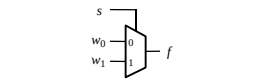
Logic Design

Combinational-Circuit Building Blocks

McMaster University

Mutliplexer

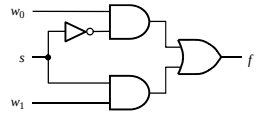
- Selects one of several inputs and directs it to a single output.



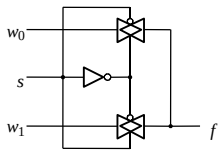
(a) Graphical symbol

s	f
0	w_0
1	w_1

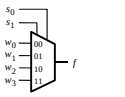
(b) Truth table



(c) Sum-of-products circuit



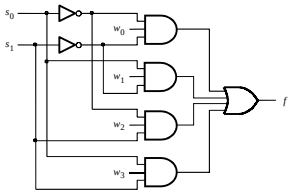
(d) Circuit with transmission gates



(a) Graphic symbol

s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

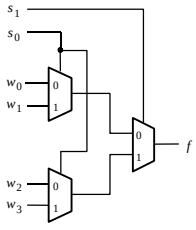
(b) Truth table



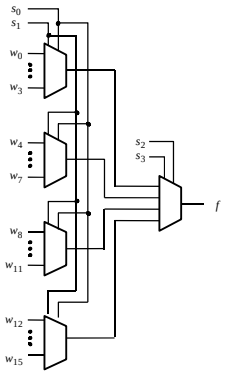
(c) Circuit

Multiplexer

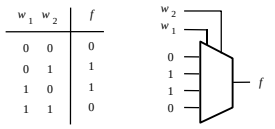
4-to-1 multiplexer build using three 2-to-1 multiplexer



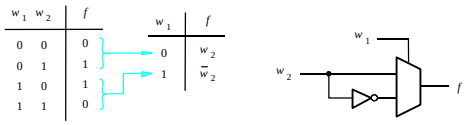
Multiplexer



Synthesis of logic functions using multiplexer



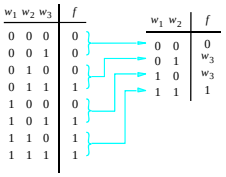
(a) Implementation using a 4-to-1 multiplexer



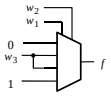
(b) Modified truth table

(c) Circuit

Synthesis of logic functions using multiplexer

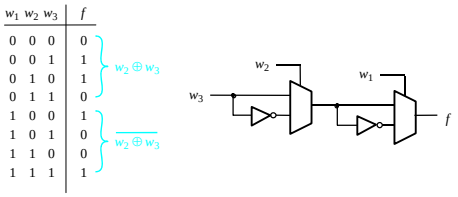


(a) Modified truth table



(b) Circuit

Figure 6.8. Implementation of the three-input majority function using a 4-to-1 multiplexer.

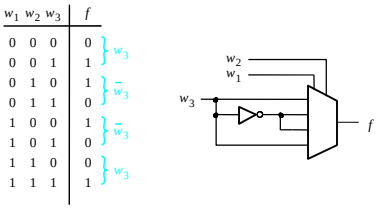


(a) Truth table

(b) Circuit

Figure 6.9. Three-input XOR implemented with 2-to-1 multiplexers.

Synthesis of logic functions using multiplexer



(a) Truth table

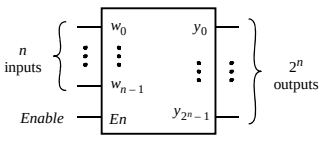
(b) Circuit

Multiplexer Synthesis Using Shannon's Expansion

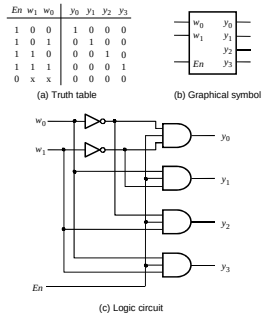
- It is possible to connect more complex circuits as inputs to a multiplexer to synthesize logic circuits
- Do the examples and Shannon's expansion on the board

Decoders

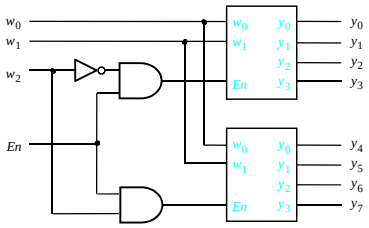
- Decoder: decodes encoded information
- A binary decoder is a logic circuit with n inputs and 2^n outputs
- Only one output is asserted at a time (corresponding to one valuation of inputs)
- Enable: En=0 none of the decoder outputs is asserted



- An n-bit binary code in which exactly one of the bits is set to 1 at a time is called one-hot-encoded



- Larger decoders can be built from smaller decoders



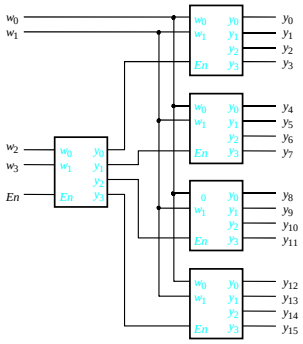


Figure 6.18. A 4-to-16 decoder built using a decoder tree.

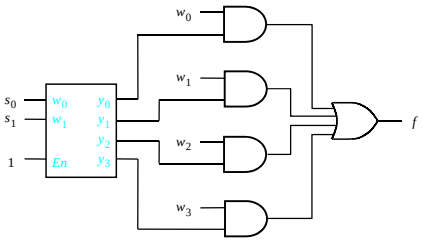


Figure 6.19. A 4-to-1 multiplexer built using a decoder.

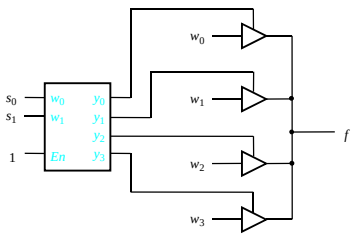


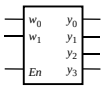
Figure 6.20. A 4-to-1 multiplexer built using a decoder and tri-state buffers.

Demultiplexers

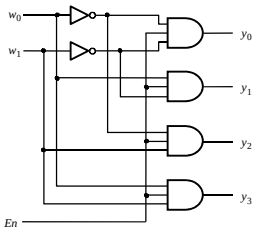
- Multiplexer: one output, n inputs, $\log_2 n$ select inputs,
- Multiplexer: multiplex n data inputs onto a single output line under the control of select inputs
- De-mulitplexer: performs the opposite function
- A decoder can be used as a de-multiplexer
- En serves as the data input

En	w_1	w_0	y_0	y_1	y_2	y_3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1
0	x	x	0	0	0	0

(a) Truth table



(b) Graphical symbol



- One of the most important applications of decoders is in memory blocks
- One type of memory block is read only memory ROM
- ROM: a collection of storage cells each permanently store a single bit

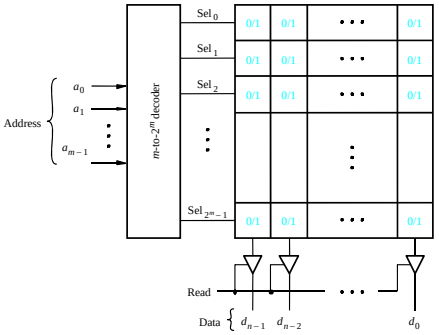
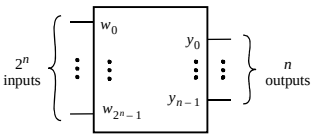


Figure 6.21. A 2^m x n read-only memory (ROM) block.

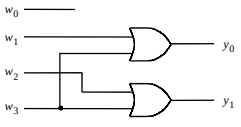
Encoder

- A binary encoder encodes information from 2^n inputs into an n-bit code
- One of the input signals should have a value of 1 and the outputs present the binary number that identifies which input is equal to 1.
- Next slide: 4 to 2 encoder, inputs one-hot encoded
- All input patterns that have multiple inputs set to 1 are treated as don't care



w_3	w_2	w_1	w_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

(a) Truth table



(b) Circuit

Figure 6.23. A 4-to-2 binary encoder.

Priority Encoder

- Each input has a priority level
- When an input with high priority is asserted, the other inputs with lower priority are ignored
- The output indicates the active input with the highest priority
- Z is used to indicate when none of the inputs is 1.

w_3	w_2	w_1	w_0	y_1	y_0	z
0	0	0	0	d	d	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

Priority Encoder

- Logic circuit for priority encoder:
- Define a set of intermediate signals i_k
- i_k is 1 if w_k is the highest priority input set to 1

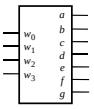
$$i_0 = \overline{w_3}\,\overline{w_2}\,\overline{w_1}w_0$$
$$i_1 = \overline{w_3}\,\overline{w_2}w_1$$
$$i_2 = \overline{w_3}w_2$$
$$i_3 = w_3$$

$$y_0 = i_1 + i_3$$
$$y_1 = i_2 + i_3$$

$$z = i_0 + i_1 + i_2 + i_3$$

Code converters

- BCD to 7-segment decoder: converts one BCD digit into information suitable for driving a digit display
- 7-segment: each segment is a light emitting diode



(a) Code converter

(b) 7-segment display

w_3	w_2	w_1	w_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

(c) Truth table

Figure 6.25. A BCD-to-7-segment display code converter.

Comparator

- Comparator inputs: two n-bit unsigned numbers A and B
- Three outputs: AeqB, AgtB and AltB
- Design: truth table approach is hard due to the size of the table
- $A=a_3a_2a_1a_0$, $B=b_3b_2b_1b_0$
- Define i_3,i_2,i_1,i_0: i_k is one if a_k and b_k are the same

$$i_k = \overline{a_k \oplus b_k}$$

$$AeqB = i_3 i_2 i_1 i_0$$

$$AgtB = a_3\overline{b_3} + i_3 a_2\overline{b_2} + i_3 i_2 a_1\overline{b_1} + i_3 i_2 i_1 a_0\overline{b_0}$$

$$AltB = \overline{AeqB + AgtB}$$