Midterm 2

Dr. Jordan Hanson - Whittier College Dept. of Physics and Astronomy

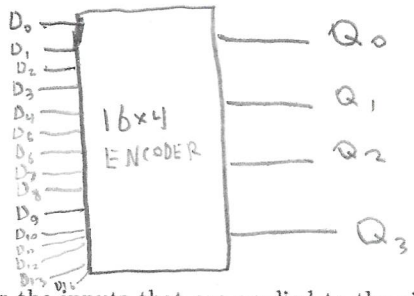
April 29, 2020

1 Chapter 5 - Combinatorial Logic Analysis

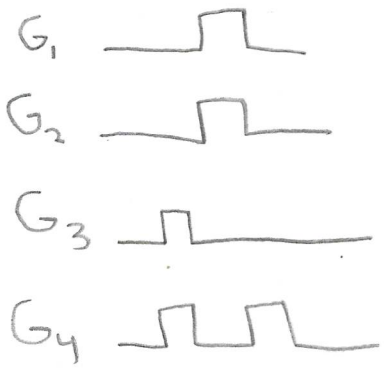
1. Imagine we have a keypad that has digits 0-9 and and letters A-F. Assume each key sends an active-LOW signal to an encoder that converts the single key value to binary. For example, if one presses A, the 4-bit output is 1010. Write the conversion table for the digits and letters to binary below, and develop the encoder logic that produces the binary outputs.

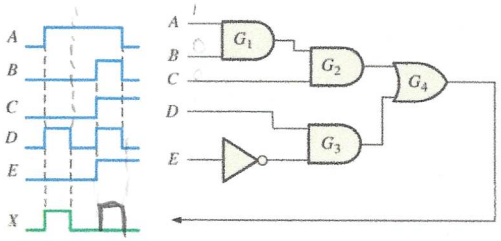
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

A	1010
B	1011
C	1100
D	1101
E	1110
F	1111



2. The output waveform in Fig. 1 is incorrect for the inputs that are applied to the circuit. Assuming that one gate in the circuit has failed, with its output either an apparent constant HIGH or a constant LOW, determine the faulty gate and the type of failure (output open or shorted).





G1 or G2 is the faulty gate

output open

Figure 1: A domain-5 logic function needs debugging.

2 Chapter 6 - Functions of Combinational Logic

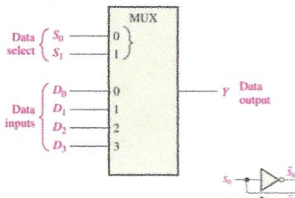
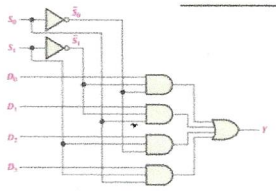


TABLE 6–8
Data selection for a 1-of-4-multiplexer.

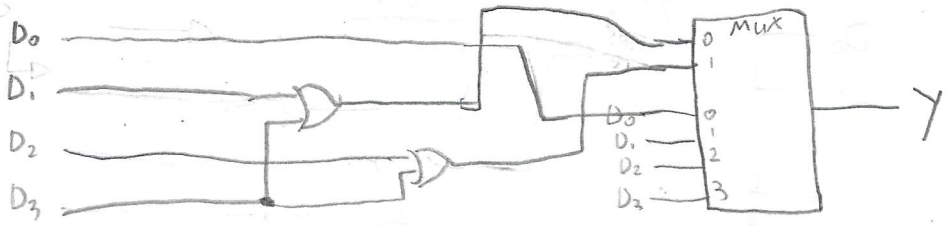
Data-Select Inputs		Input Selected
S_1	S_0	
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3



D_0	D_1	D_2	D_3	S_1	S_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Figure 2: A 1-of-4 multiplexer (mux) logic function: function block, truth table, and gate diagram.

1. Consider Fig. 2, in which the 1-of-4 mux is reviewed. Imagine a camera system (like Zoom only with firmware) in which you are switching between four data feeds from four cameras by holding down one of four different keys. Design a circuit using an encoder and a 1-of-4 mux to switch between four bitstreams on active-LOW keypresses from four keys. Show the encoder logic at the gate level.



2. Using XNOR gates, we could create a comparator that is true if two 2-bit binary numbers are equal. Starting with the 2-bit comparator, add logic that is true if $A > B$. Add one final output that is the opposite of $A > B$ (so $A < B$)[1].

XNOR

A	B	X
0	0	1
1	0	0
0	1	0
1	1	1

A > B

A B

1 1 0 0

A < B

A B

0 0 1 1

3 Chapter 7 - Latches, Flip-flops, and Timers

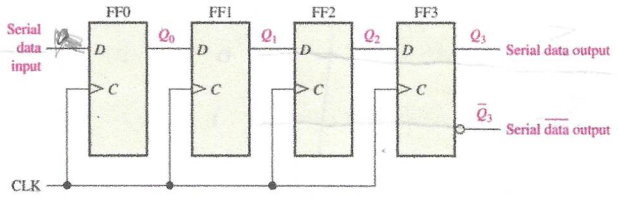
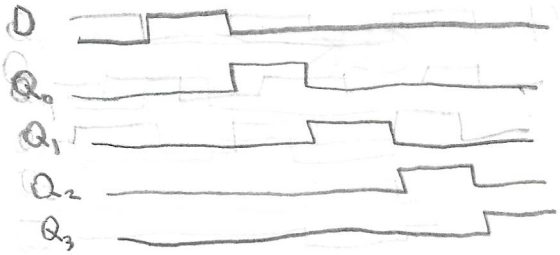


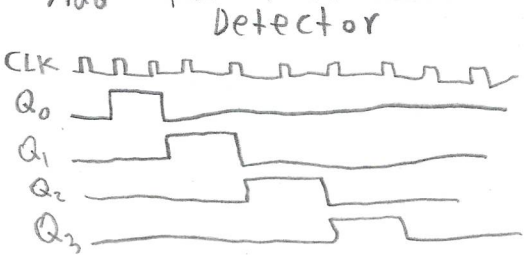
Figure 3: This is an example of a serial-in serial-out shift register.

1. Consider Fig. 3, in which four D flip-flops are connected in series to form a *shift register*. (a) Produce the timing diagram that occurs on the serial data output when all four D flip-flops are initially in the RESET state, and a single data bit arrives at the input with a constantly-running clock signal. (b) Add an input that disables the clock to halt the shift register so that the data is stored. Draw a timing diagram showing the shifting of 4 bits into the register and halting on the right clock cycle to store them. Add pulse Transition Detector

a)



b)



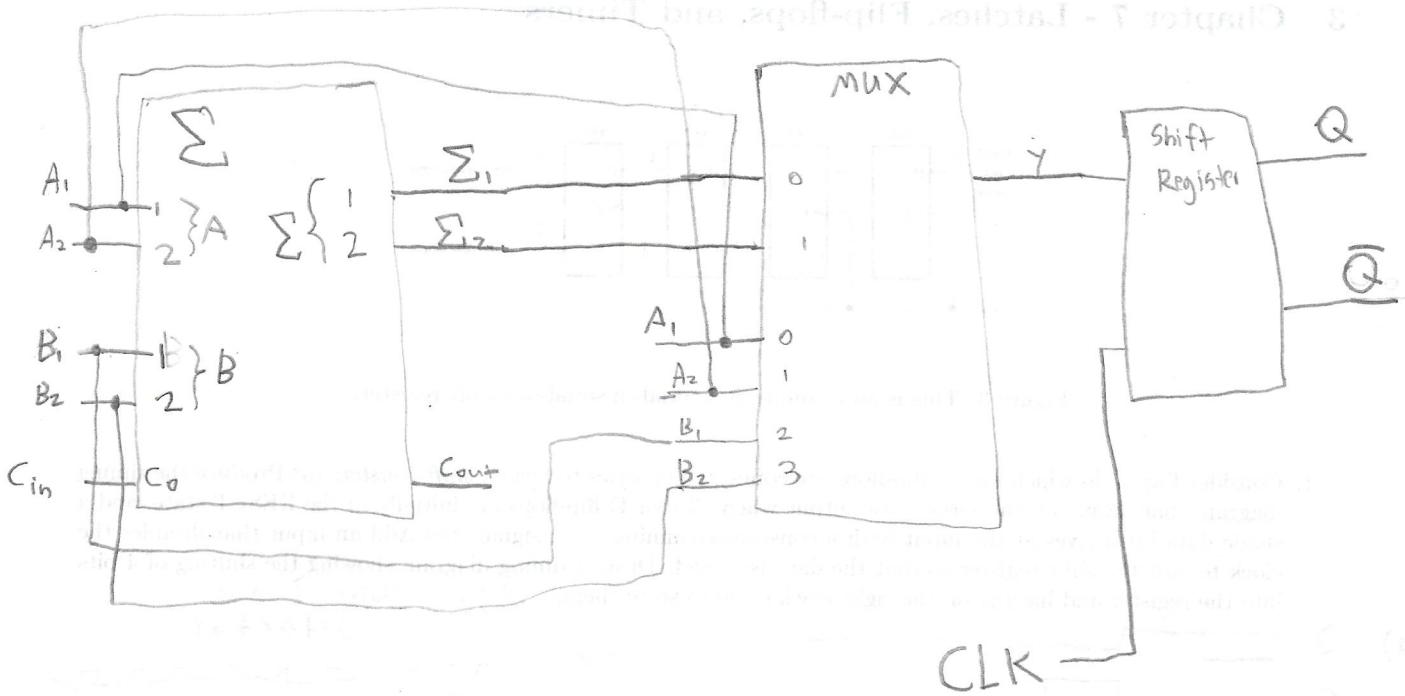
2. Create a function block to represent the 4-bit shift register from the previous problem. Now design a system that adds two binary numbers and stores the result in the shift register via a 1-of-4 multiplexer. The mux is there to convert the *parallel* data out of the adder to *serial* data into the register. (Continue response on back).

- two bit adder

on back

[1]Since this is a 2-bit system, you can enumerate all possible states, if that helps.


Shift register
D_0
Q
$\overline{Q}$
CLK
C


Σ
A_1
A_2
A
Σ_1
Σ_2
B_1
B_2
B
C_{in}
C_0
C_{out}
MUX
A_1
A_2
B_1
B_2
Y
Shift Register
Q
$\overline{Q}$
CLK