts with **front-end manufacturing** through to **final assembly** and **testing**. SC manufacturing is a highly complex and ultra-precise process that requires specialized chip factories called **wafer fabs**. Key metrics of chip manufacturing are die area and batch yield.

Within the industry, a particular manufacturing process is called a **technology node**, **process node**, or simply **node**. The term node refers to the minimum feature size for a given generation of process technology and usually is specified by the source-drain distance or equivalently the gate length. The most advanced node is the 2nm node which Samsun plans to put into production in 2025.

It is instrumental to map out global manufacturing capacity by node segmentation and chip segmentation (see Figure 4.1, page 59) where chip segmentation is broken up into memory, logic, and Analog + OSD (discrete, analog, optoelectronics, and sensor devices [DAO]).

A **wafer run** is a single run-through of the SC manufacturing process, from initial **wafer fabrication** to when the individual die are cut apart from each other during **wafer dicing**. The manufacturing process is divided into two subsections – **front-end** manufacturing (puts the desired circuit onto a silicon wafer) and **back-end** manufacturing (gets the individual chips on that wafer ready for a customer’s system).

**Front-End Manufacturing**

The first step in front-end manufacturing is called **wafer fabrication**. A wafer is a thin slice of **substrate** (melting a combination of silica and carbon down into cylindrical **ingots**) cut from an ingot on which any number of chips are built. In the 1980s wafers use to be 150mm and today is 300mm with a push for 450mm wafers.

The front-end is meant to build intricate combinations of substrates, circuitry, and other materials one on top of the other with the precision and accuracy needed to make a fully functioning chip. There a four major stages in the front-end:

1. **Deposition** – addition of thin films onto a wafer’s surface by way of atomic layer deposition, molecular bean epitaxy, physical vapor deposition, and electrochemical deposition (this is not an exhaustive list of methods). Deposition equipment is very intricate and expensive. In the case of physical vapor deposition, the substrate is placed in a vacuum across from a sputtering target that deposits sputtering gases onto the substrate.
2. **Patterning/Lithography** – a process that shapes or alters the material on the wafer. **Photoresist** deposited on the wafer breaks down in a reaction to light (with a **specific wavelength**) that passes unimpeded through a mask (some designs can require up to 75 masks) that is aligned using a **stepper** (the stepper also functions as a light source). The photomask is specific to a single layer of the process for a given chip design. Electron bean (e-beam) lithography is another method used to manipulate photoresist. Once the photoresist has been removed, metal or other materials can be deposited into the remaining areas to form wires connecting individual transistors and functional features to one another.

Lithography is a critical bottleneck technology that’s enabled geometric scaling to keep pace with Moore’s prediction over the last several decades since each generation of lithographic equipment enables fabs to etch smaller substrate features and thus pack more transistors on each chip.

In order to move to more advanced nodes, lithographic equipment suppliers have had to continually find new and creative ways to make ever smaller patterns and transistors. One primary mechanism for doing this has been to use shorter wavelength light sources (e.g. EUV lithography, see the case of Intel and its huge investment in EUV starting in the 1980s).

Light can only directly etch features as large as its own wavelength. Optical workarounds and the use of multiples of photomasks have allowed fabs to etch smaller patterns that the wavelength of light being used would directly allow. However, as SC feature sizes kept decreasing, performing photolithography at the then 193nm became increasingly difficult. This is one of the reasons by EUV has been critical to maintaining the geometric scaling demanded by Moore’s Law.

1. **Removal** – processes such as wet etching or dry etching and chemical mechanical planarization (CMP) are used to wash away the photoresist material that is no longer needed, leaving an area that can be later filled with the desired metals, oxides, transistors, or passive components on the underlying wafer material.
2. **Physical Property Alteration** – processes, like doping (ion [impurities] implantation, ion introduction), rapid thermal annealing, ultraviolet light processing, and others – that modify the electrical or physical properties of the wafer responsible for the behavior and performance of transistors and other functional components.

**Cycling – Pre- and Post-Metal**

Each of these four process types are repeated many times before enough layers are properly fabricated in our wafter “layer cake”. Some high-end chips require hundreds of steps for a single production run. The six steps wafer surface -> deposition -> photolithography -> removal -> doping -> repeat can be used across both FEOL (front-end of the line, pre-metal processes) and BEOL (back-end of the line, post-metal processes).

Modern devices can have as many as 15 layers, with upper layers connected using vertical **via structures**. Additionally, there are lower-level local interconnects and upper-level global interconnects.

The entire front-end manufacturing process for a complex wafer can require several dozen mask layers and take weeks to finish. These challenges are reflected in the industry’s cost distribution, with front-end manufacturing machinery comprising 60% of the $62B in 2020 spent on SC production equipment.

Each successive node brings an added layer of complexity which makes wafer probing, yield, and failure analysis even more crucial to meeting production goals and keeping unit costs low.

**Wafer Probing, Yield, and Failure Analysis**

Wafer probing is the process after FE and before BE (dicing, packaging, assembly, and final testing) that uses a wafer prober (electrically tests the wafer die) to see if the wafer actually works. Wafer probing is done in instances where the BE is lengthy and expensive. In a **chip-scale packaging**, the entire wafer is packaged, and testing is done afterwards.

Two modes of tests – 1) **parametric testing** of the wafer fabrication process (designed to check if the entire process is working by checking [using a spacers and **scribe line**] to see if basic parameters such as resistances and device thresholds are within standard tolerances) and 2) **wafer testing** to ensure each individual die is defect-free and fully functional by identifying dysfunctional die for disposal, performance measurement, and monitoring recurring errors. Wafer testing cash identify issues with fab equipment which may be contributing to systemic failures in a fabrication run so that the equipment can be tuned to improve subsequent yields.

Testing allows failure analysis engineers to derive the fabrication process’ **line yield** (also known as **wafer yield** measure how many wafers as a whole successfully make it through wafer probing without having to be thrown out) and **die yield** (measures the number of functional die divided by the total number of potential die that make it through wafer probing). The **end-to-end yield** is the composite of line yield and die yield and holistically accounts for the efficacy of the entire FE manufacturing process.

For a new manufacturing line of chips, yield will generally start lower (advanced processes can have initial yield around 50%) and gradually increase as the equipment is properly calibrated and manufacturing engineers have time to adjust process steps. **Yield optimization** has long been considered one of the most critical performance objectives because even small yield increases (through both line and die yield) can drive down manufacturing costs and boost margins and this can provide significant competitive advantage. Work out examples of yield optimization and its impact on profitability (see page 72 for example).

Yield is usually directly proportional to the number of die on a wafer because tiny contaminants (air filtration outfitted **clean rooms**) or slight movements (vibrations are so critical that fab equipment is frequently mounted on springs and air suspension systems) can permanently ruin a given die, smaller die sizes typically result in higher yields, since failure is more likely to be contained in a smaller portion of the overall wafer area. Defective die are often marked with a black dot so they can be tossed our of sold at a discounted price if still semi-functional (this is called **inking**).

Specialized air purification and construction requirements add enormous cost to fabrication equipment and persistent re-tooling, which makes up the majority of a new fab’s price tag. A single 3nm equipped fab can cost $6-7B to as much as $20B and become obsolete withing 5 to 6 years. To put these costs in perspective, capex for US SC companies amounted to roughly 30% of sales in 2020 compared to 4% for the manufacturing sector as a whole.

Older fabs can sometimes be sold “down-market” to mixed-signal or analog companies that are not at the bleeding edge of the tech curve, but these sales are frequently at pennies on the dollar.

**Back-End Manufacturing**

After a wafer has been tested, dies on the wafer are ready to be enclosed in IC packaging and begin BE assembly and testing. Most A&T work is done by third parties called OSATs (Outsourced Assembly, Test, and Packaging Suppliers) which are largely based in East Asia and enjoy significant labor cost advantages. There are several steps in A&T:

1. **Wafer bumping** – this step is not always performed, but in cases where the bare die is connected directly to other components, this initial step places small solder balls (**bumps**) directly onto the wafer.
2. **Wafer dicing** – **die cutting** cuts individual die from a wafer by a diamond saw and sent to a BE facility for final packaging and assembly.
3. **Die bonding** – freshly cut die at an OSAT are either attached to either a packaging substrate, directly to a PCB (**die attach**) or simply packaged as bare die (**flip chip**). **Epoxy die attach** is the most common bonding process and **flip-chip bonding** functions as a die attach method as well as a method for forming system interconnects between the die and the rest of the system.
4. **External interconnect formation** **(flip chip or wire bond)** – the attached die are connected to the rest of the system through little wires from the die that lead out to the periphery of the package forming **interconnects (I/O)** with the rest of the system. This process is called **wire bonding** and results in fewer I/O than more advanced flip-chip technology.[[2]](#footnote-3) Flip-chip packaging involves flipping the die over and soldering it to a **ball grid array** or directly to the PCB thus forming I/O’s throughout the chip’s area thus increasing the overall speed of the system.
5. **Encapsulation and sealing** – In the encapsulation stage, **surface mount technology** is used to mount the die onto the IC package enclosure. Next a **transfer molding** machine heats encapsulant compounds or **molded underfills** before injecting them into the packaging mold, sealing in the **die-package assembly**. At this point we have a fully “assembled” die-package assembly that is ready for final testing.
6. **Final testing** – the die-package assembly is tested one final time before shipping to the end customer or integrated into an intermediate system or product. Note that in some well-established technologies with very high yield (>90%), final testing may be the only testing of the individual die. When wafer testing is too expensive, it can be more economical to package every single die and test afterwards, simply discarding the die that fail at final testing.

**Semiconductor Equipment**

We can see the extent of the manufacturing costs break-down by examining figure 4.17 (page 78) of the $64B SC equipment market. This figure breaks down 2019 sales data into 11 types of SC equipment (Deposition [$12B], RTP + Oxidative Diffusion [$1B], Lithography [$12B], Photoresist Processing [$2B], Removal Processing [$16B][[3]](#footnote-4), Doping [$1B], Metrology and Inspection [$6B], Manufacturing Automation [$3B], Other Wafer Fabrication Equipment [$2B], Testing Equipment [$6B], Assembly Equipment [$3B]) with FE equipment accounting for 86% of overall equipment sales.

As chips continue to shrink, the manufacturing difficulty grows, pushing up costs of such equipment. This is especially true for core FE technologies responsible for delivering increasingly smaller patterns like deposition, lithography, and removal. Figure 4-18 (page 79) is an excellent summary for the end-to-end manufacturing process. Tying all of these technologies together, the modern SC fabs are incredible feats of human ingenuity responsible for the proliferation of computing devices across the world.

**Chapter 5 Tying the System Together [81 – 95]**

**What is a System?**

A well-designed IC is only as good as the system it’s a part of. As Moore’s Law slows down and companies lean more heavily on functional scaling to meet the demands of their customers, system integration (advanced I/O, next generation IC packaging, signal integrity, power distribution networks) has become increasingly important.

Many devices are made from a collection of different ICs and components that may have been designed by entirely separate companies using proprietary methods and unique microarchitectures. Integrating all of them is a complicated and difficult task. Additionally, connection points between each module are often a bottleneck for data flow that lead to latency and an overall slowdown in the system as a whole.

To connect the various sub-systems and make sure power and data get to the right place, a well-designed network of interconnects, strong packaging technologies, and good signal and power integrity analysis are vital to developing high-functioning systems.

**Input/Output (I/O)**

Within a single chip, interconnects are the wiring that connects different components, like transistors, with one another to form logic gates and other functional building blocks. At a higher level, interconnects refer to the connections between the chip and the PCB, other components or chips on the PM, and other parts of the system as a whole. Larges chips can has as much as 30 miles of stacked interconnects. System designers have many different options in connecting the I/O within a system, which all starts with selecting the proper package for each IC.

**IC Packaging**

Electronic packaging protects the chip from the outside world and supports the electrical interconnects that connect the device to the rest of the system. There are many packaging options including:

1. **Wire bond** – when ICs were first gaining traction, packaging interconnects were limited to wire bonded connections from IC bonds and landing pads inside the packaging to pins that soldered onto the external board or substrate that supported the rest of the system. Wire bonding limits the number of possible interconnects from an individual chip, since they can only be located on the border of the die and packaging.
2. **Flip-chip packaging** – the option solves the limited interconnect issues of wire bonding by enabling interconnects to be placed within the borders of the die itself by adding steps to the back of the manufacturing and assembly process (this is a six step process).
3. **Wafer-level (chip-scale) packaging** – begins the packaging process before the wafer is diced. Doing this results in a smaller die-package assembly that is approximately the size of the chip itself and this is why it’s referred to as chip-scale packaging. This option only works for small die sizes, which further limits its usage.
4. **Multi-chip-modules and System-in-package** – integrate multiple die into a single package and also useful for limited-space applications. MCM is 2D only whereas SIP is 3D packaging. These two options allow engineers to modulate parts of the design and more easily incorporate licensed IP. However, mixing and matching multiple die as opposed to a system-on-chip approach (different functional units in a single die), leads to performance and power disadvantages. While suffering some integration disadvantages, MCM/SIP architectures allow the use of a cheaper silicon process for the analog functions of the system, and only use the expensive lower-geometry processes for the critical high-speed processing and memory functions. SIPs also allow for passive devices like capacitors and inductors to be integrated into a single package which can improve performance by minimizing the distance between components.

The tradeoffs between SoC chip-level **monolithic integration** and package-level **heterogeneous integration** is a pressing question facing many system architects today.

1. **2.5D/3D Packaging** – It used to be that all packages and their enclosed chips were attached to a PCB or other substrate through metal pins, which then connected that chip to another part of the system through a network of wires. Today advances such as **die stacking** have allowed design teams to stack multiple die on top of one another using vertical interconnects called **through silicon vias (TSVs)** to for 2.5/3D packaging architectures.

In 2.5D packaging, die are connected to a shared substrate called an interposer, which in turn is connected to a PCB while in 3D packaging, die are stacked directly on top of one another. 2.5D is not as tightly integrated but is less costly and more tightly integrated than a separately packaged wire bond configuration.[[4]](#footnote-5)

New copper hybrid bonding technology uses copper-to-copper interconnects to connect stacked die with greater interconnect density and lower resistance, enabling quicker data transfer and faster processing speeds than tradition TSVs. These advanced packaging technologies have also enabled the design of integrated modules including die from different silicon processes. For example, a 22nm processor chip and a 180nm high-power audio amplifier can be included together in a single plastic module.

Stacking technology was first applied to memory systems like hybrid memory cube and high bandwidth memory in a memory-on-logic mode and memory-on-memory mode. Stacking chips vertically enables increased I/O density which reduces the amount of processing time it takes to move a signal or information around a circuit and system while also saving valuable real estate. This can be especially useful for more compact devices like smartphones where space is limited and valuable.

While comprising a relatively small portion of our total system cost, making up just $30B out of $440B in total 2020 sales, IC packaging has an outsized impact on system performance and has seen a resurgence of interest from engineers desperate to squeeze more performance out of existing process nodes.[[5]](#footnote-6)

**Signal Integrity**

As components are packed tighter and tighter in integrated systems, signal integrity (digital signal represented by pulses that travel from a transmitter to a receiver along a transmission line) has become increasingly important.[[6]](#footnote-7)

In complex circuits where wires can be a few nanometers apart from one another, neighboring signals can interfere with one another and the surrounding environment. These transmission line effects on signal integrity can result in data loss, accuracy issues, and system failure.

Signal integrity engineers can mitigate these effects by conducting electromagnetic simulations and analysis to identify and resolve common forms of interference including **noise**, **crosstalk**, **distortion**, and **loss** (**resistive loss**, **dielectric loss**, **radiation loss**). At high bit rates and great distances, signals can degrade to a point where an electronic system fails completely.

**Bus Interfaces**

One of the key performance bottlenecks in electronic devices is the transmission of data between each of the system’s constituent parts. To unlock the power of advanced circuits, bus interfaces, the physical wires through which data travels between system and/or PCB components have become increasingly important. Bus interfaces have three primary functions:

1. **Data bus** – transmits data
2. **Address bus** – find specific data
3. **Control bus** – control operations of different parts of the system

Together these three buses form the system bus and collectively control the flow of information to and from a CPU or microprocessor (8 to 64 bits at a time or during one clock cycle), memory, and I/O.

To increase speed and improve performance, PC companies migrated to an integrated structure that reduced the number of interface junctures from a cluster of haphazardly connected components and modules down to two chips – the **Northbridge** and **Southbridge** **chipset architecture**.

The Northbridge interfaces directly with the CPU via the front-side bus (FSB) and connects it (bridges it) with components that have the highest performance requirements (PCI-E and memory). Northbridge also connects to Southbridge which in turn connects to all of the lower priority components and interfaces such as Ethernet, USB, and other low speed buses (collectively known as the **peripheral buses**). NB and SB are connected to one another at a juncture known as the **I/O Controller Hub** and together they are known as a **chipset**.

Bus interfaces are classified by the way they transmit data between components. Parallel interfaces run multiple wires between two components and transmit data simultaneously. This works well over short distances, but signal integrity issues can arise as the distance between components increases. **Parallel interface buses** include Double Data Rate (DDR, for memory) and Peripheral Component Interface (PCI). **Serial interfaces** transmit one bit at a time across a single wire but at a much higher speed. Serial buses reduces the chances of signal integrity issues and is less likely to experience crosstalk. However serial data is susceptible to **Intersymbol Inference** where data bits can be affected by the bit transmitted just before it. Common serial interface buses include PCIe (PCI Express Bus), USB (Universal Serial Bus), SATA (Serial Advanced Technology Attachment Bus), and Ethernet Bus.

Beyond signal integrity advantages, serial interfaces are less costly due to lower wire count, but suffer slower transmission speeds. Parallel buses, by contrast, enables faster data communication, but is costlier and has limited efficacy over long distance when operating at high frequencies.

**Power Flow in Electronic Systems**

Harnessing the power of “power” is the subject of a vast field of engineering and numerous sub-disciplines. Power starts with a **voltage source** (battery or power line) that serves as a reservoir of electrical charge. A **power converter** converts AC power (is much better for transmitting power over long distances) into DC electric current (much better for power transmission over short distances). Next the charges or electric voltage from the battery are transported through a network of metal planes called a **Power Distribution Network** to the different processing centers of the system.

Power engineers must build a network of **voltage regulators** and **power converters** to ensure that voltage is never too high or too low at any point in the system. **Power integrity** studies the flow of voltage throughput a system to ensure that voltage is getting to the right place, in the right amount, at the right time.

Voltage regulators are circuits designed to maintain a fixed voltage output regardless of the input voltage they receive. Voltage regulators ensure that all components in the system receive a stable voltage, even if the battery voltage is moving around. There are different varieties of voltage regulators including DC/DC converters, power management units, Buck converters, and Boost and Flyback converters.

A company may source and design the perfect assortment of ICs and components to create a market-leading product, but having the right parts is not enough on its own. Building high caliber devices requires tight system integration with plenty of interconnects, the right IC packaging architecture, and strong signal and power integrity performance.

**Chapter 6 Common Circuits and System Components**

Though our discussion has helped us build a holistic model of electronic systems, thus far it has largely treated SCs as a monolith devoid of differentiating features. We will now break apart this monolith, exploring the numerous types of common circuits and system components that comprise the SC family.

**Digital vs. Analog**

There are two main types of components that get their name from the type of signals that they use – **digital** and **analog**.

**Digital signals** are usually synchronous, that is, they run on a reference clock to coordinate the processing of different functional blocks and ensure proper timing. Though their predictability and synchronous timing makes them great for storing and processing information, digital circuits are unable to transport information over any sort of distance without physical wiring to move their signal from place to place.

**Analog devices** process information continuously as a range of values and their ability to capture and transmit electromagnetic energy makes them well suited for applications like wireless communication. Much of the energy from real-world signals are analog in nature (e.g. sound and light). Analog signals are distinguished by their **frequency** (measured in Hertz, Hz, Heinrich Hertz) and this frequency is inversely related to the signal’s **wavelength** and directly related to the signal’s **power**. By receiving and processing different frequencies, analog electronics can perform all kinds of useful things from detecting external stimuli (**sensors**) to wireless data transmission and communication (**RF tehnology**).

The difference between analog (sensors) and digital (storage and processing) signals make them more useful for distinct parts of electronic systems. In many electronic systems, analog and digital components work together to “translate” real-world analog signals into digital signals that computers can understand, then re-translate the digital response from the computer back into analog signals we humans can understand. To accomplish this, mixed-signal devices called data converters are used to convert between signal types (**ADC** and **DAC**).

**Common System Components – The SIA Framework**

Driven by demand in the six end-use markets (communications, computing, consumer, automotive, government, and industrial) the component market is diverse and highly competitive. The markets are so vast, it is helpful to use the SIA’s framework for breaking the market into five constituent segments.

**Micro Components**

Includes all non-custom digital devices that can be plugged into another system and used for computation or signal processing. These are generic sub-components and specifically include microprocessors, microcontrollers, and digital signal processors (DSP).

*Microprocess and Microcontrollers*

A process is a chip that receives input, processes that input, and produces an output that can be used for some intended purpose. Microprocessors (MPU) is generally used to describe more complex digital circuits, like CPUs, that connect to a larger system. They perform a general computing function and require an external bus to connect to memory and other peripherals components. MPUs handle general computing tasks. PCs and servers account for the larger fraction of microprocessors sales.

Microcontrollers perform specific functions and are integrated with memory and I/O all on one chip. Microcontrollers are smaller and less powerful processors that can serve as plug-and-play computing power for simple operations. They are widely used in low power IoT devices and embedded systems. Automotive, industrial, and computing account for most of microcontroller sales.

It is important to distinguish microprocessors and microcontrollers in the Micro Component segment from processors in the Logic segment. Logic devices are custom designed for a specific application, which Micro Components provide more generic processing that can be combined with other components in all kinds of systems.

*Digital Signal Processors (DSP)*

Used to process multimedia and real-world signals like audio, video, temperature, pressure, position, etc. Digital components have trouble accurately representing the real world in 1’s and 0’s and this is why DSPs are needed to translate real-world signal is a form that digital components can understand. The most common pathway for signals is ADC to DAC. DSPs are adept at high-speed, real-time data processing and are highly programmable, which makes them easy to implement in a wide variety of devices and systems.

*Micro Component Market Summary*

MPU, MCU, and DSPs accounted for $69B of $440B on 2020 sales (16%, SIA and WSTS’s 2020 End Use Survey). Comprising 57% of the segments’ end-use applications, Micro Components are much more heavily weighted towards computing applications than other components in the SIA Framework.

**Logic**

Encompasses all non-micro component digital logic and refers to specialized circuitry including application specific ICs (ASICs), field programmable gate arrays (FPGAs), and more versatile, but application-specific digital logic devices.

*Special Purpose Logic*

Encompasses all ICs designed and sold as standard products ranging from wireless controllers like Ethernet and WLAN, Modem SoCs, image and audio processors, PC core logic, and GPUs. SPL devices are application-specifc standard parts (ASSPs) that are designed and integrated into a system the same way ASICs are. Standard means that the same part can be used in many different products (Ring doorbell, LCD TV, handheld game console). A custom part is specifically designed for a single device. High-volume consumer products like iPhones utilize many different custom chips since volume warrants customization that is intended to squeeze every last bit of performance out of the specialized silicon.

ASICs are designed and optimized for a specific use in a single system (Samsung designs an ASIC CPU for its smartphone, AMD designs an ASIC GPU to power Xbox) as opposed to a more generalized application (Intel designs a server-based CPU aimed at all data center computers). Standardized product types such as I/O circuits like USB or PCIe are also classified as ASSPs.

*Central Processing Unit*

A CPU is a special type of microprocessor. They are most commonly used in PCs and laptops but as a general class are not limited to these end-products. CPU are found in all sorts of products like smart speakers, automotive control systems, and any device that processes information (even in your coffee maker). A CPU is the digital brains of the system, processing and executing instructions as needed. The core processing of a CPU is handled by the **arithmetic logic unit** (ALU), which performs numerical and logic-based operations necessary to run all the software it was intended to deliver.

CPUs are typically connected to other modules through a bus or chipset that feeds information to the CPU through **registers** for processing and directs output data to memory for storage or to other system components. Each CPU has a fixed number of registers through which data can flow with typical registers having capacities measuring 8, 16, 32, or 64-bit wide.

An individual CPU or GPU may be referred to as a core that can be combined with other cores to form **multi-core architecture**. Together, the CPU and other components are “integrated” onto a single IC (SoC) or a larger system. A system may include a separate CPU, memory, GPU, power source, and multimedia processor, while another system may integrate all of these onto a single SoC or Multi-Chip-Module (MCM).

*Graphics Processing Units*

Best known for driving the graphics and 3D visual processing in electronic devices. They implement parallel processing as opposed to serial processing that is used by CPUs. Parallel processing enables a processor to break down more complex problems into smaller constituent parts. GPUs can perform thousands of specialized operations using hundreds of cores, though they are not as efficient at handling more diverse operations. In summary, CPUs are better at performing a high variety of tasks like running all the programs and functions of a PC while GPUs lend themselves to applications that require high-volume, repetitive calculations like graphics processing.

Some of the recent and exciting applications of GPUs are in AI and ML because GPUs are ideal matrix processors and since GPUs can break down complex problems into smaller, constituent problems, they are well equipped to handle the millions/billions of small trial-and-error calculations necessary to deliver arduous AI solutions. The same way companies created custom GPUs for graphics years ago, companies are now creating processors custom-built just for cryptocurrency mining.

While most of the SCI has been consolidating, innovations in AI-centric GPUs have led to a significant growth area where new companies have been able to compete. Specific applications ideal for GPY process include autonomous driving, machine vision and facial recognition, high performance computing (HPC), complex simulation and modeling, data science and analytics, bioinformatics, and computational finance, just to name a few.

*ASICs vs. FPGAs*

Represent two different approaches to chip design and development each with their own pluses and minuses. Since ASICs are designed for a specific purpose, ASIC chips operate at higher speeds with lower power consumption, smaller area, and at lower variable manufacturing costs at high volumes. The main disadvantage for ASIC chips is the significant upfront development costs associated with their design due to high levels of upfront capital and labor. Even if a chip is taped-out, there is always the risk of low yield and the chip not functioning the way it needs to for its intended function. Another ASIC drawback is that they are so customized for a given application that they can’t be used in another area. Thus each application needs a different chip which adds cost and complexity.

Field Programmable Gate Array (FPGA) are programmable chips which means they can be customized to serve a specific function after they have already been manufactured. Most FPGAs can be erased and re-programmed to serve a new purpose and this make them ideal for prototyping new designs.

Emulators are essentially a box with a bunch of FPGAs working together that allow ASIC designers to iterate their design before moving it to manufacturing. Emulators are becoming more important as chip manufacturing costs increase. Emulation is a way to further verify how a given design performs in the real world and be sure it will function correctly when it comes back from the fab. FPGAs are used in many applications including – video & image processing, security systems, scientific instruments, wireless communications, aerospace & defense, medical electronics, and consumer electronics.

For the last two decades, Xilinx (recently acquired by AMD for $35B, 55% market share) and Altera (acquired by Intel in 2015, 35% market share) have dominated the FPGA market.

ASIC or FPGA – Which to Choose… A key decision for many companies in need of an IC is whether to build a custom ASIC or use an off-the-shelf FPGA because the critical trade-off involves performance and price. If you’re working with short time-to-market constraints or lower than expected manufacturing volumes, FPGAs are usually the better choice, assuming there is some wiggle room for poorer performance (cannot have stringent PPAC constraints).

As volumes increase, ASIC becomes more attractive since you can spread the upfront capital and labor costs over greater volume and also capitalize on long-term yield improvements thus reducing net spend on materials and eventually costing companies less per device. Apple, Facebook, Google, and Tesla all develop custom ASICs for their devices in-house. How do you account for technological obsolescence in this product development and manufacturing dynamics?

One option that combines FPGA and ASIC is to use FPGAs initially to prove out a new product idea and demo a solution to generate customer interest. Once that hurdle is cleared, it can be easier to secure funding and corporate buy-in for the much larger expense for ASIC development.

*System on Chip*

SoC are complex and highly integrated type of ASICs that contain an entire functional device all on one IC substrate. To be considered an SoC, an IC must have at a minimum, microprocessor and/or microcontrollers, DSPs, on-chip memory, and peripheral functions like hardware accelerators. For smaller applications like cell phones (tablets, smartwatches, and other battery-powered devices where space is very limited), multiple chips may require too much space and power. In these cases, tighter integration mitigates these problems, enabling engineers to fit entire computing systems in the palm of your hand. Though SoCs are used most in embedded and mobile devices, they have been increasingly used for laptops and other devices that can still leverage their performance advantages.

Differentiating between ASICs, ASSPs, and SoCs can be a bit confusing. The main difference between the two is that ASSPs are designed to serve multiple companies and end systems, while ASICs are designed to for a single use by a single company or product. ASSPs and ASICs that contain a processor are considered SoCs while those that don’t are not. ASSPs and ASICs are designed using the same tools, methodologies, and flows.

*Logic Market Summary*

CPUs, GPUs, ASICs, FPGAs, SoCs, and other Logic devices accounted for $118B out of $440B (27%) of total industry sales in 2020. Comprising 44% of the segments’ end use applications, Logic is more heavily weighted toward Communications.

**Memory**

Classified based on whether memory storage can be done with (volatile memory, or RAM, requires power but enables quicker access) or without (non-volatile, or ROM) power. Dynamic RAM is the most common type of volatile memory, while NAND flash is the most common type of non-volatile memory.

Since the 1960s and 70s, the market needs for data storage has skyrocketed, driving the demand for more advanced memory chips to new heights year after year.

*Memory Stack*

Memory’s primary function is to store data and information for use in the processing centers of larger systems. Storage capacity is no longer the dominant performance constraint but rather the bridge between memory and the core system processors has become the key bottleneck for device performance. This has driven the development of new memory chips and microarchitectures.

Instruction, data, and information flow between the CPU and the memory stack starts with an input source, and its related input stimuli, and flows through the memory hierarchy for processing and storage. Input triggers core instructions to be readied by the long-term, non-volatile ROM memories, which are then sent to volatile RAM memories higher up the stack and thus closer to the CPU. These instructions are quickly transferred to L1 and L2 cache (quick access) memories, which directly interface, through a data bus, with the CPU registers. The CPU processes the data according to the instruction set and returns output instructions that can either be held in up-stack cache for quick re-use of delivered to down-stack permanent storage for another time.

Memory designers are always balancing memory capacity with access speed. This is why designers implement a memory hierarchy with smaller, faster cache used to store frequent time-sensitive operations for quick access and larger but slower memories used to store broader datasets that are needed less often.

Memory is classified into two broad categories: 1) **temporary storage volatile memory** (needs power, used for operations that underlie any live running programs or applications) and 2) **permanent storage non-volatile memory** (doesn’t require power, used for operations that never change like booting instructions). The second memory distinction is in read/write ability with RAM allowing both read and write whereas ROM only allow read operations.

In general, ROM memories are used for permanent data storage, while RAM memories are used for running programs and storing temporary data close to the CPU for quick access.

CPU registers serve as the bit interface where data is physically transferred from memory to processor for any given clock cycle. CPU registers 🡪 Cache (L1 and L2, volatile) 🡪 RAM, DRAM, and SDRAM (volatile, quick access, transient, low capacity) 🡪 ROM (non-volatile, slow, long-term, high capacity).

*Volatile Memory*

Most common volatile memories are DRAM (often used as temporary working memory) and SRAM (often used as cache memory). DRAM can hold more data than SRAM but is slower. SRAM has faster access speed but requires more power (this is a typical trade-off between performance and power). Computing speeds are also limited by the transfer speeds between memory DRAM and cache SRAM. This bottleneck can have significant performance implications that have been mitigated by DDR (double data rate) SDRAM.

*Non-Volatile Memory*

There are two primary categories of non-volatile memories: **primary memory** and **secondary memory**. All RAM memories are considered primary while some ROM memories are secondary. Primary memories are the main working memories of a computer, they can be accessed more quickly by the processor but have limited capacity and are usually more expensive. Secondary memories also known as **backup** or **auxiliary** memories can only be accessed through interconnects and are much slower.

*Primary Non-Volatile Memory*

1. Standard ROM cannot be adjusted or rewritten and must be programmed when it is created. Data is literally hard-wired when the chip is manufactured.
2. PROM can be programmed after manufacturing but cannot be changed once programmed.
3. EPROM can be erased and rewritten many times but erasing requires UV and all data on the chip has to be erased instead of selectively erasing parts of the data.
4. EEPROM the whole chip does not have to be erased and no UV is required. However erasing is done one bit at a time, making them relatively slow to erase and reprogram.
5. NAND Flash is a type of EEPROM that can erase information, write data in chunks and works considerably faster than EPROM. It is the primary type of ROM used to store data in electronics today.

*Secondary Memory (HDD vs. SSD)*

External non-volatile RAM memories used for permanent storage and core device functions like boot drive. **Hard disk drives** (**HDD**) and **Solid state drives** (**SSD**) are the most common types of secondary memory. HDDs use magnetic disks where as SSD used interconnected NAND flash which are faster and more reliable than magnetic storage but are more expensive and lower capacity. Moore’s Law, however, is making NAND flash cheaper and denser every year and has made all our portable electronics possible.

If cost and capacity are your main drivers then HDD is probably the better choice. If versatility and reliability are your main drivers then SSD is the better option.

*Stacked Die Memory (HBM vs. HMC)*

In many systems it’s the interconnects between chips that limits performance. New die stacking technologies connect chips directly to each other without the performance degradation of wiring from chip to chip. In addition to die stacking, 2.5/3D packaging architectures have enabled new, tightly integrated memory architectures with significant performance advantages.

**High-bandwidth memory** (**HBM**) and **Hybrid memory cube** (**HMC**) are industry standards used to build 3D memory devices. HMC uses a 3D memory-on-logic architecture where DRAM is stacked on top of logic and connected throughout using **Through Silicon Via** (TSV). HBM stacks memory on top of device logic but separates this functional stack from the CPU/GPU + Host Logic function using a interposer in a 2.5D packaging configuration. This allows HBM methods to use silicon from a more diverse set of suppliers.

*Memory Market Summary*

DRAM, SRAM, NAND Flash, Stacked memory, and other Memories accounted for $117B out of $440 (27%) of 2020 total industry sales. Memories are distributed across the different end-use applications at roughly the same proportions as the overall SIA Framework. This is not surprising given that nearly all end-use applications require memory for core functionality.

**Optoelectronics, Sensors and Actuators, Discrete Components (OSD)**

Optoelectronics includes lasers, displays, and other photonics-based electronics. Sensors include all kinds of specialized devices used to measure signals found in nature. Actuators include devices that initiate movement or take other actions in response to a stimulus detected by a sensor. Discrete components are individually packaged, specialized transistors or other basic components, like resistors, capacitors, and inductors.

*Optoelectronics*

SC devices that produce and receive light waves and are used for a variety of applications, including light detection and image sensors, LEDs, information processing, fiber-optic telecommunications, displays, and laser technologies.

**Photonic integrated circuits** (**PIC**) are commonly used as optical transceivers for data center optical networks, which enable data centers to transmit information more effectively and efficiently across greater distances than copper cabling. Photonics and Optoelectronic IC applications include data centers, high performance computing, telecommunications, consumer goods, sensors & bio-sensors, aerospace & high end, and quantum computing.

*Sensors and Actuators*

Sensors detect real-world inputs (heat, pressure, light, sound, other physical phenomenon) and convert them into electrical signals. Sensors are either passive (does not require power) or active (require power). Sensors are often used in control systems, like altimeters used to adjust airplane flight patterns or proximity sensors that trigger a car’s ABS. All sorts of SCs are used in sensors for a multitude of applications including optical, pressure, gas, speed, weight, etc.

Actuators are like reverse sensors that revert electrical signals back into real world outputs. Actuators are primarily used in industrial and manufacturing applications like robotics but are beginning to see applications in consumer and automotive markets as well. The modern revolution in industrial automation and autonomous driving are both made possible by the rapid proliferation of silicon-based sensors and actuators.

A smartphone is a great example of the diverse applications of sensors and actuators that include: accelerometer, gyroscope, electronic compass, pressure sensor, BAW filters, BAW duplexers, RF switch / variable capacitor, TCXO oscillators, MEMS micro-mirror, CMOS image sensor, auto-focus actuator, front camera, ALS & proximity sensor, microdisplay, and silicon microphone.

*MEMS*

Micro-electro-mechanical systems (MEMS) are tiny mechanical devices that operate gears or levers at a microscopic scale and are manufactured using SC fabrication techniques. MEMS are technically not SC devices since they don’t use electricity to process and store information, but are often grouped together because they compete with SC based sensors and are manufactured using similar technology. MEMS and Sensor applications include: positional and inertial sensing, optical transmission, biological and medical, power and energy technology, RF and wireless transmission, and pressure sensing.

*Discrete Components*

High volume, individually packaged components (resistors, capacitors, inductors, power transistors, switching transistors, diodes, and rectifiers) used as enabling devices for more complex systems. They generally help route signals and power to different processing centers in a given device.

*Discrete Components vs. Power Management ICs (PMIC)*

Power deliver used to be handled exclusively by discrete components that performed functions like voltage regulation, power conversion, battery management, and so on. Power management involves high voltages and signals moving around at high frequencies which can create massive issues with interference.

Therefore integration power management functionality onto an IC alongside critical sensors is a difficult problem that the growth in PMICs and PMUs has sought to address. Tighter integration of components has costs since densely packed components experience higher parasitic interference and power integrity issues. For example, when a “noisy” PM chip is placed next to a sensitive circuit like a microphone, it can cause performance issues and reduce audio quality.

Upfront design costs are higher for PMICs, but their overall performance advantages and efficiency improvements often make PMUs (type of microcontroller) a competitive long-term option. PMICs tackle the different voltages required by unique components in a system.

*Optoelectronics, Sensors and Actuators, and Discrete Components Market Summary*

OSAD and MEMS accounted for $79B out of $440B (18%) of total industry sales in 2020. Comprising 77% of the overall segment, communications, industrials, and automotive industries make up a much higher percentage of the end use market than the overall SIA Framework.

**Analog Components**

Analog ICs process analog signals and are either standard linear ICs (SLICs, generic plug-and-play analog devices that can be integrated into a larger system) or application-specific standard products (ASSPs, components designed for a specific application, but that can still be integrated into multiple systems within an application category).

While digital chips dominate the processing and information storage market, the real world remains analog. This requires analog chips to help make sense of it. These devices include sensors, wireless technology, and power supplies.

*General Purpose Analog ICs vs. ASSPs*

GPAs are used as broad plug-and-play analog components that may be optimized to perform a specific function, but can be used across many different systems much like micro-components on the digital side. GPAs include comparators, voltage regulators, data converters, amplifiers, and interface ICs. In complex systems, GPA ICs frequently sit between an analog sensor and a processor, amplifying and converting the analog sensor signal into a digital signal for use by the processor.

ASSPs and analog ICs designed for a specific application similar to the logic segment of the SIA Framework. Many ASSPs have digital components in them and are effectively mixed-signal devices. Examples of ASSPs include radio transceivers, audio amplifiers, as well as many varieties of RF (radio frequency) ICs.

*Analog Component Market Summary*

GPAs, ASSPs, and other Analog Components accounted for $56B out of $440B (13%) of 2020 industry sales. Analog components are most utilized in Automotive, Industrial, and Communications segments which each comprise 25% of their end use applications. Like OSAD and MEMS, analog components are least used for computing applications.

The market across these five systems components is led by memory -> micro components -> logic -> OSAD -> analog components.

*Signal Processing Systems – Putting Components Together*

It’s useful to think of electronic systems as signal processing devices. Let’s use the example of a music producer using a laptop to record and mix music using the five different components from the SIA Framework.

1. A microphone records an analog audio signal that is input and converted by ADCs to a digital representation in the laptop.
2. The ADC then feeds that signal to a DSP (micro component), which accepts the incoming digital stream and can apply some simple signal processing alogorithms.
3. After that, the newly converted digital signal is sent to the central processor, in this case a CPU, that runs the mixing software the producer uses for editing.
4. The central “system” processor (logic) may use volatile memory to store the collection of sound signals temporarily while it performs other tasks as directed by the mixing program.
5. Once the producer is done, she can tell the central processor to store the finished track for later using the system’s non-volatile memory, like NAND flash.
6. When the producer is ready to play the finished song, the digital signal is sent to another DSP followed by a DAC that converts the digital signal to an analog signal.
7. The analog signal is then sent to an analog processor or simply an amplifier that amplifies it out into the real world through the laptop’s speakers as music.
8. Through the system, various discrete components perform functions like system timing and power management that enable the device to run properly.

In Chapter 6, we first broke down the analog and digital electronics – capturing their differences in signal structure, data transmission methods, and power requirements. Though memory, micro components, and logic are responsible for the majority of industry revenues, all five are integral to the semiconductor ecosystem.

**Chapter 7 RF and Wireless Technologies**

**RF and Wireless**

To better understand wireless systems, it is helpful to consider two forms of energy – electrical energy and wave (wireless, airborne) energy. By manipulating these two forms of energy engineers can create, store, and communicate via the signals that are generated.

**RF Signals and the Electromagnetic Spectrum**

Radio Frequency (RF) signals are analog “wave” signals used to transmit information from one place to another without a physical cable or wired connection. RF signals can exist across a broad range of intensities and frequencies along the electromagnetic spectrum. The frequency of the signal is what electronics use to distinguish one signal from another. The end-use application (radio, television channels, telephone, ISPs, etc.) all use different frequency ranges, called frequency bands, to deliver different types of information. The Federal Communications Commission (FCC) strictly regulates frequency bands and the providers (AM, FM, CDMA, 802.11 etc.) that utilize them.

Providers only have a certain bandwidth to deliver an ever-expanding suite of services to customers. Thus, providers are incentivized to squeeze as much revenue by cramming as much information for as many people as possible from the allotted bandwidth. RF is used for radio and TV at the lower frequencies with microwave frequencies reserved for wifi, radar, and cell phones.

**RFIC – Transmitters and Receivers**

RF transmitters and receivers require at least six base components to operate: 1) source of power, 2) oscillator (sets the frequency of the transmission), 3) modulator / demodulator (a **modem** encodes / decodes digital information by making small adjustments or modulation to the carrier signal amplitude or frequency), 4) amplifier, 5) antenna, and 6) filter (low pass, high pass, band pass, and band reject). Some of these components are active while others are passive components.

**The OSI Reference Model**

System designers must ensure that their system (1) fits together as a properly function unit and (2) seamlessly communicates with other devices. Without some standardized model, every hardware company would build different systems running separate programming languages and have difficulty communicating with one another. With the Open System Interconnection, OSI, model however, you can just “follow the recipe” and build a system that both supports the needs of your software team while seamlessly connecting with other devices.

OSI describes the system layers that connect the underlying hardware to the user-facing interface that consumers interact with in order to produce high performing networks and integrated systems. The OSI system stack is comprised of seven layers with layers 1 – 3 dealing with physical transportation of information around the network and layers 4 – 7 addressing user applications. Layer 7 is the application layer that contains the user interface that a consumer interacts with like a web page or app home screen on your phone. As you descend through each subsequent layer, you get closer to the circuitry powering the application. The lowest OSI layer is the physical layer (PHY Layer), where the data itself is transmitted to the underlying hardware. What’s important is that the external facing endpoints of a system can integrate well with other devices.

We can create a universal Macro-System Stack by adding a “hardware layer” beneath the physical layer of the OSI model and it is this hardware layer that is the subject of everything that’s been studied so far. It’s the layers above it that make the hardware useful.

**RF and Wireless – The Big Picture**

How does a network of devices work together to bring you your favorite shows, talk to your friend in another city, or connect you to the Internet? To build intuition, let’s track the path of a typical long-distance phone call.

A phone call starts with transmitting a signal from the phone’s antenna that is picked up by a cell phone tower or base station (function as transceivers). The base station is a relay point that extends a service network to a specific area. service providers have spent billions building robust networks of base stations across the globe to make sure you don't lose your signal. each base station has a range of coverage called a coverage cell, and the patchwork of cells make up a service providers coverage area. Base stations come in different sizes and coverage areas: satellite (global), macro cell (suburban) , micro cell (urban), pico cell (in-building systems, femto cell or home base stations), and in-room (wifi and Bluetooth).

From the base station the phone signal is sent to a central exchange from where it can be routed to any number of locations (satellite, base station, etc.) based on the final destination of the signal. From the location the signal is routed to an information exchange center situation close to the final destination. This proximal exchange center will they route the signal to a base station, antenna relay (which then sends to a mobile device), or land line which finally connects the call to the intended receiver.

**Broadcasting and Frequency Regulation**

With so many devices and users of limited bandwidth available, a dizzying amount of RF signals are flying around at any given moment. These frequency bands are a limited and valuable resource, and it is in the service providers’ best interest to do everything they can to maximize the information they send using the fixed bandwidth they have allotted to them. A lot of complex technology goes into solving this problem.

*Digital Signal Processing*

DSP technologies use sophisticated mathematics and computation methodology (lossless or “lossy” signal compression) to fit more information into a given digital signal. By packing more information into the same signal, we can send a lot more information using the same, limited amount of frequency bandwidth.

*TDMA and CDMA*

**Multiple access standard technology** allows service providers to route multiple calls through the same base station, or across a given amount of bandwidth. **Time Division Multiple Access** (**TDMA**) separates or multiplexes calls into a time domain with the same frequency channel. TDMA technology relies on breaking up data into chunks and jamming these chunks into a given bandwidth (e.g. voice is sample at 4KHz so in a 4MHz bandwidth you can jam 1,000 calls). On the receiving end, TDMA reassembles the broken-up chunks from the same call back into a continuous voice call stream.

**Code Division Multiple Access** (**CDMA**) uses code to digitize voice bits or other data bits and transmit the encoded data across a wider bandwidth (wider frequency range). The receiver of CDMA encoded signals decodes the signal to reconstruct the original signal. CDMA can send data from numerous senders to numerous receivers simultaneously and use algorithms and DSPs to ensure that the data gets to the right place intact. Lightening fast DSP allows both TDMA and CDMA to establish uninterrupted signals and strong uninterrupted service. TDMA is a technology that underlies **GSM** (**Global System for Mobile Communications**) and is the primary standard for communication networks globally. In the US, AT&T uses GSM while Verizon uses CDMA.

**1G to 5G(eneration) – An Evolution**

Telecommunications and wireless technologies is always evolving – the original 1G could transmit 4kbps whereas today’s 5G can transmit up to 1,000,000kbps (1Gbps) – a 250K fold increase!

1G cell phone technology first came into existence in the late 1970s. The first cell phones were big and clunky with terrible battery life. They used analog technology to send RF analog “wave” signals between two points wirelessly. (1979)

2G cells phones began using modulation to transmit digital data wirelessly. CDMA and GSM technologies were developed allowing service providers to connect more devices more affordably, though services were limited to voice and SMS text. (1991)

3G building on voice and text, 3G expanded wireless capabilities to email, video streaming, web browsing, and other technology that make the “smartphone” possible. (1998)

4G high-speed connectivity built on advancements in hardware technologies has allowed for faster data transmission, mobile gaming, video conferencing, high-definition content delivery, and cloud computing capabilities. (2008)

LTE long-term evolution is an industry standard used to make sure the various devices, access points, base stations, satellites, and other components that make up our telecommunications network are able to work with each other to create one big, fully functioning system. Standards like LTE help ensure that different technology companies can develop products that are able to work with other parts of the system, which is important if you don’t want your service to cut out every time your cell phone is routed to a cell tower operated by a different network provider.

5G is still under development and is designed to make 4G much faster and more efficient. To deliver higher “data throughput” connection speeds, 5G technology will require a robust network of thousands of cell towers and tens of thousands of small cell antenna cells deployed across coverage areas. (2019)

**Wireless Communication and Cloud Computing**

While the evolution from 1G to 5G has delivered exponential improvements in data transmission and brand-new technologies like mobile gaming, video conferencing, and high-definition video streaming, these faster rates have driven an insatiable demand for **cloud computing**. The cloud is really just countless servers housed in giant rooms called **data centers**. Data centers drastically vary in size: Microsoft has a shipping contained sized data center that it uses to process and redistribute Bing Map data, Google has a 200,000 square foot data center in Council Bluffs, Iowa, while China Telecom currently boasts the largest data center in the world in Hong Kong at over 10 million square feet.

By storing, or hosting, applications on higher performance computers, companies and consumers can store information, run applications, and boost capacity without having to invest in and manage all their own infrastructure. This is only possible because communication networks are so fast today.

Prior to the boom in wireless innovation over the last couple of decades, the bottleneck to centralized computing operations like data centers was moving data to and from end users. With this bottleneck alleviated, cloud computing is here to stay. The problem has now has shifted from limited bandwidth to building and powering data center infrastructure that can support ballooning demand.

**Chapter 8 System Architecture and Integration**

**Macroarchitecture vs. Microarchitecture**

Microprocessors, the “brains” of computing systems, serve as the computational workhorses of the SC ecosystem and contain the arithmetic, logic, and control circuitry necessary to execute instructions, process data, and run sophisticated software programs. The most advanced PC microprocessors today contain over 100B transistors and this leads to immense complexity and design challenges. To overcome such challenges, design leaders must pay close attention to both the micro- and macro-architectural decisions, carefully balancing the trade-offs between flexibility and performance with the cost and complexity of tighter system integration.

The term “architecture” can be used in SC engineering to mean one of two things – system architecture or microarchitecture. System Level, or macro-level silicon architecture that are used to define entire chip families, are technically described by different kinds of Instruction Set Architectures (ISAs) which describe the way in which instructions are delivered from the programmer to the computer. Microarchitecture describes the way an ISA is actually implemented into the hardware design itself. A CPU, GPU, and PMU may all be designed using a single ISA that governs the design of the whole SoC, but still have unique microarchitecture.

**Common Chip Architectures**

***Von Neumann Architecture*** is a macroachitecture developed by **John Von Neumann** in the 1940s and is what most modern computers are based on today. It relies on the CPU, I/O interfaces, and memory. Inside the CPU there exists (1) **registers**, where data and instructions are delivered by and given to the memory, (2) a **control unit** that determines which instructions should be executed, and (3) the **arithmetic and logic unit**, where instructions are carried out and information is actually process.

**Harvard Architecture** is similar to Von Neumann however differs in how it accesses (input) and distributes (output) information. The Harvard Architecture parses instructions and data into two separate memory banks, with a unique bus for each type of input. This separation theoretically allows parallel access of data and instruction sets thus reducing the clock cycles necessary to perform a single instruction. This **pipelining** is difficult to implement and increases complexity and cost over the Von Neumann architecture. Harvard is used primarily in microcontrollers and specialized DSPs while Von Neumann is used in computers, mobile SoCs, and complex digital electronics. Both microarchitectures are more theoretical than practical. The vast majority of IC’s use a Von Neumann “type” macroachitecture – system designers must weigh the pros and cons to decide which kind of Von Neumann macroarchitecture is best for the device they are building.

**Instruction Set Architecture (ISA) vs. Microarchitecture**

ISA determine the set of instructions (lines of code written by a programmer that a computer follows) that a given processor can support. Microarchitecture, on the other hand, determines how the processor receives and executes those instructions at an implementation level. The ISA and microarchitecture is dually tied to the Universal Architecture Stack that consists of seven layers. Each level is dependent on the collection of levels below it and must be designed to support the levels above it. Whatever you’re inputting into the system via the top-level user interface layer, ultimately your input must be carried out by the transistors and functional components that comprise an integrated circuit. The translation process that makes this possible is difficult and requires engineering skill spanning numerous areas and disciplines. Bridging this gap is why abstraction layers are so vital to building high performance systems.

**Instruction Pipelining and Processor Performance**

It’s important to review the key factors that determine processor performance because this is helpful for understanding the trade-offs between CISC versus RISC. From an architectural perspective there are three things we do to increase the speed of a processor:

1. **Increase processor’s oscillator clock** frequency – between reducing the clock cycle and shortening the distance signals have to travel through our critical we can significantly boost performance.
2. **Synchronize task performance** – this is done by adding more processor cores (co-processors) – GPUS are particularly well suited for this kind of parallel processing. This strategy has an upper bound and depends on the number of registers (electronic counter space) available to accept inputs and deliver outputs to the cache memory – this is a primary bottleneck for processor throughput.
3. **Pipeline** is a method of speeding up tasks by breaking up a lengthy workflow into smaller constituent parallel tasks. Unlike synchronous processing that relies on tasks being performed in parallel along separate paths, pipelining breaks up a process into smaller constituent task performed in parallel along one path. Pipelining can significantly improve our processing throughput (RISC is better suited for pipelining versus CISC) enabling us to finish more instructions per cycle than before. This can prove to be a substantial advantage giving one ISA an edge over another depending on the application. Pipelining is a complicated scheduling operation that must ensure both the data and the instruction necessary to execute a given task or operation arrive at the CPU at the proper time and thus RISC processors aim to maintain an instruction-to-clock cycle ratio of 1-to-1.

**CISC vs. RISC**

The two main types of ISAs are **CISC** (**complex instruction set computing**) and **RISC** (**reduced instruction set computing**). There are numerous differences between RISC and CISC though the most notable is the way in which they process instructions. For many applications, RISC is often seen as a more effective option. Unlike CISC processors which use lengthy, multi-clock cycle instruction sets, RISC processors break up instructions into smaller, standardized instruction sets that are better optimized for instruction pipeline and more easily digested by **compilers**.

The reason for these difference is ISAs is rooted in SC history with CISC originating around 1970 with RISC following about a decade later. This early on in the industry’s development, compilers were unreliable and programmers often wrote directly in assembly code. Assembly languages are much closer to the hardware but they have many disadvantages compared to compiled languages including increased complexity, difficulty of use, and reduce portability. As compilers improved, RISC architecture was more widely adopted by design teams across the industry. In summary CISC is hardware centric (less dependent on compilers) whereas RISC is software centric (more dependent on compilers).

Because they process simplified instructions, RISC tend to consume less power, making them idea for applications where power is particularly limited, such as mobile phones or other battery-powered devices. However, this advantage is reduced in higher-performance applications, like servers or personal laptops, which tend to use CISC more often.

RISC processors do require more RAM (to access additional code) and greater programming efficiency (shorter instructions means more lines of code). Thus in the early days when memory was a lot more expensive, CISC dominated but RISC gained significant traction as memory chips became smaller and less expensive.

**Choosing an ISA**

Choosing an ISA is a difficult and consequential decision. MIPS and ARM are licensable ISAs which others like RISC-V are proprietary or open source. A licensed ISA will typically come with a pre-designed processing core while an open-sourced ISA is not. Licensing fees and royalties are key factors in deciding whether to license, build, or borrow.

Oftentimes more important than direct costs are the risks that come with each ISA. Engineers must consider the time and costs of physically developing a core processor in-house. Most ISA licensing companies license processors that serve as the core of any customized end-product. Even if a unique core processor design is completed within a feasible time frame, there are always manufacturing risks that can cause new processor and architecture to fail.

Even more important are the downstream software implications. ISAs like ARM and x86 have mature software “ecosystems” with fully developed software development stacks. Building a new processor with a proprietary architecture requires the development of new firmware, operating systems, and development tools. Even if hardware and software execution is seamless, time-to-market is likely to be an issue – while you’re spending time developing a core processor and proprietary architecture, the competition is releasing new products.

One question to ask yourself is – are customers buying the processor, or are they buying what it does? If your customers are buying your new algorithm or integrated processor and sensor system, then licensing an existing ISA may be more attractive. But if the processor itself forms the core of your business or adds some unique value not available in the market, then a custom design may be the right answer.

There are numerous considerations for choosing between proprietary, licensed, and open-source ISAs and these are: 1) hardware design costs, 2) software engineering costs, 3) existing software ecosystem, 4) time-to-market, 5) manufacturing risk, 6) design flexibility, and 7) royalties and licensing fees.

**Heterogeneous vs. Monolithic Integration – From PCBs to SoCs**

Historically, shrinking transistor sizes due to advances in manufacturing and lithographic technologies has enabled SC design companies to continue increasing performance of their devices without having to pay much attention to device architecture and integration (geometric scaling versus functional scaling). There was enough “wiggle room” so that even if a design included millions of unnecessary transistors, it could still deliver greater performance using less power and space at a lower or equal cost.’

Even if a company was motivated to make its chip more efficient, the additional design effort required to make fully integrated devices was costly and the rapid pace of Moore’s Law made time-to-market a critical constraint. This was enabled by the fact that the cycle time for manufacturing advances (getting to the next node) was shorter than the design effort time for architecture redesign at the previous node.

Geometric scaling continued unabated up until the last decade or so when the industry ran into three main problems:

1. As transistors shrank and logic became denser, power management issues became a primary design constraint and this displaced frequency as the dominant factor. Modern CPUs and other advanced devices often cannot use their full firepower since the heat required to run at such speeds would literally fry the IC.
2. Lithographic technologies, like EUV, depend on the wavelength of light being used and as the wavelength required gets smaller, it has become that much more difficult to push the technological envelope.
3. The thickness of the materials that transistors are made from is now just a few atoms thick with little room for further shrinking. There won’t be enough atoms to make usable feature patterns below a certain threshold.

Each successive node requires disproportionately more expensive process technology and this increases the costs to build new fabs and drives up manufacturing unit costs. This is shifting engineers’ mindsets towards squeezing out more performance from older nodes, that is, towards functional scaling. In essence, Moore’s Law is slowing down and this is pushing engineers to optimize designs for specific applications and shifting system architectures to include more heterogeneous and monolithic integration.

In **heterogeneous integration**, numerous chips are integrated with one another on the same PCB or within the same package, called system-in-package or SiP. In monolithic or homogeneous integration, numerous functional modules are included on a single IC, yielding a fully functional system called an SoC (system-on-chip). The more integrated a system is, the less distance a signal must travel to reach other parts of the chip. However, fully integrated systems like SoCs are complex to design and have numerous drawbacks that must be considered.

Heterogeneous (SiP) versus Homogeneous (SoC) have significant trade-offs. SoCs have small form factors and thus impart area and power efficiency advantages making them popular in smaller, battery powered devices like cell phones. They can suffer on performance, however, based on the application. Each functional portion of a chip may require different materials and process technologies for peak performance, which is difficult or impossible to do on a single wafer. By integrating all components on a single wafer, some parts may function very well while other parts perform poorly given suboptimal materials and processes for those poorly performing parts. Greater levels of integration have higher design costs and greater manufacturing complexity.

Cost differences play a big role in whether to use monolithic or heterogeneous integration. Greater levels of integration require more design work and thus higher design costs and greater manufacturing complexity. However, this may not lead to lower unity costs for heterogeneous devices in all cases. Monolithic devices require less area in aggregate, which enables more chips to be printed on a single wafer and can lower net manufacturing costs.

At the same time, heterogeneous integration enables a sort of “manufacturing arbitrage”, where different process nodes may be used for different parts of the system. In a SiP, for example, advanced modules like memory or core logic could be manufactured using advanced nodes while parts like analog or RF components could be manufactured using older nodes. This blended node technique could lead to overall lower manufacturing costs. Additionally, companies can build a next-gen system by only changing the memory and logic devices to enable more features, while keeping power management or RF components unchanged.

Manufacturing executives must take great care to balance the additional design costs and manufacturing cost differences when deciding which architecture to employ. Monolithically integrated SoCs consume less power and take up less space than heterogeneously integrated systems integrated at the board level. System-on-Boards (SoBs), however, have greater design flexibility, lower design costs, and can be designed more quickly. SiPs straddle between SoCs and SoBs offering greater design flexibility while reaping higher performance, power, and area advantages due to greater integration.

In addition to the core PPAC factors, system architects and design teams must keep time-to-market in mind. As a rule of thumb, the more integrated a system is, the more design time is needed. If a competitor is pressuring your team with a new product release, perhaps it’s best to shy away from desiging a new SoC from scratch.

**Chapter 9 The Semiconductor Industry – Past, Present, and Future**

Since its beginnings in the 1960s, the SCI has faced two ongoing major challenges – 1) rising design costs and 2) rising manufacturing costs – that have driven the industry from yesterday’s fully integrated companies to today’s multi-faceted fabless design models.

**Design Costs**

IC design pipeline: architecture and IP qualification -> design verification -> physical design -> software licensing (EDA) -> prototyping -> validation -> fabrication / manufacturing. This is an extensive and expensive pipeline to manage fraught with implementation and strategic risks at every stage.

In the early days, there was no standardization and scalable tools and most companies made ICs in-house. This drove up design costs and at the end of the day one company’s IP wasn’t guaranteed to work with another company’s IP. Electronic data automation (EDA) companies changed this paradigm. It substantially reduced the design cost footprint by modularizing significant chunks of IC design. This benefit was furthered by the promulgation and adoption of hardware description languages (HDL) like VHDL and Verilog. This enabled design firms to build larger systems at aggregated, more manageable levels of abstraction (e.g. C++ instead of assembly language). Separating design from EDA allowed companies to refocus on the core competencies of software and hardware – in the early days this benefited advanced analog chip companies that could now focus on chip design using licensed EDA tools. Today Cadence, Synopsis ($50B market cap), and Mentor Graphics are the three big players in EDA with EDA accounting for 50% of the total design costs.

Throughout the 1980s, most IC were medium size and complexity with systems companies purchasing different chip types and integrating them into their device by soldering them to a PCB or other connecting device. As SC technology grew more advanced, systems companies strived for integration and thus the demand for SoCs skyrocketed and this in turn increased the complexity of each system component, making design even more difficult and expensive. Out of this cycle sprang **SC IP companies** (business model involved upfront licensing fees, per-product royalties, library access subscriptions) that reduced this complexity by 1) offering base designs of common modules that could be used as a foundation for an application specific circuit and 2) offering cell libraries that could be used to generate even more complex designs. Companies could quickly acquire the undifferentiated parts of a new design and thus allow them to focus on the parts that made their chip unique.

There are three categories of SC IP companies: 1) microprocessors like those offered by ARM, 2) communication architecture that enable different parts of an SoC to talk to one another (Arteris), and 3) Analog IP which has become more difficult to design at each successive node. SCIPCs play an important role in keeping design costs under control. If you’re an analog company, SCIPCs can provide all the digital functions you need like microprocessors and memory and if you’re a digital company, SCIPCs can provide the analog components like oscillators and power reference circuits.

As lithography and manufacturing equipment suppliers push to the next node, SC R&D and design have become progressively more difficult and further complicated by phenomena like quantum tunneling and current leakage. Design costs alone (doesn’t include manufacturing) for 3nm SoCs ranges from $0.5B to $1.5B (International Business Strategies) while 7nm and 10nm can be upto $0.3B and $0.175B respectively.

**Manufacturing Costs**

Though design costs have posed significant challenges to profitability, further down the value chain, dramatic increases in manufacturing costs have had even greater repercussions. Each successive node requires increasingly complex and expensive manufacturing equipment like dicing machines, polish grinders, masks, and steppers all enclosed in specialized air filtration and vibration control environments. Advanced SC fabs with a usable life of five years can range from $7B to $20B and this does not include the variable costs of chemicals and materials.

Front-end manufacturing technology like lithography equipment has historically account for most of the manufacturing costs, though back-end has received greater attention as the industry shifts focus to advanced packing architectures and heterogeneous integration to boost performance. Figure 9.1 (page 179) is an excellent analysis of both design and manufacturing costs (IBS).

Together, supply-side design and manufacturing costs and demand-side pressures for greater variety of customized designs have transformed the industry from the **fully integrated SC companies** of the 1960s through 1980s to the mix of **fabless design companies**, **specialized IP and EDA tool companies**, **integrated device manufacturers (IDMs)**, **pure-play foundries**, **system design companies**, we see today. This evolution embodies David Ricardo’s theory of comparative advantage – the market functions optimally when everyone focuses on what they are best at providing.

**Evolution of the Semiconductor Industry**

*1960s – 1980s Fully Integrated Semiconductor Companies*

These companies would forecast demand for a product, then design, manufacture, and package it before marketing it to potential customers. Each company invested significant resources into its own fabs. High fixed costs left firms vulnerable to volatile demand swings – if sales for their products dipped even slightly, fab utilization could drop significantly, spreading revenue across a greater cost basis and cutting into profit margins.

*1980s – 2000 IDM, Fabless Design, Pure-Play Foundry*

The rising costs of building and owning a fab and retool with each progressive node drove fragmentation of the SCVC in the 1980s and 1990s. Once built, companies were driven to maximize their output and contribution market, even if they weren’t covering their fixed costs. As a result, IDMs like Intel, which design, manufacture, and sell their own ICs have started to lease out parts of their fab capacity while former IDMs like TI and AMD have become fabless. Processor companies would sell their old fabs down-market to analog companies thus extending the effective useful like of these older node fabs.

In the mid-1980s, companies like Xilinx (1984) and Qualcomm (1985) began with a business model designed to take advantage of this excess capacity. These companies designed their own chips and signed contracts with an IDM to utilize additional manufacturing capacity thus foregoing upfront capital investment and in effective converting fixed costs into variable costs that would then be built into the price charged for each wafer. Not only did this fabless model reduce required capital for new entrants, it also better matched variable supply economics to variable demand economics.

In 1987, TSMC, a pure-play foundry entered the market with an entirely new business model, focusing completely on manufacturing other companies’ design. Foundries are able to focus on a smaller set of core competencies and also take advantage of more stable demand and consistent volumes since they draw orders from multiple customers instead of just their own designs. Additionally, foundries strategically position themselves in markets with lower labor costs like Taiwan and other parts of Southeast Asia.

US dominance in fabless design and Southeast Asian dominance in manufacturing and assembly. Xilinx and other fabless design companies quickly shifted away from IDMs to lower-cost domestic and overseas foundries, a model that has dominated the market ever since. Even the few remaining IDMs like Samsung and Intel still depend on pure-play foundries for more advanced technology nodes, typically using their own manufacturing capacity to produce devices requiring less demanding, older process nodes.

In the late 80s and early 90s an important shift was happening further up the SCVC. EDA tools made building front-end design easier and systems companies that had historically depended on out-of-the-box products or custom silicon from fabless design companies and IDMs began building their own chips better suited to their specific needs in essence bypassing chip design companies and working directly with foundries.

However, there was still a gap in back-end expertise necessary to carry out physical design and downstream SCVC activities. New players like VLSI Technologies and LSI Logic as well as established players like Qualcomm met this demand with innovative ASIC-centric business models. Pure ASIC companies like LSI handled front-end designs develop by systems companies and then coordinated by manufacturing and assembly suppliers to bring their product to market.

Design services companies like Xilinx and Qualcomm drew much of their revenue from a similar model, though they also began to focus on specific markets, developing unique expertise and growing product portfolios of their own. Xilinx (acquired by AMD) controls over 50% of the FPGA market while Qualcomm is a world leader in custom silicon for wireless connectivity and infrastructure products.

*2000 – Present Fabless Design Companies, Foundries, IDM Stragglers, System Company In-House Design*

Today the SCI is driven by fabless design companies and pure-play foundries and a few remaining IDMs supported by an ecosystem of EDA tool developers, equipment manufacturers, and IP providers. Systems companies are playing a growing role as they continue to eat into what used to be traditional fabless semiconductor business. Examples of systems companies designing their own chips are numerous and growing – Apple (iPhone, Mac), Facebook (Oculus), Tesla (driver assist and autonomous driving platforms). Figure 9.3 (page 183) is an excellent chart of the evolution of the SCI from the 1970s to 2010s. The motivation of systems companies to develop their own silicon may primarily be driven by cost but also to protect their IP.

**Fabs vs. Fabless Design – The Case Against IDMs**

Owning a fab can provide numerous advantages, including greater process control, faster time-to-market, and tighter design integration. But these advantages have increasingly been outweighed by costs.

One of pure-play foundries’ biggest advantages over IDMs are their ability to pool demand across multiple customer and fully utilize capacity thus smoothing out revenues of time and preventing underutilization of fixed assets. Underutilization is a killer in SC, because it scales directly with fabrication costs and deteriorates unit economics. Operating an IDM today is truly a feast-or-famine endeavor. If you hit all your schedule targets and release a manufacturing process that’s superior to your competitors’ you have a huge advantage. If you miss those targets, you’ve sunk billions into failed process development, and may be forced to manufacture your products in the same foundries that your competitor is using and your internal development may never catch up once you fall behind.

Proprietary manufacturing IP and data capture is another common reason IDMs may use to justify the costs of fab ownership. Though this may have proved an advantage historically, foundries have unique technological advantages that are hard to match as an IDM. Because they manufacture a wider range of components and ICs, foundries can iterate process technology at a quicker rate, building important competencies that they can then democratize and provide to fabless design houses and other key customers.

Tighter integration between design and manufacturing is a substantial advantage that can result in significant cost savings and performance improvements. Though fab-owning IDMs may have a material edge here, software tools, design suites, and remote-work networking technologies have made integration between foundries and their customers highly efficient. Fabless design companies like Qualcomm and Broadcom can seamlessly design new chips with fabrication costs and **production optimization** squarely in mind. This isn’t only limited to big companies – small startups can even access leading integration and manufacturing technologies using **shuttle runs** where a foundry combines the designs of **multiple customers onto a single mask set** and manufacture just of few wafters with several hundred chips that the start-up can use to demonstrate a minimally viable product in order to raise later-round capital. By combining shuttle runs of multiple customers together, several firms can share and split the total cost (e.g. a round of wafters that may have $5M for a single customer can be cut to $500K if ten customers are combined onto a single mask set.

Looking at the pros and cons of Fabless versus IDM models can be assessed on various criteria: 1) manufacturing, 2) time-to-market, 3) organizational complexity, 4) overhead and variable costs, and 5) technology and integration.

When you analyze revenue to capex and PP&E, we find that revenue to PPE ratios heavily favor fabless design companies. For all the reasons we discussed earlier, design companies are able to produce more income per dollar of capital than IDMs, which may explain why a fabless design company like Nvidia has a market value of $60B greater than Intel, despite having only a fraction of its sales. See Figure 9.5 (page 187) for excellent cross-sectional data on various business models and their associated sales and ROA metrics. For these reasons, the two major IDMs, Intel and Samsung have moved to a **Fab-Lite** model.

An X-factor to keep in mind here is the recent global SC shortage and competition between the U.S. and China. The few remaining U.S. IDMs like Intel and GlobalFoundries are poised to benefit substantially from the $50B in government incentives recently passed by Congress to strengthen the industry and shore up domestic supply chains. It is yet to be seen, however, if this renewed support will counteract the fundamental market disadvantages IDMs suffer from today.

**Industry Outlook**

Shaped by a multitude of overlapping variables and market forces, the SCI is constantly changing and so we’ll highlight five key trends that shaped the industry’s history, influence its present health, and drive future growth prospects.

*Cyclical Revenues and High Volatility*

SC sales are largely driven by the electronics industry and a highly cyclical and volatile characterized by boom and bust cycles that can last many years. Nominal sales growth / contraction can stem from pricing dynamics – memory SC, like commodities, are highly price sensitive and are considerably more volatile than non-memory SC.

To manage year-to-year silicon cycles, SC companies must be able to control costs, while not sacrificing key investments in R&D. Manufacturing represents the greatest expenditures by US-based SC companies, comprising on average nearly a third of total costs, followed by R&D, D&A, and SG&A. For IDMs, it can be a real challenge to weather an industry downturn if you have massive fixed costs from a manufacturing facility on your balance sheet. Production expenditures have grown significantly as a percentage of total costs over the past two decades.

*High R&D and Capital Investment*

The rapid pace of technological advancement has driver SC companies to invest markedly high amounts of capital into core R&D – since 1999, U.S. SC companies have invested 15 – 20% of sales into R&D (highest after pharma and biotech) – this is necessary if firms want to maintain their competitive edge. Also, in the U.S. SC companies plow 8 – 20% of sales into new PP&E, second only to Alternative Energy (SIA Databook 2021). SC R&D and capex have grown 5.6% year-over-year, which US companies spending $75B alone in 2020 (SIA Factbook, 2021).

*High Compensation and Positive Productivity Growth*

Over the last several decades, a range of factors has led to wages and productivity in the SCI outpacing other sectors – 2001 $80K annual compensation has grown to $160K by 2019 (SIA Databook, 2020). In addition, sales per employee, a strong indicator of productivity, has doubled over the past twenty years - $571K in 2020. Unlike most industries where average selling prices increase at or above the rate of inflation, per unit costs have decreased at a rate fast enough for companies to maintain profitability without excessive price increases. In lieu of price drops in vital input materials or manufacturing costs, consistent profitability would only be possible by increasing production efficiency.

*Long-Term Profitability*

Earnings have concentrated among the biggest competitors with the scale to weather downturns and who have capitalized on disruptive technologies like PCs and smartphones, collecting pre-tax profits averaging about 20% of sales since 1999 with gross margins ranging from 37 to 57%. Major growth drivers include:

1. **Increased demand** for consumer electronics due to rising household disposable incomes, increasing urbanization, and rapid population growth (Fortune Business Insights, 2021).
2. **Fast-expanding emerging economies** with growing demand for ICs.
3. **Technological drivers** including IoT, smartphones, 5G communications, and AI/ML.

Reduced demand in automotive, industrial, and parts of the consumer market were in turn offset by areas like servers, PCs, and long-term growth areas like AI and 5G. This is a means for attempting to preserve growth by hedging demand. How consistently can demand hedging work in different silicon cycles?

Nvidia, the leader in processors for graphics, AI, and crypto, has set new records for quarterly revenue for every quarter of 2021, peaking at $6.5B quarterly revenue in August 2021. Though new technologies and the explosion of PCs, servers, and cell phones over the last couple of decades have been a boon for the industry, high consolidation has adversely affected many companies.

*High Consolidation*

As SoC designs become increasingly complex and transistors are pushed to their physical limits, design and manufacturing costs have never been higher and thus the industry’s profitability has depended on R&D breakthroughs targets at cutting costs. This has been realized so for with $0.98 per chip costs in 2001 dropping to $0.63 per chip in 2019.

In addition to pressure to decrease per-die unit costs, there has been enormous pressure on companies to get more out of each device – a boon for consumers who have gained access to greater computing power at lower costs. These counter-opposing forces have driven immense competition between rivals and resulted in the consolidated behemoths we see today. This pattern is natural in capital intensive industries that inherently favor size since ballooning fixed costs can be spread or amortized across high annual revenues. COGS as a percentage of revenues is significantly higher for smaller firms who lack the overhead and economies of scale to compete with bigger players. The annual deal volume for consolidation has averaged $69B per year since 2015.

While there were 29 companies offering advanced fab services in 2001, today there are only five, including just two main foundries, a handful of EDA companies, and a single lithographic equipment supplier (ASML). There has been significant consolidation momentum since 2015. In such a capital-intensive industry and with the top 10 SC companies owning 55% of the market, smaller companies struggling to survive have had to grow to stay competitive. (IC Insights, SIA Databook, SIA Factbook, Design and Reuse, McClean Report) For a comprehensive list of sources, see a fully annotated version of Figure 9.12 (page 195) in Appendix B.

Major SCI Dynamics Wheel

**Market Volatility** – cyclical revenues, boom & bust “silicon cycles”, heavily influenced by memory prices.

**High Capital Intensity** – high R&D costs, hight PP&E costs, 2nd highest industry R&D plus capital costs.

**Compensation and Productivity Growth** – cost decreases due to productivity growth, 2x revenue per EE growth over last 20 years, 3x average compensation of all manufacturing jobs.

**Long-Term Profitability** –new technologies like IoT and AI, expanding emerging economies, increased demand for electronics.

**Industry Consolidation** – rapid M&A activity, small number of big winners, high capital intensity.

**US vs. International Semiconductor Market**

Since 1999, the U.S. has maintained approximately 50% of the SC market share and currently controls about 47% of sales ($208B out of $440B in 2020). The U.S. continues to house most design companies and IDMs and while some design companies are growing in Europe and Asia, these companies are still predominantly US-based design efforts.

Manufacturing and assembly firms are located predominantly in Taiwan and Asia Pacific, although manufacturing may be moving more to India and South Asia. Long-term market dominance may be related to the high rates of R&D investments by US companies, which spend more on R&D as a percentage of sales than any other country. Figure 9.15 (page 199) shows detailed analysis of SCVC activities and consumption broken down by region (BSG and SIA’s report on Strengthening the Global Semiconductor Supply Chain). The US, for example, leads in areas that require intensive R&D like Logic, EDA, and core IP, while Asia focuses more on areas that are labor-intensive and require significant capex like materials, wafer fabrication, assembly, packaging, and testing.

It is important not to analyze the SCVC in a monolithic manner. The go from demand assessment all the way to product delivery requires massive amounts of specialization, timing, and coordination between different SCVC participants across the entire globe. Create a flowchart of the high-level cascade of coordinated activities discussed on page 201. You should note that real SCVCs are much more dense, with complex electronic devices requiring thousands of components, tools, equipment, and effort provided by hundreds of suppliers and sub-suppliers.

The US controls about 60% of logic and analog markets, though it trains in memory and discrete components with about 20-25% market share in each. Contrary to popular belief, the US still manufactures a considerable portion of the world’s SC. Though still a manufacturing leaders, its modern manufacturing capacity has grown at a rate less than five times that of firms overseas, although recent supply chain concerns may slow this decline.

From a demand perspective, Asia Pacific is by far the largest consumer of SC, comprising over 60% of worldwide demand with $271B out of the $440B in 2020 sales – see figure 9.17 (page 202). Though China has been the largest consumer of ICs since 2005, chips that are designed and produced within the country account for only 15% of total purchases. To avoid any possible consequences of future trade tensions or over-reliance on the US, the Chinese government has devoted considerable attention and resources to grow the country’s domestic SCI. In 2014, China pumped $20B into government-backed private equity funds, like Tsinghua Unigroup (SC technology development) and in 2019 $29B fund with the aim of reducing China’s dependence on foreign suppliers and developing IC design and manufacturing technology.

**COVID-19 and the Global Semiconductor Supply Chain**

Covid set in motion a global chip shortage that exposed many of the vulnerabilities of the SCVC. The shortage has had significant economic consequences, including production cutbacks in autos, consumer electronics, medical devices, and networking equipment extending lead times of many SC beyond one year. Most of these issues were a consequence of insufficient capacity but rather distortions on the demand-side (e.g. slumping auto sales) driving chip order from one end-use market to a different end-use market. It’s important to note that SC manufacturing utilization remained optimized throughout the pandemic. By the time auto demand rebounded, manufacturing lines had already retooled to produce other consumer products, leaving autos high and dry with no chips to power their vehicles.

Though pandemic, inaccurate demand forecasting, and titled capacity allocation, with added idiosyncratic risks like fires at two key Japanese factories, may seem like one-time, unavoidable hiccups, the reason they were able to cause so much disruption is structural in nature. The core weaknesses of the SCVC today are 1) regional stratification of key activities and 2) resulting mutual interdependence. What has more impact – design or manufacturing – in an imbalanced SCI?

US leadership in R&D and design, for example, can largely be attributed to its existing talent pool and access to steady stream of new engineers from US universities and talent pool dynamics is likely to help the US retain a significant edge in engineering talent for the foreseeable future. The US also benefits from a vast pool of VC funding, which has the capacity and will to make ambitious bets in the SCI.

On the manufacturing side, East Asia holds competitive advantages, including skilled and affordable manufacturing talent, robust infrastructure, and higher levels of government incentives. Though talent and infrastructure are both important to manufacturing, the role of government incentives should not be underestimated – government incentives may account for up to 30 – 40% of the 10-year total cost of ownership of a new start-of-the-art fab. Comparatively, the cost of ownership of the same fab in the US is between 20 to 50% greater than in Asia and between 40 – 70% of that difference is due to the lower incentives offered by the US government as compared to Asian competitors. While one might assume that lower construction and labor costs might be the main reason that Asia has an advantage in manufacturing within the SCVC, the data confirms that it is in fact government incentives that make up the difference, explaining between 40 and 70% of the 25 to 50% TCO advantage.

While free trade and specialization have enabled the global SC ecosystem to thrive, delivering performance at lower cost to global customers over the last 50 years, the price paid has been the evolution of a fragile unstable supply chain. Today, there exists greater than 50 points on the SC supply chain where a single region controls more than 65% of the global market share (SIA whitepaper, 2021). This concentration exacerbates three key risk factors – random natural variability (cannot be avoided), geographically clustered manufacturing capacity, and geopolitical conflict.

In order to strengthen the supply chain and make it more resilient, experts do not believe every country needs to become completely self-sufficient – this path would be prohibitively expensive and significantly inflate prices. However, targeted investment in US SC manufacturing capacity and a greater balance between efficiency and redundancy would go a long way in protecting against future chip shortages and economic downturns, while reducing our over-reliance on other countries for components vital to national security.

**Chinese Competition**

Manufacturing capacity concentration in East Asia and China puts the US economy and national security at risk. As such, shoring up the SC supply chain has become a recent political priority in the US. Of the $250B Science and Technology bill recently passed by Congress, $52B was earmarked for SC manufacturing – this bill drew bipartisan support as a way to counter China’s growing economic and military power. This is an attempt to buck the trend of the past two decades that have seen rising manufacturing costs compelling US players to sell, spin off, or abandon their foundries and given market share to TSMC. Today there remains only five major companies that manufacture SC in the US – Intel, Samsung, Micron, Texas Instruments, and Global Foundries (spun off from AMD in 2009). There are smaller analog companies that maintain manufacturing capacity at older nodes and niche processes but the five big player comprise most of the capacity.

Though US manufacturing capacity has remained relatively stable, over half of the 27 new fab construction projects targeting advanced technology nodes are projected to be built in China over the coming years. Figure 9.19 (page 208) depicts the forecasted trajectories of US and Chinese chip manufacturing market share according to BCG and SIA’s report on Government Incentives and US Competitiveness in Semiconductor Manufacturing (Varas et al., 2020).

China has made development of its domestic SCI a key piece of its five-year (2020-2025) plan and invested heavily in the space – pledging over $150B in investment from 2014 – 2030 (SIA Whitepaper, 2021). However capital infusions can only go so far, however. Advance SC manufacturing requires significant pools of relevant engineering talent, a seasoned base of companies with technical know-how, and access to advanced manufacturing toolsets. Despite nearly $50B in government incentives delivered over the last 20 years, Chinese companies only account for about 8% of global SC sales today, with little footprint in advanced logic, cutting-edge memory, or higher-end analog chips. Despite these lackluster results so far, capital investments have driven annual growth rates of 15 – 20% and positioned China as a leader in the labor-intensive OSAT market, but the country is still likely a decade of more away from advanced technology nodes like those in Taiwan.

Ultimately, a more diversified and robust manufacturing base in both countries could reduce over-reliance by either country on the other and boost global manufacturing capacity, which would lower costs for consumers across the globe. Though rising design and manufacturing costs constrain growth and squeeze margins, the SCI faces a far more existential threat – the fundamental limits of transistor sizes and the slowing down of Moore’s Law.

**Chapter 10 The Future of Semiconductors and Electronic Systems**

**Prolonging Moore’s Law – Sustaining Technologies**

There are currently many promising research areas, both within existing technology architecture and brand-new sources of computing power, that will continue the technological march of the SCI for years to come. Within traditional silicon engineering, renewed focus has shifted from shrinking component size (geometric scaling) toward improving design efficiency and integration (functional scaling), as well as exploring new materials and design methods to prolong our ability to keep pace with Moore’s prediction. Some key technologies as we head into the future include:

1. **2.5D and 3D die stacking** – stacking multiple die and connecting them by through silicon via. By building up instead of out, data transfer rates are vastly improved, costs are reduced, power is conserved, and space is preserved by increasing transistor density on the substrate. The trade-off is that design complexity of existing designs in greatly increased due to more intricate data flow schemas and system architectures. At an unstacked (1D) 3nm node, design costs range from $0.5B to $1.5B. What would be the same incremental cost/benefit for going from 1D to 2.5/3D keeping design constant? Stacked logic will undoubtedly increase design costs further, though high-performance applications like AI will likely continue to drive demand for increasingly complex system architecture.
2. **Gate-All-Around (GAA)** transistors and new channel materials are a promising next step in the evolution of transistor technology and offer the most concrete path to prolonging geometric scaling. Planar transistors, the dominant transistor technology from the 1960s through 2000s, is satisfactory at the 20nm or larger node, however sub-20nm, planar transistors suffer from crippling leakage current issues that hamper overall system power. In 2011, planar was phased out by FinFET for high advanced ICs. FinFETs reduce leakage issues and enable greater control at lower voltages. However, even with FinFETs, some current leaks out even if the transistor is turned off. Leakage has become a prominent issue as we move towards 3nm and 2nm nodes. With gate control across four dimensions, Nanowire and Nanosheet GAA transistors aim to solve these issues. GAA present unique deposition and etch challenges that likely require new channel materials like strained Silicon Germanium (SiGe) to mitigate electron mobility issues.

As transistors become smaller and more densely packed, heat dissipation becomes a major performance constraint, forcing many devices to run below their maximum speed to avoid overheating. Traditionally silicon has been used as the main channel material, but has power density constraints that limit many of the performance advantages from small GAA transistors. This challenge can be tackled by implementing different channel materials with greater electron mobility. This is what strain engineering is focused on and research into SiGe, GaAs, GaN, and other III-V elements in a hotbed area.

1. **Custom Silicon and Specialized Accelerators** – have shown promise for functional scaling of application categories and performance improvements of specific products. Functional scaling has been clearly demonstrated in GPUs where Nvidia’s GPU’s ability to perform key AI calculations have approximately doubled every year for a 317x improvement through May 2020. This has been dubbed Huang’s Law named after Nvidia’s CEO Jensen Huang.

Functional improvements in key areas like memory and GPUs can continue improving IC performance even if geometric scaling slows down. In addition to functional scaling of widely adoptable subsystems, custom silicon design can boost performance through tighter integration using existing technology. Systems companies like Apple, Google, Facebook, and Tesla are all developing custom chip. Though cost prohibitive of smaller players that depend on fabless design companies to design their chips, large product companies are increasingly building customer chips in-house. These companies can afford the cost of building internal engineering groups, and by shifting from commodification to full customization, they are able to sustain competitive advantages in performance that may not be possible when working with third-party providers which also allowing for quality control advantages and reducing the need to disclose sensitive information.

In addition to current leakage, quantum tunneling interference is another big problem is very small atomic scales. There are numerous candidate materials, such as **graphene sheets** and **carbon nanotubes** (strong materials that are only one atom thick), that is well suited to resist tunneling interference. Carbon nanotubes, however, are difficult to manufacture and will requires a lot for R&D before they are ready for market.

Optical Chips and Optical Interconnects use light instead of electrons as the main signal carrier within and between electronic devices. One advantage over copper is that multiple optical signals can the transmitted and received in parallel by using different wavelength. VC backed Ayar Labs and researchers from MIT and UC Berkeley working on a DARPA-funded project called the Photonically Optimized Embedded Microprocessors (POEM) project have already begun commercializing photonic chip technology. Ayar has targeted chip-to-chip communication, creating I/O optical interconnects that are much faster and more power efficient than traditional copper wiring. Optics and optical interconnects are very promising because they may resolve several significant bottlenecks in the SC architecture and its related data transmission that exist today.

**Overcoming Moore’s Law – New Technologies**

Outside of advances in traditional silicon engineering techniques, there are some truly fascinating technologies in development that could launch us into a post-Moore’s Law computing renaissance.

1. **Quantum Computing** – the physics of QC is very complicated, but in essence quantum computers combine superposition with entanglement to execute exponentially more complex calculations that modern computers can handle.
2. **Quantum Transistors** – provide an interesting twist to quantum computing by harnessing the power of quantum tunneling and entanglement to process and store information. Despite all of this R&D, tunneling and entanglement are still not fully understood and maintaining the atomic-scale control necessary to harness such forces is an incredibly difficult engineering challenge in need of considerable investment and inquiry. Quantum transistors are still in early development, though researches have been able to develop working prototypes as proof of concept.
3. **Neuromorphic Computing** – technologies are modeled after living processing structures like neurons and present a fascinating potential avenue for computing progress beyond Moore’s Law. **Neurotransistors**, spiking neural networks (**SNN**, Intel investigating SNN applications in AI), The Human Brain Project (**HBP**), **SpiNNaker** System, **BrainScaleS** Systems (employes analog and mixed-signal components), and DNA as a new paradigm for data storage are some of the cutting edge areas within neuromorphic computing.

Google, the leader in search, has been researching and developing quantum computing technology since 2016 and hopes to have a quantum computer by 2029. Quantum computing could find applications in a diverse set of fields including cryptography, machine learning, medicine and materials, and big data searching and mining.

We can think of prolonging and overcoming technologies in terms of geometric scaling (making components smaller and more efficient) and functional scaling (greater performance for a given feature size or rendering feature size irrelevant by transforming computing structure to new base components and system architectures) respectively. Unlike functional scaling using existing transistor technology, functional scaling can also be achieved by paradigm shifts in computing performance irrespective of nm size. We will need both in the coming decades to fuel the relentless pace of technological development and innovation.

**Your Personal Semiconductor Awareness Test**

**Chapter 1 Semiconductor Basics**

1. Define electricity and conductivity. How do they related to current and voltage?
2. What is the most important semiconductor and why?
3. Which two inventions are responsible for the modern semiconductor industry? Why were these innovations so important?
4. After fabrication and manufacturing, what step in the SC value chain is necessary before system integration? Can you name all six?
5. What does PPAC stand for? Can you name which key design factor is missing?

**Chapter 2 Circuit Building Blocks**

1. Name the five types of discrete components we covered in this chapter. What function does each one perform? What are their differences?
2. Describe how a transistor in structure. What are its major components and how do they work?
3. What is the difference between a MOSFET and a FinFET transistor?
4. What is CMOS and what can it refer to?
5. How do logic gates work? What kind of logic do they use?

**Chapter 3 Building a System**

1. What five levels of electronics did we cover? On which level are all higher levels built?
2. Can you name each stage of the IC design flow? How does each step relate to a construction analogy?
3. What are the differences between emulation, functional, and formal verification? Pros and cons for each?
4. Which stage of the silicon design flow represents the transition between front- and back-end design? What happens at this stage?
5. How do EDA tools help hardware designers build better systems?

**Chapter 4 Semiconductor Manufacturing**

1. What are the four kinds of processes used for wafer fabrication?
2. Which core process technology is seen as a bottleneck to the rest of the industry? Why?
3. Can you tell the difference between front-end-of-the-line (FEOL) and back-end-of-the-line (BEOL)? How does this differ from front-end and back-end manufacturing?
4. Why is yield such an important metric? What is it used for?
5. What are the five core steps in the assembly and testing process?

**Chapter 5 Tying the System Together**

1. What are interconnects and what makes them so important?
2. What is the difference between wire bonding and flip-chip bonding? How do these impact the number of interconnects and transmission speed of a system?
3. Why do signal integrity engineers exist? What kinds of interfaces and transmission methods might they encounter?
4. In a chipset, why are the CPU, the northbridge, and southbridge arranged the way they are? What do each of these bridges handle and how are they different?
5. Describe the four main stages of power flow in an electronic system. Which components or modules handle each stage? Can you related each of them to similar components in a water utility system?

**Chapter 6 Common Circuits and System Components**

1. Compare and contrast analog and digital signals.
2. In the SIA Framework, what is the difference between Micro Components and Logic?
3. Which memory type sits closest to the CPU in the memory hierarchy? Why?
4. What makes MEMS similar and different from integrated circuits? If you were to categorized MEMS devices, which SIA component family would you choose?
5. Why are analog components well suited for wireless communications? Why not digital components?

**Chapter 7 RF and Wireless Technologies**

1. What do we mean when we say “RF”? Which key characteristics set one RF signal apart from another?
2. How does the FCC manage shared frequency bandwidths? What technologies are used to fit more information into a given amount of bandwidth? How does each technology accomplish this task?
3. Name the five key base components to any transmitter or receiver. Which component is number six and what makes it special?
4. Why is the physical layer so important in the OSI model? What are the other six layers and how do they function?
5. What makes each generation of telecommunications technology unique? How have these advancements enable the growth and success of cloud computing?

**Chapter 8 System Architecture and Integration**

1. What is the key difference between macro- and micro- level architecture? Where do ISAs fit in?
2. Which theoretical advantages does Harvard Architecture have over Von Neumann? Why does this not play out in real life?
3. Name four key differences between CISC and RISC.
4. Which ISA strategy is the most advantageous across the most design- and market-based constraints? What are its shortcomings?
5. Why are cost advantages not so straightforward between monolithic and heterogeneous integration? How do these factors relate to performance?

**Chapter 9 The Semiconductor Industry – Past, Present, and Future**

1. How have burgeoning design and manufacturing costs shaped the evolution of the semiconductor industry?
2. Make the case of IDMs. What are some competitive advantages they may have that fabless companies do not? Do you think these are sustainable?
3. List three key current industry trends. Which do you think is most important?
4. What are core drivers of consolidation? Can you think of any drawbacks to being too big (think IDMs)?
5. Describe the distribution of global value chain activities and consumption across the United States and Asia. Which surprising factor is responsible for much of Asia’s manufacturing cost advantages?

**Chapter 10 The Future of Semiconductors and Electronic Systems**

1. What advantages does stacking have over traditional “two-dimensional” ICs? Disadvantages?
2. Compare and contrast the structure of planar, FinFET, and GAA transistors. What gives GAA an advantage?
3. Name three promising channel materials. What are these materials so important?
4. Describe the difference between a bit and a qubit. Which applications might quantum computers be well suited for over traditional digit devices?
5. What questions do geometric scaling and functional scaling ask us? Classify each of the technologies covered in this chapter as more geometrically oriented or functionally oriented in nature.

1. ***SCI – semiconductor industry*** [↑](#footnote-ref-2)
2. What are the pros and cons of wire molding versus flip-chip technology? [↑](#footnote-ref-3)
3. Why is removal processing such a big slice? [↑](#footnote-ref-4)
4. Can you put packaging options on a spectrum across various manufacturing parameters and see how this lends itself to competitive advantage and valuation? [↑](#footnote-ref-5)
5. Can this be the basis for making a valuation trade based on the slowing of Moore’s Law and geometric scale with a shift toward functional scaling? Is there any economic proof that such a valuation shift has or is taking place? [↑](#footnote-ref-6)
6. What of objective ways to measure signal integrity as a function of transistor size and density? [↑](#footnote-ref-7)