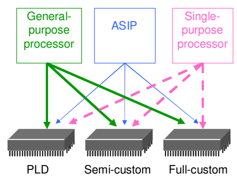
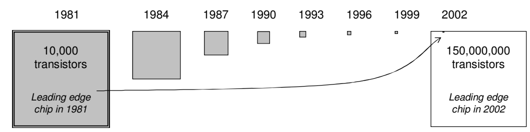
Recall…

A processor (any processor) is a way of working with and transferring data, it can be implemented in any IC technology…



Recall…



- Something that doubles frequently grows more quickly than most people realize!
  - A 2002 chip can hold about 15,000 1981 chips inside itself
  - Intel Itanium: 325 Million Transistors!

Design Technology

- The manner in which we convert our concept of desired system functionality into an implementation
  - Compilation: Takes a high level language and translates it into assembly code
  - Synthesis: Takes a high level language and translates it into gates

  To help us speed things up we have:
  - Libraries of already made functions
  - IP (Intellectual Property) blocks of hardware

Design Technology

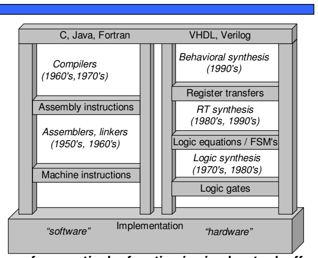
Test and Verification are also more automated now to speed the design process.

Some new trends that are helping speed the design process:

- Well designed standards that help designers move between different tools
- Tools that can synthesize to hardware or compile to assembly using the same language
- Simulators that can simulate software and hardware at the same time

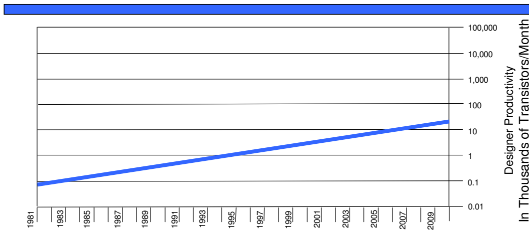
The co-design ladder

- In the past:
  - Hardware and software design technologies were very different
  - Recent maturation of synthesis enables a unified view of hardware and software
- Hardware/software "codesign"



The choice of hardware versus software for a particular function is simply a tradeoff among various design metrics, like performance, power, size, NRE cost, and especially flexibility; there is no fundamental difference between what hardware or software can implement.

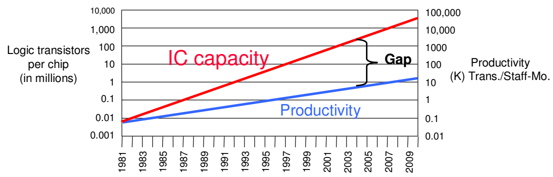
Design productivity exponential increase



- Exponential increase over the past few decades

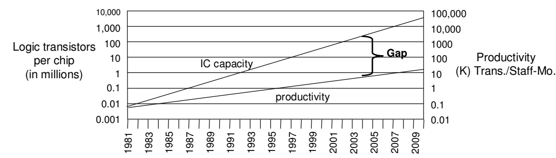
Design productivity gap

- While designer productivity has grown at an impressive rate over the past decades, the rate of improvement has not kept pace with chip capacity



Design productivity gap

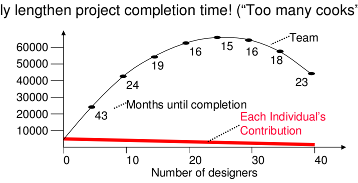
- 1981 leading edge chip required 100 designer months
  - 10,000 transistors / 100 transistors/month
- 2002 leading edge chip requires 30,000 designer months
  - 150,000,000 / 5000 transistors/month
- Designer cost increase from $1M to $300M



The mythical man-month

- The situation is even worse than the productivity gap indicates
- In theory, adding designers to team reduces project completion time
- In reality, productivity per designer decreases due to complexities of team management and communication
- In the software community, known as "the mythical man-month" (Brooks 1975)
- At some point, can actually lengthen project completion time! ("Too many cooks")

- 1M transistors, 1 designer=5000 trans/month
- Each additional designer reduces for 100 trans/month
- So 2 designers produce 4900 trans/month each



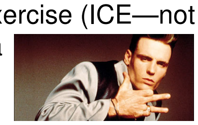
Here is a different look at the same problem:

Project size (in KLOC)	Best programmer (months/KLOC)	Worst (months/KLOC)
1	1	6
8	2.5	7
64	6.5	11
512	17.5	21
2048	30	32

KLOC = 1000 lines of code

source: "The Embedded Muse"; www.ganssle.com

In Class Exercise

- **I**n **C**lass **E**xercise (ICE—not to be confused with Vanilla )
- It will count as part of your score on the next homework, but unlike homework, you get credit if you are at least close.

ICE

- Write a general equation that describes one engineers productivity if each time an engineer is added to the project their productivity is 5% less.
- Use the variables:
  - $\mathbf{P_o}$ = Original productivity
  - $\mathbf{P_n}$ = New productivity
  - **n** = number of engineers on project
- So, to check your equation, if $\mathbf{P_o = 100}$
  - For n = 1, $\mathbf{P_n}$= 100; for n = 2, $\mathbf{P_n}$= 95; for n = 3, $\mathbf{P_n}$= 90.25

ICE

- If a digital camera takes the following steps
  - Acquire picture: 100 ms
  - Convert picture: 300 ms
  - Write picture to memory: 1 s
- What is the latency?
- What is the throughput assuming it cannot take a second picture until it is done writing the first one to memory?
- In VHDL, what is the entity for?

Summary of Chapter 1

- Embedded systems are everywhere
- Key challenge: optimization of design metrics
  - Design metrics compete with one another
- A unified view of hardware and software is necessary to improve productivity
- Three key technologies
  - Processor: general-purpose, application-specific, single-purpose
  - IC: Full-custom, semi-custom, PLD
  - Design: Compilation/synthesis, libraries/IP, test/verification

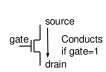
Chapter 2: Custom single-purpose processors

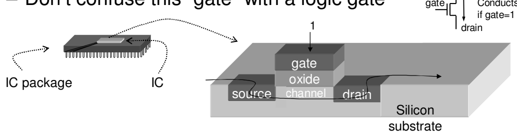
Outline

- Introduction
- Combinational logic
- Sequential logic
- Custom single-purpose processor design
- RT-level custom single-purpose processor design

CMOS transistor on silicon

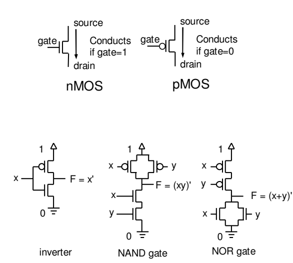
- Transistor
  - The basic electrical component in digital systems
  - Acts as an on/off switch
  - Voltage at "gate" controls whether current flows from source to drain
  - Don't confuse this "gate" with a logic gate



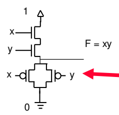


CMOS transistor implementations

- Complementary Metal Oxide Semiconductor
- We refer to logic levels
  - Typically 0 is 0V, 1 is 5V
- Two basic CMOS types
  - nMOS conducts if gate=1
  - pMOS conducts if gate=0
  - Hence "complementary"
- Basic gates
  - Inverter, NAND, NOR

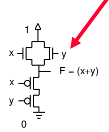


Additions…

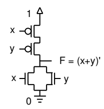


AND gate

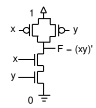
These work in theory, but as it turns out, NMOS transistors are good for connecting to ground, but not so good at connecting to positive voltage. PMOS is the other way around. So, in actual implementation, NAND's NOR's are preferred.



OR gate

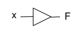


NOR gate



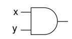
NAND gate

Basic logic gates



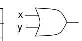
F = x
Driver

x	F
0	0
1	1



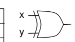
F = x y
AND

x	y	F
0	0	0
0	1	0
1	0	0
1	1	1



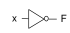
F = x + y
OR

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1



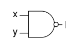
F = x ⊕ y
XOR

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0



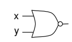
F = x'
Inverter

x	F
0	1
1	0



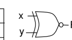
F = (x y)'
NAND

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0



F = (x+y)'
NOR

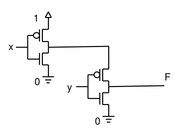
x	y	F
0	0	1
0	1	0
1	0	0
1	1	0



F = x ⊙ y
XNOR

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

Final ICE! (for today)



Fill in the missing outputs:

x	y	F
0	0	
0	1	
1	0	
1	1	

To do before Thursday…

- Sign up for e-mail list if you haven't already.
  – Remember www.cs.ucr.edu/cs120B
- Go to lab at 11AM or 3PM