Shefali E

CSCI 350 Digital Logic and Computer Organization
G. Pothering
Spring 2020
Assignment 4 **Due February 13, 2020**

1. Implement a 16-1 multiplexer using **only** 4x1 multiplexers. $2^n = 4$ $n=2$ on bottom

2. Implement a 4 x16 decoder with enable using **only** three 3x8 decoders (again, with enable).

3. Use ***only*** an 8-1 multiplexer to implement the function f(x,y,z) = Π(2,5,6). (see page 161 for an example)

The purpose of the rest part of the assignment is to familiarize you better with the notion of propagation delays for signals in a circuit and for you to see some of the tradeoffs that arise in circuit design because of attention to propagation delay. Remember, the *propagation delay* for a gate is the time it takes for a gate to (possibly) change its output value following a change in its input values. time 4 gate 2 change output

4. Consider the full adder circuit in your textbook on page 165.
   a. Assuming each of the logic gates used in this circuit has a propagation delay of 4 ns (nanoseconds), what are the delays for generating the sum and carry-out for this full adder?
   b. Using the "ripple adder" approach described on page 165 to implement a 4-bit adder from four 1-bit adder , calculate the propagation delays for the sum and carry-out for this 4-bit adder.

5. In this exercise we are going to show how we can speed up the propagation delays for ripple adders.
   a. Your textbook describes an alternate type of adder known as a ***carry-select adder*** on page 166. Show how you could implement a 2-bit adder from 1-bit adders using this approach and calculate the propagation times for the sum and final carry-out.
   b. Now use carry-select to design a 4-bit adder from 2 bit adders that were themselves designed by carry-select. Calculate the propagation times for the sum and final carry-out for this 4-bit adder.

6. The approach described in problem 14 on pages 238-39 of your textbook implements what is known as a carry-look-ahead adder. Use carry-look-ahead to implement a 4-bit adder from four 1-bit adders and calculate the propagation times for the sum and final carry-out for this 4-bit adder.

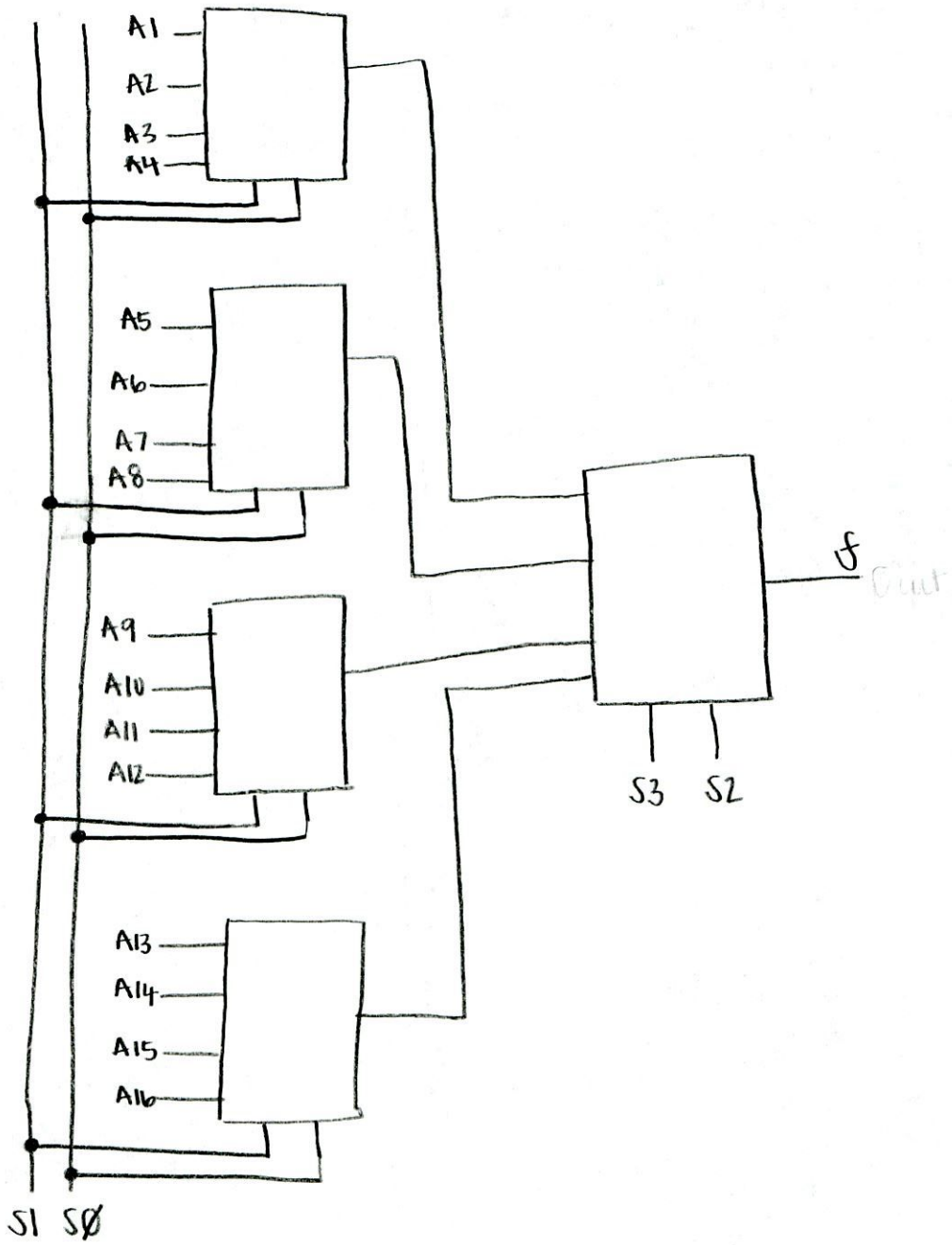
8 input

Assignment 4 Shefali E

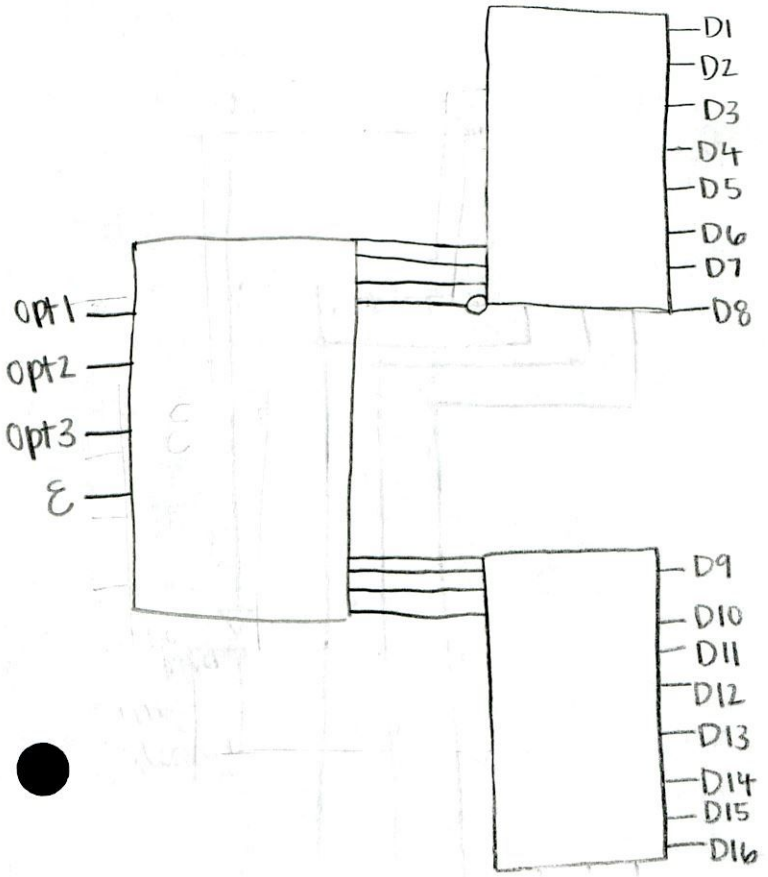
#1 Implement a 16-1 Mux using 4x1 Mux's

$2^n = 4$

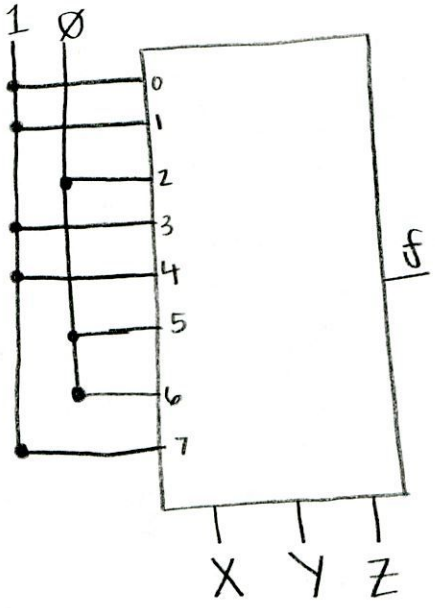
$n = 2$



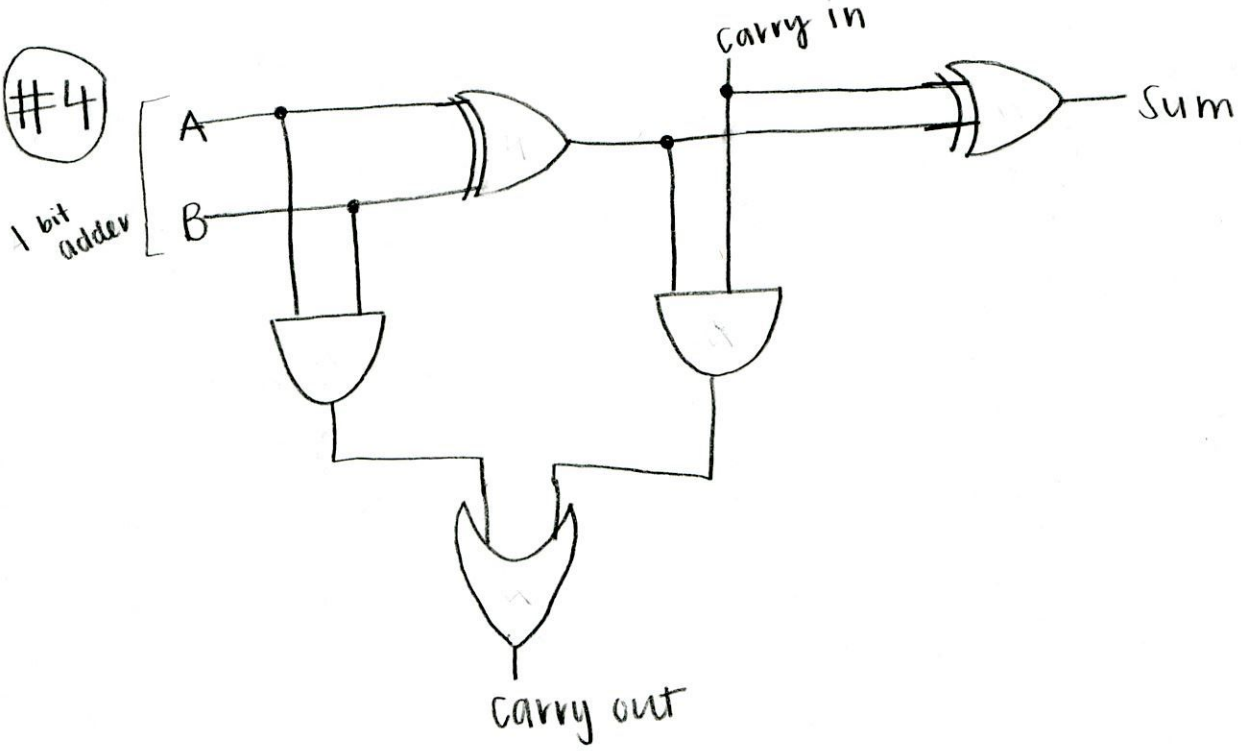
#2 4 x 16 decoder w/ enable using 3 x 8 decoders



(#3) 8-1 Mux to implement $f(x,y,z) = \Pi(2,5,6)$ where it is Ø



(#4)



(a) Sum Delays = 4 + 4 = 8 nano second delay

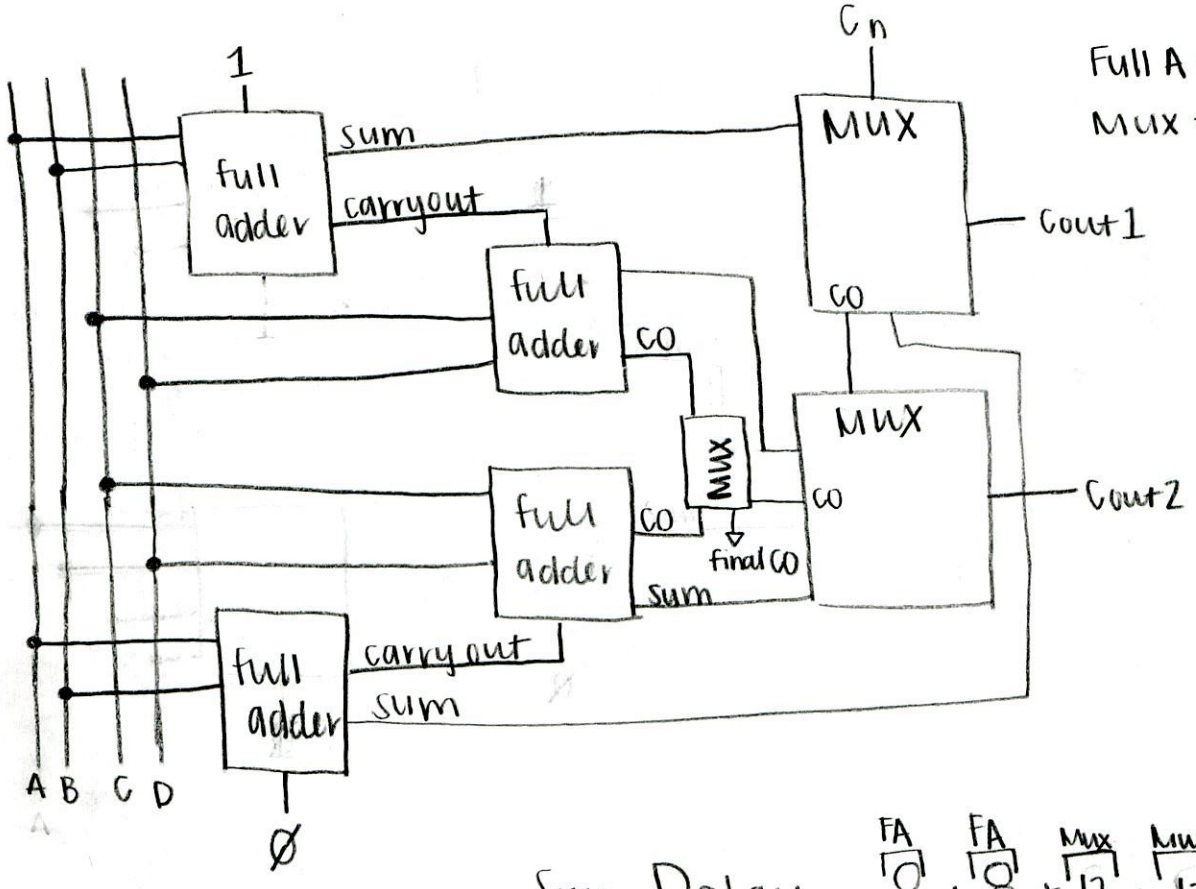
Carry Out Delay = 4 + 4 + 4 + 4 = 16 nano second delay

4b Ripple Adder (4 bit)

$$\text{Sum Delay} = \underbrace{\overset{\text{Sum}}{8} + \overset{\text{CO}}{16}}_{1} + \underbrace{8 + 16}_{2} + \underbrace{8 + 16}_{3} + \underbrace{8}_{4} = 80 \text{ nano seconds}$$

$$\text{Carry Out Delay} = \underbrace{\overset{\text{CO}}{16}}_{1} + \underbrace{16}_{2} + \underbrace{16}_{3} + \underbrace{16}_{4} = 64 \text{ nano seconds}$$

5a Carry Select Adder



Full A = 8 nano

MUX = 12 nano

$$\text{Sum Delay} = \overset{\text{FA}}{8} + \overset{\text{FA}}{8} + \overset{\text{Mux}}{12} + \overset{\text{Mux}}{12} = 40 \text{ nanosecond}$$

$$\text{Final Carry Out Delay} = \underbrace{16 + 16 + 12 + 12}_{} + \overset{\text{last MUX}}{12} = 68 \text{ nanoseconds}$$

5b

Sum delay = 40 + 40 + 12 + 12 = 104 nano sec

- 40: 2 full adders
- 40: 2 FA
- 12: Mux
- 12: Mux

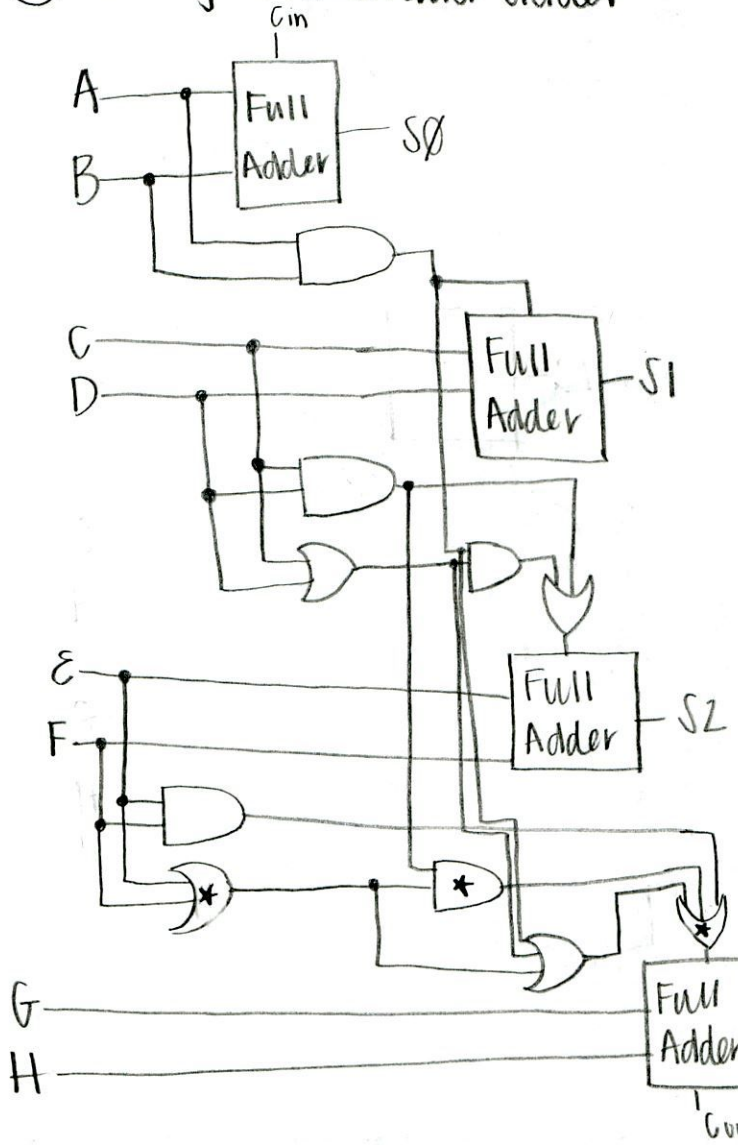
↓

Stay the same time no matter the # of inputs

Carry out

delay = 68 + 68 + 12 + 12 + 12 = 172 nanosec

6 Carry look-ahead adder



calculate 1+ carry bit b4 sum = ↓ wait time
will 1 appear in prev value?
∴ use Cin

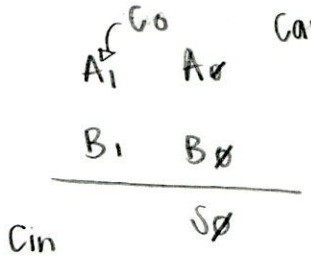
Sum delay =

(3 gates w/* × 4 sec) + 8 last FA = 20 nano seconds

Carryout delay =

3 gates × 4 sec + 16 last FA = 28 nano seconds

test if any before produced a carry out of 1



$$S1 = C0 \oplus (A0 + B0)$$

$$C1 = A1\,B1 + B1\,C0$$

$$C0 = A0\,B0 + B0\,Cin$$