CSE 140 Lecture 12
Combinational Standard Modules

CK Cheng
CSE Dept.
UC San Diego
Part III. Standard Modules

Interconnect Modules:
1. Decoder, 2. Encoder
3. Multiplexer, 4. Demultiplexer
Multiplexer

- Definition
- Logic Diagram
- Application
Interconnect: Decoder, Encoder, Mux, DeMux

- **Decoder**: Decode the address to assert the addressed device
- **Mux**: Select the inputs according to the index addressed by the control signals
iClicker: Multiplexer Definition

A. A device that interleaves two or more activities
B. A communications device that combines several signals for transmission over a single medium
C. A logic circuit that sends one of several inputs out over a single output channel.
D. The circuit that uses a common communications channel for sending two or more messages or signals.
E. All of the above
3. Mux (Multiplexer) Definition: A digital module that selects one of data inputs according to the binary address of the selector.

Description
If $E = 1$
$$y = D_i$$
where $i = (S_{n-1}, ..., S_0)$
Else
$$y = 0$$

$E$ (Enable)
$D_{2^n-1}-D_0$ (Data input)
$S_{n-1,0}$ (Selector or Address)
**Multiplexer (Mux): Definition**

- Selects between one of $N$ inputs to connect to the output.
- $\log_2 N$-bit select input – control input

![Diagram of a multiplexer with inputs $D_0$, $D_1$, selector $S$, enable $E$, and output $y$.]
PI Q: What is the output of the following MUX?

A. 0
B. 1
C. Can’t say
Multiplexer (Mux): Definition

• Selects between one of $N$ inputs to connect to the output.
• $\log_2 N$-bit select input – control input
• Example: 

![Diagram of 2:1 Mux]

<table>
<thead>
<tr>
<th>$S$</th>
<th>$D_1$</th>
<th>$D_0$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$S$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$D_0$</td>
</tr>
<tr>
<td>1</td>
<td>$D_1$</td>
</tr>
</tbody>
</table>
Multiplexer Definition: Example

\[ y = D_S \]

\[ \begin{array}{ccc}
S_1 & S_0 & y \\
0 & 0 & D_0 \\
0 & 1 & D_1 \\
1 & 0 & D_2 \\
1 & 1 & D_3 \\
\end{array} \]
Multiplexer Definition: Example

E=1:
If $D_0 = 0$ and $S_1 S_0 = 00 \Rightarrow y = 0$
If $D_0 = 1$ and $S_1 S_0 = 00 \Rightarrow y = 1$
Multiplexer: Logic Diagram

- Logic gates
  - Sum-of-products form

\[
Y = D_0 S + D_1 S
\]

- Tristates
  - For an N-input mux, use N tristates
  - Turn on exactly one to select the appropriate input
Multiplexer Application

- Mux for a Boolean function with truth table as input

\[
\begin{array}{ccc}
A & B & Y \\
0 & 0 & 0 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\]

\[Y = AB\]
Multiplexer: Application

\[ Y = A B \]

\[ Y = A + B \]

\[ \begin{array}{c|c|c|c|c}
A & B & Y & A & Y \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 \\
\end{array} \]

\[ B \]

\[ 0 \]

\[ 1 \]

\[ \overline{S} \to Y \]

\[ A \]

\[ 0 \]

\[ 1 \]

\[ 0 \]

\[ 1 \]

\[ 0 \]

\[ 1 \]

\[ 0 \]

\[ 1 \]

\[ 1 \]

\[ 0 \]

\[ 1 \]

\[ 1 \]

\[ 0 \]

\[ 1 \]
Multiplexer Application: universal set \{Mux\}

We use selector to decompose the function into smaller functions (less number of variables), which follows Shannon’s expansion.
We simplify the decomposed functions using K-map, which follows consensus theorem.
Multiplexer Application: universal set \{Mux\}

Example 1: Given $f(a,b,c) = \Sigma m(0,1,7) + \Sigma d(2)$, implement with an 8-input Mux.

<table>
<thead>
<tr>
<th>Id</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Example 2: Given \( f(a,b,c) = \Sigma m(0,1,7) + \Sigma d(2) \), implement with 4-input Muxes.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c = 0</th>
<th>c = 1</th>
<th>D(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>D₀(c) = 1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>D₁(c) = 0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>D₂(c) = 0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>D₃(c) = 0</td>
</tr>
</tbody>
</table>

\[ (a b c) = (010) \Rightarrow y = 0 \]

Example: \( (a b c) = (111) \Rightarrow y = 0 \)

\[ (a b c) = (110) \Rightarrow y = 0 \]
Example 3: Given $f(a,b,c) = \Sigma m(0,1,7) + \Sigma d(2)$, implement with 2-input Muxes.

<table>
<thead>
<tr>
<th>$a$</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
<th>$D(b,c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>$D_0(b,c)$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>$D_1(b,c)$</td>
</tr>
</tbody>
</table>

$l_0 = b'c$

$l_1 = bc$

By $(abc) = (010)$

$(abc) = (101) \Rightarrow y = 0$
Example 3: Given $f(a, b, c) = \Sigma m(0, 1, 7) + \Sigma d(2)$, implement with 2-input Muxes.

<table>
<thead>
<tr>
<th>a</th>
<th>00 01 10 11</th>
<th>D(b,c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 1 - 0</td>
<td>$D_0(b,c)$</td>
</tr>
<tr>
<td>1</td>
<td>0 0 0 1</td>
<td>$D_1(b,c)$</td>
</tr>
</tbody>
</table>

$D_0(b,c) = b'$

$D_1(b,c) = bc$

![Diagram of Muxes](image)
Example 3: Given $f(a,b,c) = \Sigma m (0,1,7) + \Sigma d(2)$, implement with 2-input Muxes.

<table>
<thead>
<tr>
<th>$b$</th>
<th>$c = 0$</th>
<th>$c = 1$</th>
<th>$l_1(0) = 0$</th>
<th>$l_1(c) = c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram of Muxes](image)
Example 3: Given $f(a,b,c) = \Sigma m(0,1,7) + \Sigma d(2)$, implement with 2-input Muxes.

\[
\begin{array}{l|cc|c}
D_1 (b,c) & c = 0 & c = 1 \\
\hline
b & 0 & 0 & 0 & l_1(0) = 0 \\
& 1 & 0 & 1 & l_1(c) = c \\
\end{array}
\]
Example 4: Given $f(a,b,c) = \Sigma m(0,2,4,7) + \Sigma d(3,5)$, implement with 2-input Muxes.

<table>
<thead>
<tr>
<th>$f$</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$ 0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>$a$ 1</td>
<td>1</td>
<td>-</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$D_0(b,c) \quad D_1(b,c)$

Diagram:

```
  E
 /|
/  |
/   |
/    |
/     D_0(b,c)
/       |
/        D_1(b,c)
/          y
/            a
```
4. Demultiplexers

Control Input
4. Demultiplexers

\[ y_i = x \text{ if } i = (S_{n-1}, \ldots, S_0) \& E = 1 \]
\[ y_i = 0 \text{ otherwise} \]
Shifters

- **Logical shifter:** shifts value to left or right and fills empty spaces with 0’s
  - Ex: \( 11001 \gg 2 = 00110 \)
  - Ex: \( 11001 \ll 2 = 00100 \)

- **Arithmetic shifter:** same as logical shifter, but on right shift, fills empty spaces with the old most significant bit (msb).
  - Ex: \( 11001 \gg>> 2 = 11110 \)
  - Ex: \( 11001 \ll<< 2 = 00100 \)

- **Rotator:** rotates bits in a circle, such that bits shifted off one end are shifted into the other end
  - Ex: \( 11001 \text{ ROR} 2 = 01110 \)
  - Ex: \( 11001 \text{ ROL} 2 = 00111 \)
Shifter

Can be implemented with a mux

\[ y_i = \begin{cases} 
    x_{i-1} & \text{if } E = 1, s = 1, \text{ and } d = L \\
    x_{i+1} & \text{if } E = 1, s = 1, \text{ and } d = R \\
    x_i & \text{if } E = 1, s = 0 \\
    0 & \text{if } E = 0 
\end{cases} \]
Shifter Design

\[ A_{3:0} \rightarrow 4 \rightarrow \text{shamt}_{1:0} \rightarrow 4 \rightarrow Y_{3:0} \]
Barrel Shifter

- **S₀**: O or 1 shift
- **S₁**: O or 2 shift
- **S₂**: O or 4 shift

- Input: x
- Output: y

- Bit sequence: 0 1 0 1 0 1 0 1
- Shift sequences:
  - S₀: 0 1 0 1 0 1 0 1
  - S₁: 0 1 0 1 0 1 0 1
  - S₂: 0 1 0 1 0 1 0 1
Shifters as Multipliers and Dividers

- A left shift by $N$ bits multiplies a number by $2^N$
  - Ex: $00001 << 2 = 00100$ ($1 \times 2^2 = 4$)
  - Ex: $11101 << 2 = 10100$ ($-3 \times 2^2 = -12$)

- The arithmetic right shift by $N$ divides a number by $2^N$
  - Ex: $01000 >>> 2 = 00010$ ($8 \div 2^2 = 2$)
  - Ex: $10000 >>> 2 = 11100$ ($-16 \div 2^2 = -4$)