Engr354: Digital Logic Circuits

Chapter 6: Combinational Blocks

Dr Curtis Nelson

Chapter 6 Overview

In this chapter you will learn about:

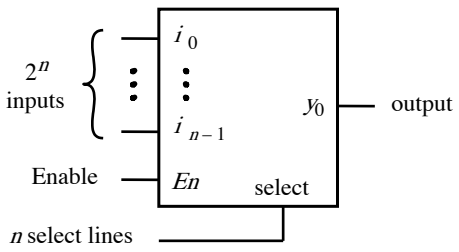
- Commonly used combinational sub-circuits;
- Multiplexers, which can be used for selection of signals and for implementation of general logic functions;
- Circuits for encoding, decoding, and code-conversion purposes.

Combinational Circuits

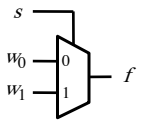
- Combinational circuits are those whose outputs are completely defined by the inputs, i.e. no memory is involved.
- Examples include
  - Basic gates like Nands, Nors and Inverters;
  - Multiplexers, de-multiplexers;
  - Decoders, encoders, code converters;
  - Arithmetic circuits;
  - Magnitude comparators;
  - Parity checkers;
  - Etc.

Multiplexers

- Devices that select one of many inputs to be routed to one output based on the binary value of select lines
  - Enable – used to enable or disable the complete function;
  - Select – used to select which one of the inputs gets routed to the output.



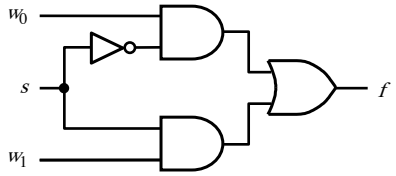
2-to-1 Multiplexer



(a) Graphical symbol

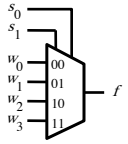
s	f
0	w_0
1	w_1

(b) Truth table



(c) Sum-of-products circuit

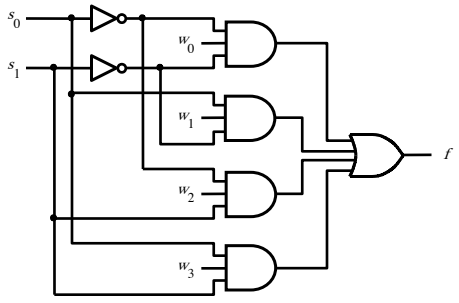
4-to-1 Multiplexer



(a) Graphical symbol

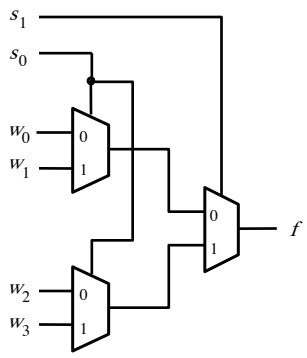
s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

(b) Truth table

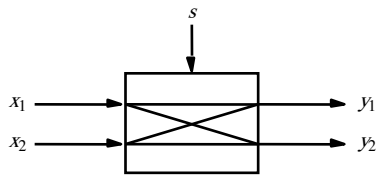


(c) Circuit

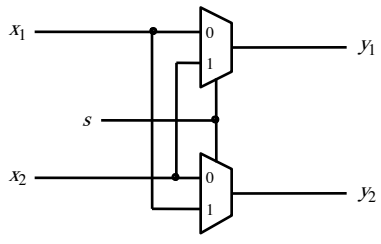
Using 2-to-1 Mux's to Build a 4-to-1 Mux



Practical Application



(a) A 2x2 crossbar switch



(b) Implementation using multiplexers

Multiplexer Data Sheet

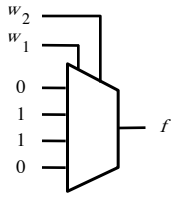
- 74HC153 Dual 4-Line to 1-Line Data Selectors/Multiplexers

Synthesis of Logic Functions Using Mux's

- Mux's can be used to synthesize logic functions as follows
  - Create truth table;
  - Compress as necessary;
  - Implement.
- In general, an *N* variable function can be implemented with one *N-1* multiplexer and at most, one inverter.

Example 2-Input Function

w_1	w_2	f
0	0	0
0	1	1
1	0	1
1	1	0

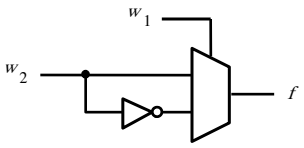


(a) Implementation using a 4-to-1 multiplexer

w_1	w_2	f
0	0	0
0	1	1
1	0	1
1	1	0

w_1	f
0	w_2
1	$\overline{w}_2$

(b) Modified truth table



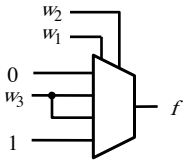
(c) Circuit

Example 3-Input Majority Function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

w_1	w_2	f
0	0	0
0	1	w_3
1	0	w_3
1	1	1

(a) Modified truth table



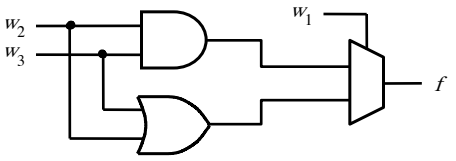
(b) Circuit

Example 3-Input Majority Function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

w_1	f
0	$w_2 w_3$
1	$w_2 + w_3$

(a) Truth table

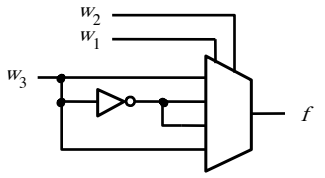


(b) Circuit

Example 3-Input XOR Function

w_1	w_2	w_3	f	
0	0	0	0	w_3
0	0	1	1	
0	1	0	1	$\bar{w}_3$
0	1	1	0	
1	0	0	1	$\bar{w}_3$
1	0	1	0	
1	1	0	0	w_3
1	1	1	1	

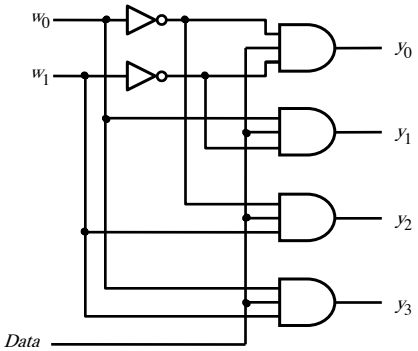
(a) Truth table



(b) Circuit

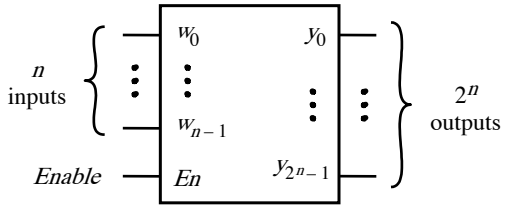
De-Multiplexers

- A de-multiplexer is a circuit which places the value of a single data input onto one of a number of outputs.



An *n*-to-2^n Decoder

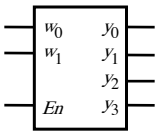
- A decoder is a device that activates one output, whose outputs are usually active low, based on the binary value of the inputs;
- A decoder is a minterm generator;
- Enable – used to enable or disable the complete decoder function.



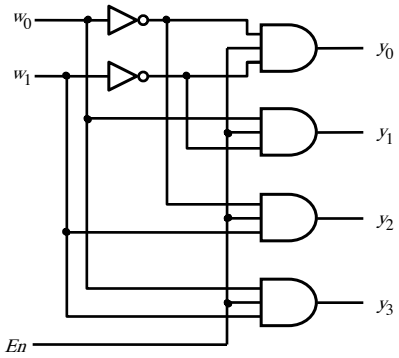
A 2-to-4 Decoder

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

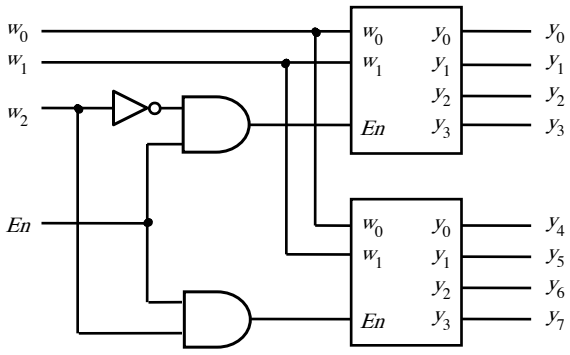


(b) Graphic symbol

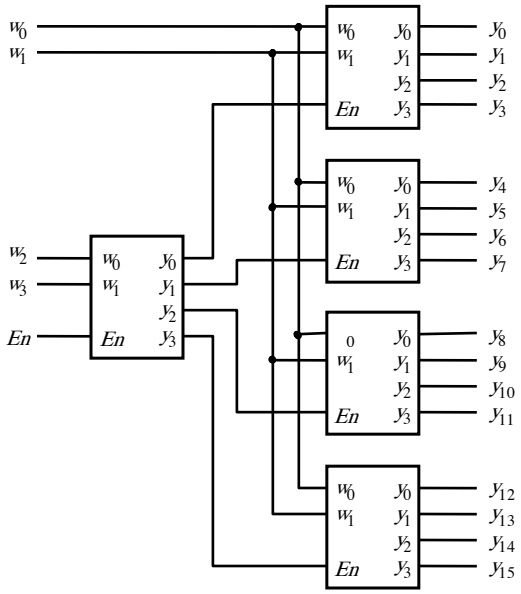


(c) Logic circuit

3-to-8 Decoder Using 2-to-4 Decoders



Decoder Tree



Decoder Data Sheet

- 74HC138 Single 3-Line to 8-Line Decoders/Demultiplexers

Combinational Design with Decoders

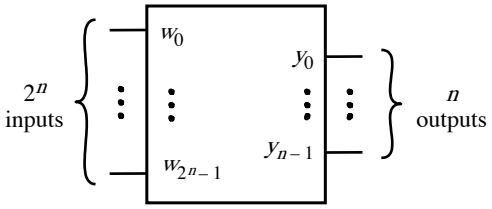
- Implement the function $f(a,b,c) = \Sigma m(0,3,4,5)$ using a decoder and random logic.
- Implement the function $f(a,b,c) = \Pi M(0,1,4,6)$ using a decoder and random logic.

Combinational Design with Decoders and Mux's

- Design a digital circuit which has a 4-bit input, A = $A_3A_2A_1A_0$ and a single output Z. Z is high if A is exactly divisible by 3 and A is not equal to 12. Note that 0 is not divisible by 3 in this circuit. Use a 2-to-4 decoder, a 4-to-1 MUX, and minimum logic gates, if necessary. Note that all inputs and outputs of the MUX and decoder are asserted HIGH.

A 2^n-to-n Binary Encoder

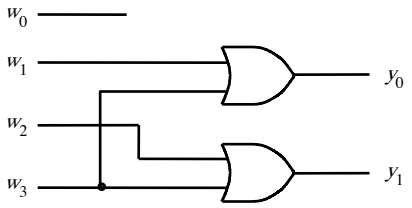
- An encoder is a device that outputs a binary code representing which one of many inputs is active. The outputs are usually active low.
- Priority encoder – assigns priority to certain inputs
  - Used in embedded computer systems to service interrupts.



A 4-to-2 Binary Encoder

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



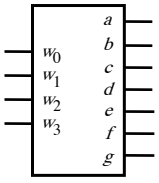
(b) Circuit

Truth Table for a 4-to-2 Priority Encoder

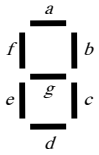
- Create a circuit using the following truth table for a 4-to-2 priority encoder (z is an output that is asserted whenever any output is asserted).

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

BCD to 7-segment Code Converter



(a) Code converter

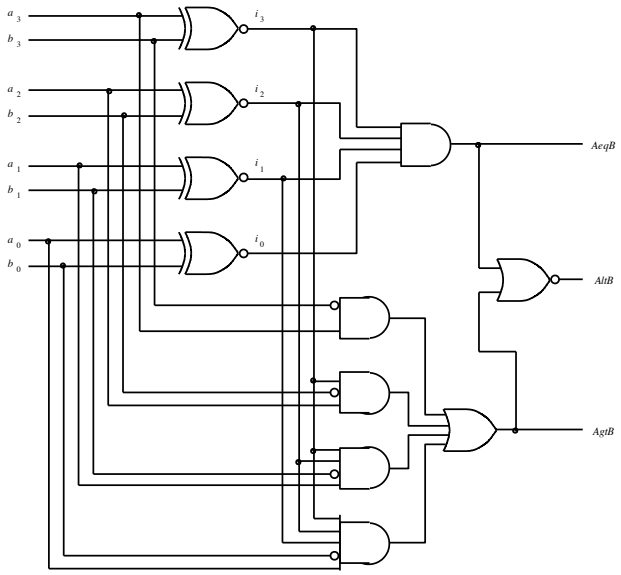


(b) 7-segment display

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

Four-bit Magnitude Comparator



Arithmetic Logic Unit (74HC381)

Operation	Inputs s_2 s_1 s_0	Outputs F
Clear	0 0 0	0 0 0 0
B=A	0 0 1	$B = A$
A=B	0 1 0	$A = B$
ADD	0 1 1	$A + B$
XOR	1 0 0	A XOR B
OR	1 0 1	A OR B
AND	1 1 0	A AND B
Preset	1 1 1	1 1 1 1

Summary

In this chapter you learned about:

- Commonly used combinational sub-circuits;
- Multiplexers, which can be used for selection of signals and for implementation of general logic functions;
- Circuits for encoding, decoding, and code-conversion purposes.