

Chapter 3 Digital Logic Structures

Original slides from Gregory Byrd, North Carolina State University
Modified slides by Chris Wilcox, Colorado State University

The diagram on the left shows a hierarchy of abstraction. At the bottom is a logic gate symbol. An arrow points up to a CPU chip icon. Another arrow points up to a more complex system icon containing a CPU, memory, and other components.

Computing Layers

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Problems

Algorithms

Language

Instruction Set Architecture

Microarchitecture

Circuits ←

Devices

CS270 - Spring Semester 2014 2

The diagram on the left is identical to the one in the first slide, showing the progression from logic gates to a CPU to a full system.

Combinatorial Logic

◆ Cascading set of logic gates

The circuit diagram shows three inputs: A, B, and C. Input A goes to an AND gate with input B, producing output W. Input B goes to an AND gate with input C, producing output Y. Input C goes to an OR gate with input B, producing output X. Output X goes to an OR gate with input W, producing output Z. There is also a NOT gate connected to the output of the first AND gate (W).

What is the truth table?

CS270 - Spring Semester 2014 3

Truth Table (from circuit)

◆ Truth table for circuit on previous slide

A	B	C	W	X	Y	Z
0	0	0	0	0	0	1
0	0	1	0	1	1	1
0	1	0	0	1	1	1
0	1	1	0	1	1	1
1	0	0	0	0	0	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	1	1	0	0

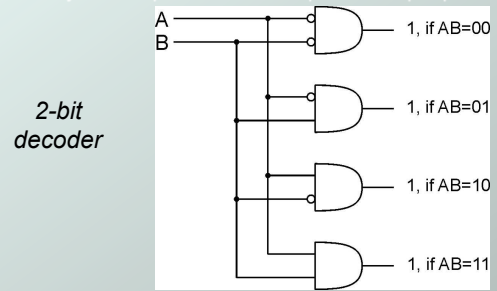
CS270 - Spring Semester 2014 4

Logisim Simulator

- ◆ Logic simulator: allows interactive design and layout of circuits with AND, OR, and NOT gates
- ◆ Simulator web page (linked on class web page) <http://ozark.hendrix.edu/~burch/logisi>
- ◆ Overview, tutorial, downloads, etc.
- ◆ Windows or Linux operating systems
- ◆ Logisim demonstration

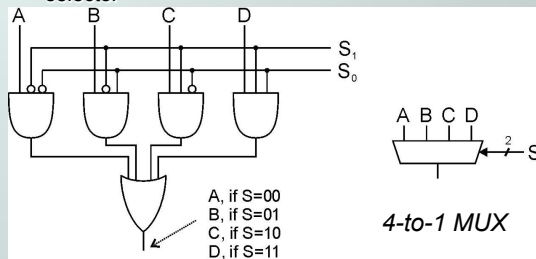
Decoder

- ◆ n inputs, 2^n outputs
 - exactly one output is 1 for each possible input pattern



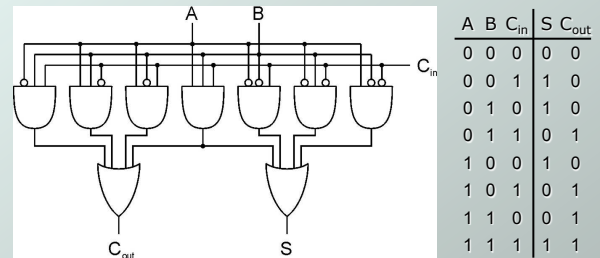
Multiplexer (MUX)

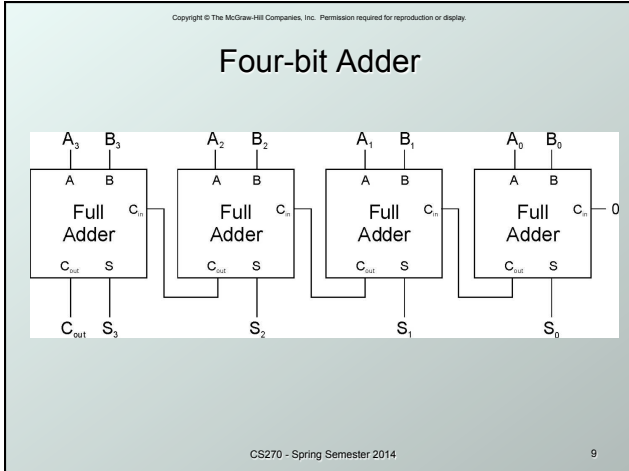
- ◆ n -bit selector and 2^n inputs, one output
 - output equals one of the inputs, depending on selector



Full Adder

- ◆ Add two bits and carry-in, produce one-bit sum and carry-out.





Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Logical Completeness

- Can implement **ANY** truth table with combo of AND, OR, NOT gates.

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

- AND combinations that yield a "1" in the truth table.
- OR the results of the AND gates.

CS270 - Spring Semester 2014 10

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Truth Table (to circuit)

- How do we design a circuit for this?

A	B	C	X	Y
0	0	0	1	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

CS270 - Spring Semester 2014 11

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Programmable Logic Array

- Front end is a input decode
- Back end selects outputs
- Not necessarily minimal circuit!
- Logic arrays are prebuilt

CS270 - Spring Semester 2014 12

Looking Ahead: C Structures

- Useful for data structures

```
struct student
{
    char *lastName;
    char *firstName;
    Date birthDate;
    ...
};
struct student s;
s.lastname = (char *)malloc(80);
strcpy(s.lastname, "Smith");
```

Looking Ahead: Dynamic Memory

- Static versus dynamic memory allocation:

```
// static allocation
char name[80];
strcpy(name, "Smith");
printf("Name: %s\n", name);

// dynamic allocation
char *name = (char *)malloc(80);
strcpy(name, "Smith");
printf("Name: %s\n", name);
free(name);
```

Looking Ahead: String Tokens

- How to extract tokens from a string:

```
char *token = strtok(string, " \t");
while (token != null)
{
    tokens[numTokens] = (char *)
        malloc(strlen(token)+1);
    strcpy(tokens[numTokens], token);
    token = strtok(NULL, " \t");
    numTokens++;
}
```