

inst.eecs.berkeley.edu/~cs61c
CS61C : Machine Structure
Lecture 23 – Introduction to Synchronous Digital Systems (SDS)
Switches, Transistors, Gates



Senior Lecturer SOE Dan Garcia
 www.cs.berkeley.edu/~ddgarcia

Password Ring? ⇒ Google wants to “put an end to all passwords”; a small USB key (with code inside, not a password) that would provide credentials, also would work with mobile devices.



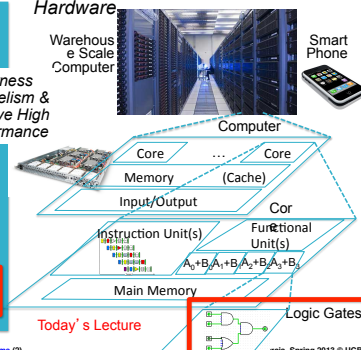
www.technologyreview.com/news/512051/google-wants-to-replace-all-your-passwords-with-a-ring

New-School Machine Structures (It’s a bit more complicated!)

Software | Hardware

- Parallel Requests**
Assigned to computer e.g., Search “Garcia”
- Parallel Threads**
Assigned to core e.g., Lookup, Ads
- Parallel Instructions**
>1 instruction @ one time e.g., 5 pipelined instructions
- Parallel Data**
>1 data item @ one time e.g., Add of 4 pairs of words
- Hardware descriptions**
All gates @ one time

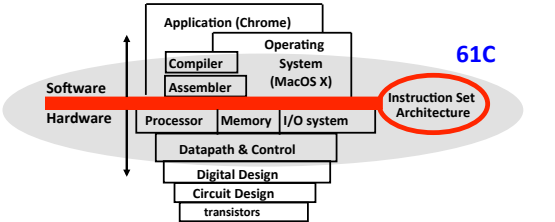
Harness Parallelism & Achieve High Performance



Today’s Lecture

CS61C L23 Synchronous Digital Systems (2) Garcia, Spring 2013 © UCB

What is Machine Structures?



61C

Coordination of many levels of abstraction
 ISA is an important abstraction level:
 contract between HW & SW

CS61C L23 Synchronous Digital Systems (3) Garcia, Spring 2013 © UCB

Levels of Representation/ Interpretation

High Level Language Program (e.g., C)

Compiler

Assembly Language Program (e.g., MIPS)

Assembler

Machine Language Program (MIPS)

```

temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;

lw $t0, 0($2)
lw $t1, 4($2)
sw $t1, 0($2)
sw $t0, 4($2)
  
```

Anything can be represented as a number, i.e., data or instructions

0000 1001 1100 0110 1010 1111 0101 1000
 1010 1111 0101 1000 0000 1001 1100 0110
 1100 0110 1010 1111 0101 1000 0000 1001
 0101 1000 0000 1001 1100 0110 1010 1111

Machine Interpretation

Hardware Architecture Description (e.g., block diagrams)

Register File

ALU

Architecture Implementation

Logic Circuit Description (Circuit Schematic Diagrams)

CS61C L23 Synchronous Digital Systems (4) Garcia, Spring 2013 © UCB

Synchronous Digital Systems

Hardware of a processor, such as the MIPS, is an example of a Synchronous Digital System

Synchronous:

- All operations coordinated by a central clock
 - “Heartbeat” of the system!

Digital:

- All values represented by discrete values
- Electrical signals are treated as 1s and 0s; grouped together to form words

CS61C L23 Synchronous Digital Systems (5) Garcia, Spring 2013 © UCB

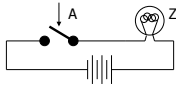
Logic Design

- Next several weeks: we’ll study how a modern processor is built; starting with basic elements as building blocks
- Why study hardware design?
 - Understand capabilities and limitations of hw in general and processors in particular
 - What processors can do fast and what they can’t do fast (avoid slow things if you want your code to run fast!)
 - Background for more in depth hw courses (CS 150, CS 152)
 - There is just so much you can do with standard processors: you may need to design own custom hw for extra performance

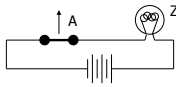
CS61C L23 Synchronous Digital Systems (6) Garcia, Spring 2013 © UCB

Switches: Basic Element of Physical Implementations

- Implementing a simple circuit (arrow shows action if wire changes to "1"):



Close switch (if A is "1" or asserted) and turn on light bulb (Z)



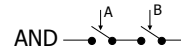
Open switch (if A is "0" or unasserted) and turn off light bulb (Z)

$$Z = A$$

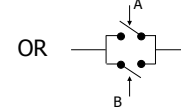


Switches (cont' d)

- Compose switches into more complex ones (Boolean functions):



$$Z = A \text{ and } B$$



$$Z = A \text{ or } B$$



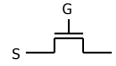
Transistor Networks

- Modern digital systems designed in CMOS
 - MOS: Metal-Oxide on Semiconductor
 - C for complementary: normally-open and normally-closed switches
- MOS transistors act as voltage-controlled switches

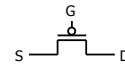


<http://youtu.be/zALiciesOU> MOS Transistors

- Three terminals: drain, gate, and source
 - Switch action: if voltage on gate terminal is (some amount) higher/lower than source terminal then conducting path established between drain and source terminals



n-channel
open when voltage at G is low
closes when:
 $\text{voltage}(G) > \text{voltage}(S) + \epsilon$

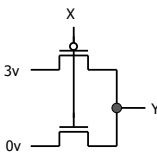


p-channel
closed when voltage at G is low
opens when:
 $\text{voltage}(G) < \text{voltage}(S) - \epsilon$



MOS Networks

"1" (voltage source)



"0" (ground)

what is the relationship between x and y?

x	y
0 volts	
3 volts	



Transistor Circuit Rep. vs. Block diagram

- Chips are composed of nothing but transistors and wires.
- Small groups of transistors form useful building blocks.



"1" (voltage source)

"0" (ground)

a	b	c
0	0	1
0	1	1
1	0	1
1	1	0

- Block are organized in a hierarchy to build higher-level blocks: ex: adders. (You can build AND, OR, NOT out of NAND!)



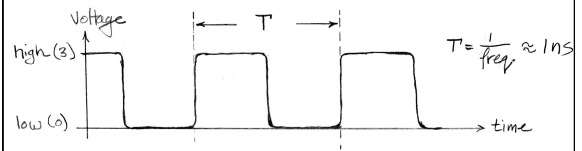
How many hours h on Project 2a?

- a) $0 \leq h < 5$
- b) $5 \leq h < 10$
- c) $10 \leq h < 15$
- d) $15 \leq h < 20$
- e) $20 \leq h$

Other administrivia?



Signals and Waveforms: Clocks

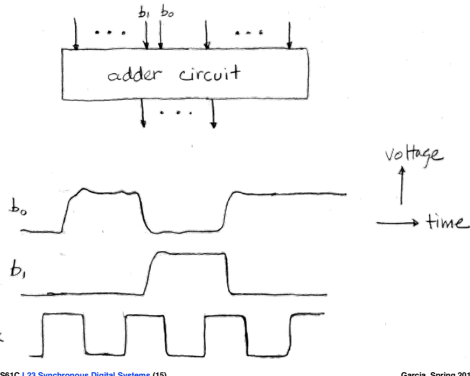


• Signals

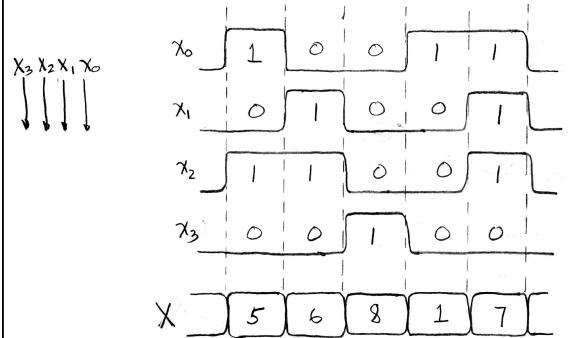
- When **digital** is only treated as 1 or 0
- Is transmitted over wires continuously
- Transmission is effectively instant
 - Implies that any wire only contains 1 value at a time



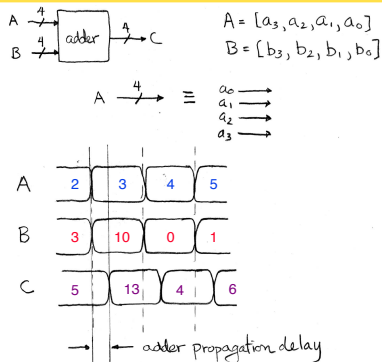
Signals and Waveforms



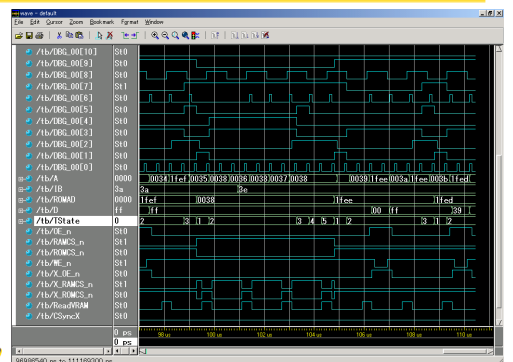
Signals and Waveforms: Grouping



Signals and Waveforms: Circuit Delay



Sample Debugging Waveform



Type of Circuits

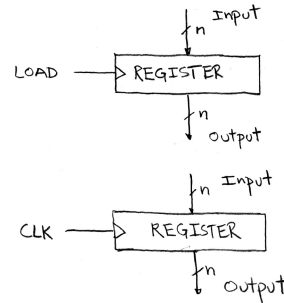
- Synchronous Digital Systems are made up of two basic types of circuits:
- **Combinational Logic (CL) circuits**
 - Our previous adder circuit is an example.
 - **Output is a function of the inputs only.**
 - Similar to a pure function in mathematics, $y = f(x)$. (No way to store information from one invocation to the next. No side effects)
- **State Elements:** circuits that store information.



CS61C L23 Synchronous Digital Systems (19)

Garcia, Spring 2013 © UCB

Circuits with STATE (e.g., register)



CS61C L23 Synchronous Digital Systems (20)

Garcia, Spring 2013 © UCB

Peer Instruction

- 1) SW **can peek** at HW (past ISA abstraction boundary) for optimizations
- 2) SW **can depend** on particular HW implementation of ISA

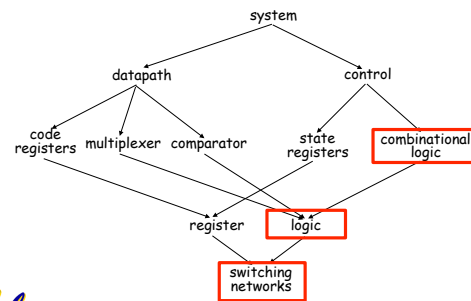
- | | |
|----|----|
| | 12 |
| a) | FF |
| b) | FT |
| c) | TF |
| d) | TT |



CS61C L23 Synchronous Digital Systems (21)

Garcia, Spring 2013 © UCB

Design Hierarchy



CS61C L23 Synchronous Digital Systems (22)

Garcia, Spring 2013 © UCB

And in conclusion...

- ISA is very important abstraction layer
 - Contract between HW and SW
- Clocks control pulse of our circuits
- Voltages are analog, quantized to 0/1
- Circuit delays are fact of life
- Two types of circuits:
 - Stateless Combinational Logic (&,!,~)
 - State circuits (e.g., registers)



CS61C L23 Synchronous Digital Systems (23)

Garcia, Spring 2013 © UCB