SoC IC Basics

COE838: Systems on Chip Design
http://www.ecb.torontomu.ca/~courses/coe838/
Dr. Gul N. Khan
http://www.ecb.torontomu.ca/~gnkhan
Elect., Computer & Biomedical Engineering
Toronto Metropolitan University

Overview

- SoC Chip/IC Overview
- Cycle Time and Performance
- Chip Area and Yield
- Power and Reliability
- Configurability

Chapter 2 of the text book by M.J. Flynn & W. Luk as well as some additional material

SoC Design Tradeoffs
Five Big Issues for SoC Design

1. **Time:** Cycle time relates to Performance
2. **Chip Area:** It also determines the IC cost
3. **Power Consumption:** Performance as well as Implementation. **Some Instruction Sets need more chip area are less valuable than those requiring less area.**
4. **Reliability:** It relates to deep submicron effects.
5. **Configurability:** Standardization in manufacturing and customization for application.

Long-term *cost-performance ratio* is the basis of most design decision.

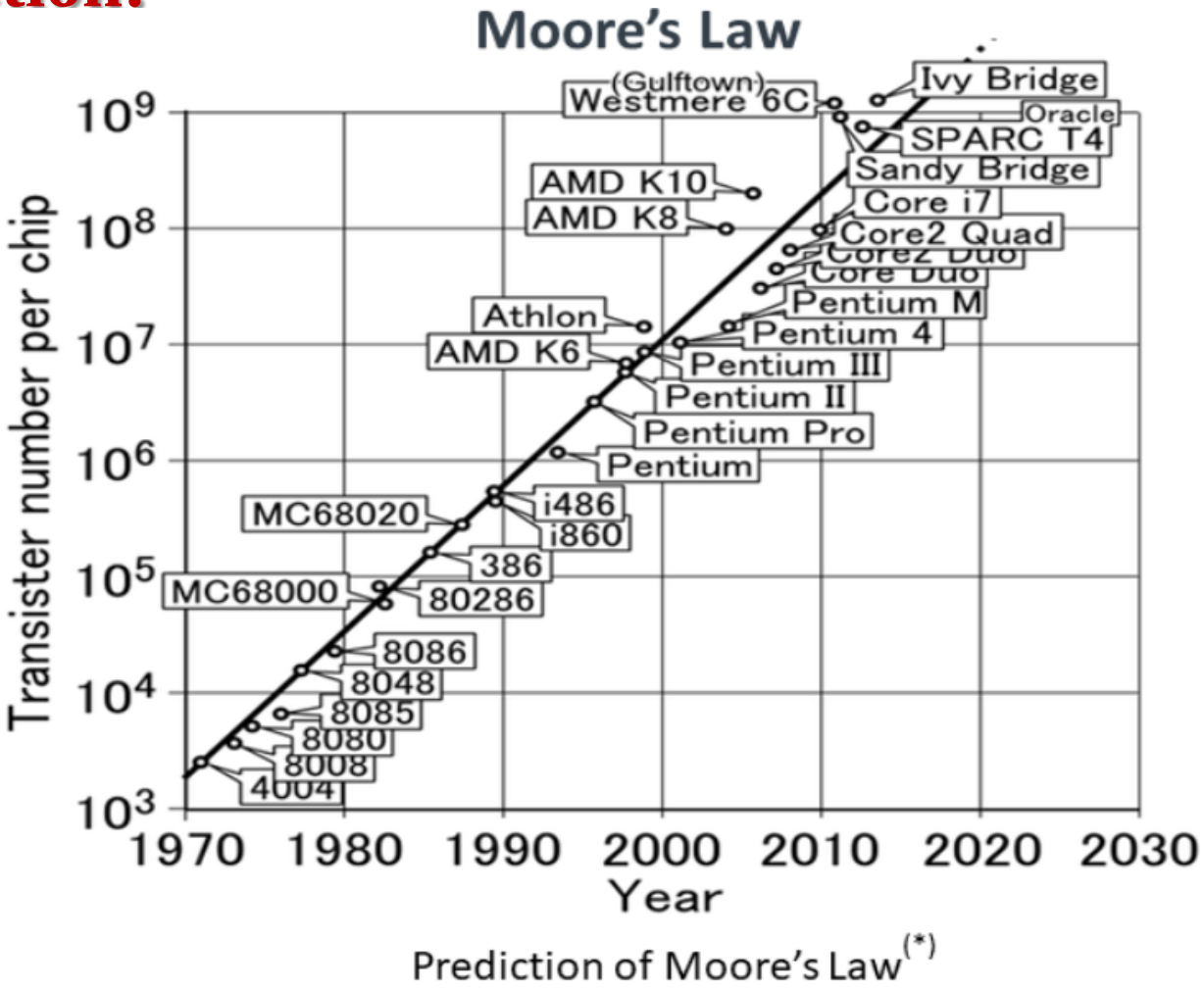
Chip/IC Technology Roadmap

Projections:

Year	2010	2013	2016
Technology generation (nm)	45	32	22
Wafer size, diameter (cm)	30	45	45
Defect density (per cm^2)	0.14	0.14	0.14
μP die size (cm^2)	1.9	2.6	2.6
Chip frequency (GHz)	5.9	7.3	9.2
Million transistors per square centimeter	1203	3403	6806
Max power (W) high performance	146	149	130

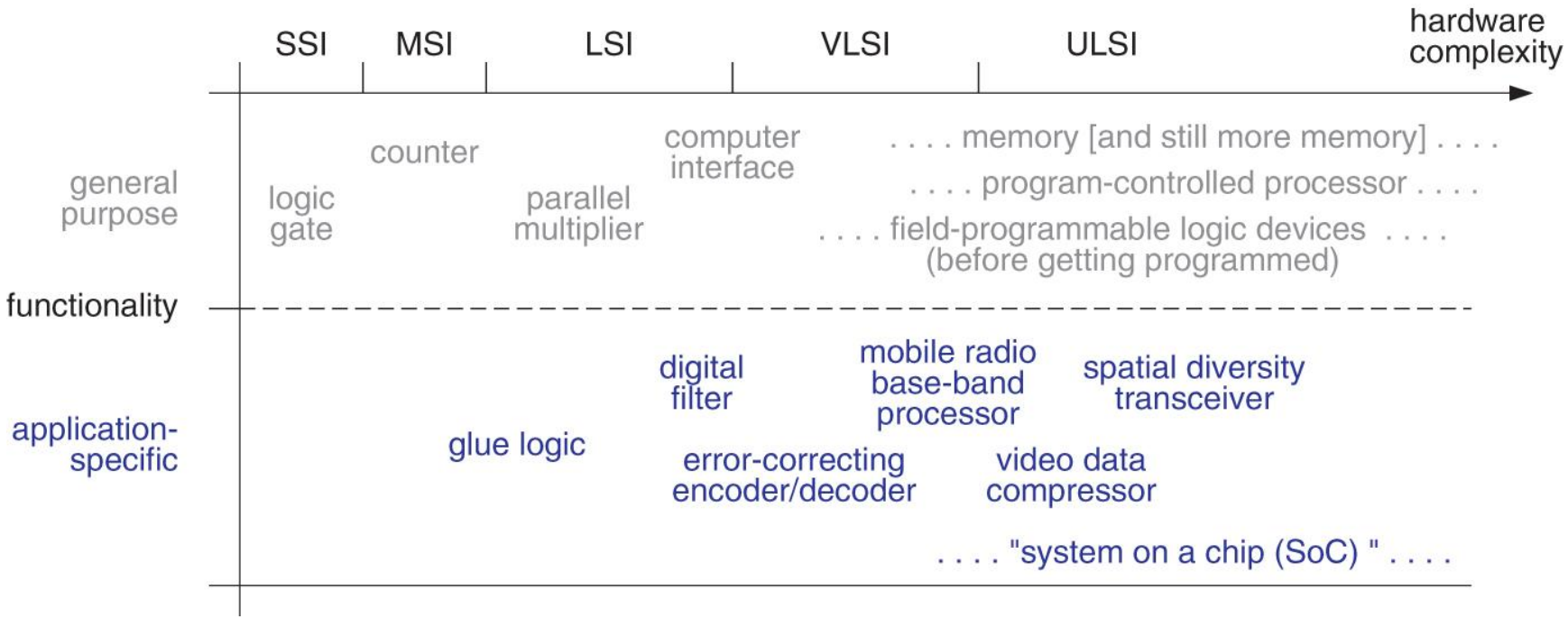
Chip/IC Technology Roadmap

Prediction:

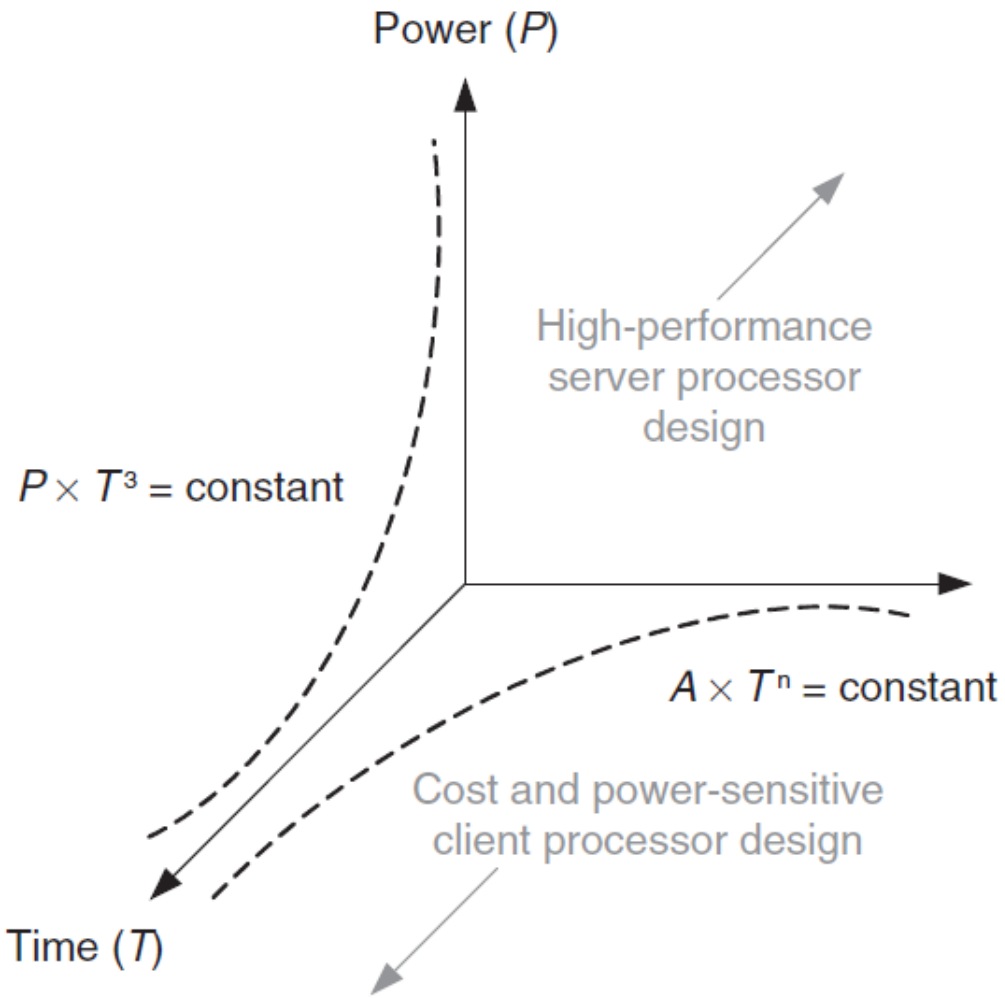


(*) Data are based on international semiconductor technology road map (http://www.itrs.net/)

SoC Hardware Complexity



CPU Design Tradeoffs



- Increase time **to complete a task**, decrease power.
- **Decrease area, decrease power consumption.**

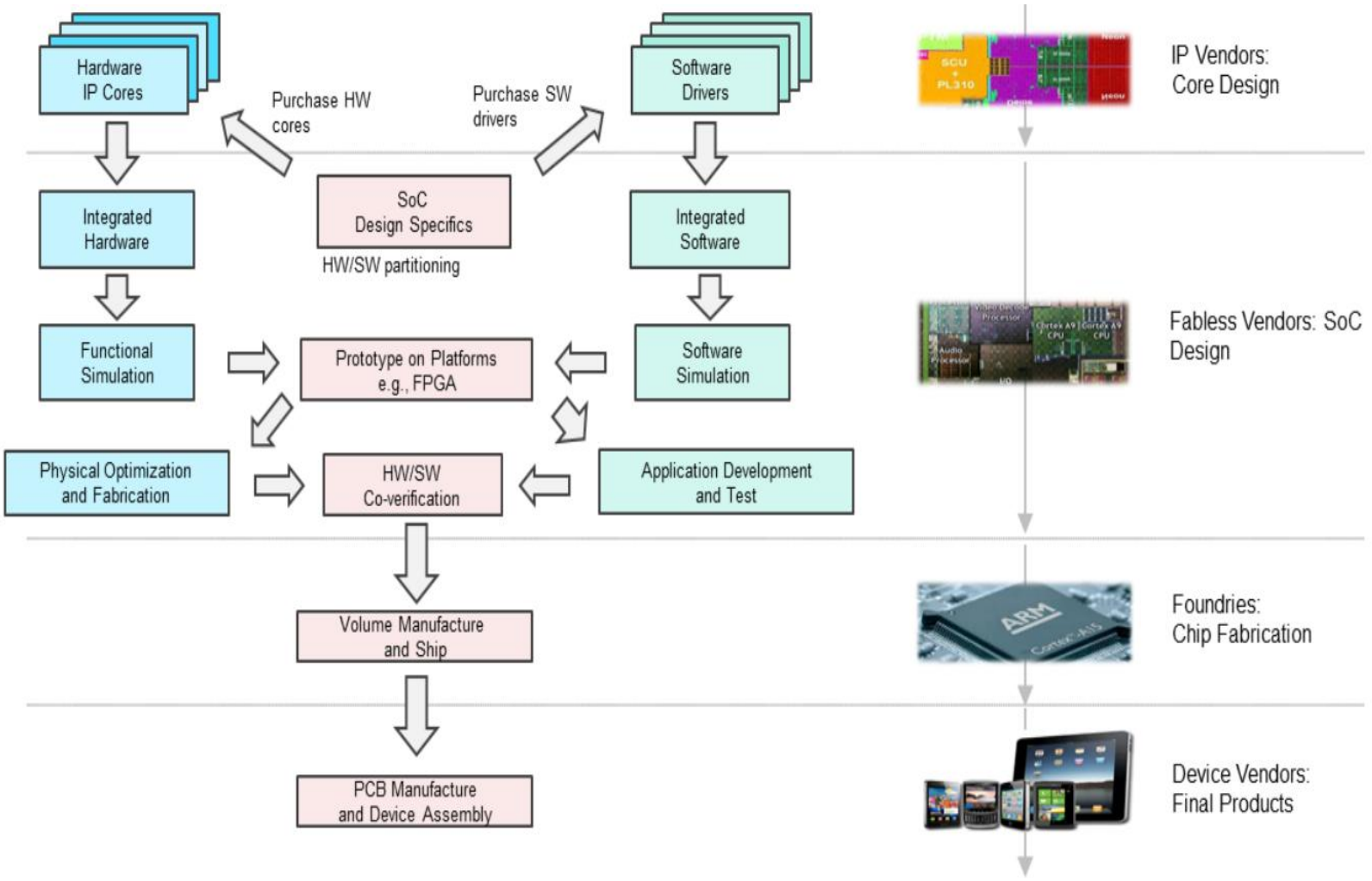
- Decrease SoC area, possible increase the time.

SoC Requirements & Specifications

Basic SOC design trade-offs provide a mechanism to analyze and translate SOC requirements into specifications.

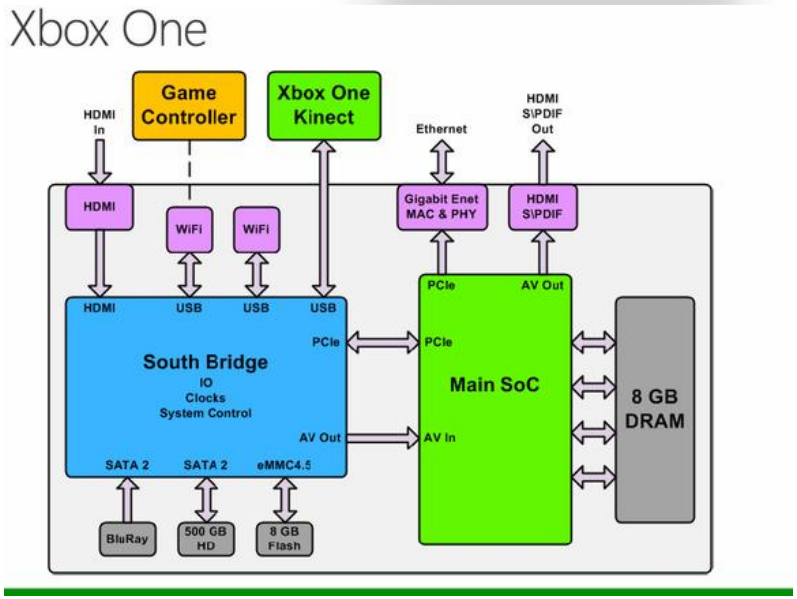
- Low-cost systems will optimize die cost, design reuse and may be low power.
- Gaming systems have low cost - especially the production cost. However, performance with reliability is a lesser consideration.
- Wearable systems stress on low power leading to lower weight of power supply. These systems, such as cell phones, have real-time constraints and their performance cannot be ignored.
- Embedded systems used in planes (aerospace) and other safety-critical applications require reliability, along with performance and design for lifetime (configurability).

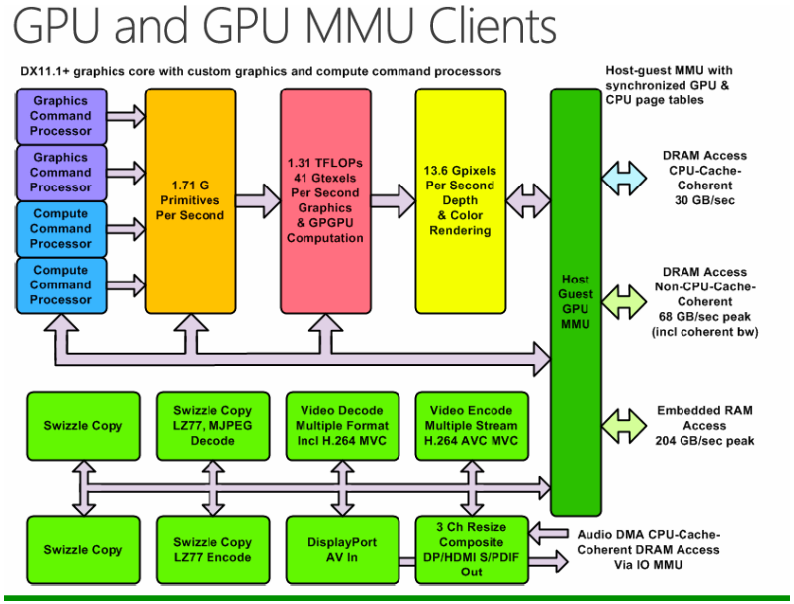
SoC Desgin and Implementation



SoC Design – 5 Big Issues

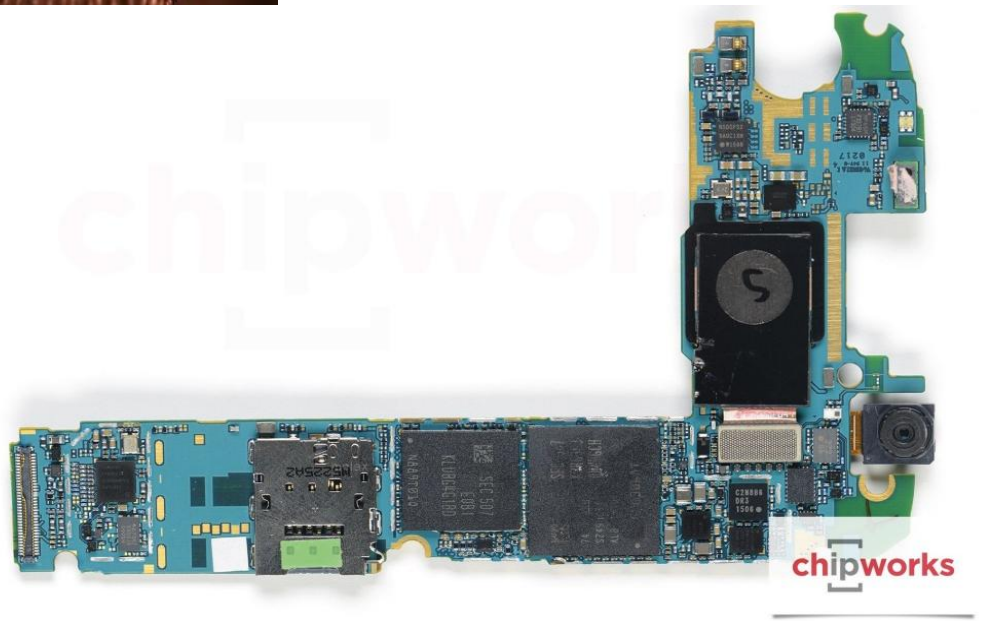
1. **Time**
2. Chip Area
3. Power Consumption
4. **Reliability**
5. *Configurability*





SoC Design – 5 Big Issues

1. Time
2. Chip Area
3. Power Consumption
4. Reliability
5. Configurability



Cycle Time

The basic measure of performance.

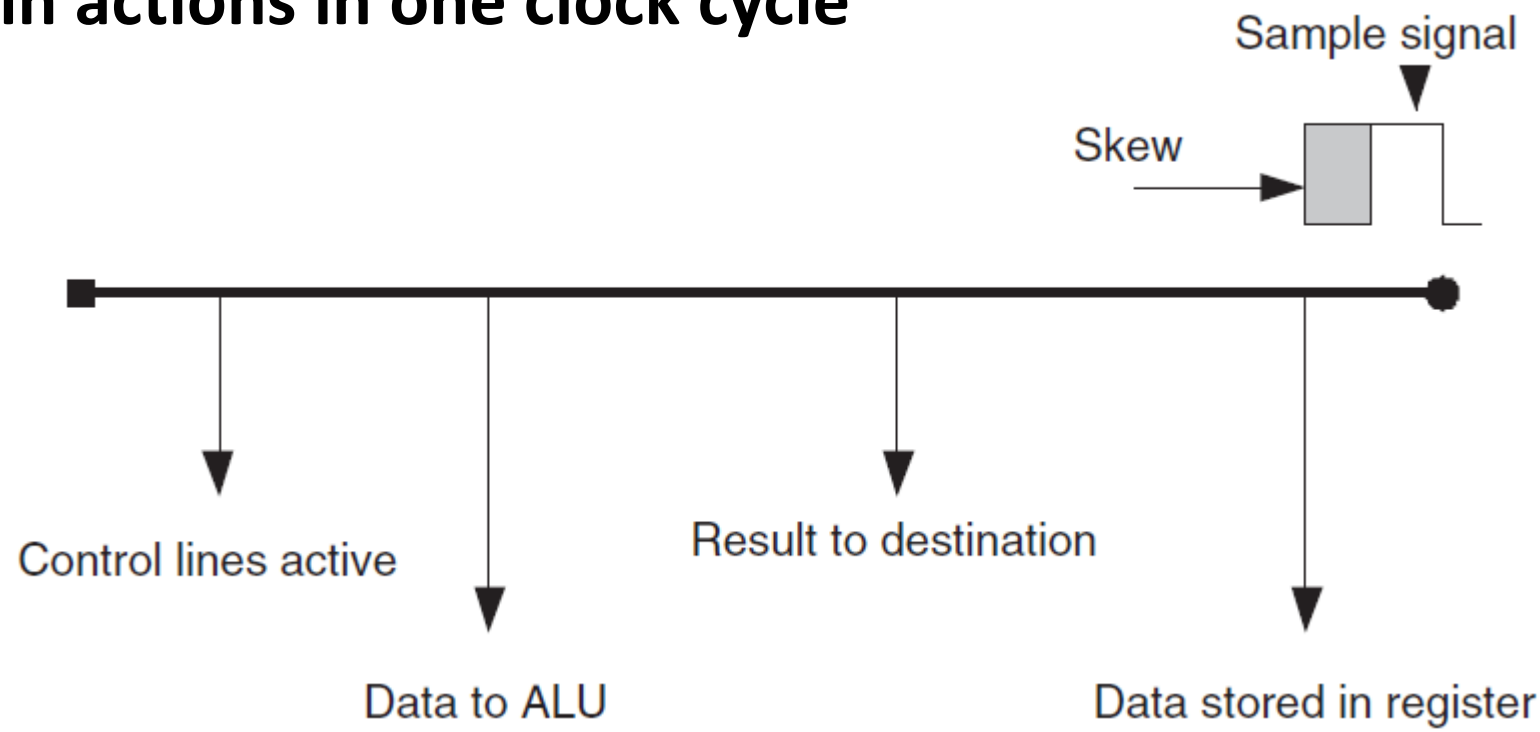
A cycle (of the clock) is the basic time unit for processing information.

- Clock rate is a fixed value and the cycle time is based on the maximum time to accomplish a frequent operation.
- Less frequent operations that require more time to complete? Use multiple clock cycles.

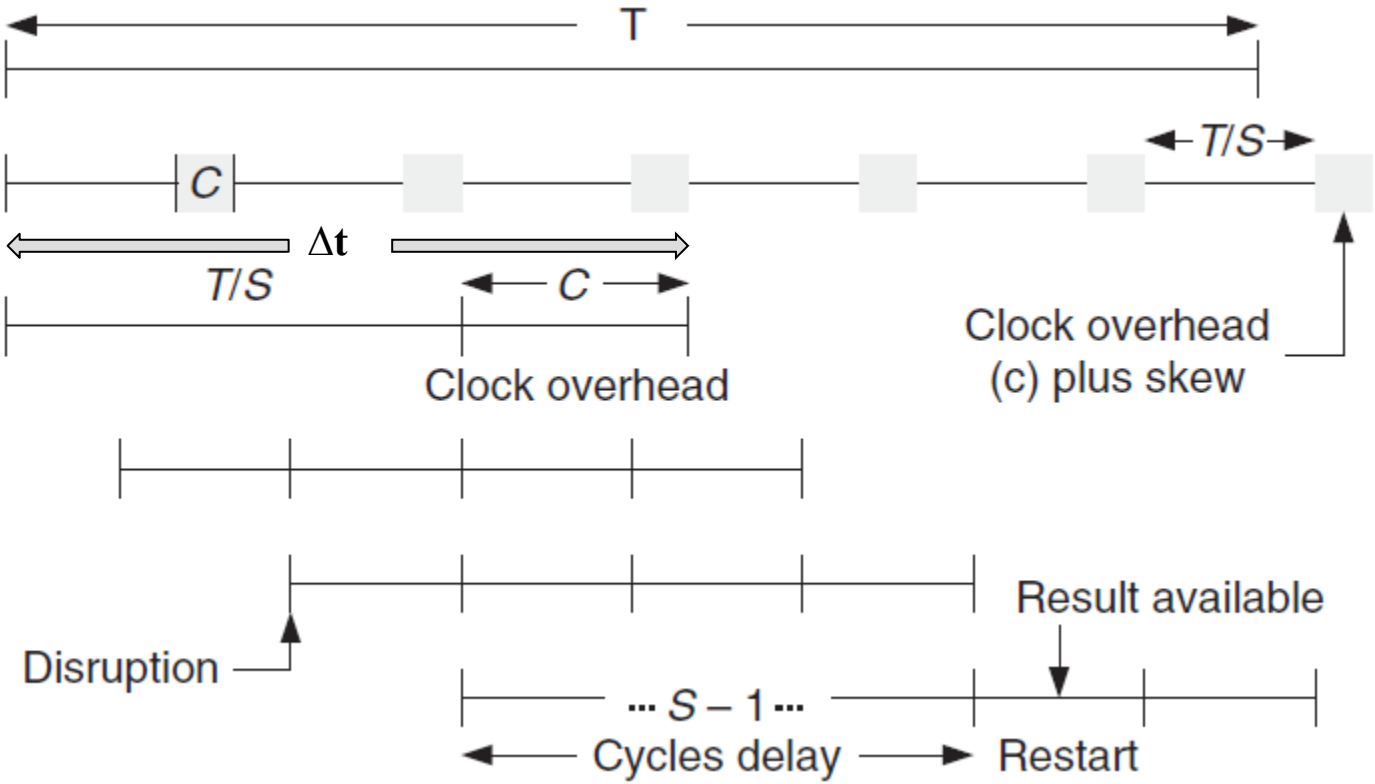
CPU Clock Cycle

Clock skew – clock arrives at a different time to different components **--- Also causes setup and hold violations.**

Main actions in one clock cycle



Pipelining and Clock Cycle



For S, segments; Pipeline Cycle : $\Delta t = T/S + C$

Where C (skew, setup time/hold time delays)

Much smaller than non-pipelined Cycle Time, T

Optimum Pipeline

Performance = 1/[1+ (S − 1)b] insts./cycle

where ***b*** is the number of pipeline disruptions

Throughput (G) = performance/Δt insts./ns

$$G = \{1/[1+ (S - 1)b]\} \times \{1/(T/S + C)\}$$

For dG/dS = 0, Optimal number of pipeline segments

$$S_{opt} = \sqrt{\frac{(1-b)T}{bC}}$$

However, a new pipeline segment will introduce an additional cost that is not considered.

Performance

- High clock rates with small pipeline segments may (or may not) produce better performance.
- Two basic factors enabling clock rate advances:

  (1) Increased control over clock overhead.

  (2) Increased number of segments in the pipelines.

- **Low clock overhead (small *C*) *may cause higher pipeline segmentation*, but performance does not correspondingly improve unless we also decrease pipeline disruption, *b***

DIE Area and Cost

Marginal cost to produce an SoC can be determined by the die area.

- There are significant side effects that die area has on the fixed and other variable costs.
- SOCs usually have die sizes of about 10-15 mm on a side.
- The die is produced in bulk from a larger wafer, perhaps 30 cm in diameter.
- Silicon wafers and processing technologies are not perfect. Defects randomly occur over wafer surface.

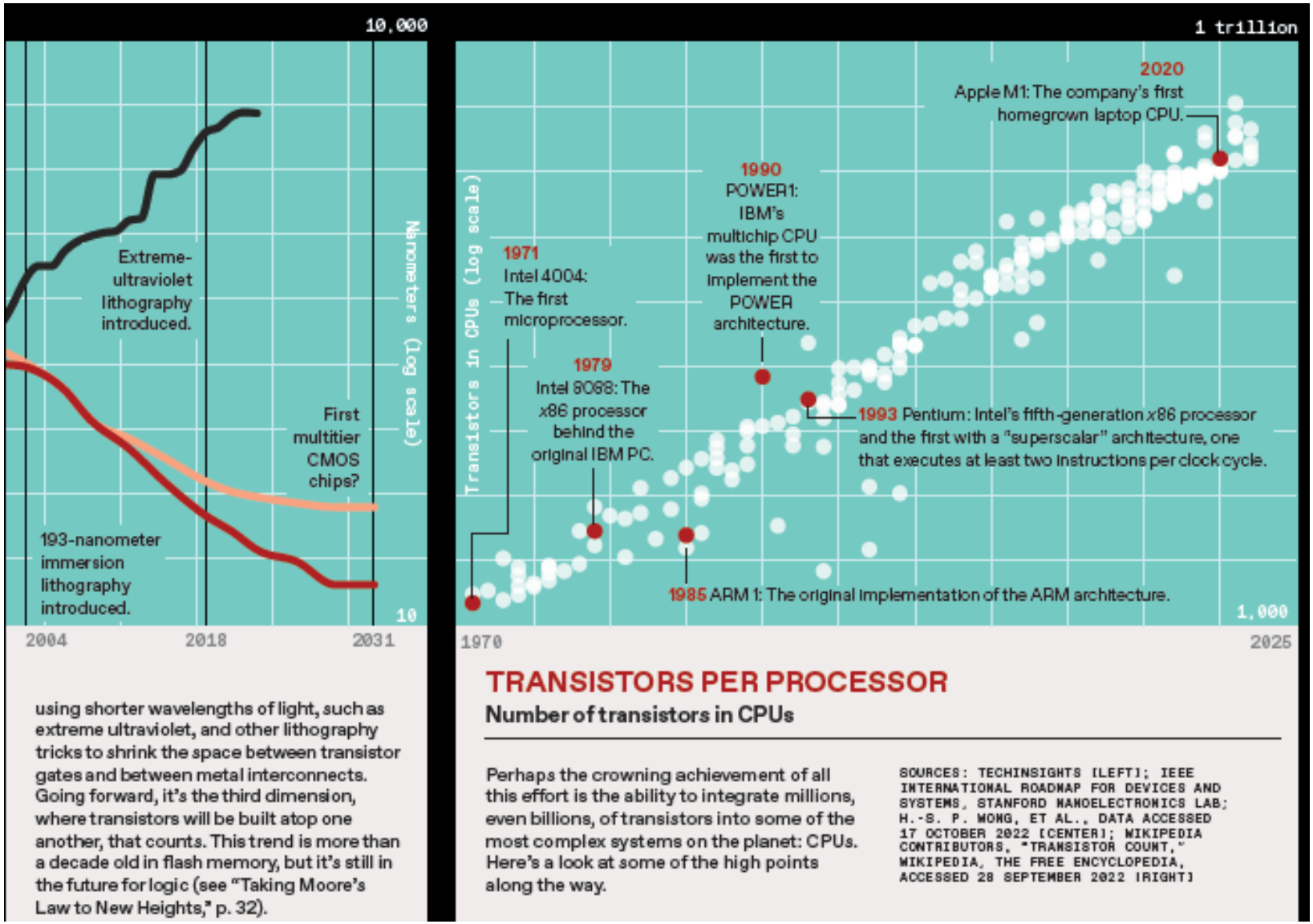
Die, Wafer size and other Technology Parameters for the last Five Years

Year	2010	2013	2016
Technology generation (nm)	45	32	22
Wafer size, diameter (cm)	30	45	45
Defect density (per cm^2)	0.14	0.14	0.14
μP die size (cm^2)	1.9	2.6	2.6
Chip frequency (GHz)	5.9	7.3	9.2
Million transistors per square centimeter	1203	3403	6806
Max power (W) high performance	146	149	130

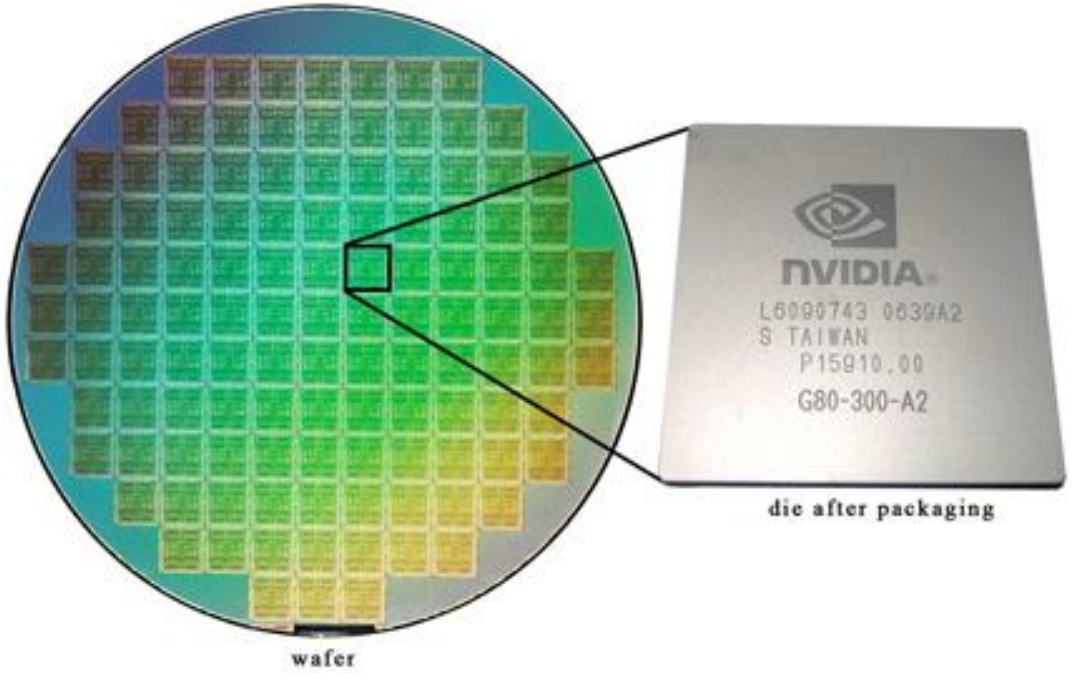
Technology Roadmap - Latest

Processor/SoC	Year	Designer	Technology	SoC Area (mm^2)	Transistor density ($tr./mm^2$)
AMD Zeppelin SoC Ryzen	2017	AMD	14 nm	192 mm^2	25,000,000
Xbox One X main SoC	2017	Microsoft/AMD	16 nm	360 mm^2	19,440,000
HiSilicon Kirin 970 (octa-core ARM64 mobile SoC)	2017	Huawei	10 nm	96.72 mm^2	56,900,000
Snapdragon 845 octa-core SoC	2017	Qualcomm	10 nm	94 mm^2	56,400,000
Apple A12 Bionic (hexa-core ARM64 mobile SoC)	2018	Apple	7 nm	83.27 mm^2	82,900,000
Snapdragon 865(octa-core ARM64 mobile SoC)	2019	Qualcomm	7 nm	83.54 mm^2	123,300,000
Apple M1 (octa-core 64-bit ARM64 SoC)	2020	Apple	5 nm	119 mm^2	134,500,000
M1 Pro (10-core, 64-bit)	2021	Apple	5 nm	245 mm^2	137,600,000
Snapdragon 8 Gen 2 (8-core ARM64 mobile SoC)	2022	Qualcomm	4 nm	268 mm^2	59,701,492
Apple M2 Ultra (24-CPU Cores and 76-GPU Cores)	2023	Apple	5 nm	?	134 Billon Transistors per SoC
Apple A17	2023	Apple	3 nm	103.8 mm^2	183,044,315
M4 (10-core ARM64 SoC)	2024	Apple	3 nm	?	

Transistors on CPU/SoC Chips



DIE Area and Cost



Each die (core, etc.) is produced in bulk from a wafer.

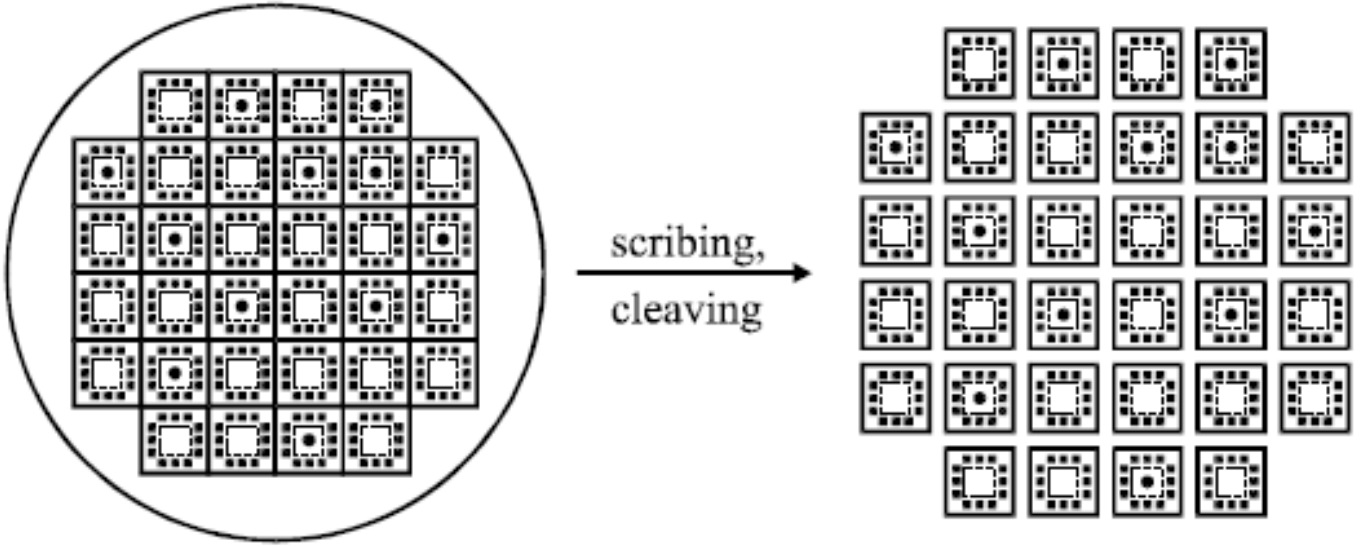
Moreover, silicon & technology processes are imperfect.

https://www.youtube.com/watch?v=qm67wbB5GmI

Scribing and Cleaving

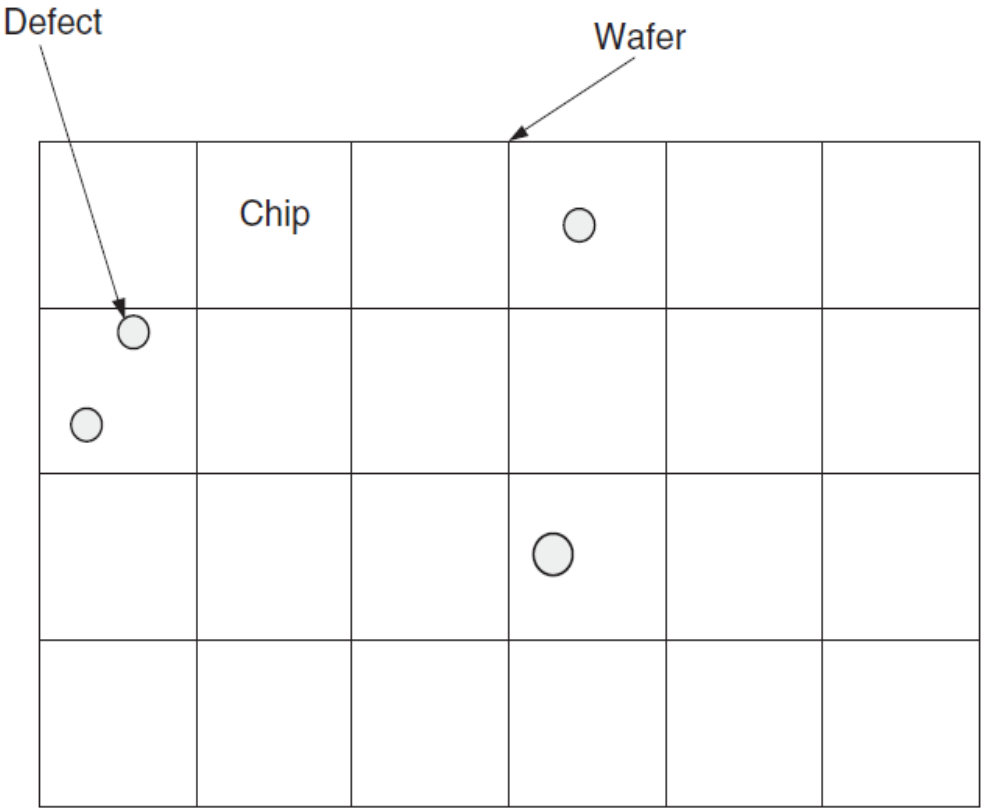
Fabricated wafers are separated into individual dice by scribing and cleaving.

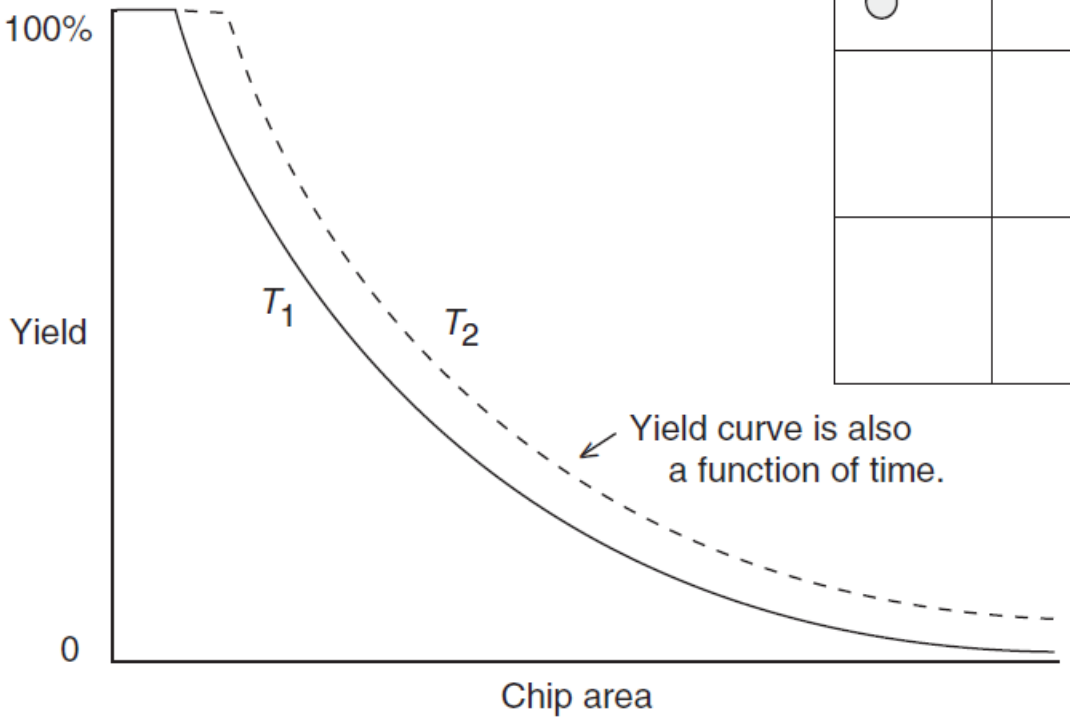
- Scribing is to create a groove along scribe channels which have been left between the rows and columns of individual chips.
- Cleaving is the process of breaking the wafer apart into individual dice between the adjacent dies on a wafer.



Wafer Defects

Defects randomly occur over the wafer surface.





Large SoC chip area requires an absence of defects over that area

Die Area and Yield

A good SoC design is not necessarily the one that has the maximum yield.

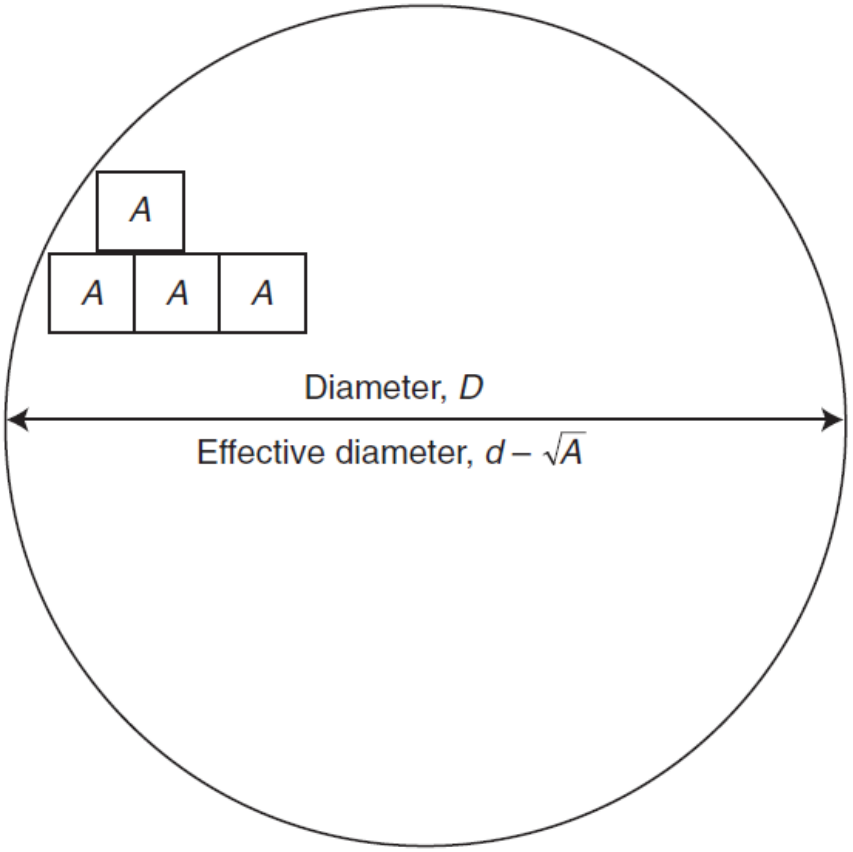
- **Reducing the area of a design below a certain amount has only a marginal effect on yield.**
- **Small designs waste chip area.**
  - There is an overhead area for pins and separation between the adjacent dies on a wafer.
- **Area available to a designer is a function of the manufacturing processing technology.**
  - **Purity of the silicon crystals,**
  - Absence of dust and other impurities,
  - Overall control of the process technology.

Improved manufacturing technology allows larger dice to be realized with higher yields.

Die Area and Yield

N number of die (of area *A*) on a wafer of diameter *d*

$$N \approx \frac{\pi}{4A}\left(d - \sqrt{A}\right)^2$$



N_G good chips and N_D point defects on the wafer.

***If $N_D > N$,* one can still expect several good chips.**

N_G / N is the probability that the defect damages a good die.

Note-if a defect hits a defected die then it will not affect yield

$dN_G/dN_D = -N_G/N$ or $1/N_G\,(dN_G) = -1/N\,(dN_D)$

Integrating and solving or $\ln N_G = -N_D/N + C$

Die Yield

$\boldsymbol{ln}\ N_G = -N_D/N + C$

$N_G = N$ *means* $N_D = 0$*; then* C must be *ln(N)*

$$\text{Yield} = \frac{N_G}{N} = e^{-N_D/N}$$

For ρ_D is the defect density per unit area, then

$$N_D = \rho_D \times \textit{(wafer area)}$$

For large wafers where $d >> \sqrt{A}$ ***;***

$$\left(d - \sqrt{A}\right)^2 \approx d^2 \quad \textit{and} \quad N_D / N = \rho_D A$$

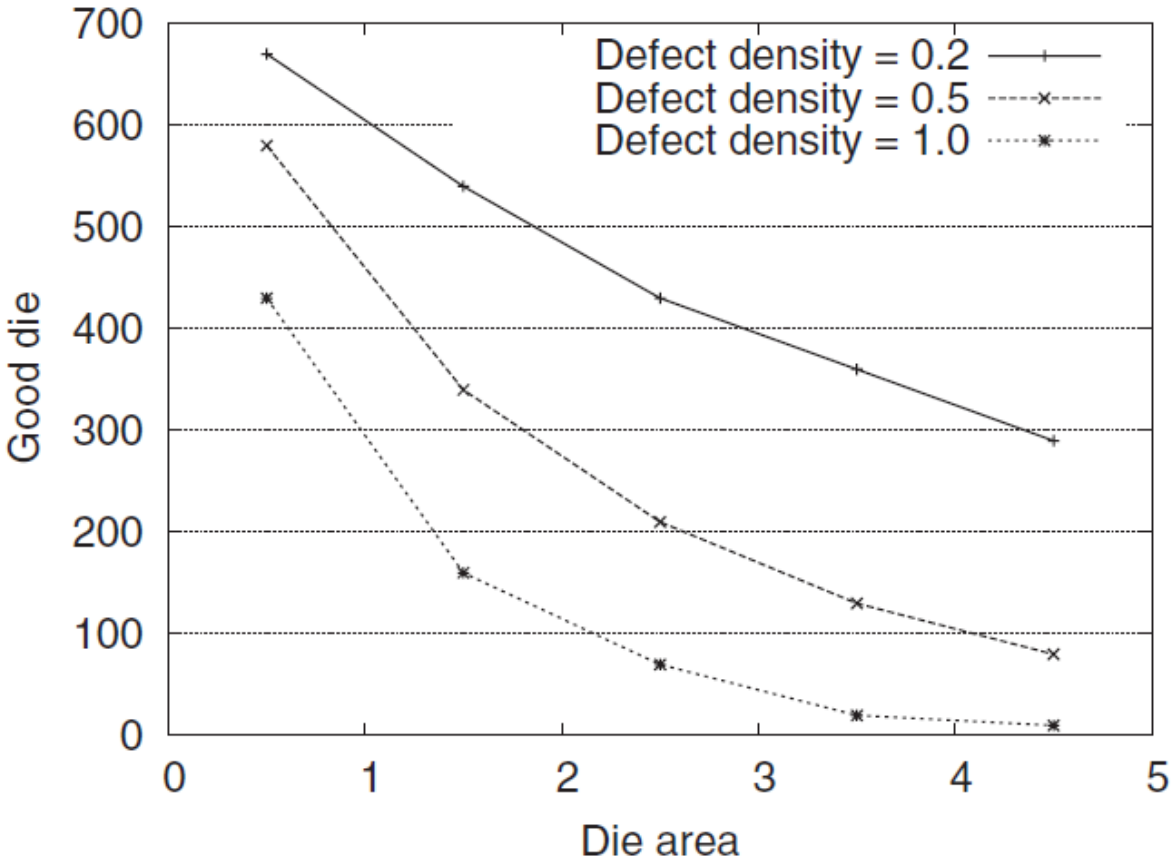
So that $\text{Yield} = e^{-\rho_D A}$ **= good dies obtained from the wafer**

Wafer Defects

Large die sizes are very costly. Doubling the die area has a significant effect on the yield for a large $\rho_D \times A$ *(≈ 5 – 10 or more).*

A modern fab. facility would have a ρ_D *of* (0.15 → 0.5)

It depends on the maturity of the process and the expense charges by the fab. facility



Feature and Area Unit - Details

- A mm^2 area unit is good, but photolithography and geometries' resulting minimum feature sizes are constantly shifting, a dimensionless unit is preferred.
- A unit λ is the distance from which a geometric feature on any one layer of mask may be positioned from another.
- A transistor is $4\lambda^2$, positioned in a minimum region of $25\lambda^2$.
- The minimum feature size, ***f*** is the length of one Polysilicon gate, or the length of one transistor, $\mathbf{f} = 2\lambda$.
- Register bit equivalent (rbe) is a useful unit defined to be a 6-transistor register **(memory)** cell and represents about $2700\lambda^2$.
- Even larger unit, ***A*** is defined as 1 mm^2 of die area at $\boldsymbol{f} = 1\mu m$. This is also the area occupied by a 32×32 bit three-ported register file or 1481 rbe.

Feature and Area Unit

1 register bit (rbe)	1.0 rbe
1 static RAM bit in an on-chip cache	0.6 rbe
1 DRAM bit	0.1 rbe
rbe corresponds to (in feature size: f)	1 rbe = $675f^2$
Item: Size in A Units	
A corresponds to 1 mm^2 with $f = 1\,\mu m$.	
$1\,A$	$=f^2 \times 10^6$ (f in μm)
or about	$\approx$1481 rbe
A simple integer file (1 read + 1 read/write) with 32 words of 32 bits per word	=1444 rbe
or about	$\approx 1\,A$ ($=0.975\,A$)
A 4-KB direct mapped cache	=23,542 rbe
or about	$\approx 16\,A$
Generally a simple cache (whose tag and control bits are less than one-fifth the data bits) uses	$=4\,A/KB$
Simple Processors (Approximation)	
A 32-bit processor (no cache and no floating point)	$=50\,A$
A 32-bit processor (no cache but includes 64-bit floating point)	$=100\,A$
A 32-bit (signal) processor, as above, with vector facilities but no cache or vector memory	$=200\,A$
Area for interunit latches, buses, control, and clocking	Allow an additional 50% of the processor area.

Baseline SoC Area Case Study

Consider a manufacturing process that has a defect density of 0.2 defects per cm^2; we target an initial yield of 95%

Chip Area A = $25mm^2$ by employing $\mathbf{Y = e^{-\rho_D A}}$

Feature Size: The smaller the feature size, the more logic that can be accommodated within a fixed area.

For f = 65 nm, we have about **5200A** or area units in 22 mm^2

The Architecture:

- **a small 32-bit core processor with an 8 KB I-cache and a 16 KB D-cache;**
- **two 32-bit vector processors, each with 16 banks of 1 K×32b vector memory; an 8 KB I-cache and a 16 KB D-cache for scalar data;**
- **a bus control unit;**
- **directly addressed application memory of 128 KB ; and**
- **a shared L2 cache.**

Baseline SoC Area Model

An Area Model:

Unit	Area (*A*)
Core processor (32 b)	100
Core cache (24 KB)	96
Vector processor #1	200
Vector registers and cache #1	256 + 96
Vector processor #2	200
Vector registers and cache #2	352
Bus and bus control (50%)	650
Application memory (128 KB)	512
Subtotal	2462

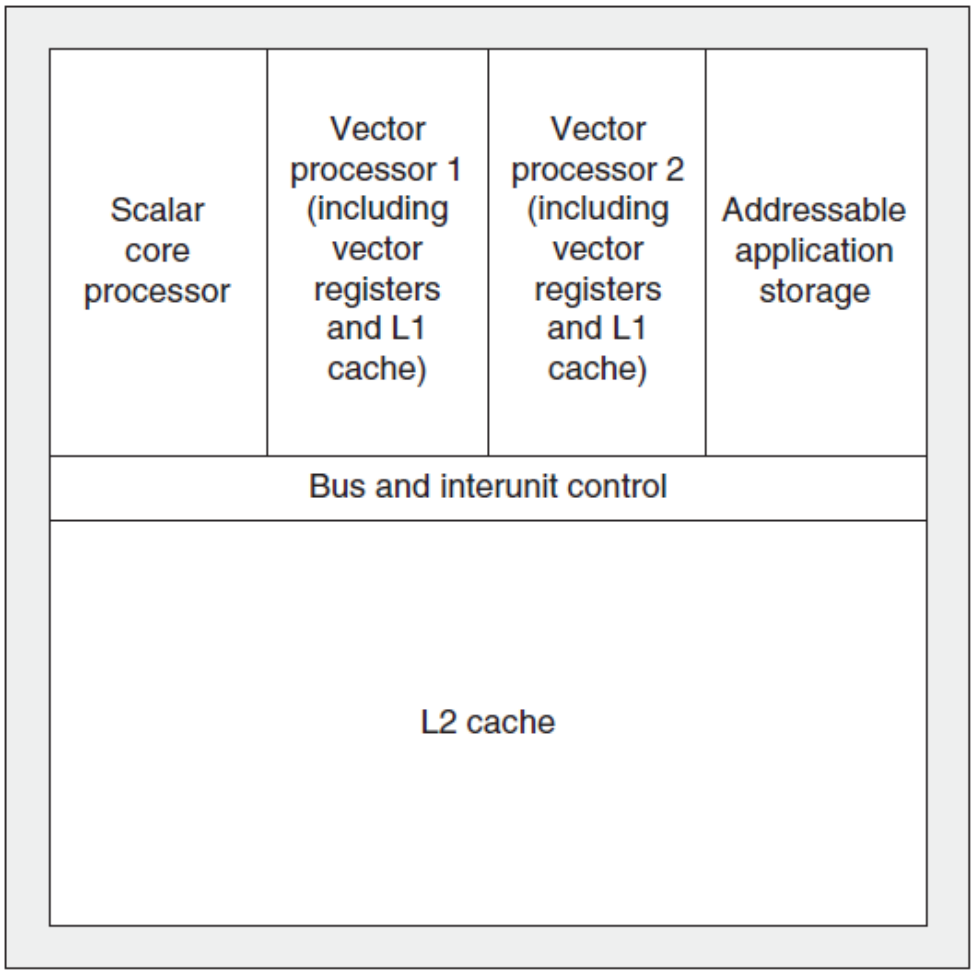
Latches, Buses, and (Inter-unit) Control: 10% overhead for latches and 40% overhead for buses, routing, clocking, and overall control

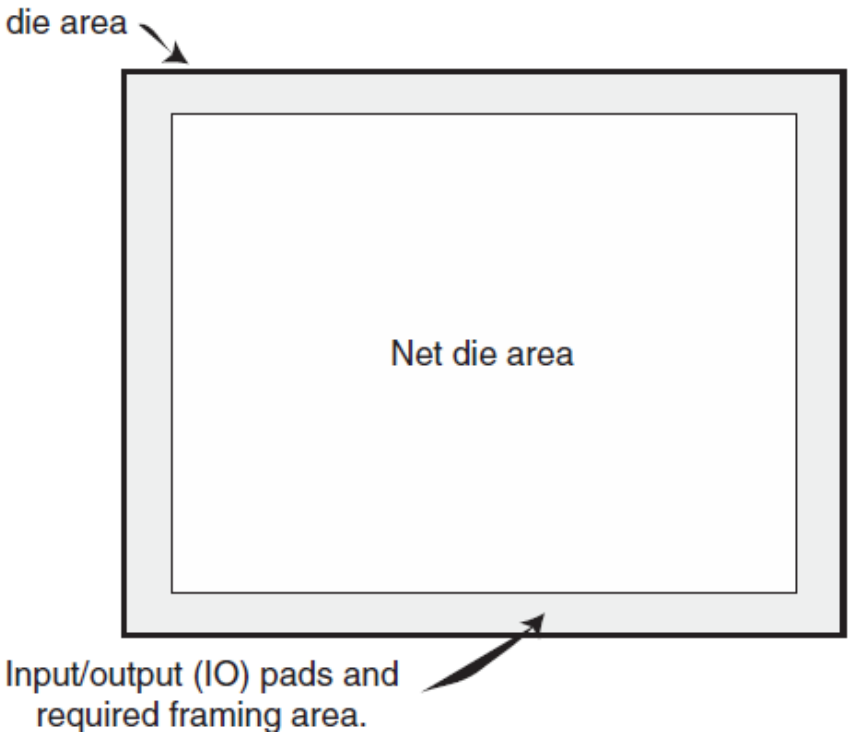
Total System Area: 5200-2462 = 2738A for Cache

Cache Area: 2738A

Baseline SoC Area Case Study

Baseline die floor plan





We allow 12% of the chip area - around the periphery of the chip **- then the net area will be 22 mm^2 rather than 25 mm^2**

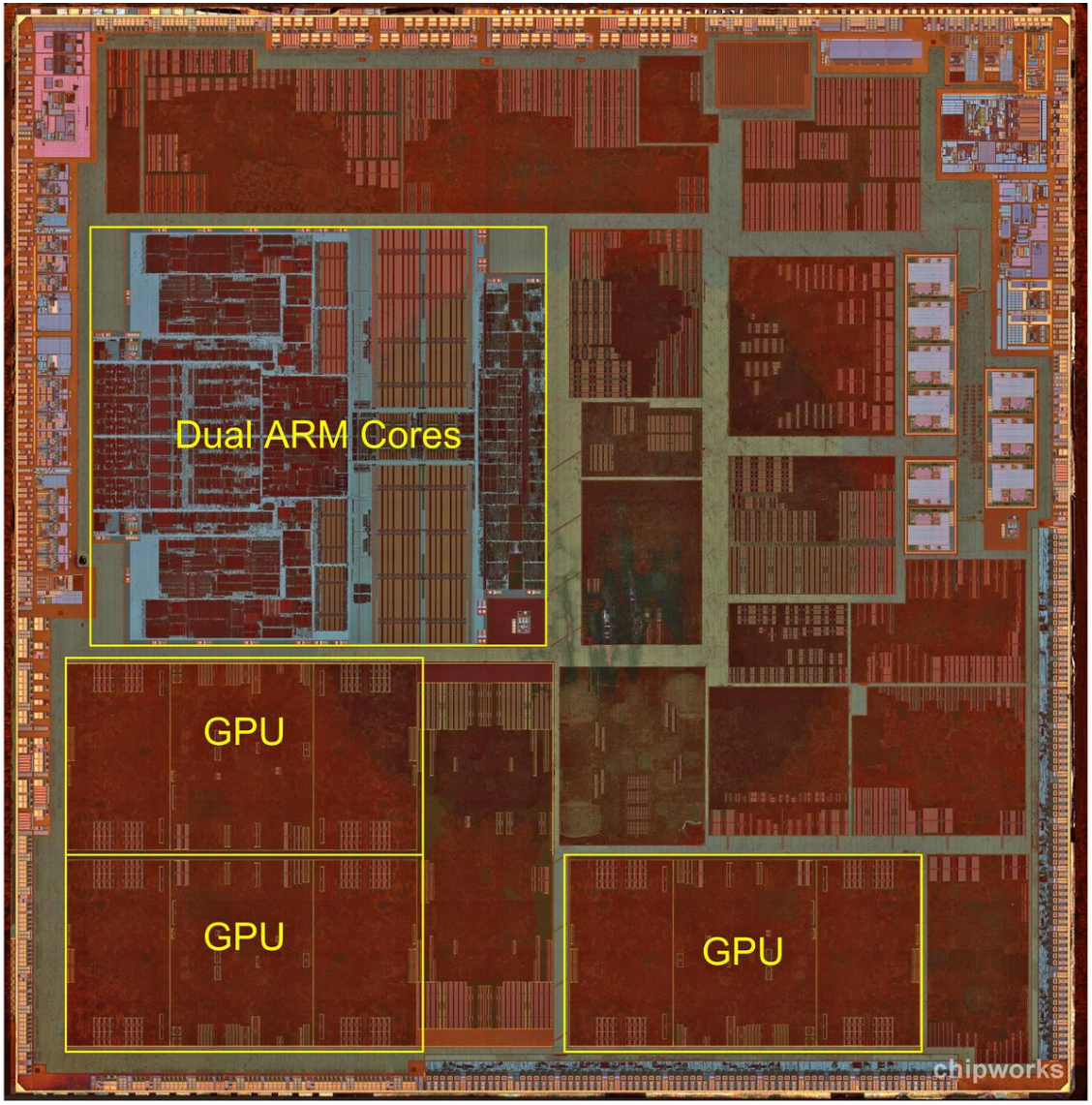
SoC Area Design Rules

Feature Size (μm)	Number of A per mm^2
1.000	1.00
0.350	8.16
0.130	59.17
0.090	123.46
0.065	236.69
0.045	493.93

Design Rules:

1. Compute the target chip size using the yield and defect density.
2. Compute the die cost and determine whether it is satisfactory.
3. Compute the net available area. Allow 10 – 20% (or other appropriate factor) for pins, guard ring, power supplies, etc.
4. Determine the **rbe size from the minimum feature size.**
5. Allocate the area based on a trial system architecture until the basic system size is determined.
6. Subtract the basic system size (5) from the net available area (3). This is the die area available for cache and storage optimization.

Apple A6 SoC



Apple SoC Examples

SoC Example: Apple SoC Families

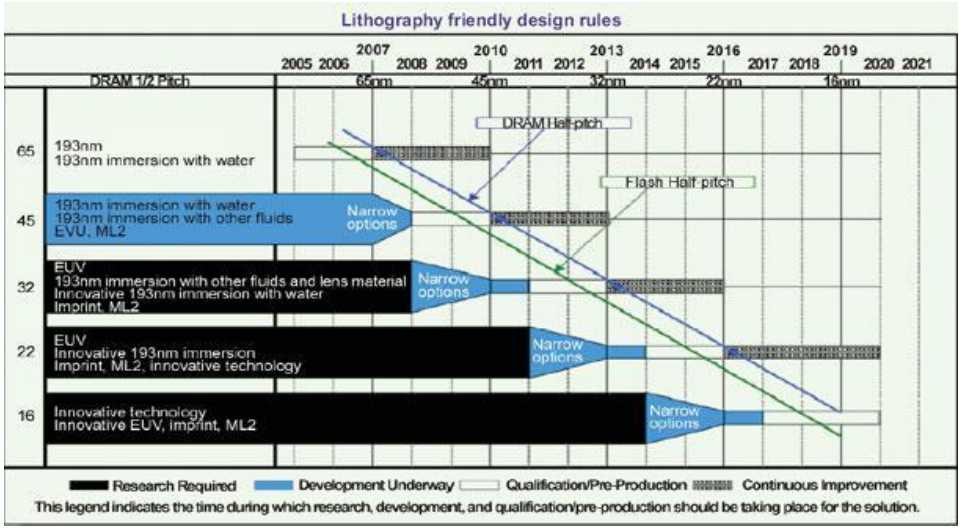
SoC	Model No.	CPU	CPU ISA	Technology	Die size	Date	Devices
N/A	APL0098	Arm11	Armv6	90 nm	N/A	6/2007	iPhone, iPod Touch (1st gen.)
A4	APL0398	Arm Cortex-A8	Armv7	45 nm	53.29 mm^2	3/2010	iPad, iPhone 4, Apple TV (2nd gen.)
A5	APL0498	Arm Cortex-A9	Armv7	45 nm	122.6 mm^2	3/2011	iPad 2, iPhone 4S
APL2498	Arm Cortex-A9	Armv7	32 nm	71.1 mm^2	3/2012	Apple TV (3rd gen.)	
APL7498	Arm Cortex-A9	Armv7	32 nm	37.8 mm^2	3/2013	Apple TV 3	
A5X	APL5498	Arm Cortex-A9	Armv7	45 nm	162.94 mm^2	3/2012	iPad (3rd gen.)
A6	APL0598	Swift	Armv7s	32 nm	96.71 mm^2	9/2012	iPhone 5
A6X	APL5598	Swift	Armv7s	32 nm	123 mm^2	10/2012	iPad (4th gen)
A7	APL0698	Cyclone	Armv8-A	28 nm	102 mm^2	9/2013	iPhone 5S, iPad mini (2nd gen)
APL5698	Cyclone	Armv8-A	28 nm	102 mm^2	10/2013	iPad Air	
A8	APL1011	Cyclone	Armv8-A	20 nm	89 mm^2	9/2014	iPhone 6, iPhone 6 Plus

(Die) Area and Costs

- Rapid advances in process technology are driving forces in design innovation
- ITRS and SIA road maps – make projections of process technology advancements
  - Companies base their products on these projections

ITRS: Int. Tech. Roadmap for Semiconductors

SIA: Semiconductor Industry Association



(Die) Area and Costs

When we increase area, we will more than likely be:

- Increasing complexity of the design
- Increasing the HW design effort
- Increasing power
- Increasing time-to-market
- *Increasing documentation for the product*
- *Increasing the effort to service the system*

SoC Power

Higher is due to higher SoC operating frequency, higher overall capacitance, and larger size.

Power scales indirectly with feature size, as it primarily determines the frequency.

Type	Power/Die	Source/Environment
Cooled high power	70.0 W	Plug - in, chilled
High power	10.0 – 50.0 W	Plug - in, fan
Low power	0.1 – 2.0 W	Rechargeable battery
Very low power	1.0 – 100.0 mW	AA batteries
Extremely low power	1.0 – 100.0 μW	Button battery

Power dissipation, P_{total} is made of dynamic/switching power and static power caused by leakage current

$$P_{\text{total}} = \frac{CV^2 \text{freq}}{2} + I_{\text{leakage}} V$$

Gate delays are roughly proportional to $CV/(V - V_{th})^2$, where V_{th} is the threshold voltage (for logic - level switching) of the transistors.

SoCs and Power

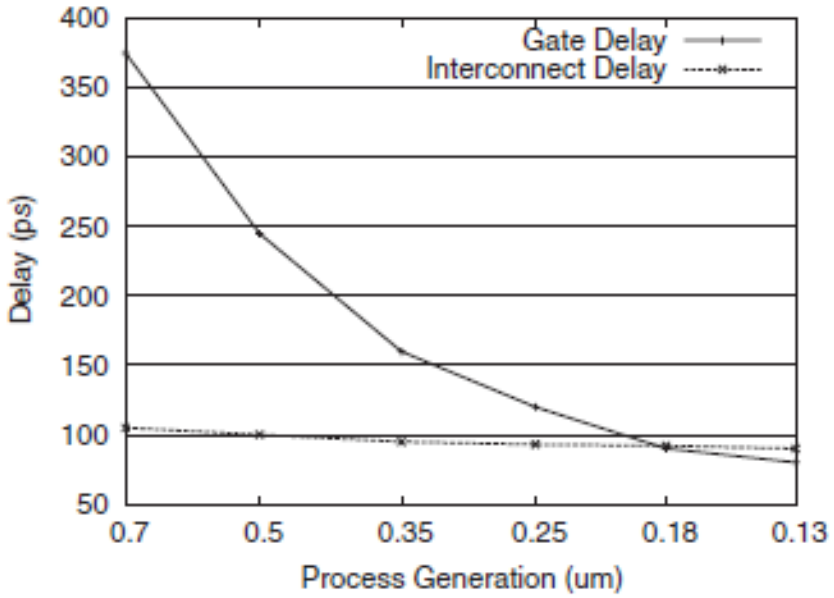
- Especially important in portable electronics, need low power consumption
- However there is a trade-off with respect to performance, power, and the technology node used

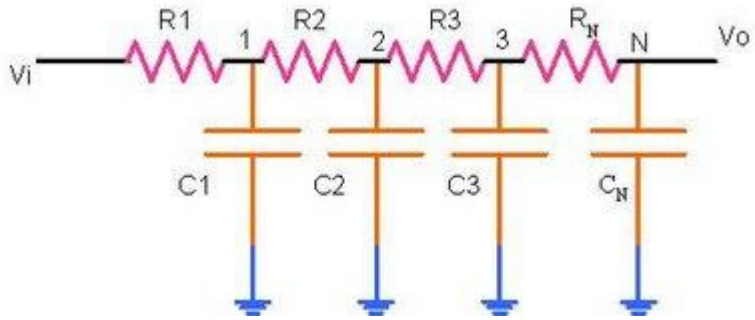
Power and Feature Size

- A feature size decrease results in lower device size.
- Smaller device sizes will reduce the capacitance.
- **Low capacitance decreases the dynamic power consumption and gate delays (both).**
- As device size decreases, the electric field applied becomes destructively large.
- To increase the device reliability, we need to reduce the supply voltage, V.
- **Low Voltage will effectively reduce the dynamic power consumption but results in an increase in gate delays.**
- Gate delays increase can be avoided by reducing, V_{th}
- On the other hand, reducing V_{th} will increase the leakage current and, therefore static power consumption.

SoCs and Power

- Although gate delay scales with technology generation, wire delays do not scale at the same rate





SOC Power and Frequency

Assume V_{th} = 0.6 V; and we reduce voltage by one-half, (3.0 to 1.5 V), Operating frequency is also reduced by half. The total power consumption is 1/8th of the original.

We can optimize an existing design for frequency and modify that design to operate at a lower voltage.

Frequency can be reduced by approximately the cube root of original (dynamic) power:

$$\frac{\text{freq}_1}{\text{freq}_2} = \sqrt[3]{\frac{P_2}{P_1}}.$$

- **Power dissipation (not performance) is the critical issue for SOC applications such as portable ones running on batteries.**

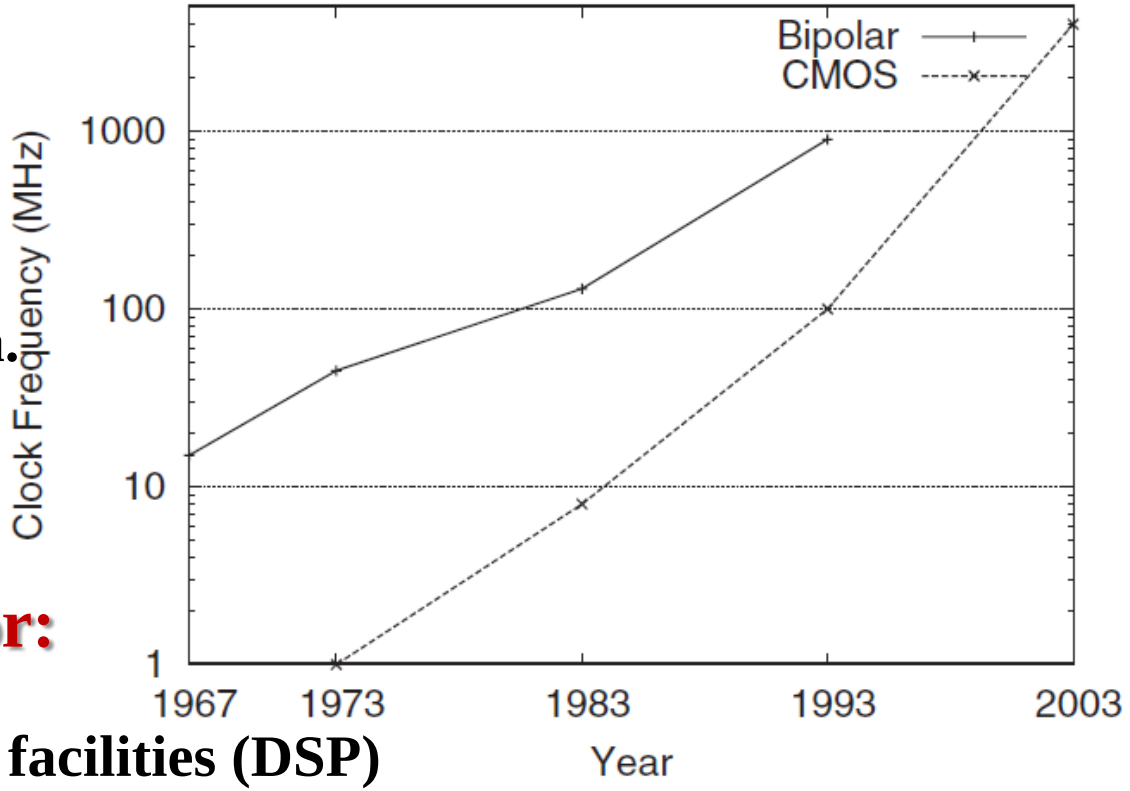
Battery Capacity and Duty Cycle

Type	Energy Capacity (mAh)	Duty Cycle/Lifetime	At Power
Rechargeable	10,000	50 h (10–20% duty)	400 mW–4 W
2 × AA	4000	0.5 year (10–20% duty)	1–10 mW
Button	40	5 years (always on)	1 μW

Area–Time–Power Tradeoff

Workstation Processor:

Designs are high-clock based AC power sources. (excluding Tabs/Laptops)
Cache occupies large die area.
CPU designs are complex (superscalar, multi-core, etc).
Need ample power.



SoC Embedded Processor:

Generally simpler in control
May be complex in execution facilities (DSP)
Area is a factor as well as the design time and power.

A typical DIE CPU-SOC

	Processor on a Chip	SOC
Area used by storage	80% cache	50% ROM/RAM
Clock frequency	3.5 GHz	0.5 GHz
Power	≥50 W	≤10 W
Memory	≥1-GB DRAM	Mostly on-die

SOC Embedded Processors

SOC Implementations have Advantages:

- The requirements are generally known.
- Memory sizes & real-time delay constraints can be easily anticipated.
- Processors can be specialized to do a particular function.
- Clock frequency (power) can be reduced as performance is regained by introducing concurrency (multiple hardware accelerators) in the architecture.

SOC Disadvantages as compared to Processors:

- Available design time/effort and intra-die communications between functional units.
- The market for any specific system is relatively small;
- Huge custom optimization in processor dies is difficult to sustain.
- Off-the-shelf core processor designs are commonly used.
- Specific storage structures can be included on the chip.

Reliability

Known as Dependability and Fault-Tolerance

- **Reliability is related to die area, clock frequency, and power.**
- **Die area increases the amount of circuitry and the probability of a fault.**
- **It also allows the use of error correction and detection techniques.**
- **Higher clock frequencies increase electrical noise and noise sensitivity.**
- **Faster circuits are smaller and more susceptible to radiation.**

Fault-Tolerance: Definition/Design

- ***Failure*** **is a deviation from a design specification.**
- ***Fault*** **is an error that manifests itself as an incorrect result.**
- ***Physical fault*** **is a failure caused by environment: aging, radiation, temperature, etc. The probability of physical faults increases with time.**
- ***Design fault*** **is a failure caused by a bad design.**
  Design faults occur early in the lifetime of a design.

Fault-tolerant designs involve simpler Hardware:

• *Error Detection:* The use of parity, residue, and other codes are essential to reliable system configurations.

• *Action Retry:* Once a fault is detected, the action can be retried to overcome transient errors.

• *Error Correction:* Since most of the system is storage and memory, an ECC can be effective in overcoming storage faults.

• *Reconfiguration:* Once a fault is detected, it may be possible to reconfigure parts of the system so that the failing subsystem is isolated from further computation.

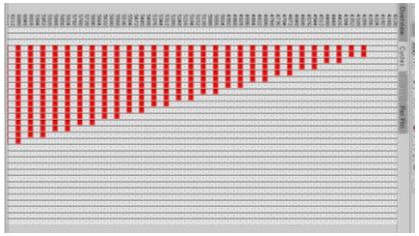
Dealing with Manufacturing Faults

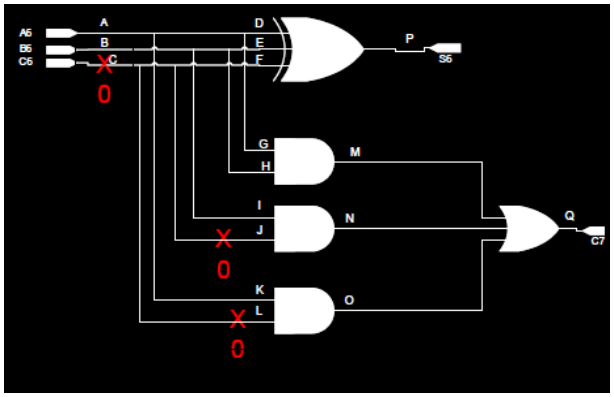
IC Testing for Manufacturing Faults

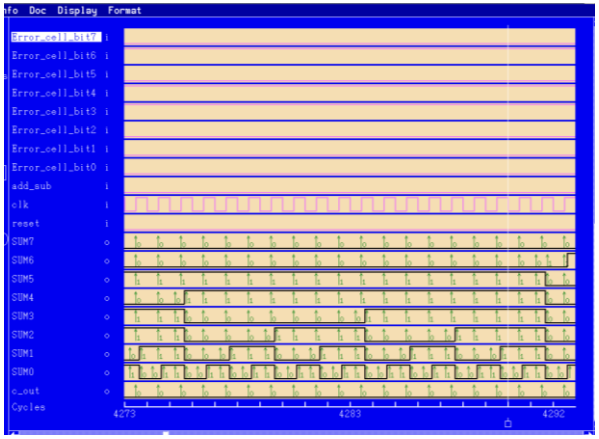
- **Transistor density or overall die transistor count increase leads to the problem of testing increases exponentially.**
- **Without a testing breakthrough, it is estimated that the cost of die testing will exceed the remaining cost of manufacturing.**
- **The hardware designer can help the testing and validation effort, through a process called *design for testability*.**
- **Scan chains require numerous test configurations for large design. Scan is limited in its potential for design validation.**
- **Newer scan techniques compress multiple test patterns and incorporate various *built-in self-test* (BIST) features.**
- ***Scrubbing* is a technique that tests a unit by exercising it when it would otherwise be idle or unavailable. It is most often used with memory - same technique is applied to all hardware units**

Reliability

- Fault-Tolerance in SoCs requires testing the die(s) for manufacturing faults:
  - Built In Self Tests (BIST)
  - Stress tests
  - Scan Chains
  - Scrubbing







Configurability

Reconfigurable designs manage complex high-performance IPs and avoid the risks and delays associated with fabrication.
Three main reasons for using reconfigurable, FPGA devices:

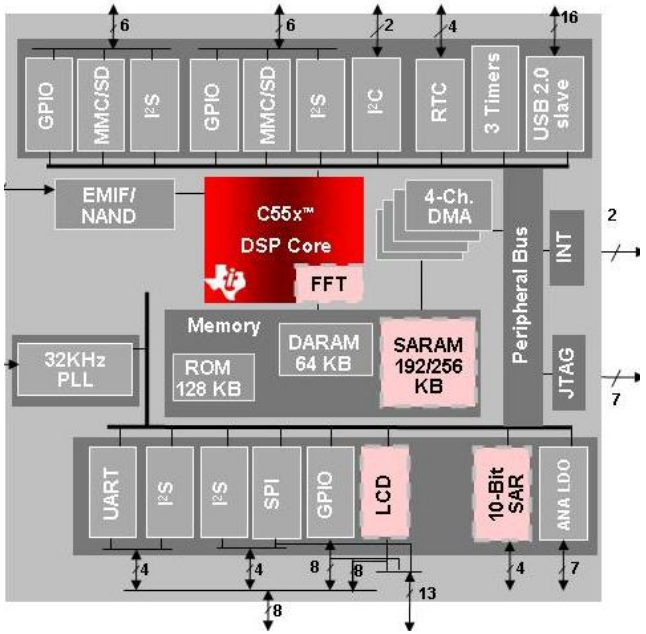
Time: **FPGAs contain large number of registers and support pipelined designs. Instead of running a CPU at a high clock rate, FPGA-based processor at a lower clock can have superior performance by using customized circuits executing in parallel.**
Area: **Regularity of FPGAs use aggressive manufacturing technologies than ASICs.**
Reliability: **Regularity and homogeneity of FPGAs help to introduce redundant cells and interconnections into their architecture.**
Various strategies have been developed to avoid manufacturing or run-time faults by means of such redundant structures.

Configurability

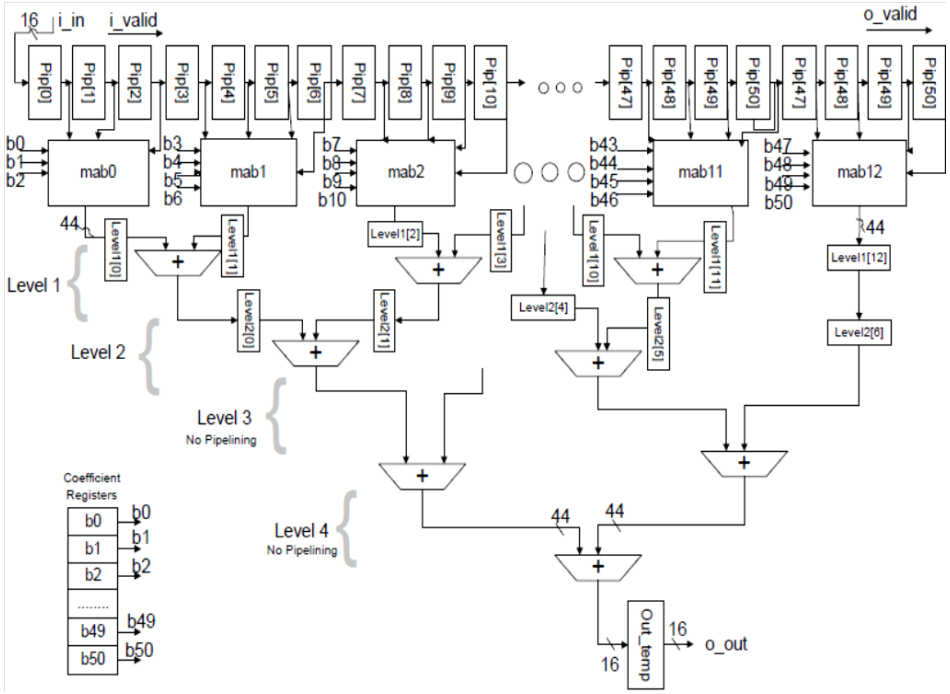
Using FPGAs in design vs ASICs

- **Time** – exceptional performance for highly pipelined and parallel designs
  - FPGAs run at lower frequencies in comparison to CPUs, however their customizability gives higher performance.
- **Area** – Flexibility contributes to fine-grained reconfigurable overhead but higher yield.
  - FPGAs consist of highly regular components which allow for aggressive manufacturing processes.
- **Reliability** – Redundant cells and interconnect make FPGAs more reliable

Configurability



VS



FIR Filter Type	Frequency	Price	Samples/s	Samples/W	Samples/$
DSP (90nm)	120 MHz	$10	$3.67x10^7$	$8.36x10^8$	$3.67x10^6$
DSP (40nm)	150MHz	**$20**	$4.9x10^7$	$2.48x10^9$	$1.65x10^6$
FPGA (40nm)	510.46 MHz	**$500**	**$1.122x10^{10}$**	$1.102x\mathbf{10^{10}}$	$2.244x10^7$