COEN 651 Dec 2019

Question 1

③ a) 1) For any fixed number of variables the circuit is the same and only the memory has to be programmed

⑤ 2) For any fixed number of variables, the area, delay and power is the same

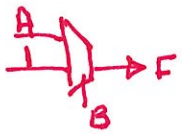
⑤ 3) Evaluation of the delay, area and power is simple.

4) Re-programing easy since only the memory has to be updated

③ b) $F(A,B,C) = A\bar{B} + \bar{A}B + AB + ABC$

$= A\bar{B} + B(A+\bar{A}) = A\bar{B} + B$

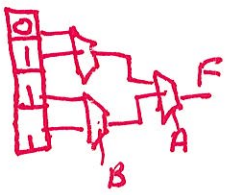
This is an OR gate



③ c)

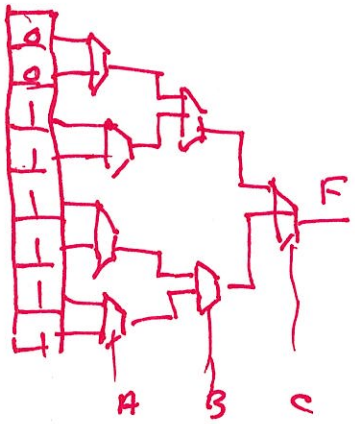
with minimization

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



without minimization

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



① d) 2 → 1 based delay → Δ

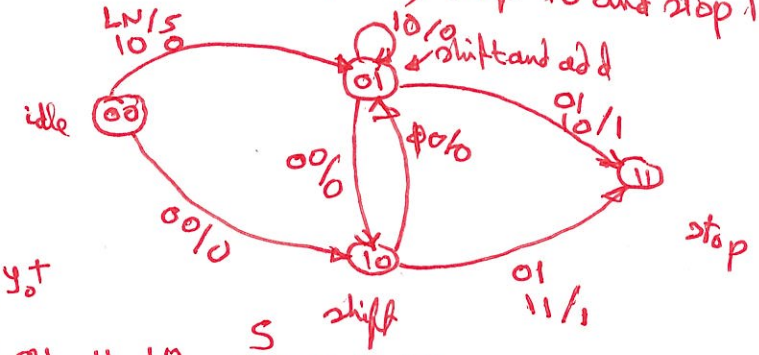
Look up delay $2\Delta + \frac{1}{2}\Delta$ minimized

$3\Delta + \frac{1}{2}\Delta$ without minimization

→ Q2 COEN 6501 Dec 2019

There are four states, idle, add and shift, shift and stop

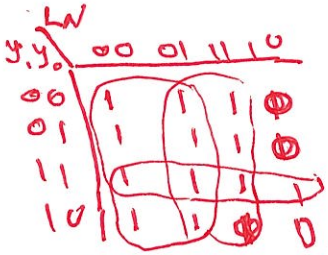
assume idle = "00", shift and add, "01", shift "10" and stop "11"



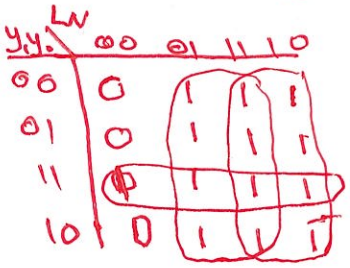
y_1y_0 \ LN	00	01	11	10
00	10	11	11	01
01	10	11	11	01
11	11	11	11	11
10	10	11	11	01

$y_1^+ y_0^+$

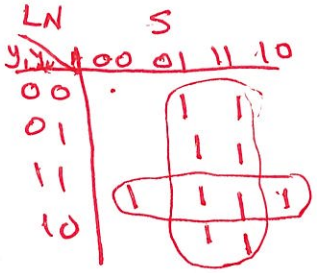
y_1y_0 \ S	00	01	11	10
0	1	1	0	
0	1	1	0	
1	1	1	1	
0	1	1	0	



$y_1^+ = \bar{L} + N + y_1y_0$

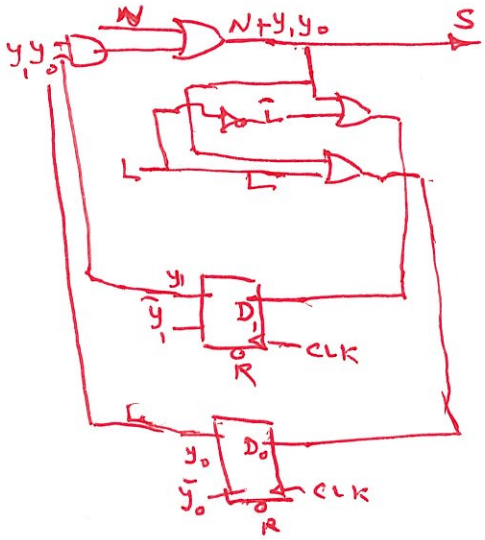


$y_0^+ = L + N + y_1y_0$



$S = N + y_1y_0$

Using D_FF

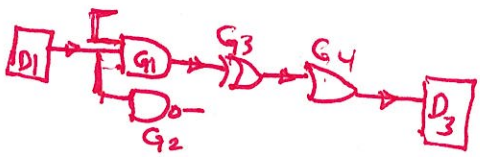


COEN 6501 Dec 2019

→ Q3

a) There are 5 paths in the circuit

Path 1 : D1 – G1 – G3 – G4 – D3
Path 2 : D1 – G2 – G3 – G4 – D3
Path 3 : D2 – G2 – G3 – G4 – D3
Path 4 : D2 – G4 – D3
Path 5 : D3 – G5 – G1 – G3 – G4 – D3



Critical Path

b) All paths share G3 and G4 . G1 & G2 have similar loading but tp of AND is twice of the NAND

That leaves the critical path to be either path 1 or path 5.

The difference between Path 1 and Path 5 is the Inverter and its loading which is much less than the loading on D1 due to G1 and G3 loading so, it is expected the path 1 will be the critical Path.

Path 1 - Combinational logic

$t_{CL} = 0.35\times2 + 0.3(1.5+2) + 0.5 + 0.2*1 + 0.15*3 + 1 + 0.3*1 + 0.25*2.5 + 0.75 + 0.25*1 + 0.2*3.5 = 6.5 + 1.5_{\text{Flip Flop}} = 8.025 - 0.25_{\text{Skew}}$

Path 5 $0.3*1 + 0.35*0.15 + 0.05*2 + 0.1 + 0.5 + 0.15*3 + 0.2 + 1 + 0.25 + 0.3 + 0.75 + 0.2(3.5) + 0.25 + 1.5_{FF} = 7.275 - 0.25_{\text{Skew}}$

Path 1 is the critical path

Delay $= 8.025 + 1^{S4} = 9.025 - 0.25^{t_{sc}} \simeq 8.8$

Max Frequency $\simeq$ 113 MHz

c) If we are running the circuit at 100 MHz the period is 10ns

Slack time for the set up time $10 - (t_{CL} + t_{S4}) + t_{sc}$ $\quad t_{slack} = t_{required} - t_{arrival}$

$= 10 - (8.025+1) = 0.975ns + 0.25 \simeq 1$

for hold time we have to calculate the fastest path. which is

Path 4 $D_2 \rightarrow G_4 \rightarrow D_3$

Path 4 $= 1.5 + 0.3(1.5*2.5) + 0.35*2 + 0.2*3.5 + 0.25 = 5.1ns$

Slack time for hold time

$5.1 - 0.5 = 4.6ns$ $\quad t_{sc} = 4.6 - 0.25 = 4.35$

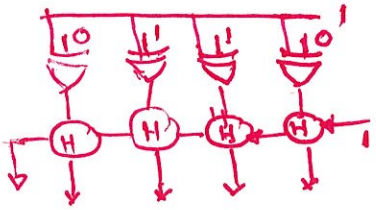
COEN 6501 Dec 2019

Q4

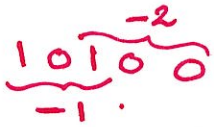
$6_{10} \rightarrow 0110_2$

$-6_{10} \rightarrow 1010$

Hardware required to get -6



Booth Multiplid

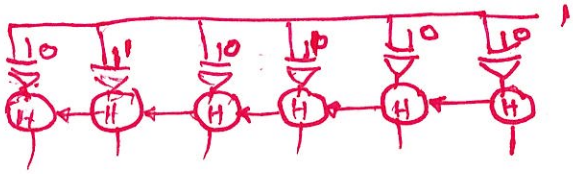


20 → 010100

-20 → 101100

-40 → 1011000

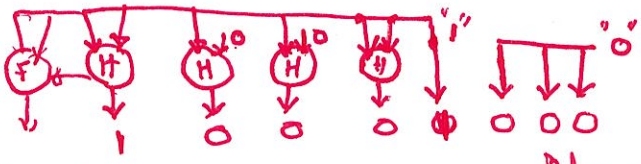
Hardware to get -20



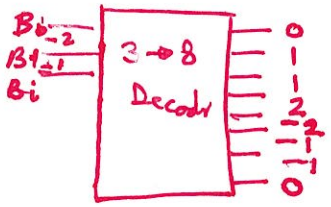
Addition

1 1 0 1 1 0 0 0

1 0 0 0 1 0

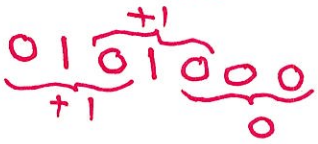


Delay = $t_a + 3t_g$



Decodr for Booth Multiplid

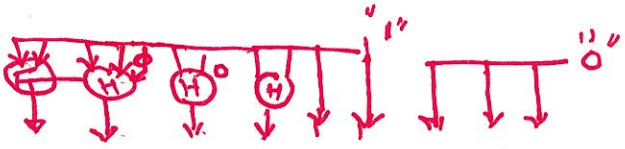
Alternate If we code 20



0 0 0 0 0 0 0 0

1 1 1 0 1 0

1 0 1 0



Delay $t_a + 3t_g$

COEN 6501 Dec 2019

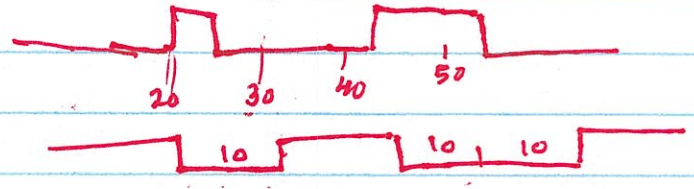
Q5

Use ieee library;
Use ieee.std_logic_1164.all;

→ entity circuit is port(A,B: in std_logic; F: out std_logic); end circuit;
architecture structural of circuit is
component inv_1 port (A: in std_logic; F: out std_logic); end component;
component XOR_2 port (A,B: in std_logic; F: out std_logic); end component;
signal A1: std_logic;
begin L1: inv_1 portmap(A, A1);
L2: XOR_2 portmap(A1, B, F); end structural;

use ieee.std_logic_1164.all
→ entity generator is port (A, B: out std_logic); end generator;
architecture behavioral of generator is
begin process A <= '0'; B <= '0'; wait for 10ns;
A <= '0'; B <= '1'; wait for 10 ns;
A <= '1'; B <= '0'; wait for 10ns;
A <= '1'; B <= '1'; wait for 10ns;
end process; end behavioral;

→ use ieee.std_logic_1164.all;
entity test_bench is end test_bench;
architecture behavioral of test_bench is
component generator (A, B: out std_logic); end component;
component circuit (A, B: in std_logic; F: out std_logic); end compt;
signal A, B, F: std_logic
L1: generator portmap(A, B);
L2: circuit portmap(A, B, F); end behavioral;



Q6

n_1	n_0	F	f_4	f_3	f_2	f_1	f_0
0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0
1	0	9	0	1	0	0	1
1	1	28	1	1	1	0	0

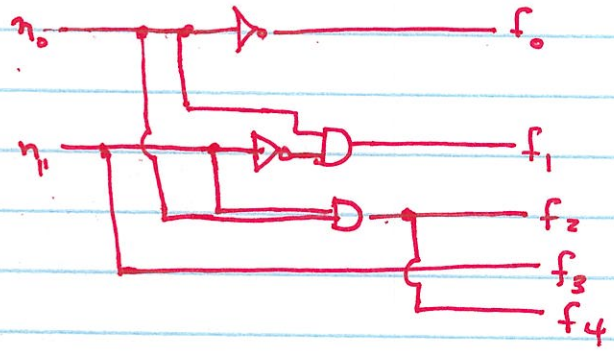
$f_0 = \bar{n}_0$

$f_1 = \bar{n}_1 n_0$

$f_2 = n_1 n_2$

$f_3 = n_1$

$f_4 = n_1 n_0$



Delay 2 t_g and for a Double rail is t_g

Alternatively

N^2 — $n_1\ n_0$ / $n_1\ n_0$

$n_1 n_0 \quad n_0$

$n_1 \quad n_1 n_0$

$n_1 + n_1 n_0 \quad 0 \quad n_0$

$n_1 n_0$

$\times N$ $n_1 n_0$

$n_0(n_1 + n_1 n_0) \quad 0 \quad \overline{n_0}$

$n_1(n_1 + n_1 n_0) \quad n_1 \times 0 \quad n_1 n_0 \quad x$

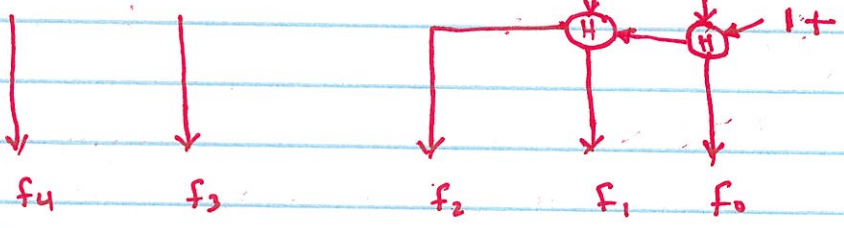
$n_1 n_0$

$\overline{n_1 n_0}$

$0 \quad n_1 n_1 + n_1 n_1 n_0 \quad n_1 n_0 + n_1 n_0 \quad n_1 n_0 \quad n_0$

N^3

$n_1 n_0 \quad n_1 \qquad 0 \quad n_1 n_0 \quad n_0$



Delay = 2 half Adder or two XOR $2t_g$ + 1 AND = $3t_g$

if we assume that t_g XOR = t_g AND then $2t_g$