Part III - Standard Combinational Modules

• Introduction
• Decoder
  – Behavior, Logic, Usage
• Encoder
• Multiplexer (Mux)
  – Behavior, Logic, Usage
• Demultiplexer (DeMux)
Part III - Standard Combinational Modules

Signal Transport
- Decoder: Decode address
- Encoder: Encode address
- Multiplexer (Mux): Select data by address
- Demultiplexer (DeMux): Direct data by address
- Shifter: Shift bit location

Data Operator
- Adder: Add two binary numbers
- Multiplier: Multiply two binary numbers

ALU, FFT, Image Conv.

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
1. Decoder

- Definition
- Logic Diagram
- Application (Universal Set)
- Tree of Decoders
Decoder: Definition

A. A device that decodes
B. An electronic device that converts signals from one form to another
C. A machine that converts a coded text into ordinary language
D. A device or program that translates encoded data into its original format
E. All of the above

Decoder Definition: A digital module that converts a binary address to the assertion of the addressed device

\[ y_i = 1 \text{ if } E = 1 \& (I_2, I_1, I_0) = i \]
\[ y_i = 0 \text{ otherwise} \]
1. Decoder: Definition

- $N$ inputs, $2^N$ outputs
- One-hot outputs: only one output HIGH at most

Example: A 3-input decoder has how many outputs?
A. 2 outputs
B. 4 outputs
C. 8 outputs
D. 10 outputs
Decoder Definition

Example: For a 3-input decoder, suppose \((E,I_2,I_1,I_0) = (1,0,0,0)\), then \((y_7,y_6, \ldots, y_0)\) is equal to:
A. (00000000)
B. (00000001)
C. (00000010)
D. (01000000)
E. (10000000)

Decoder: Logic Diagram (Inside a decoder)

\(y_0 = 1\) if \((A_1, A_0) = (0,0)\) & \(E = 1\)

\(y_i = m_i E\)

\(y_3 = A_1 A_0 E\)
1. Decoder: Definition

PI Q: What is the output $Y_{3:0}$ of the 2:4 decoder for $(A_1, A_0) = (1,0)$?

A. (1, 1, 0, 0 )
B. (1, 0, 1, 1)
C. (0, 0, 1, 0)
D. (0, 1, 0, 0)

2:4 Decoder

<table>
<thead>
<tr>
<th>$A_1$</th>
<th>$A_0$</th>
<th>$Y_3$</th>
<th>$Y_2$</th>
<th>$Y_1$</th>
<th>$Y_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>10</td>
<td>01</td>
<td>00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Decoder Application: universal set \{Decoder, OR\}

Example:
Implement the following functions with a 3-input decoder and OR gates.

i) $f_1(a,b,c) = \Sigma m(1,2,4)$
ii) $f_2(a,b,c) = \Sigma m(2,3)$,
iii) $f_3(a,b,c) = \Sigma m(0,5,6)$

\[\text{Decoder}\]
\[
\begin{array}{ccc}
\text{Decoder} & \rightarrow & y_0 \\
\text{Decoder} & \rightarrow & y_1 \\
\text{Decoder} & \rightarrow & y_2 \\
\end{array}
\]

\[\begin{array}{ccc}
\text{Decoder} & \rightarrow & f_1 \\
\text{Decoder} & \rightarrow & f_2 \\
\text{Decoder} & \rightarrow & f_3 \\
\end{array}\]
Decoder Application: universal set \{\text{Decoder, OR}\}

Decoder produces minterms when $E=1$.
We can use an OR gate to collect the minterms to cover the On-set.
For the Don’t Care-Set, we can just ignore the terms.

Decoder Application: universal set \{\text{Decoder, OR}\}

Example: Implement functions
i) $f_1(a,b,c) = \Sigma m(1,2,4) + \Sigma d(0,5)$,
ii) $f_2(a,b,c) = \Sigma m(2,3) + \Sigma d(1,4)$,
iii) $f_3(a,b,c) = \Sigma m(0,5,6)$

with a 3-input decoder and OR gates.
Decoders

- OR minterms

\[ Y = AB + \overline{AB} = A \oplus B \]

Tree of Decoders: Scale up the size of the decoders using a tree structure

Implement a $4-2^4$ decoder with $3-2^3$ decoders.

\[ E \text{ is used by signal } a. \]
Tree of Decoders

Implement a 6-2^6 decoder with 3-2^3 decoders.

when \( E = 0 \) \( \rightarrow \) all \( y_i = 0 \) \( \forall i \)
when \( E = 1 \)

Construction: A four variable switching function \( f(a,b,c,d) \) can be implemented using which of the following?

A. 1:2 decoders and OR gates
B. 2:4 decoders and OR gates
C. 3:8 decoders and OR gates
D. All of the above
E. None of the above
2. Encoder

- Definition
- Logic Diagram
- Priority Encoder

Definition of Encoder

A. Any program, circuit or algorithm which encodes
B. In digital audio technology, an encoder is a program that converts an audio WAV file into an MP3 file
C. A device that convert a message from plain text into code
D. A circuit that is used to convert between digital video and analog video
E. All of the above
Encoder Definition: A digital module that converts the assertion of a device to the binary address of the device.

Encoder Description:

At most one \( I_i = 1 \).

\[
\begin{align*}
C(y_{n-1}, \ldots, y_0) &= i \text{ if } I_i = 1 \land E = 1 \\
C(y_{n-1}, \ldots, y_0) &= 0 \text{ otherwise.}
\end{align*}
\]

\( A = 1 \) if \( E = 1 \) and one \( i \) s.t. \( I_i = 1 \)

\( A = 0 \) otherwise.

Encoder: Logic Diagram
Priority Encoder: Definition

Description: Input \((I_{2^{n-1}}, \ldots, I_0)\), Output \((y_{n-1}, \ldots, y_0)\)

\[(y_{n-1}, \ldots, y_0) = i \text{ if } I_i = 1 \& E = 1 \& I_k = 0 \text{ for all } k > i \text{ (high bit priority)} \text{ or }
\]

\[\text{for all } k < i \text{ (low bit priority).}
\]

\[\color{red}{2} E_o = 1 \text{ if } E = 1 \& I_i = 0 \text{ for all } i,
\]

\[\color{red}{3} G_s = 1 \text{ if } E = 1 \& \exists i \text{ s.t. } I_i = 1.
\]

\[(G_s \text{ is like } A, \text{ and } E_o \text{ passes on enable}).
\]

1. \(E = 0\)
2. \(E = 1 \& \forall I_i = 0\)
3. \(E = 1 \& \exists i \text{, } I_i = 1\)
Priority Encoder: Implement a 32-input priority encoder with 8 input priority encoders (high bit priority).

1. If $E=0 \Rightarrow A=0 \Rightarrow$ Address is not valid.
2. If $E=1 \Rightarrow I_{31-24} \exists i \in 31-24, I_i=1$

Then $G_s=1$, $y_{32}, y_{31}, y_{30}$ show the index of input line $\Rightarrow E_{o_3}=0$

$\forall k \in 31-24, I_k=0$

$E_{o_3}=1 \Rightarrow$ Pass the token to the next encoder.