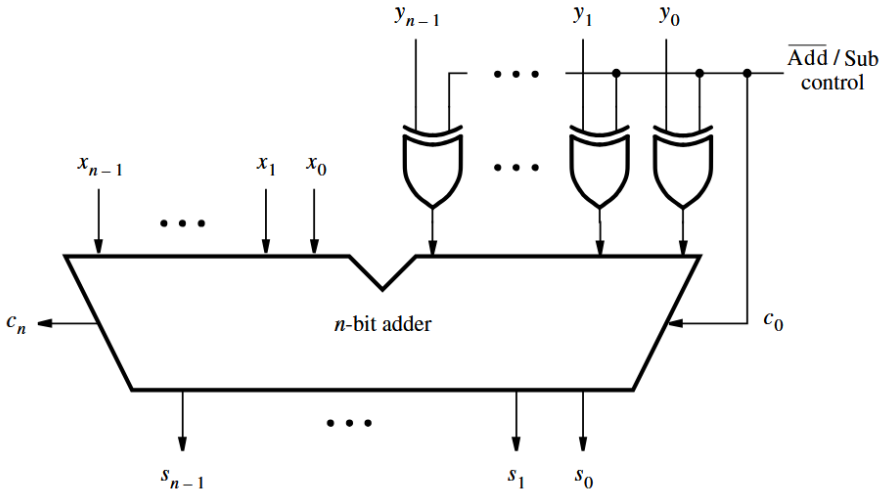
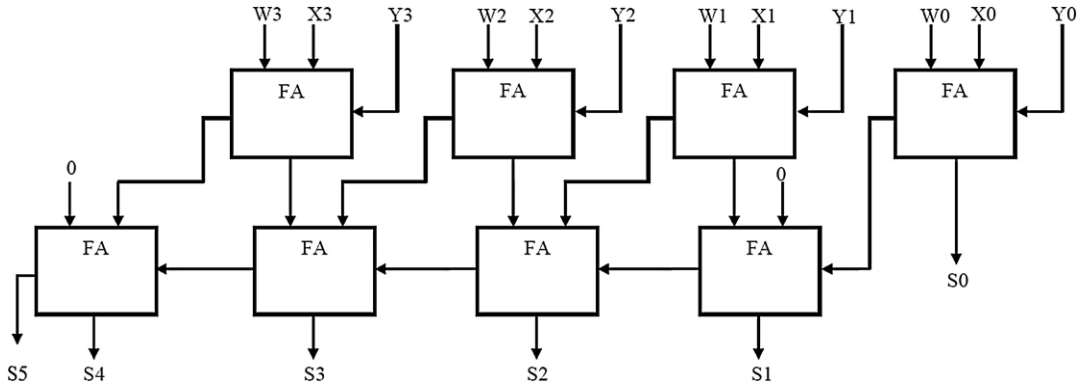
Assignment-03: Structural Modelling

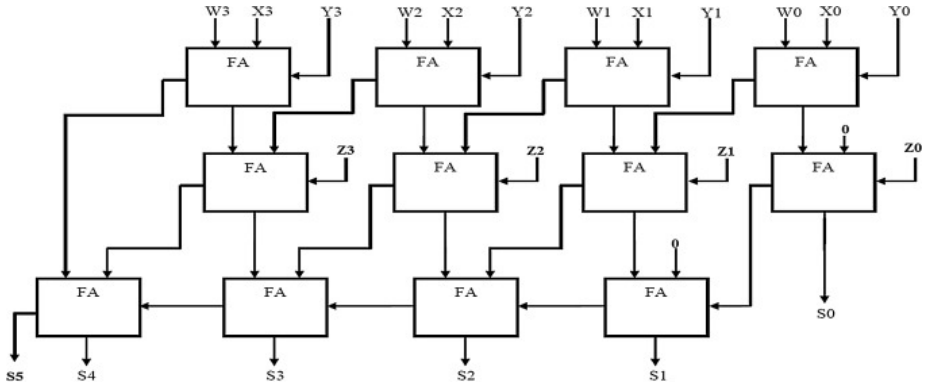
- Design 4:1 using 2:1 multiplexer
- Design 8:1 using 2:1 multiplexer
- Design 1:2 demultiplexer,
- Design 1:8 demux using 1:2 demux
- Design 5:1 Mux using 2:1 multiplexer using structural design
- Design Adder cum Subtractor using structural design (4-bit)



- Design Carry Save Adder (CSA) (Three input carry save addition)



- Design Carry Save Adder(CSA) (Four input carry save addition)



- Design 4-bit Multiplier circuits as below

Multiplicand M	(14)	1 1 1 0
Multiplier Q	(11)	× 1 0 1 1
		1 1 1 0
		1 1 1 0
		0 0 0 0
		1 1 1 0
Product P	(154)	1 0 0 1 1 0 1 0

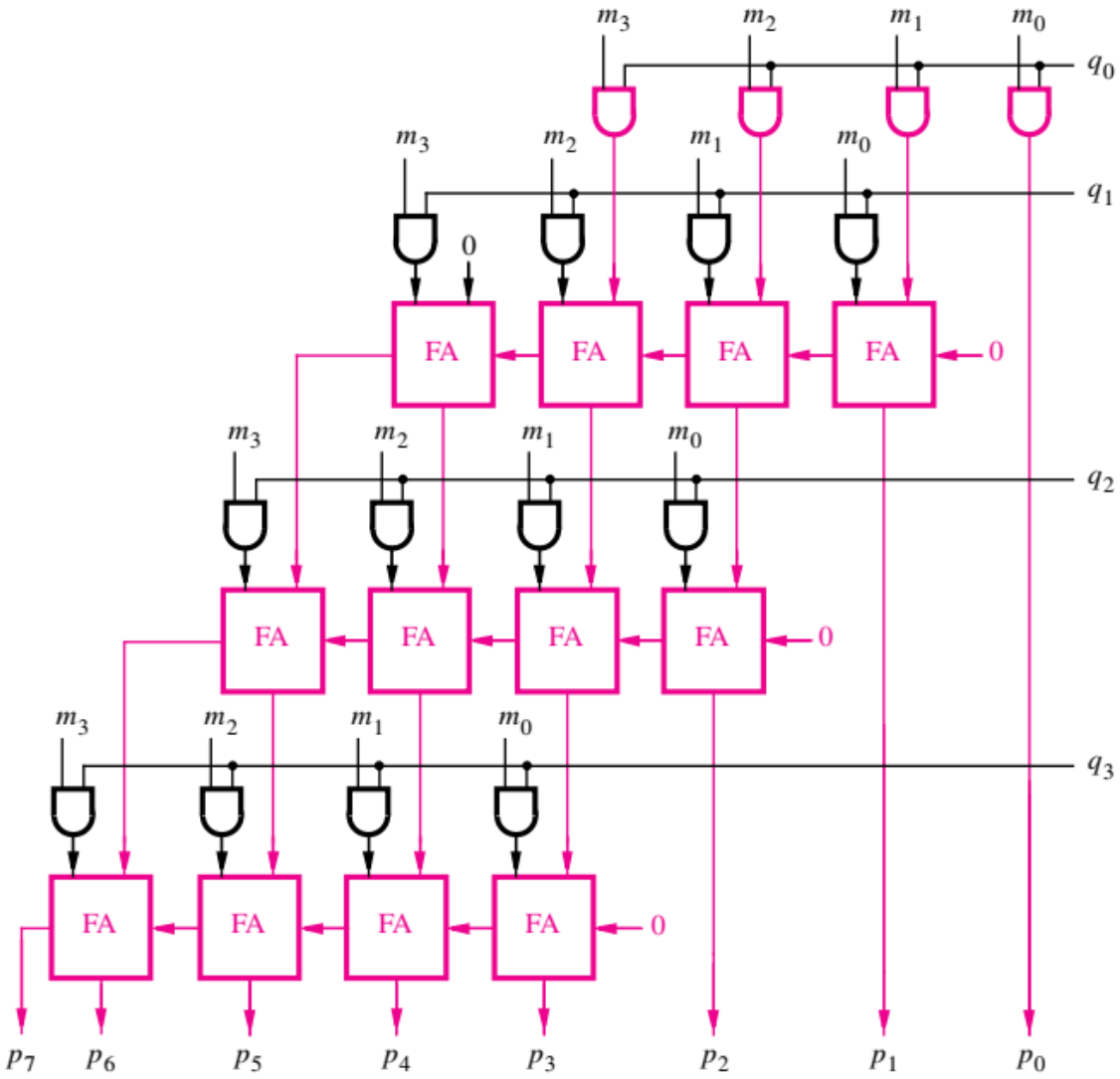
(a) Multiplication by hand

Multiplicand M	(14)	1 1 1 0
Multiplier Q	(11)	× 1 0 1 1
Partial product 0		1 1 1 0
		+ 1 1 1 0
Partial product 1		1 0 1 0 1
		+ 0 0 0 0
Partial product 2		0 1 0 1 0
		+ 1 1 1 0
Product P	(154)	1 0 0 1 1 0 1 0

(b) Using multiple adders

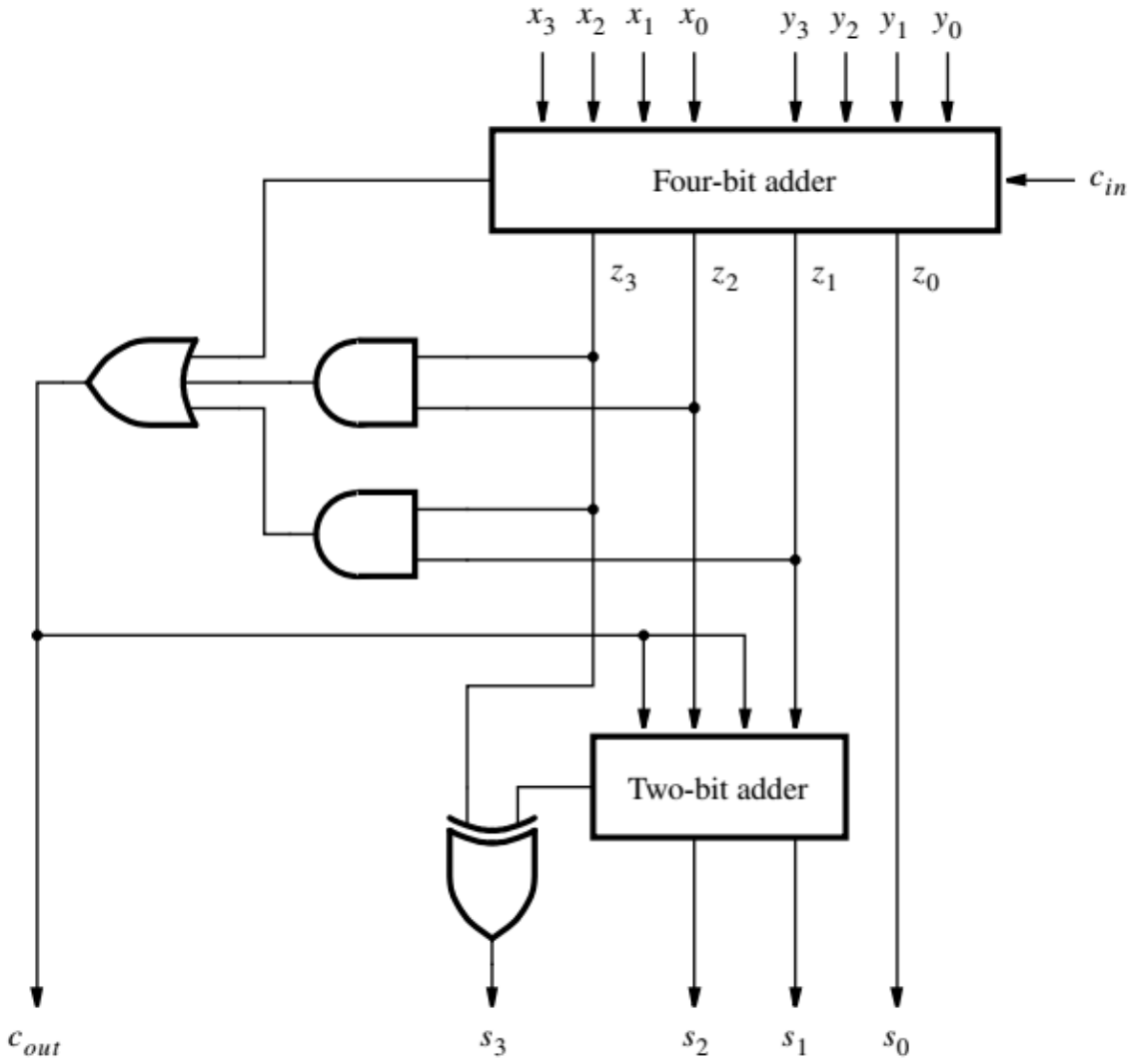
			m_3	m_2	m_1	m_0	
		×	q_3	q_2	q_1	q_0	
Partial product 0			m_3q_0	m_2q_0	m_1q_0	m_0q_0	
		+	m_3q_1	m_2q_1	m_1q_1	m_0q_1	
Partial product 1		$PP1_5$	$PP1_4$	$PP1_3$	$PP1_2$	$PP1_1$	
	+	m_3q_2	m_2q_2	m_1q_2	m_0q_2		
Partial product 2	$PP2_6$	$PP2_5$	$PP2_4$	$PP2_3$	$PP2_2$		
+	m_3q_3	m_2q_3	m_1q_3	m_0q_3			
Product P	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

(c) Hardware implementation



Hardware implementation of four-bit multiplier

- Write Verilog description for following BCD adder circuit and verify the results



Explanation for the above circuit

When X + Y ≤ 9:

- The addition is the same as adding two 4-bit unsigned binary numbers.

When X + Y > 9:

- The result exceeds the valid range of BCD (which can only represent values 0-9).
- The sum will require two BCD digits.
- The initial 4-bit sum from the adder may be **incorrect** and needs correction.

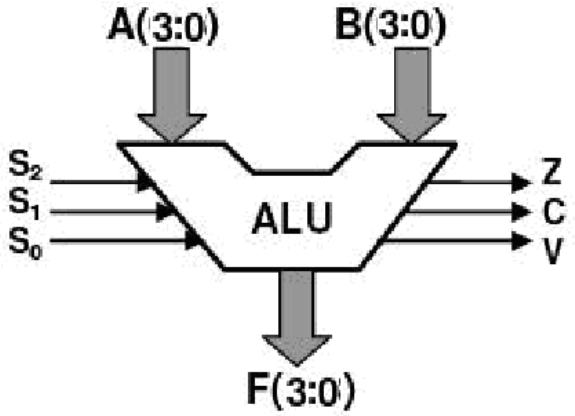
Correction Needed: If the sum exceeds 9, an adjustment is required to ensure the result is a valid BCD representation.

$$\text{Adjust} = \text{Carry-out} + z_3(z_2 + z_1)$$

X	0 1 1 1	7
+ Y	+ 0 1 0 1	+ 5
Z	1 1 0 0	12
	+ 0 1 1 0	
carry →	1 0 0 1 0	
	S = 2	

X	1 0 0 0	8
+ Y	+ 1 0 0 1	+ 9
Z	1 0 0 0 1	17
	+ 0 1 1 0	
carry →	1 0 1 1 1	
	S = 7	

- Design Four-bit ALU



Z, C and V are status flags

Z = 1 if F=0

C = Carry or Borrow

V = Overflow

S_2	S_1	S_0	Function (F)
0	0	0	A+B
0	0	1	A-B
0	1	0	A-1
0	1	1	A+1
1	0	0	A | B
1	0	1	A • B
1	1	0	NOT A
1	1	1	A ⊕ B