

1.3. What is the 16-bit FP number representation of -5.375 in hex with 1-bit sign, 4-bit biased exponent, and 11-bit fraction, where bias offset = 7?

Step 1: Convert decimal to binary

First convert the whole #: 5

$$\begin{array}{r} 2 \\ \hline 2 \overline{) 5} \\ 4 \\ \hline 1 \end{array} \rightarrow \boxed{1}$$

$$5_{10} = 101_2$$

$$\begin{array}{r} 1 \\ \hline 2 \overline{) 2} \\ 2 \\ \hline 0 \end{array} \rightarrow 0$$

$$\begin{array}{r} 0 \\ \hline 2 \overline{) 1} \\ 0 \\ \hline 1 \end{array} \rightarrow \boxed{1}$$

* Remember: Write binary number starting from last division digit to the first division digit calculation

Now convert the fraction portion: .375

$$0.375 \times 2 = 0.75 = \boxed{0} + .75$$

$$0.75 \times 2 = 1.5 = 1 + .5$$

$$0.50 \times 2 = 1.0 = \boxed{1} + 0$$

$$0.0 \times 2 = 0.0 \rightarrow \text{End}$$

$$0.375_{10} = .011_2$$

$$5.375_{10} = 101.011_2$$

Step 2: Convert Binary to Scientific Notation

$$5.375_{10} = 101.011_2$$

$$\underline{101.011}_2 \Rightarrow 1.01011 \times 2^2 \quad \text{\# of decimal places moved to the left}$$

Step 3: Calculate Biased Exponent:

$$\text{Unbiased Exponent} = 1.01011 \times 2^{\boxed{2}}$$

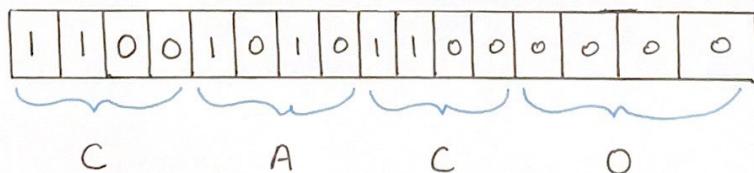
$$\text{Biased Exponent} = \text{Unbiased Exponent} + \text{Biased offset}$$

$$\text{Biased offset} = \underline{7}$$

$$\begin{aligned}\text{Biased Exponent} &= \underline{2} + \underline{7} \\ &= 9_{10} \\ &= 1001_2\end{aligned}$$

Step 4: Fusion of Floating Point

$$\begin{array}{c} 1.01011 \times 2^9 \\ \text{sign bit} \quad \text{Mantissa} \end{array} = \begin{array}{c} 1.01011 \times 2^{1001_2} \\ \text{Biased Exponent} \end{array}$$



0xCAC0

1.4. What is the real number equivalent to FP number 0x3400 with 1-bit sign, 4-bit biased exponent, 11-bit fraction, and bias offset = 7?

Step 1: Convert Hex to 16-bit binary

$$3 = 0011$$

$$4 = 0100$$

$$0 = 0000$$

$$0 = 0000$$

$$0x3400 \Rightarrow [0011 \ 0100 \ 0000 \ 0000]_2$$

Step 2: Calculate Unbiased Exponent:

sign bit = 0
mantissa = 10000000000
exponent = 0110

$$\text{Biased Exponent} = 0110_2 = 6$$

$$\boxed{\text{Biased Exponent} = \text{Unbiased Exponent} + \text{Biased offset}}$$

$$\begin{aligned}\text{Unbiased Exponent} &= \text{Biased Exponent} - \text{Biased offset} \\ &= 6 - 7 \\ &= -1\end{aligned}$$

Step 3: Convert Mantissa to Decimal

$$\text{Mantissa} = 1000000000000000_2$$

$$\text{Mantissa} = 1000000000000000$$

$$= 1 \times 2^{-1}$$

$$= 0.5$$

Step 4: Find the real number equivalent using the equation:

$$\boxed{\text{Real #} = (-1)^S \times (1 + m) \times 2^E}$$

S: Sign Bit

m: Mantissa

E: Unbiased Exponent

$$= (-1)^0 \times (1 + 0.5) \times 2^{-1}$$

$$= 1.5 \times 2^{-1}$$

$$= 0.75$$

1.5. What is the real number equivalent to FP number 0x3400 with 1-bit sign, 4-bit biased exponent, 11-bit fraction, and bias offset = 8?

Step 1: Convert Hex to 16-bit Binary

$$3 = 0011$$

$$4 = 0100$$

$$0 = 0000$$

$$0 = 0000$$

$$0x3400 \Rightarrow [0011 \ 0100 \ 0000 \ 0000]_2$$

Step 2: Calculate Unbiased Exponent

$$\text{Biased Exponent} = 0110_2 = 6$$

$$\begin{aligned}\text{Unbiased Exponent} &= \text{Biased Exponent} - \text{Biased offset} \\ &= 6 - 8 \\ &= -2\end{aligned}$$

Step 3: Convert Mantissa to Decimal

$$\begin{aligned}\text{Mantissa} &= 100\ 0000\ 0000 \\ &= 1 \times 2^{-1} \\ &= 0.5\end{aligned}$$

Step 4: Find the real number equivalent using the equation:

$$\begin{aligned}\text{Real } \# &= (-1)^S \times (1 + m) \times 2^E \\ &= (-1)^0 \times (1 + 0.5) \times 2^{-2} \\ &= 0.375\end{aligned}$$

1.14. What is a Von Neumann architecture bottleneck?

The performance of CPUs has increased at a faster rate than that of memory. Therefore, the Von Neumann architecture presents a communication bottleneck between a faster CPU and a slower memory.

1.6 What is the biggest positive FP number (in Decimal) that can be represented in 16-bit format using 1-bit sign, 4-bit biased exponent, and 11-bit fraction, where bias is 7?

largest floating number has exp value of 1110
and fractional part has all 1's

0 1110 1111111111

Convert 1110 to decimal

$$\begin{aligned} &= 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= 8 + 4 + 2 + 0 \\ &= 14 \end{aligned}$$

Calculate Biased Exponent:

$$14 - 7 = 7$$

IEEE-754 Decimal value is 1.frac $\times 2^{\text{Exponent}}$

$$\begin{aligned} &= 1.1111111111 \times 2^7 \\ &= 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} + 1 \times 2^{-6} + 1 \times 2^{-7} + 1 \times 2^{-8} \\ &\quad + 1 \times 2^{-9} + 1 \times 2^{-10} + 1 \times 2^{-11} \\ &= 1.999511719 \end{aligned}$$

So 1.999511719×2^7 in decimal is 255.9375

1.8 Do the following assuming 16-bit FP numbers with 4-bit bias exponent, bias = 7, and 11-bit fraction.

- a) What real number does an FP number with sign = 0, bias exponent = 1 and fraction = 0 represent? (Answer in 4 decimal places)

0	0001	000000000000
Sign	Exponent	Fraction

$$\text{Bias} = 7$$

$$\text{so Exponent becomes } 1 - 7 = -6$$

the binary number is

1.0000000000 ~~and all zeros after 64 (in base 2)~~
so Mantissa

$$\text{Unbiased Exponent} = \text{Biased Exponent} - \text{Bias offset}$$

Mantissa value

Find the real number equivalent using the following equation:

$$\text{Real #} = (-1)^S \times (1 + m) \times 2^E$$

S: Sign Bit

$$= (-1)^0 \times (1 + 0) \times 2^{-6}$$

m: Mantissa

$$= 0.015625$$

E: Unbiased Exponent

$$\boxed{\text{Real #} = 0.0156}$$

2.4. Proof Demorgan's Theorem $\overline{x+y} = \overline{x}\overline{y}$ by creating truth tables for $f = \overline{x+y}$ and $g = \overline{x}\overline{y}$. Are the two truth tables identical?

$$f = \overline{x+y}$$

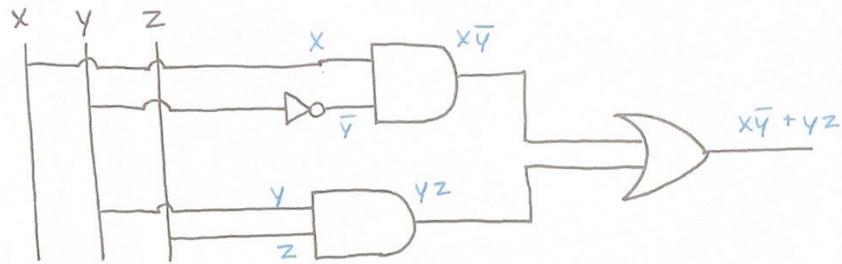
x	y	$x+y$	$\overline{x+y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

$$g = \overline{x}\overline{y}$$

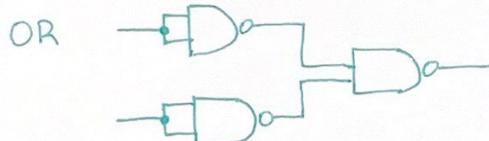
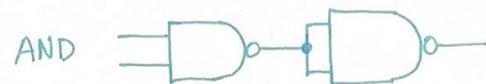
x	y	\overline{x}	\overline{y}	$\overline{x}\overline{y}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

As shown in the last column of each table, we see they match, therefore $\overline{x+y} = \overline{x}\overline{y}$

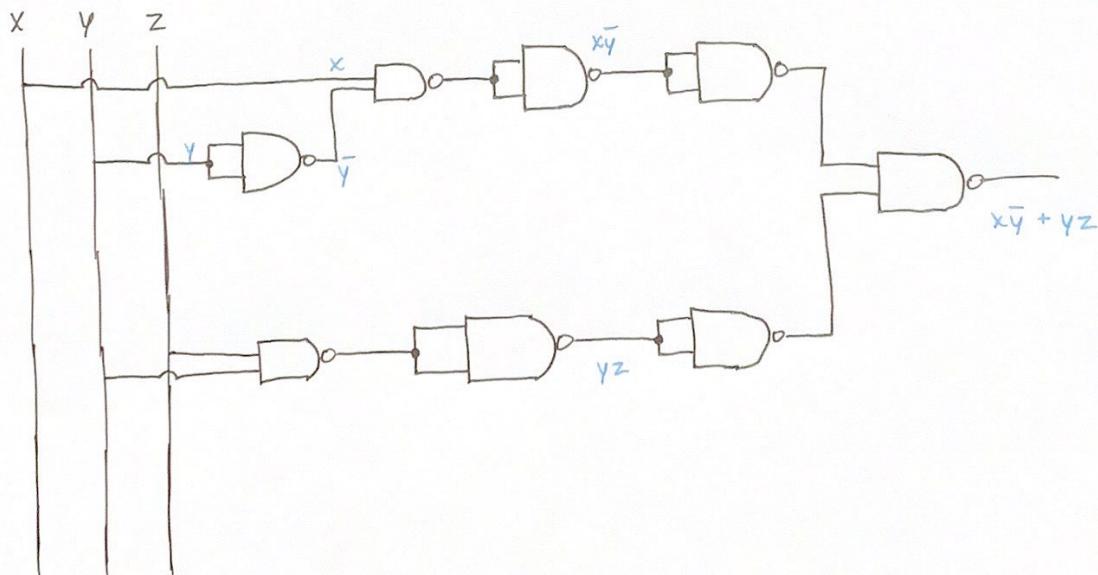
2.5. Draw the circuit schematic for $f = x\bar{y} + yz$ and then convert the schematic to NAND gates using the steps illustrated in the textbook.



We can convert NOT, AND, and OR Gates to NAND using the following configurations



Now replace each gate with the appropriate configuration

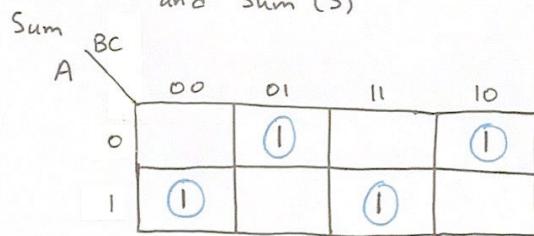


1. Design a Single cell - 1 bit Carry Propagate (Ripple Carry Adder) full adder.

a. Generate the truth table

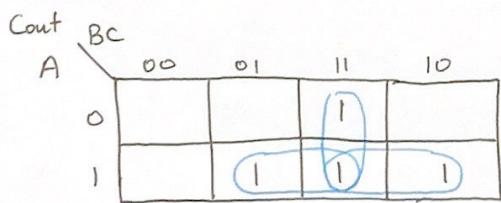
A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

b. Using K-map, determine the logical expression for Carry out (C-out) and Sum (S)



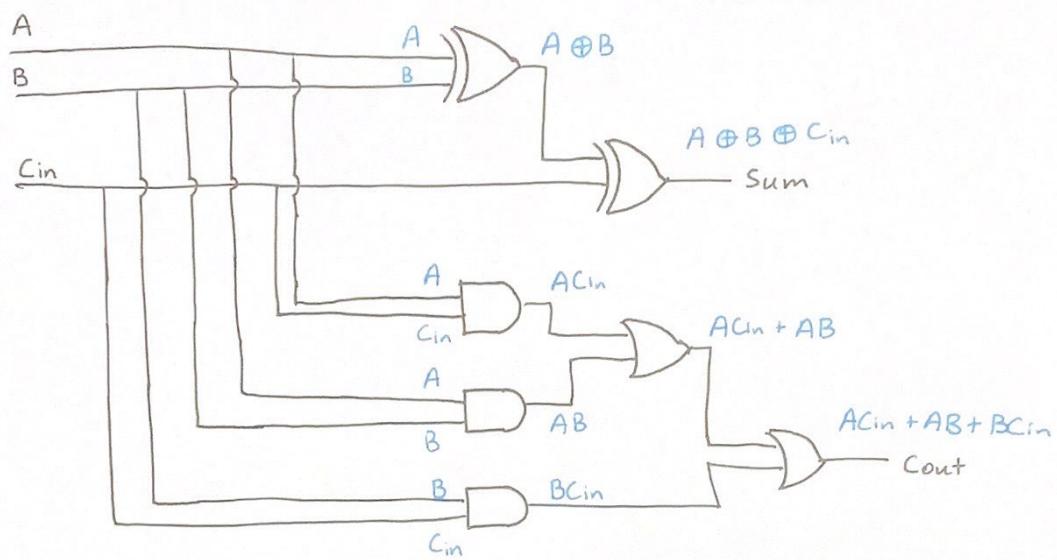
$$\begin{aligned}
 \text{Sum} &= A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}BC \\
 &= A(\bar{B}\bar{C} + BC) + \bar{A}(\bar{B}C + B\bar{C}) \\
 &= A(B \oplus C) + \bar{A}(B \oplus C)
 \end{aligned}$$

$$\text{Sum} = A \oplus B \oplus \text{Cin}$$



$$\text{Cout} = AC_{\text{int}} + AB + BC_{\text{in}}$$

c. Based on the logical expression, create the schematic diagram for full adder.

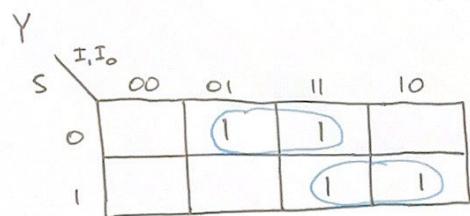


2. Design a 1 bit, 2 to 1 multiplexer (Mux). Outputs Y when S=0; X when S=1.

a. Generate the truth table.

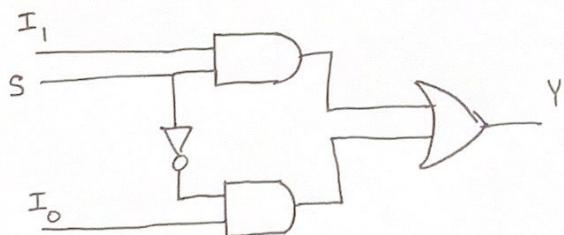
S	I ₁	I ₀	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

b. Using K-map, determine the logical expression for output.



$$Y = \bar{S}I_0 + SI_1$$

c. Based on the logical expression, create the schematic diagram for Mux.



5.8. Design a Moore sequence recognizer that detects the nonoverlapping sequence "101." Use binary encoded state labels and design and draw the circuit schematic similar to the one shown in Fig. 5.16. (4 pts)

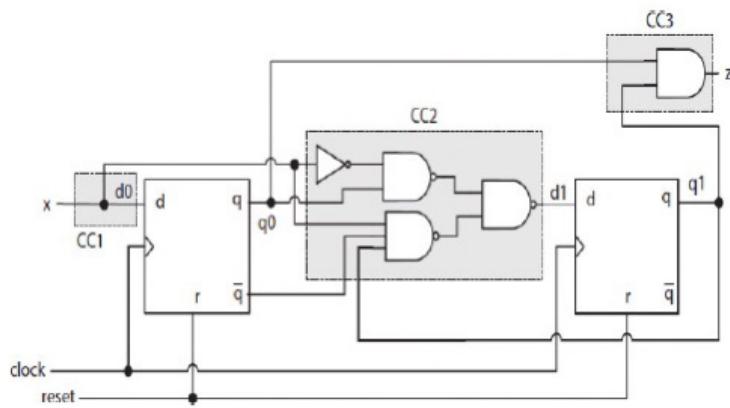
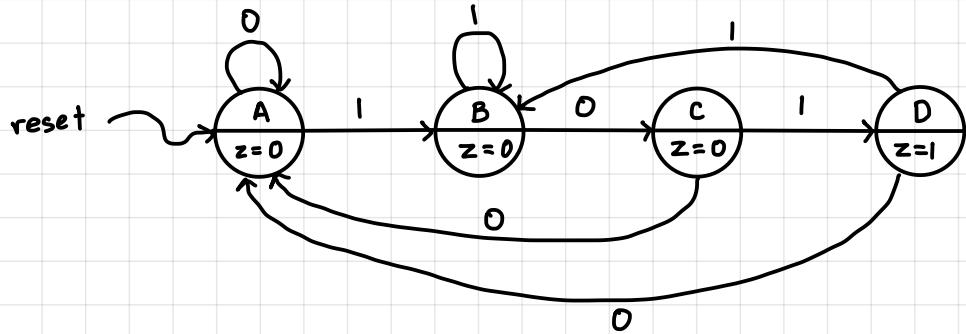


FIGURE 5.16 An alternative and typical layout for the circuit shown in Fig. 5.15.

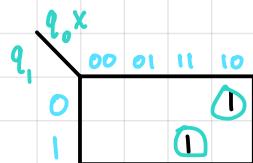


Number of bits = $\log_2 [k]$, where $k = \# \text{ of states}$

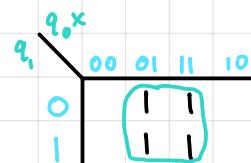
$$\log_2 (4) = 2$$

$$A = 00, B = 01, C = 10, D = 11$$

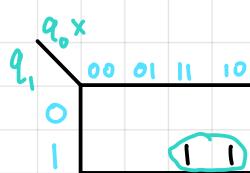
Current State	Input	Next State	Output
$q_1 \ q_0$	x	$d_1 \ d_0$	z
0 0	0	0 0	0
0 0	1	0 1	0
0 1	0	1 0	0
0 1	1	0 1	0
1 0	0	0 0	0
1 0	1	1 1	0
1 1	0	0 0	1
1 1	1	0 1	1



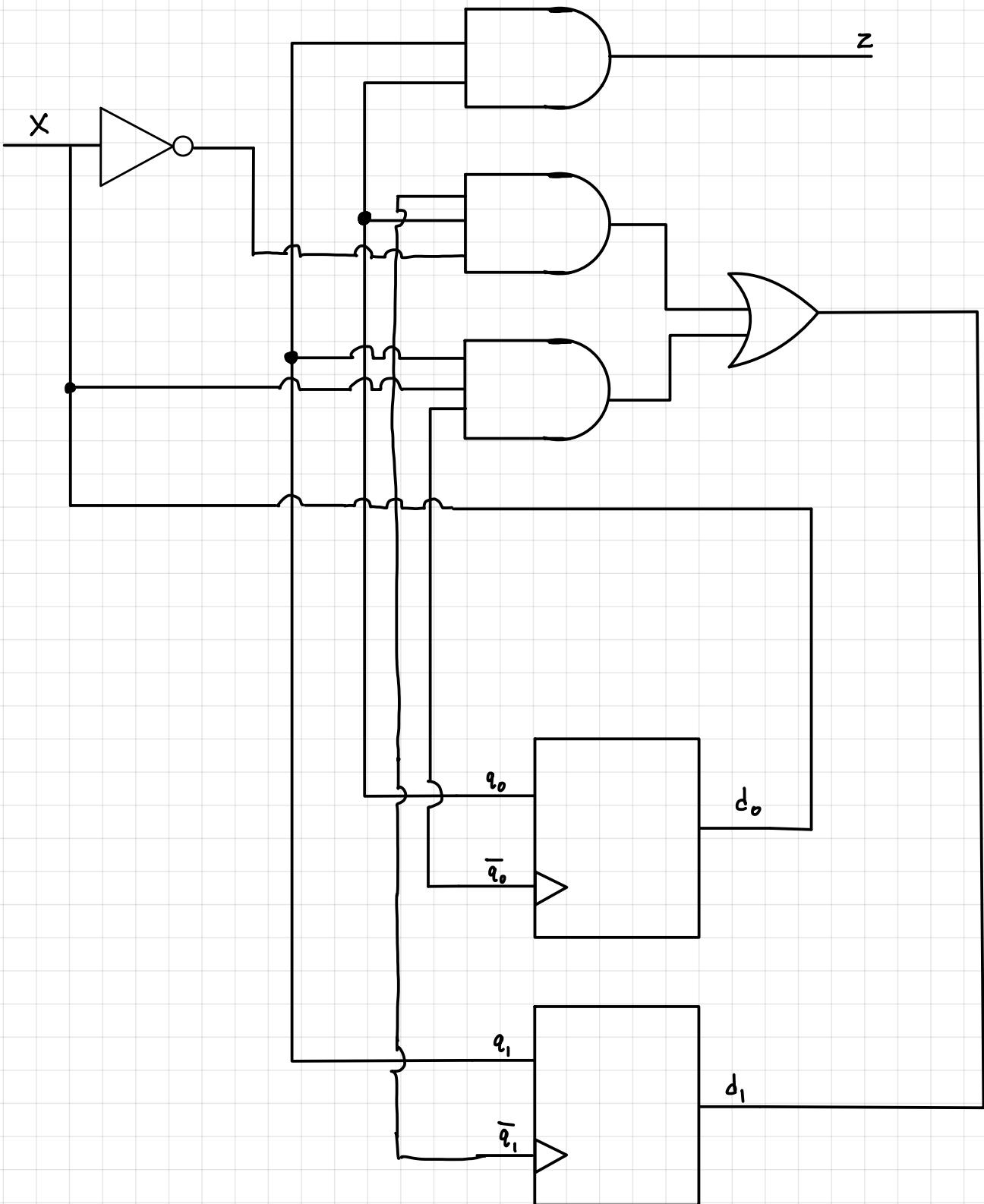
$$d_1 = \bar{q}_1 q_0 \bar{x} + q_1 \bar{q}_0 x$$



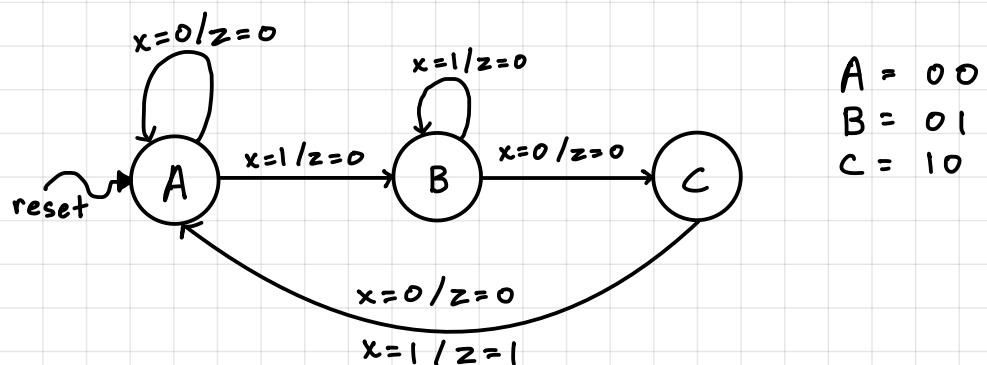
$$d_0 = x$$



$$z = q_1 q_0$$



5.9. Design a Mealy sequence recognizer that detects the nonoverlapping sequence "101." Use binary encoded state labels and draw the circuit schematic similar to the one shown in Fig. 5.16. (4 pts)



$$\begin{aligned} A &= 00 \\ B &= 01 \\ C &= 10 \end{aligned}$$

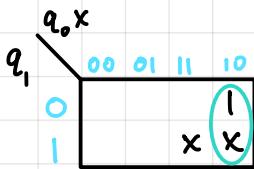
Number of bits = $\log_2 [k]$, where $k = \# \text{ of states}$

$$\log_2 (3) = 2$$

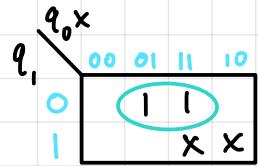
Current State $q_1 \ q_0$	Input x	Next State $d_1 \ d_0$	Output z
0 0	0	0 0	0
0 0	1	0 1	0
0 1	0	1 0	0
0 1	1	0 1	0
1 0	0	0 0	0
1 0	1	0 0	1
1 1	0	x x	x
1 1	1	x x	x

$$d_1 = \bar{q}_1 q_0 \bar{x} \quad d_0 = \bar{q}_1 \bar{q}_0 x + \bar{q}_1 q_0 x$$

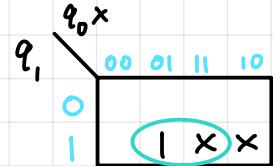
$$z = q_1 \bar{q}_0 x$$



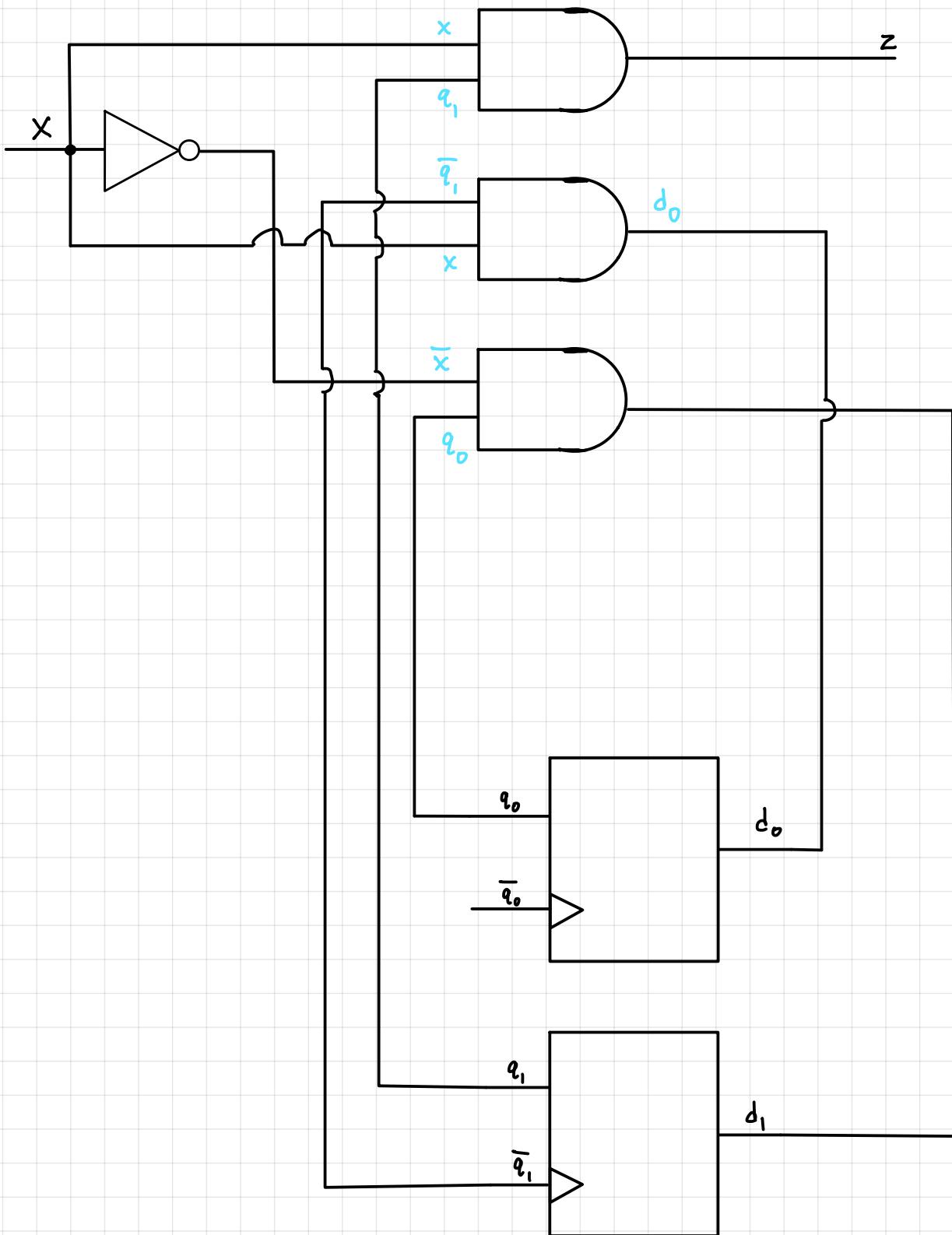
$$d_1 = q_0 \bar{x}$$



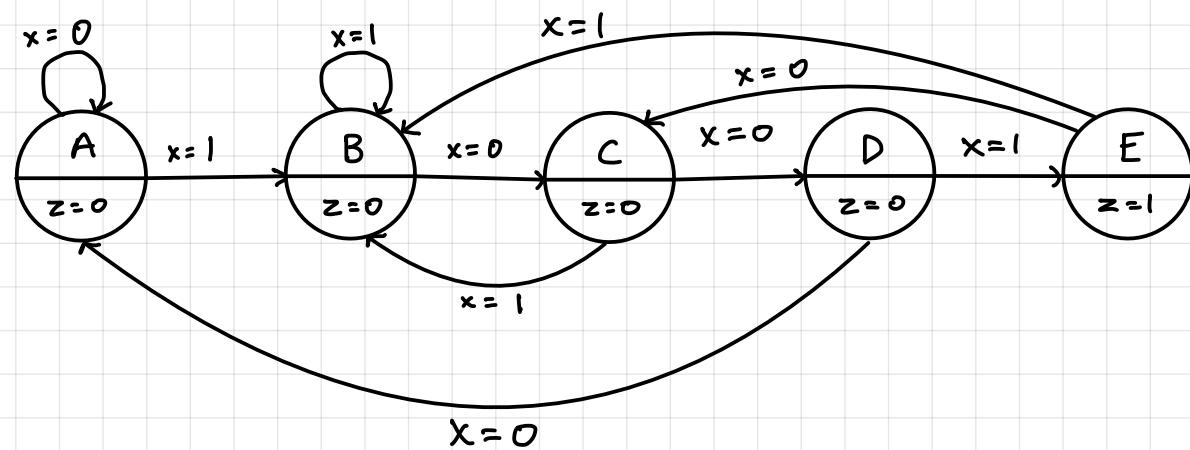
$$d_0 = \bar{q}_1 x$$



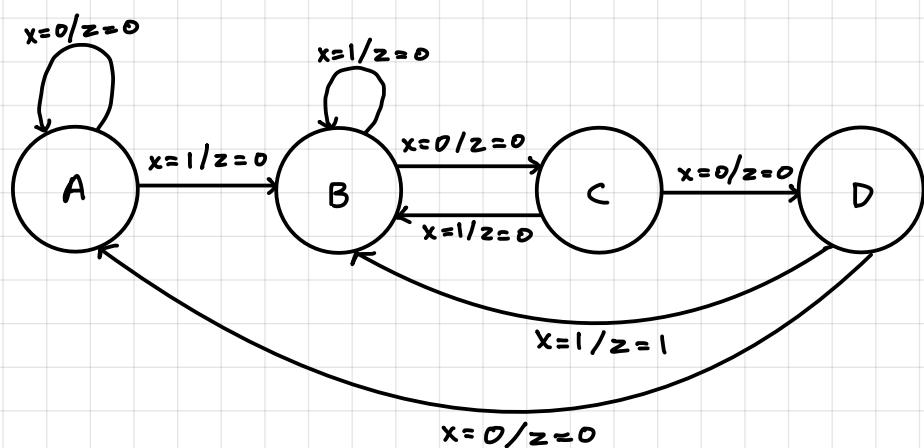
$$z = q_1 x$$



5.10. Design a Moore sequence recognizer that detects the overlapping sequence "1001." Use binary encoded state labels. . (Step 1. FSD only) (4 pts)



5.11. Design a Mealy sequence recognizer that detects the overlapping sequence "1001." Use binary encoded state labels. . (Step 1. FSD only) (4 pts)



Problem I: Consider the sequential circuit in Figure 5.31 where the adder is a CLA. Its simplified circuit is shown in Figure 5.32. Assuming that the Flip-Flops register set-up time, clock-to-q, and clock-skew are each 0.1 ns, determine the upper bound for its clock frequency. (4 pts)

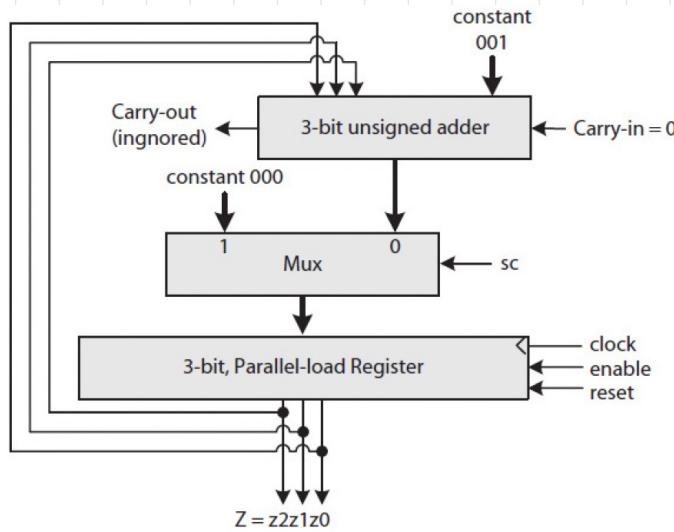


FIGURE 5.31 A synchronously cleared bit-parallel mod-8 up-counter.

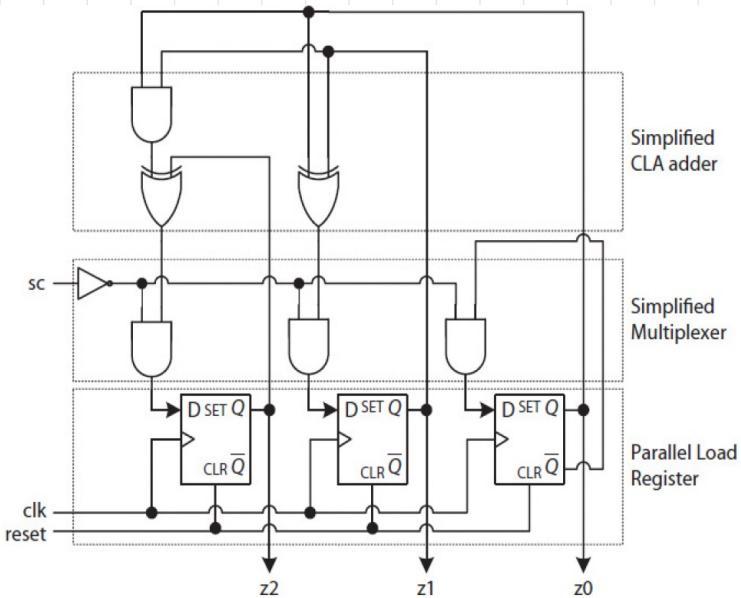


FIGURE 5.32 A synchronously cleared bit-parallel mod-8 up-counter using a simplified CLA adder and a simplified MUX.

The longest path delay for the simplified circuit is:

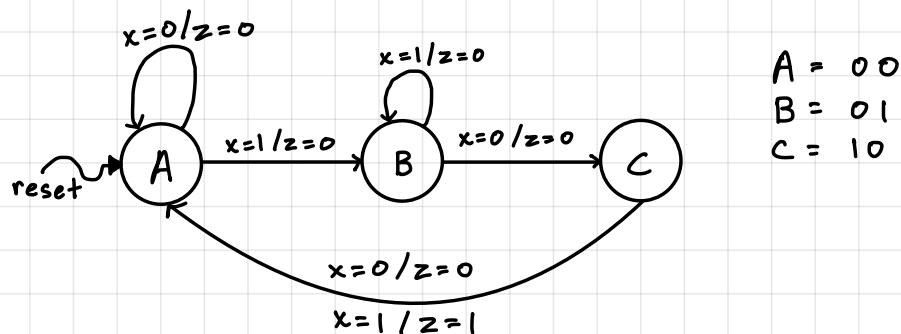
$$\Delta_{AND} + \Delta_{XOR} + \Delta_{AND} = 0.2 + 0.3 + 0.2 = 0.7 \text{ ns}$$

$$T \geq 0.7 + 3(0.1) = 1.0 \text{ ns}$$

$$F \leq 1 \text{ GHz}$$

Problem II: Textbook problem 5.9 assuming the unknown state are ignored (don't care) in the design. (5 pts)

5.9. Design a Mealy sequence recognizer that detects the nonoverlapping sequence "101." Use binary encoded state labels and draw the circuit schematic similar to the one shown in Fig. 5.16.



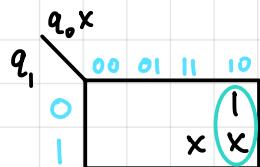
Number of bits = $\log_2 [k]$, where $k = \# \text{ of states}$

$$\log_2 (3) = 2$$

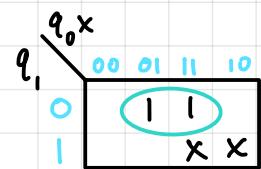
Current State $q_1\ q_0$	Input x	Next State $d_1\ d_0$	Output z
0 0	0	0 0	0
0 0	1	0 1	0
0 1	0	1 0	0
0 1	1	0 1	0
1 0	0	0 0	0
1 0	1	0 0	1
1 1	0	x x	x
1 1	1	x x	x

$$d_1 = \bar{q}_1 q_0 \bar{x} \quad d_0 = \bar{q}_1 \bar{q}_0 x + \bar{q}_1 q_0 x$$

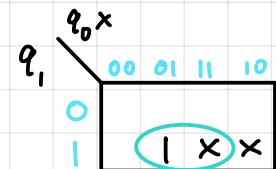
$$z = q_1 \bar{q}_0 x$$



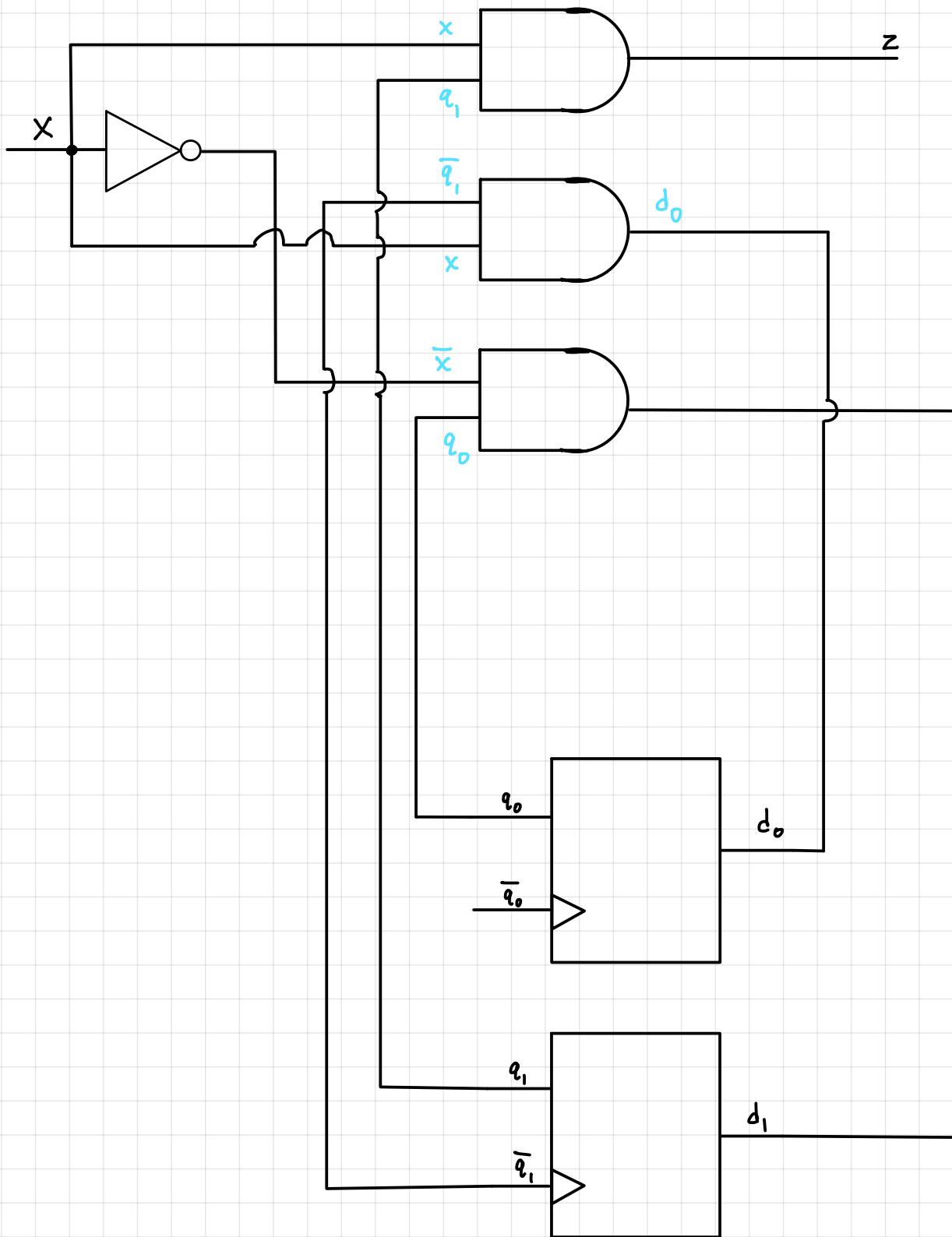
$$d_1 = q_0 \bar{x}$$



$$d_0 = \bar{q}_1 x$$

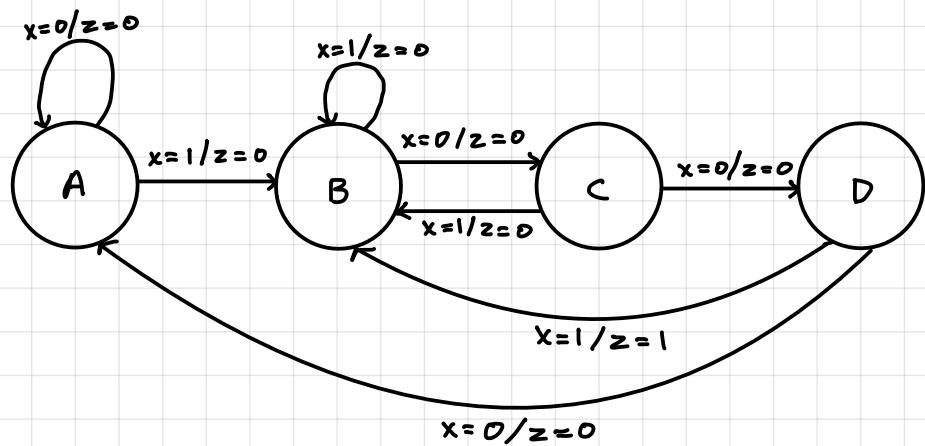


$$z = q_1 x$$



Problem III: Textbook problem 5.11 (only FSD) (3 pts)

5.11. Design a Mealy sequence recognizer that detects the overlapping sequence "1001." Use binary encoded state labels.



For problems 6.1 through 6.3: Suppose the propagation delay of an 8-bit adder is 0.8ns, adder/subtractor is 1.1ns, 2-to-1 mux is 0.3ns, and 4-to-1 mux is 0.6ns. Also, assume register setup time (0), clock-to-q (0), and clock-skew (0) are all 0.05ns. (15 points)

6.1 Calculate the required maximum clock frequency for each of the following data paths:

- Single-cycle data path in Fig. 6.2.

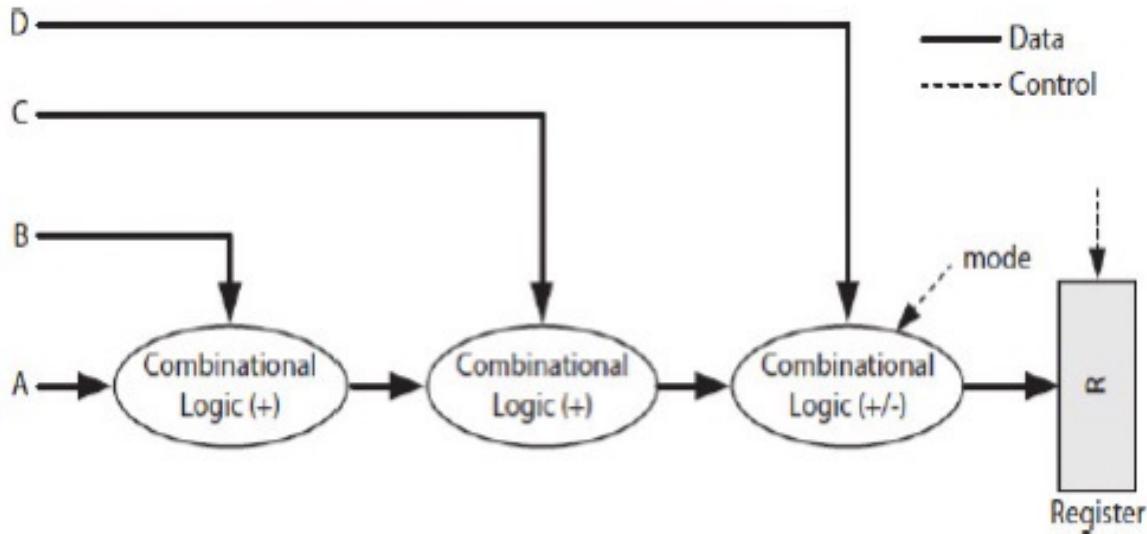


FIGURE 6.2 A single-cycle two-function data path that computes either $A + B + C + D$ or $A + B + C - D$ in one clock cycle.

$$f \leq \frac{1}{\tau} \text{ cycles/second or Hertz} \quad (4.2 \text{ equation})$$

$$\tau \geq 2\Delta_{ADD} + \Delta_{ADD/SUB} + \tau_{st} + \tau_{cq} + \tau_{cs} \quad (6.1 \text{ equation})$$

$$\tau \geq 2(0.8 \text{ ns}) + (1.1 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns})$$

$$f \leq \frac{1}{\tau} = 350.877193 \Rightarrow f \leq 350.877 \text{ MHz}$$

For problems 6.1 through 6.3: Suppose the propagation delay of an 8-bit adder is 0.8ns, adder/subtractor is 1.1ns, 2-to-1 mux is 0.3ns, and 4-to-1 mux is 0.6ns. Also, assume register setup time (0), clock-to-q (0), and clock-skew (0) are all 0.05ns. (15 points)

6.1 Calculate the required maximum clock frequency for each of the following data paths:

b. Multi-cycle data path in Fig. 6.3.

Cycle 2: $R \leftarrow R + B$

Cycle 3: $R \leftarrow R + C$

Cycle 4: If mode == 0 then $R \leftarrow R + D$; otherwise, $R \leftarrow R - D$

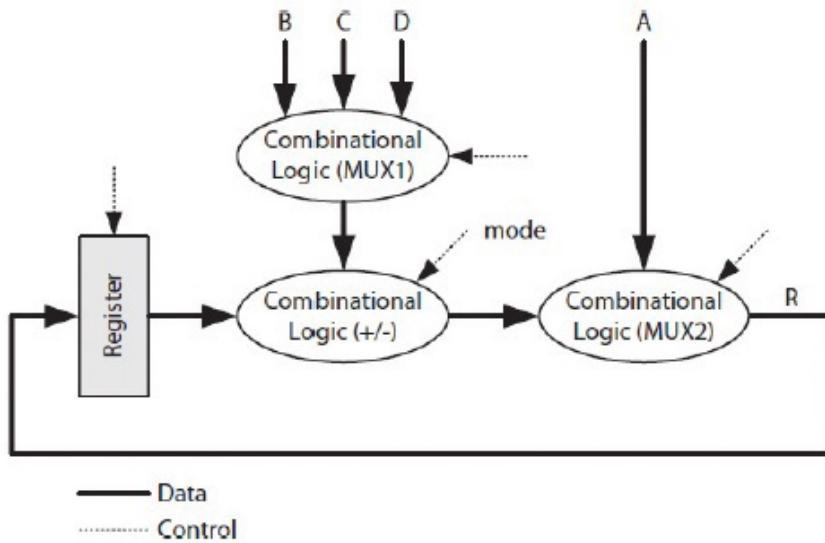


FIGURE 6.3 A multicycle data path requiring four clock cycles to compute $A + B + C + D$ or $A + B + C - D$.

$$f \leq \frac{1}{T} \text{ cycles/second or Hertz} \quad (4.2 \text{ equation})$$

$$T \geq \Delta_{\text{mux1}} + \Delta_{\text{add/sub}} + \Delta_{\text{mux2}} + \gamma_{\text{st}} + \gamma_{\text{cq}} + \gamma_{\text{cs}} \quad (6.2 \text{ equation})$$

$$\gamma \geq (0.3 \text{ ns}) + (1.1 \text{ ns}) + (0.6 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns})$$

$$f \leq \frac{1}{T} = 465.116279.1 \Rightarrow f \leq 465.116 \text{ MHz}$$

For problems 6.1 through 6.3: Suppose the propagation delay of an 8-bit adder is 0.8ns, adder/subtractor is 1.1ns, 2-to-1 mux is 0.3ns, and 4-to-1 mux is 0.6ns. Also, assume register setup time 0, clock-to-q 0, and clock-skew 0 are all 0.05ns. (15 points)

- 6.1 Calculate the required maximum clock frequency for each of the following data paths:

c. Pipelined data path in Fig. 6.4.

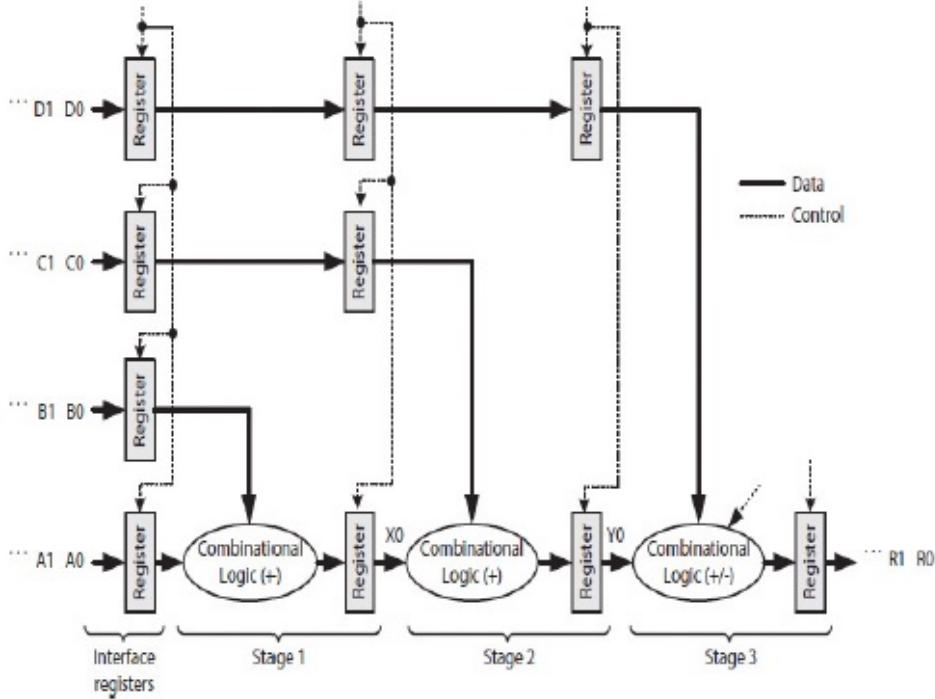


FIGURE 6.4 A two-function pipelined data path computing a stream of quantities $A_i + B_i + C_i \pm D_i$ for $i = 0, 1, 2, \dots$

$$f \leq \frac{1}{\tau} \text{ cycles/second or Hertz} \quad (4.2 \text{ equation})$$

$$\tau \geq \Delta_{ADD/SUB} + \tau_{st} + \tau_{cq} + \tau_{cs} \quad (6.3 \text{ equation})$$

$$\tau \geq (1.1 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns})$$

$$f \leq \frac{1}{\tau} = 800,000,000 \Rightarrow f \leq 800 \text{ MHz}$$

- 6.2 Estimate the speedup between the following data paths when generating $N = 1000$ quantities $A_i + B_i + C_i \pm D_i$ for $i = 0, 1, 2, \dots, 999$. Ignore the data reading and writing delays. (10 points)

a. Problem 6.1a vs. 6.1c

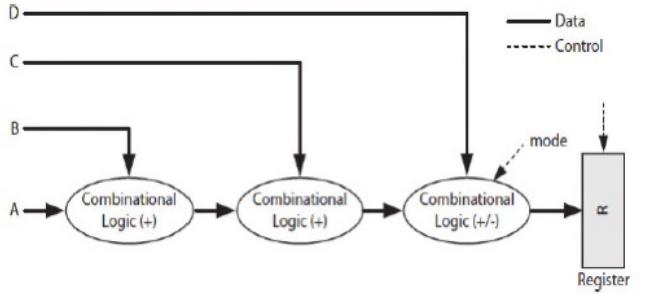


FIGURE 6.2 A single-cycle two-function data path that computes either $A + B + C + D$ or $A + B + C - D$ in one clock cycle.

VS

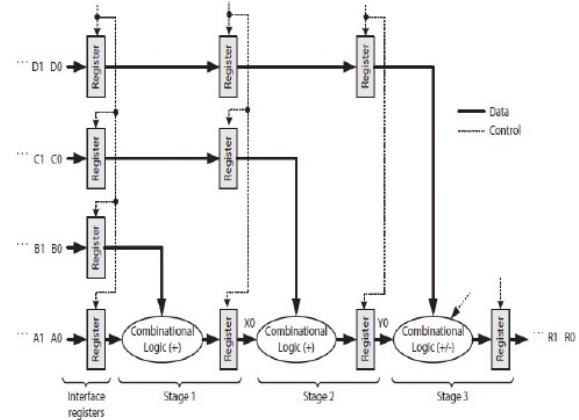


FIGURE 6.4 A two-function pipelined data path computing a stream of quantities $A_i + B_i + C_i \pm D_i$ for $i = 0, 1, 2, \dots$

$$S = \frac{\gamma_{\text{single-cycle}}}{\gamma_{\text{pipeline}}} = \frac{Nk\gamma}{k\gamma + (N-1)\gamma} = \frac{Nk}{k+N-1} \quad (6.6 \text{ Equation})$$

$$= \frac{2(0.8 \text{ ns}) + (1.1 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns})}{(1.1 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns}) + (0.05 \text{ ns})} \quad (\text{Using First eq.})$$

$S = 2.28 \text{ seconds}$

$$= \frac{(1,000)(3)}{(3) + (1,000-1)} \quad (\text{Using third eq.})$$

$S = 2.994 \text{ seconds}$

This is the correct answer

7.10 Consider a 32-bit data bus SDRAM. Given that the clock frequency of the bus is 200MHz, what is the peak memory bandwidth in megabyte per second (MBs)? **(5 pts)**

$$\text{clock frequency} = 200 \text{ MHz}$$

$$\text{width of data bus} = 32 \text{ bits} \Rightarrow 8 \text{ bits} = 1 \text{ byte} \Rightarrow \frac{32}{8} = 4 \text{ bytes}$$

$$\text{Bandwidth} = \text{Bus width} \times \text{SRAM Frequency}$$

$$= 4 \times 200 \times 10^6$$

$$= 800 \text{ MBs}$$

7.11 Consider a 64-bit data bus SDRAM. Given that the clock frequency of the bus is 200MHz, what is the peak memory bandwidth in megabyte per second (MBs)? **(5 pts)**

$$\text{clock frequency} = 200 \text{ MHz}$$

$$\text{width of data bus} = 64 \text{ bits} \Rightarrow 8 \text{ bits} = 1 \text{ byte} \Rightarrow \frac{64}{8} = 8 \text{ bytes}$$

$$\text{Bandwidth} = \text{Bus width} \times \text{SRAM Frequency}$$

$$= 8 \times 200 \times 10^6$$

$$= 1,600 \text{ MBs}$$

7.12 Consider a 32-bit data bus DDR SDRAM. Given that the clock frequency of the bus is 200MHz, what is the peak memory bandwidth in megabyte per second (MBs)? **(5 pts)**

$$\text{clock frequency} = 200 \text{ MHz}$$

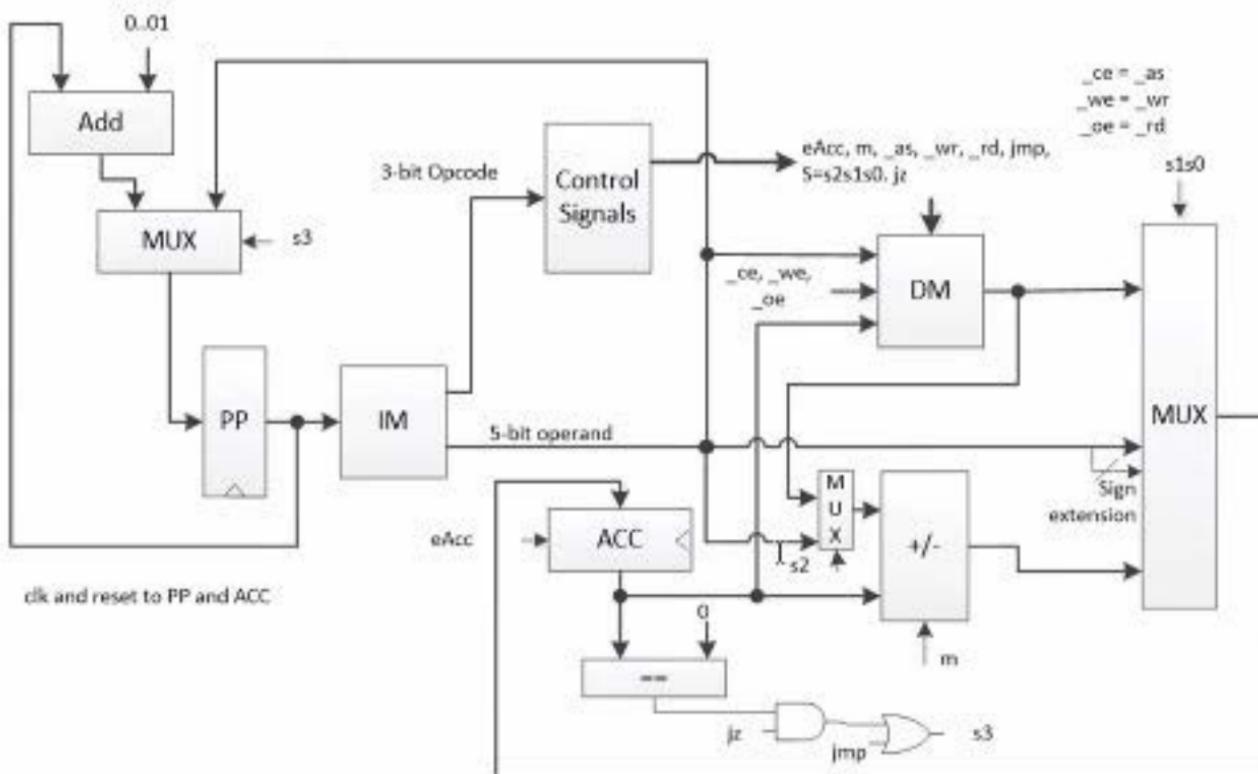
$$\text{width of data bus} = 32 \text{ bits} \Rightarrow 8 \text{ bits} = 1 \text{ byte} \Rightarrow \frac{32}{8} = 4 \text{ bytes}$$

$$\text{Bandwidth} = \text{Bus width} \times 2 \times \text{SRAM Frequency}$$

$$= 4 \times 2 \times 200 \times 10^6$$

$$= 1,600 \text{ MBs}$$

- a) Draw a data path for the CPU assuming the DM has separate input and output bus as in the data path shown in Fig. 8.7. Do not include additional data paths not used by the instructions. (15 pts)



8.3 An Acc-ISA CPU executes the following instructions using 3-bit op-codes and 5-bit address or 2's complement data. Do the following:

LD address	$\text{Acc} \leftarrow \text{Memory}[\text{address}], \text{read from LM2}$
LD data	$\text{Acc} \leftarrow \text{data} (\text{a 2's complement number, sign extended})$
ADD data	$\text{Acc} \leftarrow \text{Acc} + \text{data} (\text{data is a 2's complement number, sign extended})$
SUB data	$\text{Acc} \leftarrow \text{Acc} - \text{data} (\text{data is a 2's complement number, sign extended})$
ADD (address)	$\text{Acc} \leftarrow \text{Acc} + \text{Memory}[\text{address}]$
STM (address)	$\text{M}[\text{address}] \leftarrow \text{Acc}$
SUB (address)	$\text{Acc} \leftarrow \text{Acc} - \text{Memory}[\text{address}]$
JMP address	$\text{PP} \leftarrow \text{address}$
JZ address	

- a) Draw a data path for the CPU assuming the DM has separate input and output bus as in the data path shown in Fig. 8.7. Do not include additional data paths not used by the instructions. (15 pts)

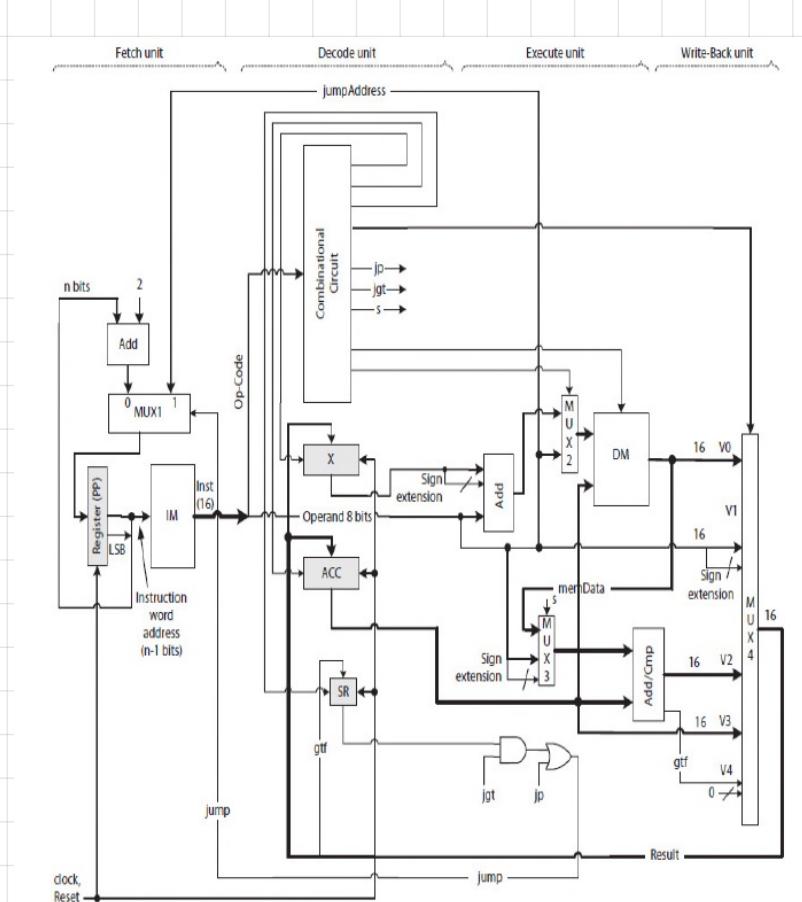
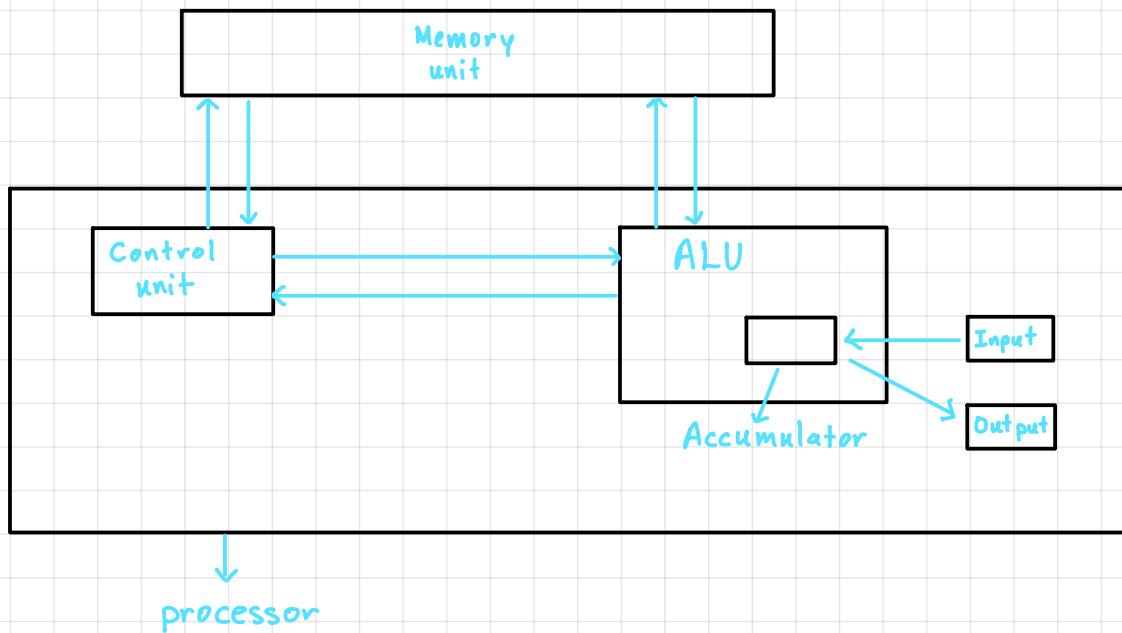
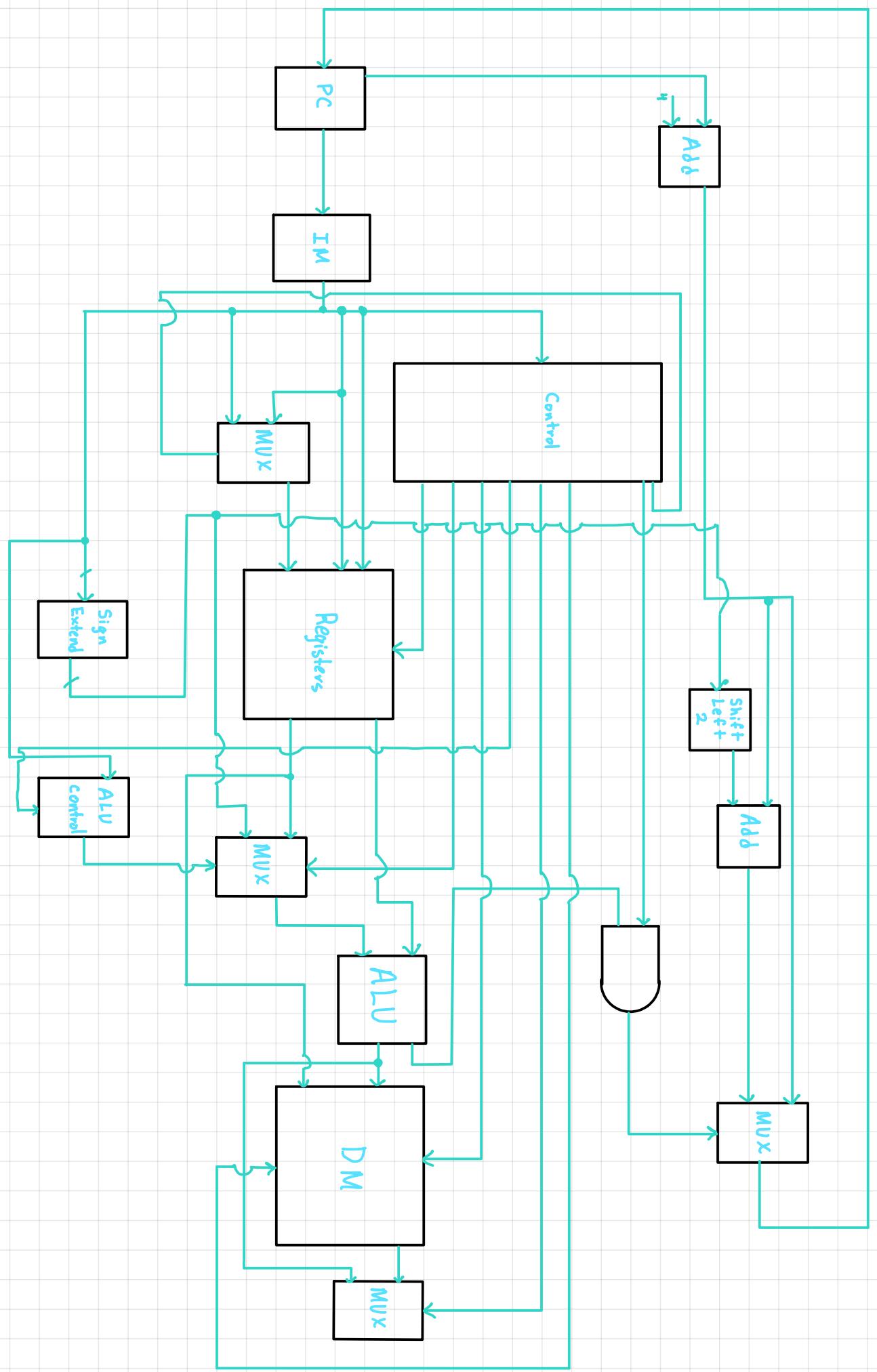


FIGURE 8.7 The Acc-ISA single-cycle data path to execute the program in Example 8.2.





Problem I. Computation is performed by a RISC ISA. $A = B * (C + D)$. What is the value in R4 after the execution of code line # 6: (B = 5; C = 10; D = 15) ie: Code line # 6 has been completed. (5 pts)

R4 = _____

RISC-ISA: Example of assembly program

1. LD R1, (C)
2. LD R2, (D)
3. ADD R3, R1, R2
4. LD R4, (B)
5. MUL R5, R3, R4
6. ST (A), R5

1. LD R1, (C) // load c=10 into R₁
2. LD R2, (D) // load d=15 into R₂
3. ADD R3, R1, R2 // $R_1 + R_2 = 10 + 15 = 25$; store result in R₃
4. LD R4, (B) // load b=5 into R₄
5. MUL R5, R3, R4 // $R_3 \times R_4 = 25 \times 5 = 125$; store result in R₅
6. ST (A), R5 // Store R₅ = 125 into A

∴ The value of R4 after the execution of code line
#6 is R4 = 5