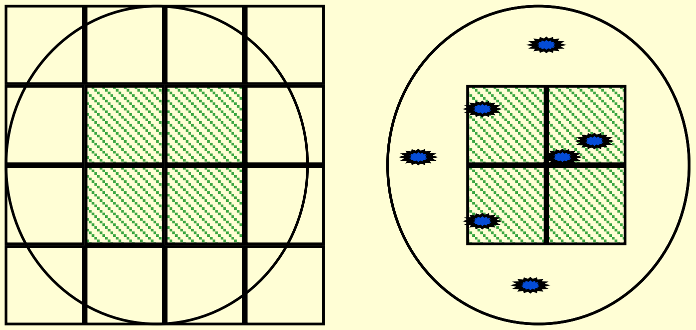
CMSC 611: Advanced Computer Architecture

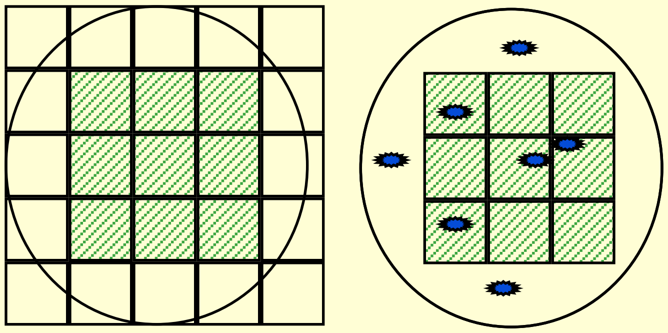
Cost & Performance

Integrated Circuits Costs

$$\text{Dies\_per\_Wafer} = \frac{\pi \times (\text{Wafer\_diameter}/2)^2}{\text{Die\_Area}} - \frac{\pi \times \text{Wafer\_Diameter}}{\sqrt{2 \times \text{Die\_Area}}}$$

$$\text{Die\_Yield} = \text{Wafer\_Yield} \times \left[1 + \frac{\text{Defects\_per\_Unit\_Area} * \text{Die\_Area}}{\alpha}\right]^{-\alpha}$$





$$\text{Die\_Cost} = \frac{\text{Wafer\_Cost}}{\text{Dies\_per\_Wafer} \times \text{Die\_Yield}}$$

Die cost roughly goes with die area^4

$$\text{IC\_Cost} = \frac{\text{Die\_Cost} + \text{Testing\_Cost} + \text{Packing\_Cost}}{\text{Final\_Test\_Yield}}$$

What Affects Cost?

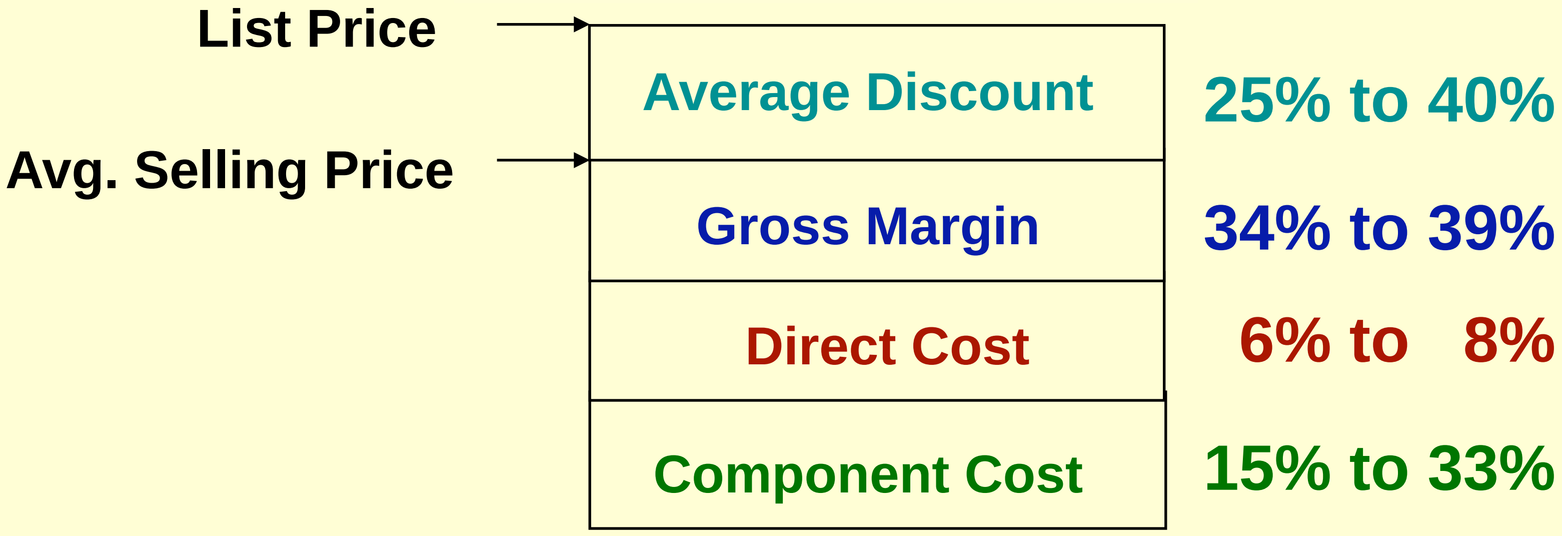
1. Learning curve:
   - The more experience in manufacturing a component, the better the yield
   - In general, a chip, board or system with twice the yield will have half the cost.
   - The learning curve is different for different components, complicating design decisions
2. Volume
   - Larger volume increases rate of learning curve
   - Doubling the volume typically reduce cost by 10%
3. Commodities
   - Are essentially identical products sold by multiple vendors in large volumes
   - Foil the competition and drive the efficiency higher and thus the cost down

Real World Examples

Chip	Layers	Wafer cost	Defect/cm^2	Area (mm^2)	Dies/Wafer	Yield	Die Cost
386DX	2	$900	1.0	43	360	71%	$4
486DX2	3	$1200	1.0	81	181	54%	$12
PowerPC 601	4	$1700	1.3	121	115	28%	$53
HP PA 7100	3	$1300	1.0	196	66	27%	$73
DEC Alpha	3	$1500	1.2	234	53	19%	$149
SuperSPARC	3	$1700	1.6	256	48	13%	$272
Pentium	3	$1500	1.5	296	40	9%	$417

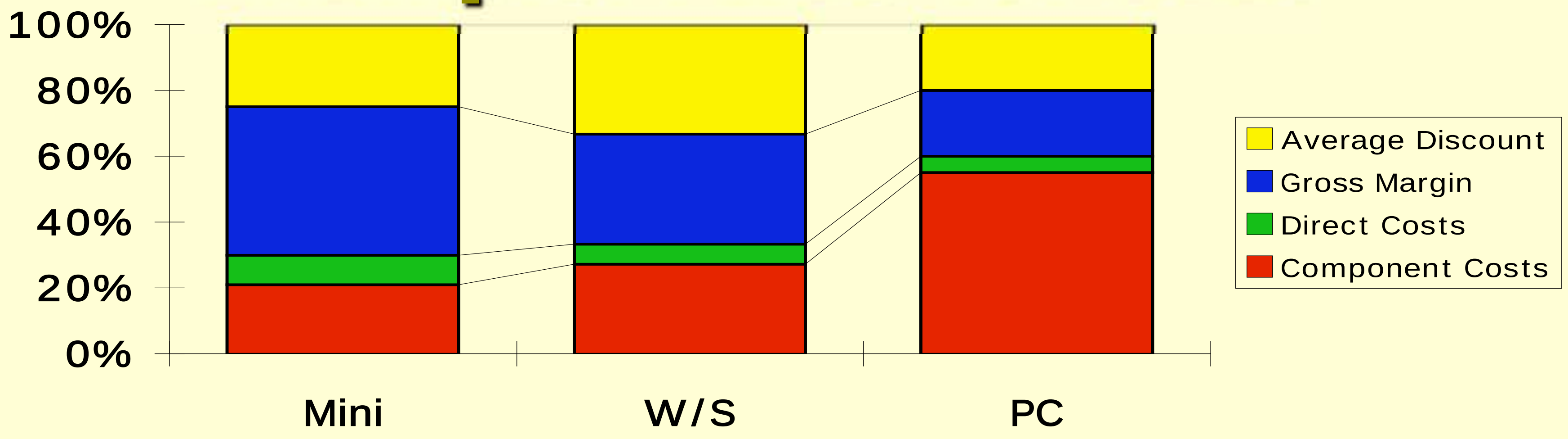
From "Estimating IC Manufacturing Costs," by Linley Gwennap,
***Microprocessor Report*, August 2, 1993, p. 15**

Cost vs. Price



- Component Costs: raw material cost for the system's building blocks
- Direct Costs (add 25% to 40%) recurring costs: labor, purchasing, scrap, warranty
- Gross Margin (add 82% to 186%) nonrecurring costs: R&D, marketing, sales, equipment maintenance, rental, financing cost, pretax profits, taxes
- Average Discount to get List Price (add 33% to 66%): volume discounts and/or retailer markup

Example: Price vs. Cost



Chip Prices (August 1993) for a volume of 10,000 units

Chip	Area (mm^2)	Total Cost	Price	Comment
386DX	43	$9	$31	
486DX2	81	$35	$245	No Competition
PowerPC 601	121	$77	$280	
DEC Alpha	234	$202	$1231	Recoup R&D?
Pentium	296	$473	$965	

Defining Performance

- Performance means different things to different people, therefore its assessment is subtle

Analogy from the airlines industry:

- How to measure performance for an airplane?
  - Cruising speed (How fast it gets to the destination)
  - Flight range (How far it can reach)
  - Passenger capacity (How many passengers it can carry)

Airplane	Passenger capacity	Cruising range (miles)	Cruising speed (m.p.h)	Passenger throughput (Passenger × m.p.h)
Boeing 777	375	4630	610	228,750
Boeing 747	470	4150	610	286,700
BAC/Sud Concorde	132	4000	1350	178,200
Douglas DC-8-50	146	8720	544	79,424

Criteria of performance evaluation differs among users and designers

Performance Metrics

- Response (execution) time:
  - The time between the start and the completion of a task
  - Measures user perception of the system speed
  - Common in reactive and time critical systems, single-user computer, etc.
- Throughput:
  - The total number of tasks done in a given time
  - Most relevant to batch processing (billing, credit card processing)
  - Mainly used for input/output systems (disk access, printer, etc.)

Response-time Metric

- Maximizing performance means minimizing response (execution) time

$$\text{Performance} = \frac{1}{\text{Execution time}}$$

Response-time Metric

$$\text{Performance} = \frac{1}{\text{Execution time}}$$

- Performance of Processor P1 is better than P2 if
  - **For a given work load L**
  - P1 takes less time to execute L than P2

$$\text{Performance}(P_1) > \text{Performance}(P_2) \text{ w.r.t } L$$

$$\Rightarrow \text{Execution time}(P_1, L) < \text{Execution time}(P_2, L)$$

Response-time Metric

$$\text{Performance} = \frac{1}{\text{Execution time}}$$

- Relative performance captures the performance ratio
  - For the same work load

$$\frac{\text{CPU Performance } (P_2)}{\text{CPU Performance } (P_1)} = \frac{\text{Total execution time } (P_1)}{\text{Total execution time } (P_2)}$$

Designer's Performance Metrics

- Users and designers measure performance using different metrics
  - Users: quotable metrics (GHz)
  - Designers: program execution

$$\text{CPU execution time for a program} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$
$$= \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

- Designer focuses on reducing the clock cycle time and the number of cycles per program
- Many techniques to decrease the number of clock cycles also increase the clock cycle time or the average number of cycles per instruction (CPI)

Example

A program runs in 10 seconds on a computer "A" with a 400 MHz clock. We desire a faster computer "B" that could run the program in 6 seconds. The designer has determined that a substantial increase in the clock speed is possible, however it would cause computer "B" to require 1.2 times as many clock cycles as computer "A". What should be the clock rate of computer "B"?

$$\text{CPU time (A)} = \frac{\text{CPU clock cycles}}{\text{Clock rate (A)}}$$

$$10 \text{ seconds} = \frac{\text{CPU clock cycles of program}}{400 \times 10^6 \text{ cycles/second}}$$

$$\begin{aligned}\text{CPU clock cycles of program} &= 10 \text{ seconds} \times 400 \times 10^6 \text{ cycles/second} \\ &= 4000 \times 10^6 \text{ cycles}\end{aligned}$$

To get the clock rate of the faster computer, we use the same formula

$$6 \text{ seconds} = \frac{1.2 \times \text{CPU clock cycles of program}}{\text{clock rate (B)}} = \frac{1.2 \times 4000 \times 10^6 \text{ cycles}}{\text{clock rate (B)}}$$

$$\text{clock rate (B)} = \frac{1.2 \times 4000 \times 10^6 \text{ cycles}}{6 \text{ second}} = 800 \times 10^6 \text{ cycles/second}$$

Calculation of CPU Time

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

Or

$$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$

$$\text{CPU time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

Component of performance	Units of measure
CPU execution time for a program	Seconds for the program
Instruction count	Instructions executed for the program
Clock cycles per instructions (CPI)	Average number of clock cycles/instruction
Clock cycle time	Seconds per clock cycle

CPU Time (Cont.)

- CPU execution time can be measured by running the program
- The clock cycle is usually published by the manufacture
- Measuring the CPI and instruction count is not trivial
  - Instruction counts can be measured by: software profiling, using an architecture simulator, using hardware counters on some architecture
  - The CPI depends on many factors including: processor structure, memory system, the mix of instruction types and the implementation of these instructions

CPU Time (Cont.)

- Designers sometimes uses the following formula:

$$\text{CPU clock cycles} = \sum_{i=1}^{n} CPI_i \times C_i$$

Where: C_i is the count of number of instructions of class *i* executed
CPI_i is the average number of cycles per instruction for that instruction class
n is the number of different instruction classes

Example

Suppose we have two implementation of the same instruction set architecture. Machine "A" has a clock cycle time of 1 ns and a CPI of 2.0 for some program, and machine "B" has a clock cycle time of 2 ns and a CPI of 1.2 for the same program. Which machine is faster for this program and by how much?

Both machines execute the same instructions for the program. Assume the number of instructions is "I",

CPU clock cycles (A) = I $\times$ 2.0 CPU clock cycles (B) = I $\times$ 1.2

The CPU time required for each machine is as follows:

CPU time (A) = CPU clock cycles (A) $\times$ Clock cycle time (A)
= I $\times$ 2.0 $\times$ 1 ns = 2 $\times$ I ns

CPU time (B) = CPU clock cycles (B) $\times$ Clock cycle time (B)
= I $\times$ 1.2 $\times$ 2 ns = 2.4 $\times$ I ns

Therefore machine A will be faster by the following ratio:

$$\frac{\text{CPU Performance (A)}}{\text{CPU Performance (B)}} = \frac{\text{CPU time (B)}}{\text{CPU time (A)}} = \frac{2.4 \times I \text{ ns}}{2 \times I \text{ ns}} = 1.2$$

Comparing Code Segments

A compiler designer is trying to decide between two code sequences for a particular machine. The hardware designers have supplied the following facts:

Instruction class	CPI for this instruction class
A	1
B	2
C	3

For a particular high-level language statement, the compiler writer is considering two code sequences that require the following instruction counts:

Code sequence	Instruction count for instruction class		
	A	B	C
1	2	1	2
2	4	1	1

Which code sequence executes the most instructions? Which will be faster? What is the CPI for each sequence?

Answer:

Sequence 1:	executes 2 + 1 + 2 = 5 instructions	
Sequence 2:	executes 4 + 1 + 1 = 6 instructions	✔

Comparing Code Segments

Using the formula: $$\text{CPU clock cycles} = \sum_{i=1}^{n} CPI_i \times C_i$$

Sequence 1: CPU clock cycles = $(2 \times 1) + (1 \times 2) + (2 \times 3) = 10$ cycles

Sequence 2: CPU clock cycles = $(4 \times 1) + (1 \times 2) + (1 \times 3) = 9$ cycles

☞ Therefore Sequence 2 is faster although it executes more instructions

Using the formula: $$\text{CPI} = \frac{\text{CPU clock cycles}}{\text{Instruction count}}$$

Sequence 1: CPI = 10/5 = 2

Sequence 2: CPI = 9/6 = 1.5

☞ Since Sequence 2 takes fewer overall clock cycles but has more instructions it must have a lower CPI

The Role of Performance

- Hardware performance is a key to the effectiveness of the entire system
- Performance has to be measured and compared to evaluate designs
- To optimize the performance, major affecting factors have to be known
- For different types of applications
  - different performance metrics may be appropriate
  - different aspects of a computer system may be most significant
- Instructions use and implementation, memory hierarchy and I/O handling are among the factors that affect the performance

Calculation of CPU Time

$$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$

	Instr. Count	CPI	Clock Rate
Program	X		
Compiler	X	X	
Instruction Set	X	X	
Organization		X	X
Technology			X

$$\text{CPU clock cycles} = \sum_{i=1}^{n} CPI_i \times C_i$$

Where: C_i is the count of number of instructions of class *i* executed

CPI_i is the average number of cycles per instruction for that instruction class

n is the number of different instruction classes

Important Equations (so far)

$$\text{Performance} = \frac{1}{\text{Execution time}}$$

$$\text{Speedup} = \frac{\text{Performance (B)}}{\text{Performance (A)}} = \frac{\text{Time (A)}}{\text{Time (B)}}$$

$$\text{CPU time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

$$\text{CPU clock cycles} = \sum_{i=1}^{n} \text{CPI}_i \times \text{Instructions}_i$$