

## Some Common Combinational Circuits

MUX (Multiplexor)  
Decoder  
Adder

CompOrg Fall 2002 - Combinational Circuits

1

---

---

---

---

---

---

---

---

## Some commonly used components

- Decoders:  $n$  inputs,  $2^n$  outputs.  
– *the inputs are used to select which output is turned on. At any time exactly one output is on.*
- Multiplexors:  $2^n$  inputs,  $n$  selection bits, 1 output.  
– *the selection bits determine which input will become the output.*
- Adder:  $2n$  inputs,  $2n$  outputs.  
– *Computer Arithmetic.*

CompOrg Fall 2002 - Combinational Circuits

2

---

---

---

---

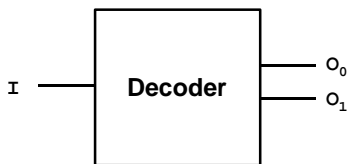
---

---

---

---

## 1 input Decoder



Treat  $I$  as a 1 bit integer  $i$ . The  $i^{\text{th}}$  output will be turned on ( $O_i=1$ ), the other one off.

CompOrg Fall 2002 - Combinational Circuits

3

---

---

---

---

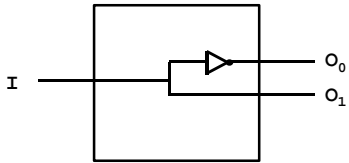
---

---

---

---

### 1 input Decoder



CompOrg Fall 2002 - Combinational Circuits

4

---

---

---

---

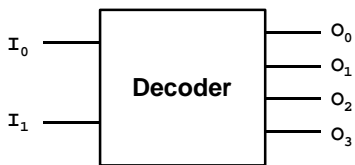
---

---

---

---

### 2 input Decoder



Treat  $I_0I_1$  as a 2 bit integer  $i$ . The  $i^{\text{th}}$  output will be turned on ( $O_i=1$ ), all the others off.

CompOrg Fall 2002 - Combinational Circuits

5

---

---

---

---

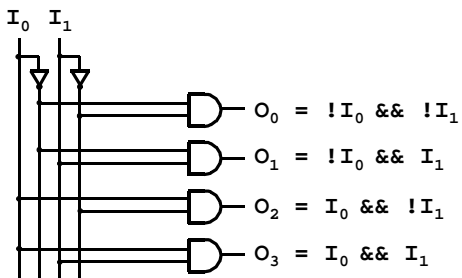
---

---

---

---

### 2 input Decoder



CompOrg Fall 2002 - Combinational Circuits

6

---

---

---

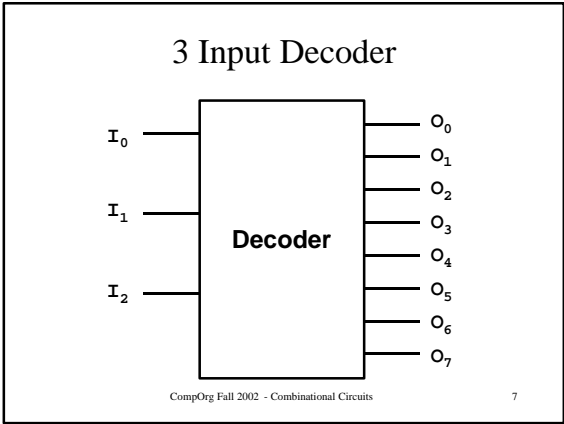
---

---

---

---

---




---

---

---

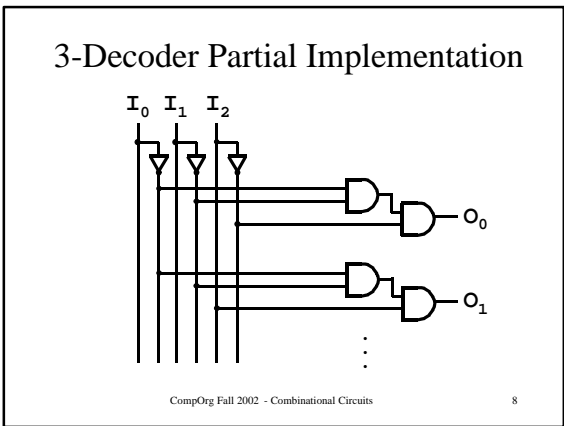
---

---

---

---

---




---

---

---

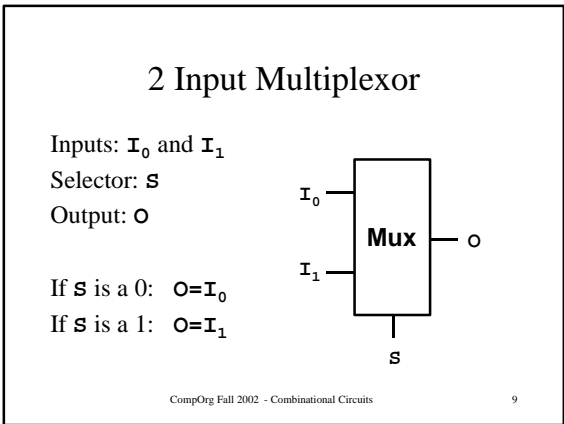
---

---

---

---

---




---

---

---

---

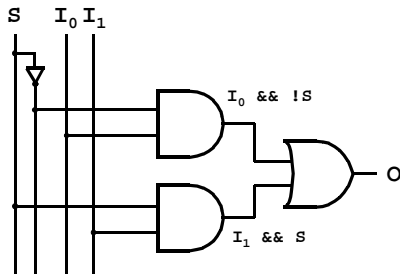
---

---

---

---

## 2-Mux Logic Design



CompOrg Fall 2002 - Combinational Circuits

10

---

---

---

---

---

---

---

---

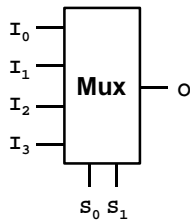
## 4 Input Multiplexor

Inputs:  $I_0$   $I_1$   $I_2$   $I_3$

Selectors:  $S_0$   $S_1$

Output:  $O$

$S_0$	$S_1$	$O$
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$



CompOrg Fall 2002 - Combinational Circuits

11

---

---

---

---

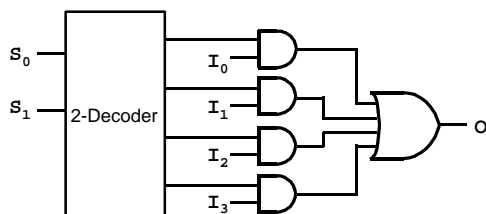
---

---

---

---

## One Possible 4-Mux



CompOrg Fall 2002 - Combinational Circuits

12

---

---

---

---

---

---

---

---

## Adder

- We want to build a box that can add two 32 bit numbers.
  - Assume 2s complement representation
- We can start by building a *1 bit adder*.

CompOrg Fall 2002 - Combinational Circuits

13

---

---

---

---

---

---

---

---

## Addition

- We need to build a 1 bit *adder*
  - compute binary addition of 2 bits.
- We already know that the result is 2 bits.

A	B	O <sub>0</sub>	O <sub>1</sub>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

This is addition!  $\begin{array}{r} A \\ + B \\ \hline O_0 O_1 \end{array}$

CompOrg Fall 2002 - Combinational Circuits

14

---

---

---

---

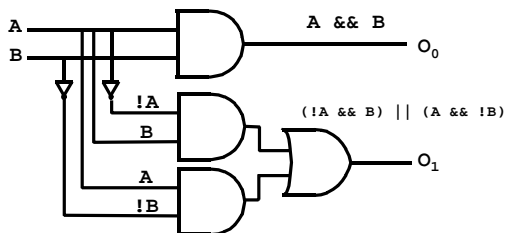
---

---

---

---

## One Implementation



CompOrg Fall 2002 - Combinational Circuits

15

---

---

---

---

---

---

---

---

## Binary addition and our *adder*

$$\begin{array}{r}
 \phantom{0}1 \phantom{0}1 \leftarrow \text{Carry} \\
 01001 \\
 + \underline{01101} \\
 10110
 \end{array}$$

What we really want is something that can be used to implement the binary addition algorithm.

- $O_0$  is the *carry*
- $O_1$  is the *sum*

---

---

---

---

---

---

---

---

## What about the second column?

$$\begin{array}{r}
 \phantom{0}1 \phantom{0}1 \leftarrow \text{Carry} \\
 01001 \\
 + \underline{01101} \\
 10110
 \end{array}$$

- We are adding 3 bits
  - new bit is the *carry* from the first column.
  - The output is still 2 bits, a *sum* and a *carry*

---

---

---

---

---

---

---

---

## Truth Table for Addition

A	B	Carry In	Carry Out	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

---

---

---

---

---

---

---

---

## 1 bit adder (3 inputs!)

- We can come up with a logic design:

```
Carry Out = (A&&B) || (A&&CarryIn) || (B&&CarryIn)
```

```
Sum = (!A && !B && CarryIn) ||  
      (!A && B && !CarryIn) ||  
      ( A && !B && !CarryIn) ||  
      ( A && B && CarryIn)
```

CompOrg Fall 2002 - Combinational Circuits

19

---

---

---

---

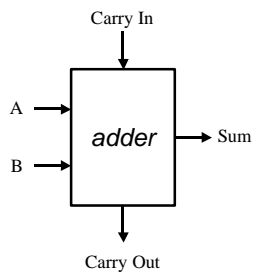
---

---

---

---

## New Component: 1 Bit Adder



CompOrg Fall 2002 - Combinational Circuits

20

---

---

---

---

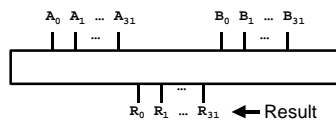
---

---

---

---

## Building a 32 bit Adder



- 64 inputs
- 32 bit output

CompOrg Fall 2002 - Combinational Circuits

21

---

---

---

---

---

---

---

---

**4 Bit Ripple Carry Adder**

$$\begin{array}{r}
 C_3 \ C_2 \ C_1 \ C_0 \\
 A_3 A_2 A_1 A_0 \\
 + B_3 B_2 B_1 B_0 \\
 \hline
 S_3 S_2 S_1 S_0
 \end{array}$$

CompOrg Fall 2002 - Combinational Circuits 22

---

---

---

---

---

---

---

---

**4 Bit Ripple Carry Adder**

$$\begin{array}{r}
 010 \\
 0011 \\
 + 0010 \\
 \hline
 0101
 \end{array}$$

CompOrg Fall 2002 - Combinational Circuits 23

---

---

---

---

---

---

---

---

### Subtraction

- Compute  $A-B$  as  $A + (-B-1) + 1$
- $-B-1$  is just all the bits of  $B$  inverted.
- Add the  $+1$  by setting  $C_0$  to 1

CompOrg Fall 2002 - Combinational Circuits 24

---

---

---

---

---

---

---

---

### Subtraction

$$\begin{array}{r} 0101 \\ - 0011 \\ \hline 0010 \end{array}$$

**B inverted**

CompOrg Fall 2002 - Combinational Circuits 25

---

---

---

---

---

---

---

---

### Two's Complement Numbers

- Nothing is different!
  - This is the advantage of using 2's complement representation.
- Overflow:
  - For addition: sign of the result is different than the sign of the operands (and they have the same sign).

CompOrg Fall 2002 - Combinational Circuits 26

---

---

---

---

---

---

---

---

### -3 + 7

$$\begin{array}{r} 111 \\ 1101 \\ + 0111 \\ \hline 0100 \end{array}$$

CompOrg Fall 2002 - Combinational Circuits 27

---

---

---

---

---

---

---

---

$-3 + -7$

```

      001
      1101
    + 1001
    -----
      0110
  
```

**Overflow!**

CompOrg Fall 2002 - Combinational Circuits 28

---

---

---

---

---

---

---

---

### Ripple Carry Timing

- All the adders are actually operating all the time (they are just combinational circuits).
- We wait long enough (until the last carry has been computed) and then pay attention to the complete answer.
- It is likely that there are intermediate values that are wrong!

CompOrg Fall 2002 - Combinational Circuits 29

---

---

---

---

---

---

---

---

### Carry Look-ahead

- Compute the carry bits right away.
  - As a function of the inputs A and B.
- Not possible for a large adder (32 bit), but realistic for a 4 bit adder.

CompOrg Fall 2002 - Combinational Circuits 30

---

---

---

---

---

---

---

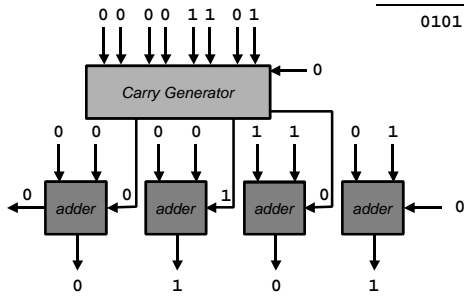
---

### 4 Bit Carry Look-ahead

```

  010
 0011
+ 0010
-----
0101

```



CompOrg Fall 2002 - Combinational Circuits

31

---



---



---



---



---



---



---