

CS 61C: Great Ideas in Computer Architecture (Machine Structures)
Muxes, Adders, and ALUs

Instructors:
 Randy H. Katz
 David A. Patterson
<http://inst.eecs.Berkeley.edu/~cs61c/fa10>

10/25/10 Fall 2010 - Lecture #24 1

Agenda

- Multiplexer
- Administrivia
- Technology Break
- ALU Design

10/25/10 Fall 2010 - Lecture #24 2

Agenda

- Multiplexer
- Administrivia
- Technology Break
- ALU Design

10/25/10 Fall 2010 - Lecture #24 3

Data Multiplexer (e.g., 2-to-1 x n-bit-wide)

10/25/10 Fall 2010 - Lecture #24 4

N Instances of 1-bit-Wide Mux

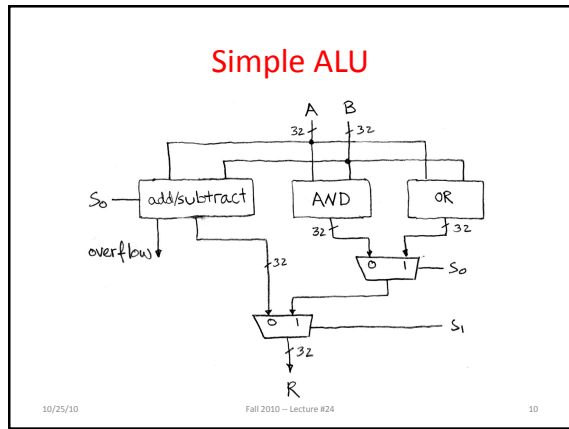
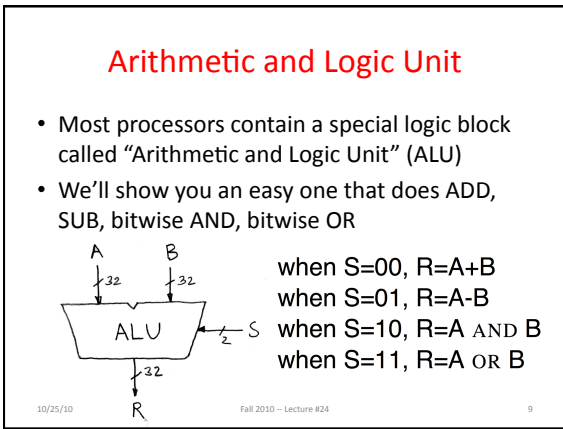
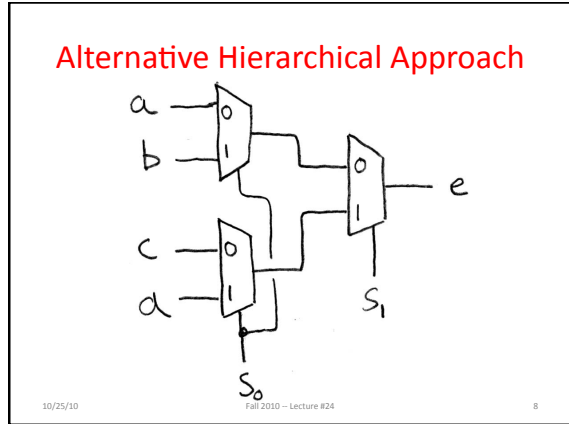
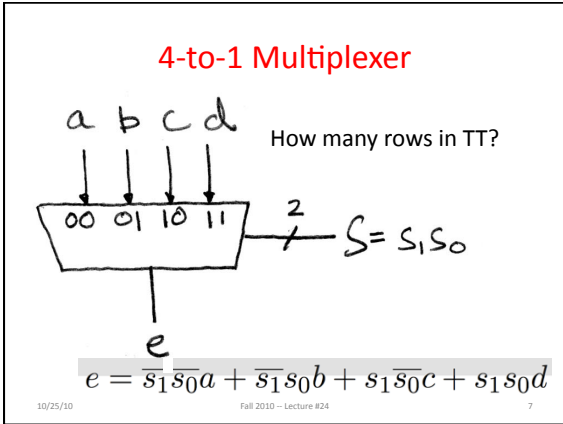
How many rows in TT?

$$\begin{aligned}
 c &= \bar{s}a\bar{b} + \bar{s}ab + s\bar{a}b + sab \\
 &= \bar{s}(a\bar{b} + ab) + s(\bar{a}b + ab) \\
 &= \bar{s}(a(\bar{b} + b)) + s((\bar{a} + a)b) \\
 &= \bar{s}(a(1)) + s((1)b) \\
 &= \bar{s}a + sb
 \end{aligned}$$

10/25/10 Fall 2010 - Lecture #24 5

How Do We Build a 1-bit-Wide Mux?

10/25/10 Fall 2010 - Lecture #24 6



- ### Agenda
- Multiplexers
 - Administrivia
 - Technology Break
 - ALU
- 10/25/10 Fall 2010 - Lecture #24 11

- ### Agenda
- Mux + Adder Design
 - Administrivia
 - Technology Break
 - ALU Design
- 10/25/10 Fall 2010 - Lecture #24 12

Agenda

- Multiplexer
- Administrivia
- Technology Break
- ALU Design

10/25/10

Fall 2010 - Lecture #24

13

Adder/Subtractor: One-bit adder Least Significant Bit

	a_3	a_2	a_1	a_0
+	b_3	b_2	b_1	b_0
	s_3	s_2	s_1	s_0

a_0	b_0	s_0	c_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$s_0 = a_0$$

$$c_1 = a_0$$

10/25/10

Fall 2010 - Lecture #24

14

Adder/Subtractor: One-bit adder (1/2) ...

	a_3	a_2	a_1	a_0
+	b_3	b_2	b_1	b_0
	s_3	s_2	s_1	s_0

a_i	b_i	c_i	s_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$s_i =$$

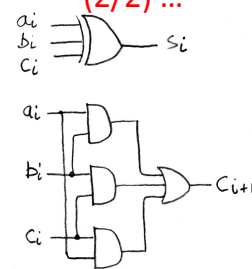
$$c_{i+1} =$$

10/25/10

Fall 2010 - Lecture #24

15

Adder/Subtractor: One-bit Adder (2/2) ...



$$s_i = \text{XOR}(a_i, b_i, c_i)$$

$$c_{i+1} = \text{MAJ}(a_i, b_i, c_i) = a_i b_i + a_i c_i + b_i c_i$$

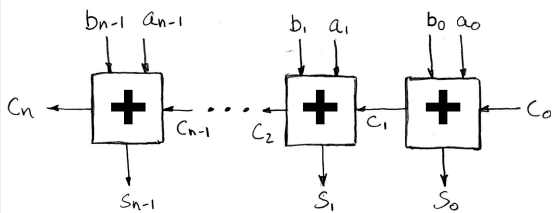
10/25/10

Fall 2010 - Lecture #24

16

N x 1-bit Adders \Rightarrow 1 N-bit Adder

Connect Carry Out $i-1$ to Carry in i :



10/25/10

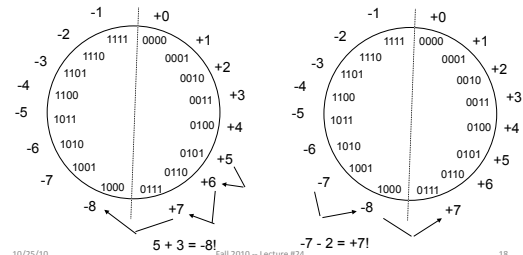
Fall 2010 - Lecture #24

17

Overflow Conditions

Add two positive numbers to get a negative number

or two negative numbers to get a positive number



10/25/10

Fall 2010 - Lecture #24

18

Overflow Conditions

5	$\begin{array}{r} 0111 \\ 0101 \\ \hline \end{array}$	-7	$\begin{array}{r} 1000 \\ 1001 \\ \hline \end{array}$
3	$\begin{array}{r} 0011 \\ \hline \end{array}$	-2	$\begin{array}{r} 1100 \\ \hline \end{array}$
-8	$\begin{array}{r} 1000 \\ \hline \end{array}$	7	$\begin{array}{r} 10111 \\ \hline \end{array}$

Overflow

Overflow

5	$\begin{array}{r} 0000 \\ 0101 \\ \hline \end{array}$
2	$\begin{array}{r} 0010 \\ \hline \end{array}$
7	$\begin{array}{r} 0111 \\ \hline \end{array}$

No overflow

-3	$\begin{array}{r} 1111 \\ 1101 \\ \hline \end{array}$
-5	$\begin{array}{r} 1011 \\ \hline \end{array}$
-8	$\begin{array}{r} 11000 \\ \hline \end{array}$

No overflow

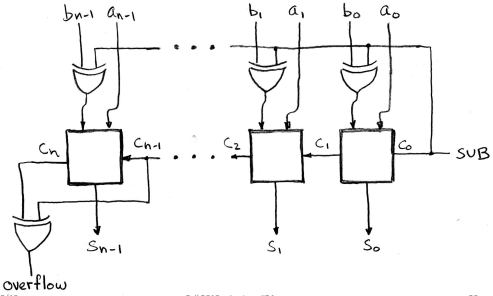
Overflow when carry in to sign does not equal carry out: $C_n \text{ xor } C_{n-1}$

10/25/10

Fall 2010 -- Lecture #24

19

Twos Complement Adder/Subtractor

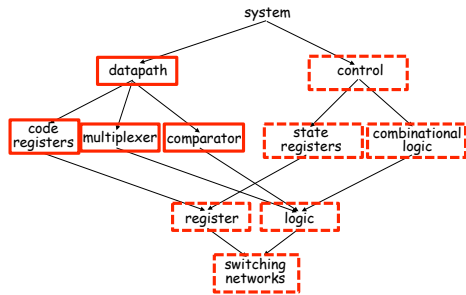


10/25/10

Fall 2010 -- Lecture #24

20

Design Hierarchy



10/25/10

Fall 2010 -- Lecture #24

21

Summary

- Use muxes to select among input
 - S input bits selects 2^S inputs
 - Each input can be n-bits wide, indep of S
- Can implement muxes hierarchically
- ALU can be implemented using a mux
 - Coupled with basic block elements
- N-bit adder-subtractor done using N 1-bit adders with XOR gates on input
 - XOR serves as conditional inverter

10/25/10

Fall 2010 -- Lecture #24

22