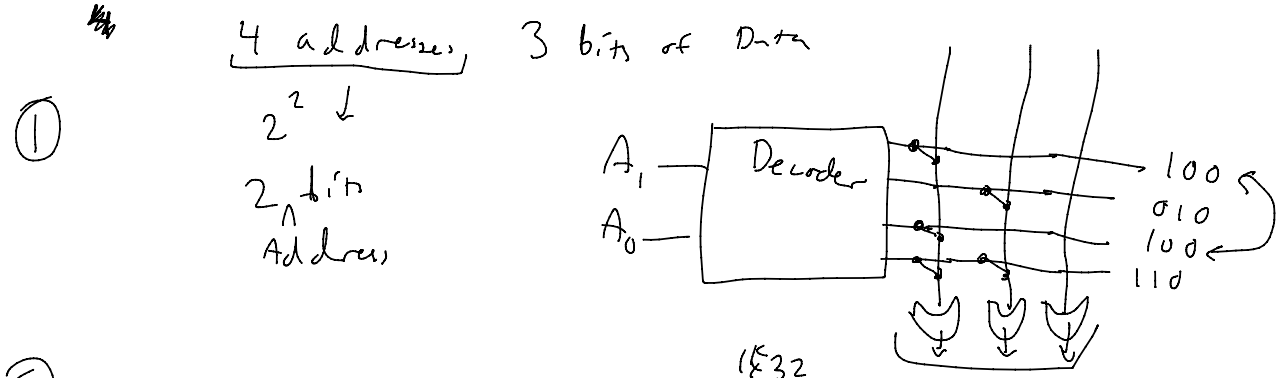
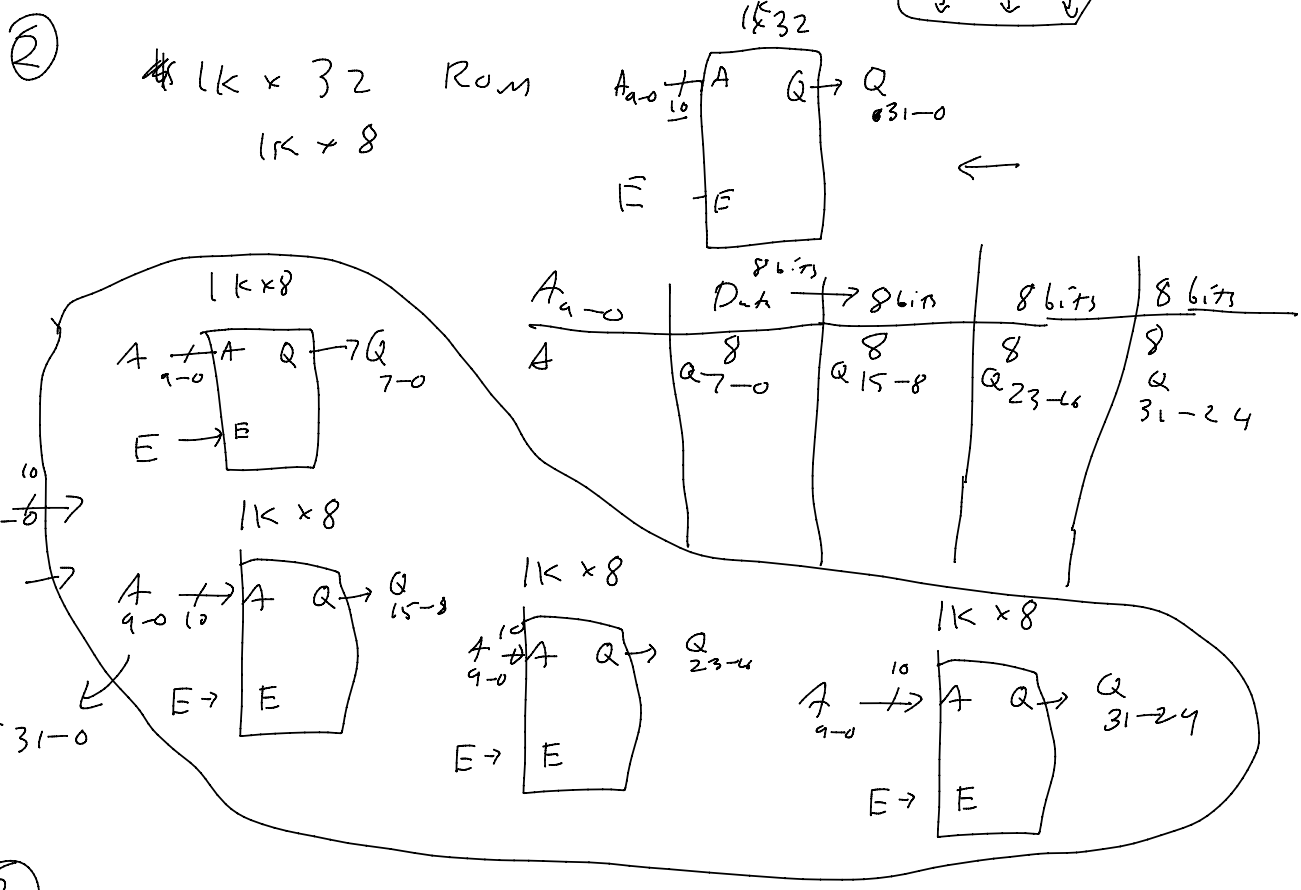
Exercise 5

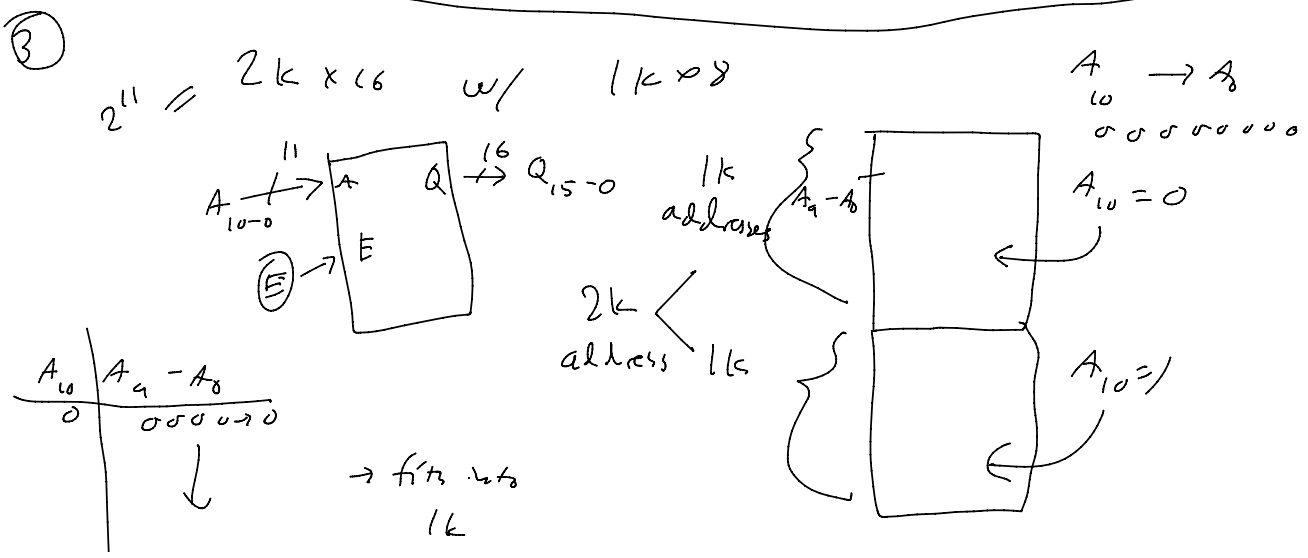
Monday, November 7, 2022 6:20 PM

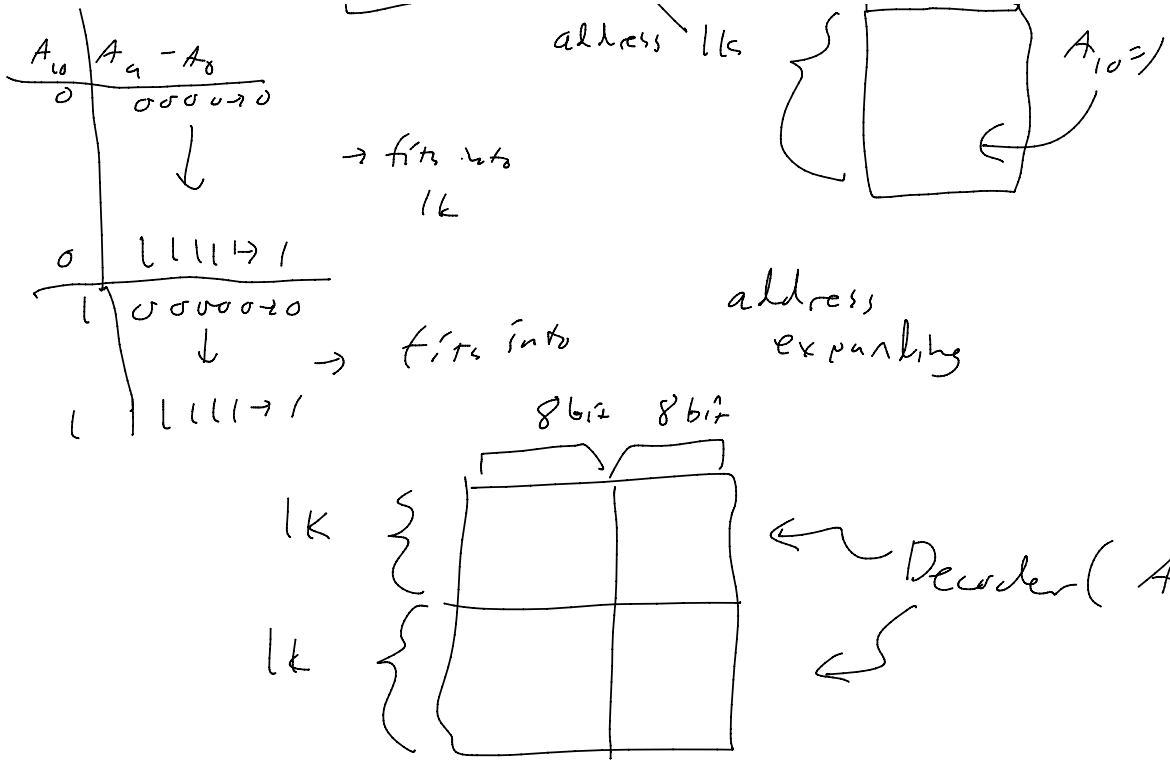
Problem I (Memory)

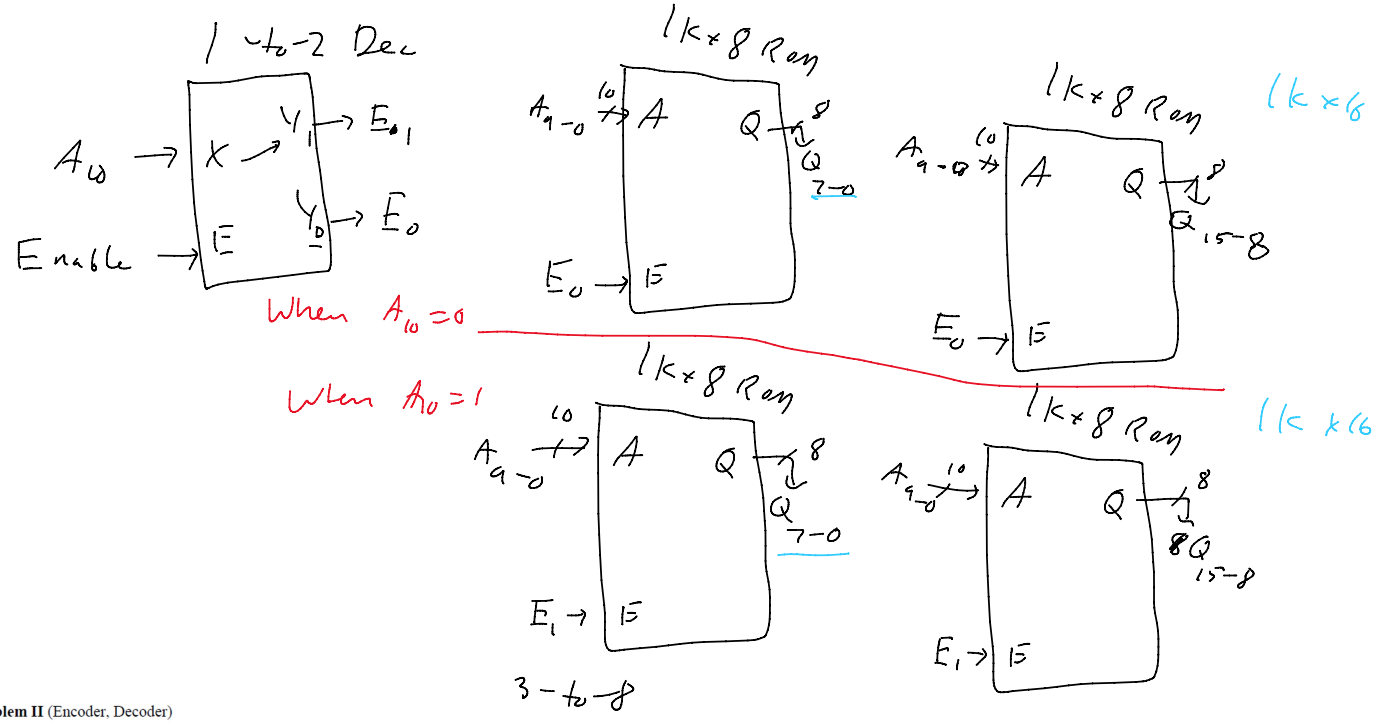
1. Sketch the internal design of a 4x3 ROM.
2. Show how to create a 1Kx32 ROM using only 1kx8 ROMs (note: 1K means 1028 words).
3. Show how to create a 2Kx16 ROM using only 1kx8 ROMs





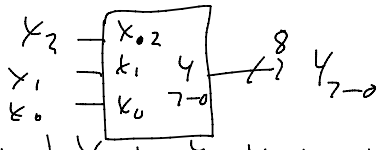




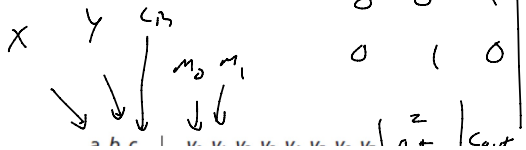


Problem II (Encoder, Decoder)

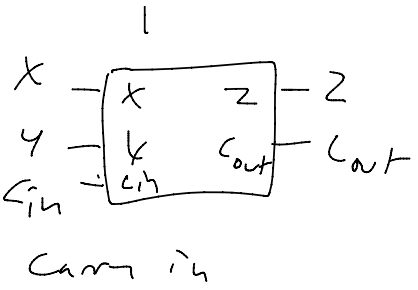
1. Design a full adder using a 3-to-8 line decoder and
   - two OR gates.
   - two NOR gates.



X_2	X_1	X_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0								

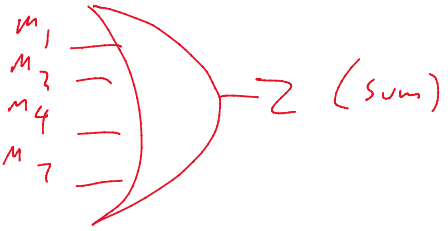


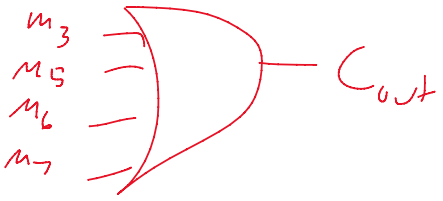
a b c	y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7	Z out	Cout
0 0 0	1 0 0 0 0 0 0 0	0 m_0	0 m_0
0 0 1	0 1 0 0 0 0 0 0	1 m_1	0 m_1
0 1 0	0 0 1 0 0 0 0 0	1 m_2	0 m_2
0 1 1	0 0 0 1 0 0 0 0	0 m_3	1 m_3 m_4
1 0 0	0 0 0 0 1 0 0 0	1 m_4	0 m_5
1 0 1	0 0 0 0 0 1 0 0	0 m_5	1
1 1 0	0 0 0 0 0 0 1 0	0 m_6	1 m_6
1 1 1	0 0 0 0 0 0 0 1	1 m_7	1 m_7



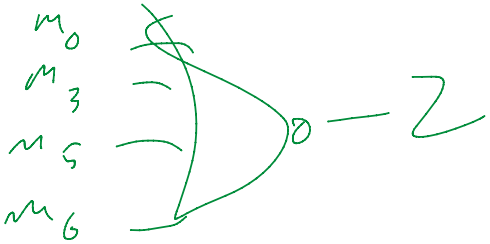
Carry in

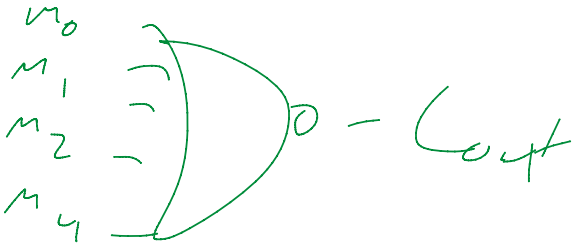
X, Y, Cin inputs → adder → decoder → OR's, Nor's → Z, Cout



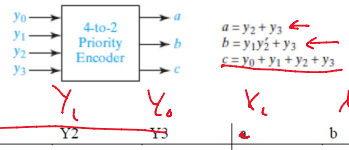


w/ Nor's → 0's





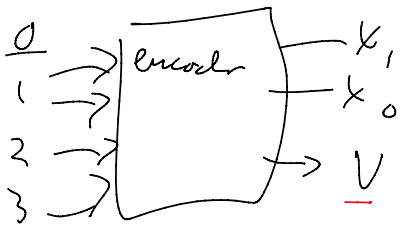
2. Derive the logic equations for a 4-to-2 priority encoder by completing the following table.



$a = y_2 + y_3$
$b = y_1\bar{y}_2 + y_3$
$c = y_0 + y_1 + y_2 + y_3$

Y_3	Y_2	Y_1	Y_0	X_1	X_0	V
Y0	Y1	Y2	Y3	a	b	c
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	—	0	1	1
0	1	—	—	1	0	1
1	—	—	—	1	1	1

$Y_1\bar{Y}_2 + Y_3 = B = X_0$

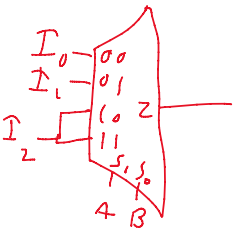


0, 1, 2, (H) 3

$X_1 \to 1$
$X_0 \to 1$
$V \to 1$

Problem III (Multiplexer, Demultiplexer)

1. Show how two 2-to-1 multiplexers (with no added gates) could be connected to form a 3-to-1 MUX. Input selection should be as follows:
   - If $AB = 00$, select Input 0.
   - If $AB = 01$, select Input 1.
   - If $AB = 1-$ (B is a don't-care), select Input 2.
2. Show how two 4-to-1 and one 2-to-1 multiplexers could be connected to form an 8-to-1 MUX with three control lines.



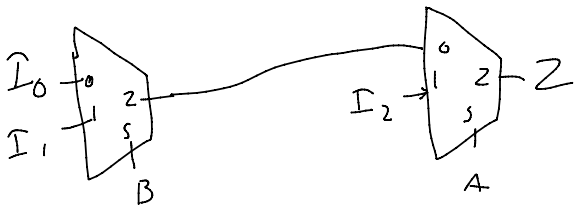
Select

A: = 0

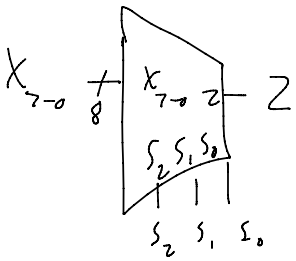
Select ↳ B b/w I0 & I1 ←

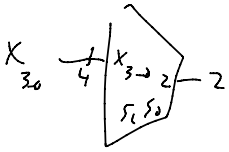
A = 1

I2



2)





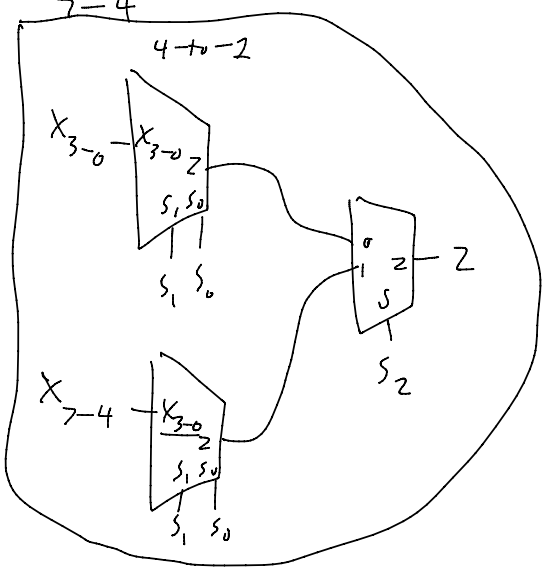
Choose: 1 4-to-1 Mux

X_{3-0}

S_2	S_1	S_0	
0	0	0	X_{3-0}
0	0	1	
0	1	0	
0	1	1	
1	0	0	X_{7-4}
1	0	1	
1	1	0	
1	1	1	

Another Mux

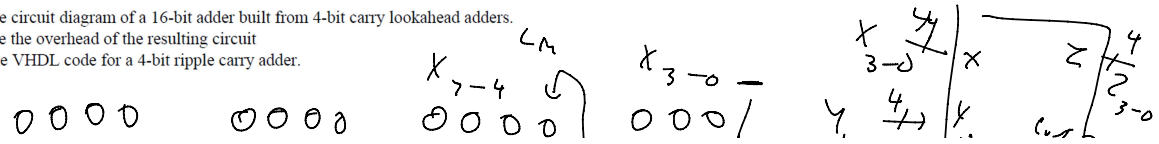
X_{7-4}



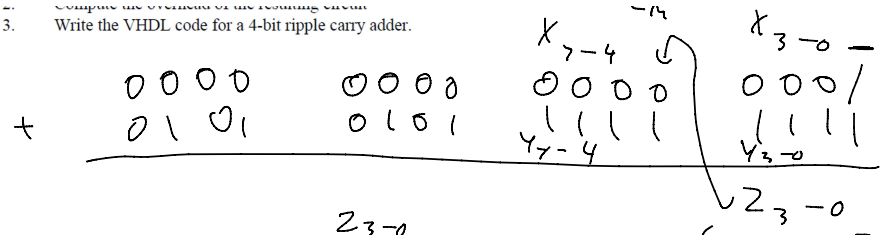
4-bit adder

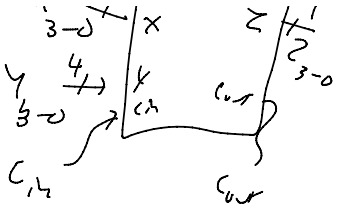
Problem IV. (Adder):

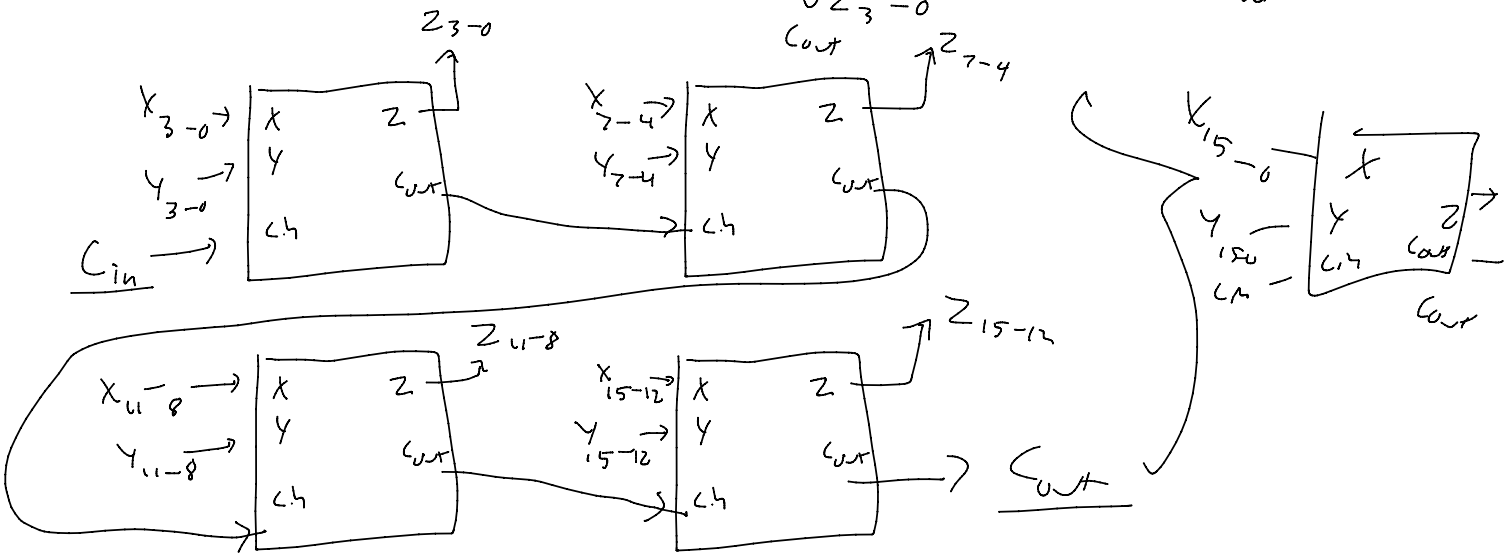
1. Draw the circuit diagram of a 16-bit adder built from 4-bit carry lookahead adders.
2. Compute the overhead of the resulting circuit
3. Write the VHDL code for a 4-bit ripple carry adder.



2. Compute the overhead of the resulting circuit
3. Write the VHDL code for a 4-bit ripple carry adder.







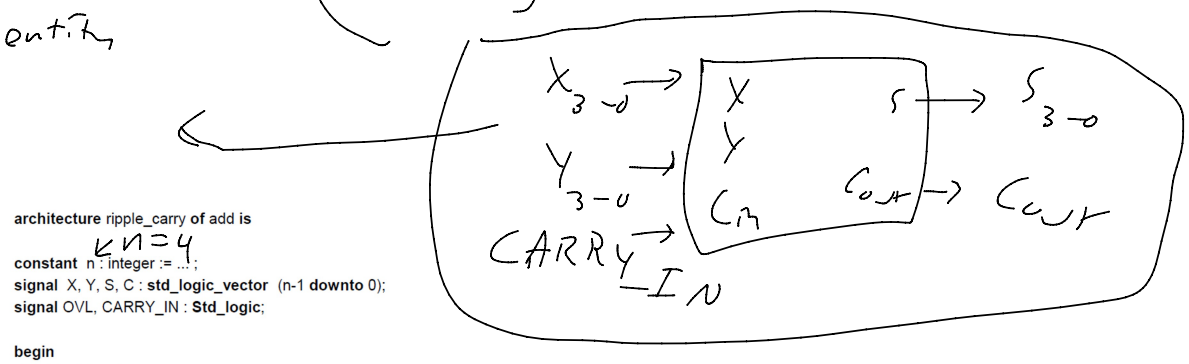
2) Overhead of 2 adder

$$\frac{n^2}{2} + \frac{q_n}{2}$$

Overhead of result

$$4\left(\frac{n^2}{2} + \frac{q_n}{2}\right)$$

3)



```
architecture ripple_carry of add is
                  ← n = 4
constant  n : integer := ... ;
signal  X, Y, S, C : std_logic_vector  (n-1 downto 0);
signal OVL, CARRY_IN : Std_logic;

begin
  for i in 1 to n-1 loop
     S(i) <= X(i) xor Y(i) xor C(i-1);
     C(i) <= (X(i) and Y(i)) or (X(i) and C (i-1)) or (Y(i) and C(i-1));
  end loop ;
S(0) <= X(0) xor Y(0) xor CARRY_IN;
C(0) <= (X(0) and Y(0)) or (X(0) and CARRY_IN) or (Y(0) and CARRY_IN);
OVL <= C(n-1);

end ripple_carry;
```

X is std_logic_vector 3 downto 0