CPE201 HW 6 (100 points)

Answer all questions completely. Put a box around the final solution. Put your name on it. Show your work.

By hand:

1. Given the waveforms in Figure 1, design a circuit that will generate the output waveform given the input waveforms.

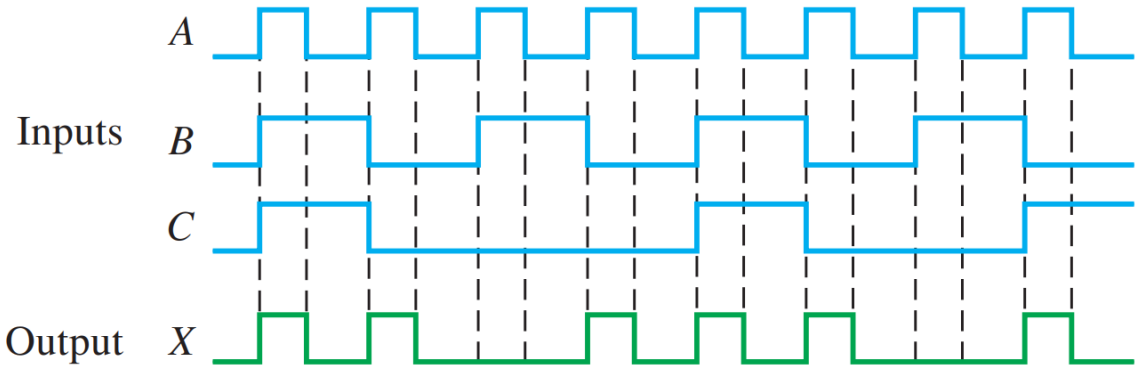


Figure 1

There are several ways to tackle this. You could looks at the inputs and try to come up with an output. I'm going to make a truth table of the inputs and then use a Karnaugh map to minimize the circuit logic.

The truth table can be created by looking for a place in the signal that has the inputs matching each row of the truth table and then recording what the output is at that point.

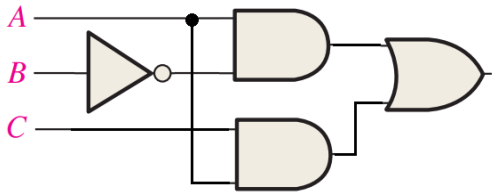
A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Putting this truth table into a Karnaugh map and making groupings of 1's gives:

AB \ C	0	1
00	0	0
01	0	0
11	0	1
10	1	1

Then the minimized output expression is:
Output = AB' + AC

Writing this as a circuit gives:



2. Given the Boolean expression X = A'(B + C'(D' + E)), give the SOP circuit that implements the minimized expression.

We need to fully distribute,
X = A'(B + C'(D' + E))
X = A'(B + C'D' + C'E)
X = A'B + A'C'D' + A'C'E

This expression is in SOP form. Making a Karnaugh map to see if it minimized:

E'

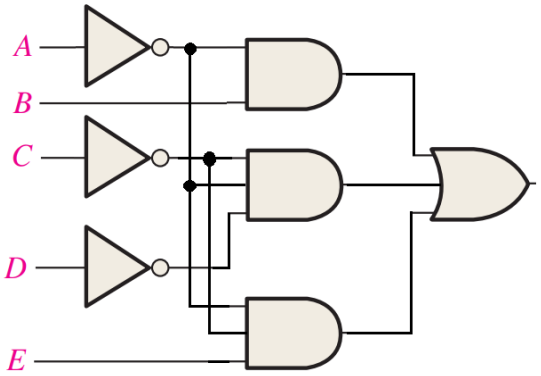
AB \ CD	00	01	11	10
00	1	0	0	0
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

E

AB \ CD	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

The three groupings are in different colors for readability. They are the original terms A'B + A'C'D' + A'C'E

This expression is minimized, so we can create the SOP circuit:



3. Given the following inputs, give the outputs for a full adder.

We can use the truth table for a full adder to get these answers

a. A = 1, B = 0, Cin = 0

Σ = 1
C_{out} = 0

b. A = 1, B = 1, Cin = 1

Σ = 1
C_{out} = 1

c. A = 0, B = 0, Cin = 1

Σ = 1
C_{out} = 0

4. Given the 5-bit parallel adder circuit in Figure 2 and the following input sequences, determine the sum outputs of the 5-bit parallel adder after each input set (i.e. A_0 = 1, A_1 = 0, etc. then A_0 = 0, A_1 = 1, etc.)
   A_0 1010
   A_1 0110
   A_2 1110
   A_3 1010
   A_4 0100
   B_0 0001
   B_1 1111
   B_2 1011
   B_3 0011
   B_4 0101

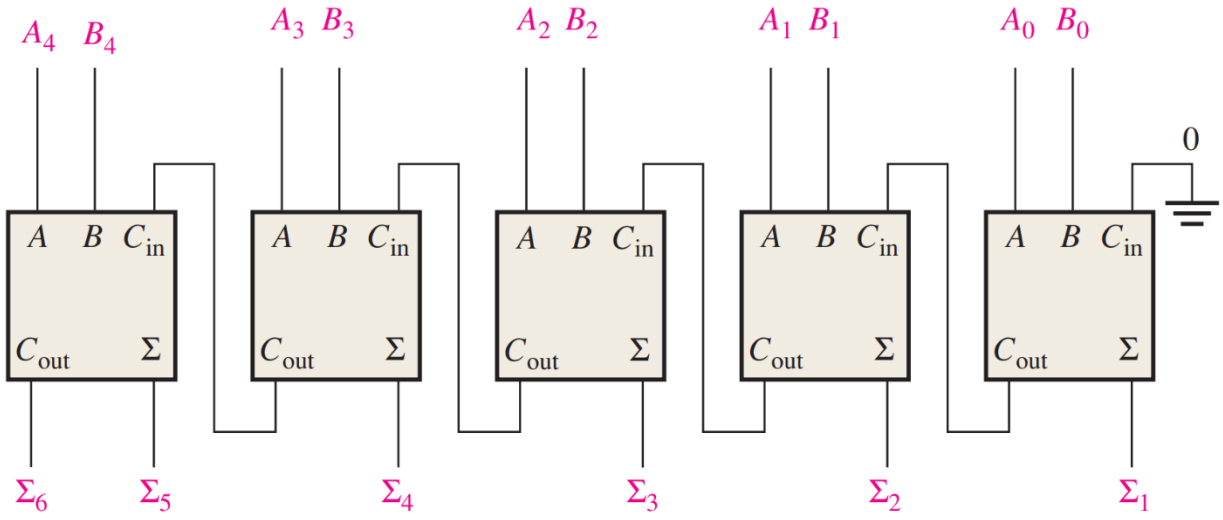


Figure 2

Each set of outputs can be made from the truth table of a full adder.

At t=1, the inputs from left to right are:

A_4	B_4	A_3	B_3	A_2	B_2	A_1	B_1	A_0	B_0
0	0	1	0	1	1	0	1	1	0

The carry bits from left to right are:

Cout4	Cout3	Cout2	Cout1	Cout0
0	1	1	0	0

Then the outputs for t=1 from left to right are:

Σ_6	Σ_5	Σ_4	Σ_3	Σ_2	Σ_1
0	1	0	0	1	1

At t=2, the inputs from left to right are:

A_4	B_4	A_3	B_3	A_2	B_2	A_1	B_1	A_0	B_0
1	1	0	0	1	0	1	1	0	0

The carry bits from left to right are:

Cout4	Cout3	Cout2	Cout1	Cout0
1	0	1	1	0

Then the outputs for t=2 from left to right are:

Σ_6	Σ_5	Σ_4	Σ_3	Σ_2	Σ_1
1	0	1	0	0	0

At t=3, the inputs from left to right are:

A_4	B_4	A_3	B_3	A_2	B_2	A_1	B_1	A_0	B_0
0	0	1	1	1	1	1	1	1	0

The carry bits from left to right are:

Cout4	Cout3	Cout2	Cout1	Cout0
0	1	1	1	0

Then the outputs for t=3 from left to right are:

Σ_6	Σ_5	Σ_4	Σ_3	Σ_2	Σ_1
0	1	1	1	0	1

At t=4, the inputs from left to right are:

A_4	B_4	A_3	B_3	A_2	B_2	A_1	B_1	A_0	B_0
0	1	0	1	0	1	0	1	0	1

The carry bits from left to right are:

Cout4	Cout3	Cout2	Cout1	Cout0
0	0	0	0	0

Then the outputs for t=4 from left to right are:

Σ_6	Σ_5	Σ_4	Σ_3	Σ_2	Σ_1
0	1	1	1	1	1