Q1 Dec. 2016

a. Several programming techniques the most used ones are:
MUX based, Look-up Table, PAL, EPROM, PLA etc.

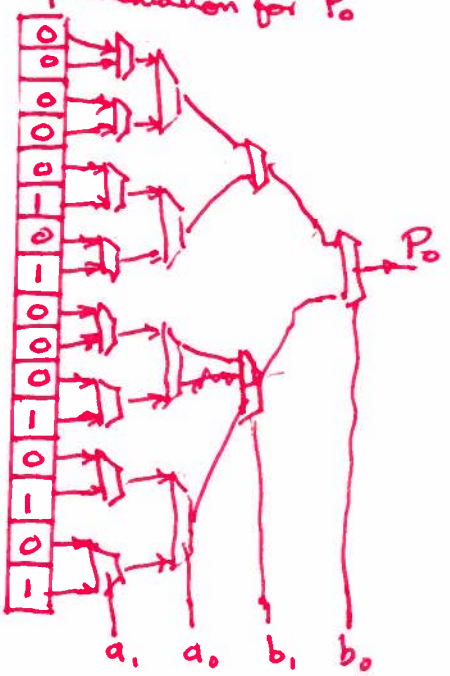
1) Mux based programming uses 2→1 Multiplexers to implement the gates as a basic unit and build on the technique to build larger functions
Example an Ex-OR is built as $A \oplus B$.

2 - Look Up Table represent a function with K-variables in a memory table with one output. It requires 2^k memory location each location is addressed by k bits with $2^k - 1$ MUXes with access time of MUX + 1 read of memory.

b.

a_1	a_0	b_1	b_0	P_3	P_2	P_1	P_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	1	0	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Implementation for P_0



The same architecture is used for P_1, P_2 and P_3 with different Memory for P_1, P_2 & P_3

Total Area = 4 * [16 cells + 15 MUX] = 4 * [16 * 2A + 15A] = 4 * 47 = 188

Total Delay = $0.5D_{cell} + 4D_{MUX} = 4.5D$

Q2 Dec 2016

$F = (N-1)^2$ ... N is a 3-bit unsigned number $N = N_2 N_1 N_0$

	N			F							
	N_2	N_1	N_0	F_6	F_5	F_4	F_3	F_2	F_1	F_0	Equivalent Decimal
0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	1	1
3	0	1	1	0	0	0	0	1	0	0	4
4	1	0	0	0	0	0	1	0	0	1	9
5	1	0	1	0	0	1	0	0	0	0	16
6	1	1	0	0	0	1	1	0	0	1	25
7	1	1	1	0	1	0	0	1	0	0	36

$F_0 = \sum_m(0,2,4,6) = \bar{N}_0$

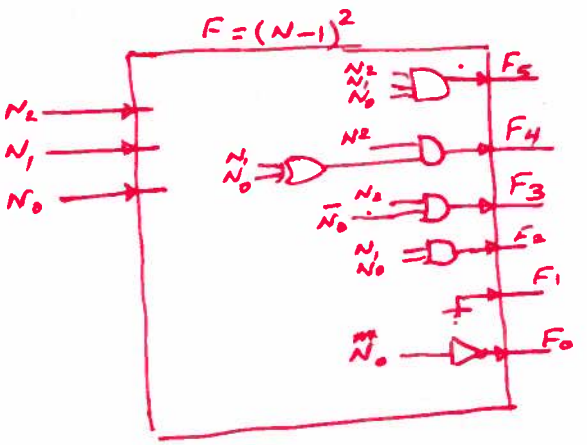
$F_1 = \sum_m 0 = 0$

$F_2 = \sum_m(3,7) = N_1 N_0$

$F_3 = \sum_m(4,6) = N_2 \bar{N}_0$

$F_4 = \sum_m(5,6) = N_2 \cdot (N_1 \bar{N}_0 + \bar{N}_1 N_0)$

$F_5 = \sum_m(7) = N_2 N_1 N_0$



Ddelay = 2 * 2 = 4ns

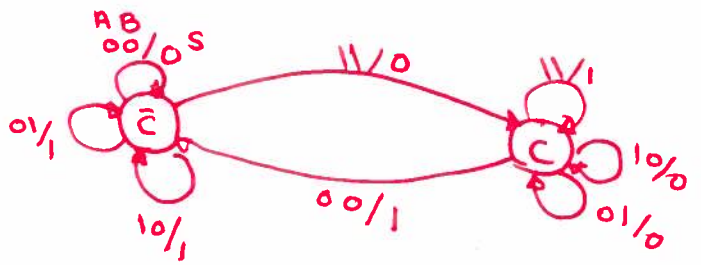
Area = 6 Agate

An Alternative to implement the function by a LUT. But that is requires more delay & Area.

Also it can be implemented by a multiplier using carry save adders as $N^2 - 2N + 1$ but the delay & Area is prohibitive.

Q3 Dec 2016

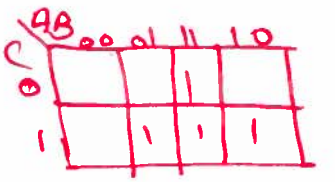
This adder has 2 states to remember C There is a Carry
$\bar{C}$ " " no "



Present State	Next state / output			
	00	01	11	10
$\bar{C}$ = 0	0/0	0/1	1/0	0/1
C = 1	0/1	1/0	1/1	1/0

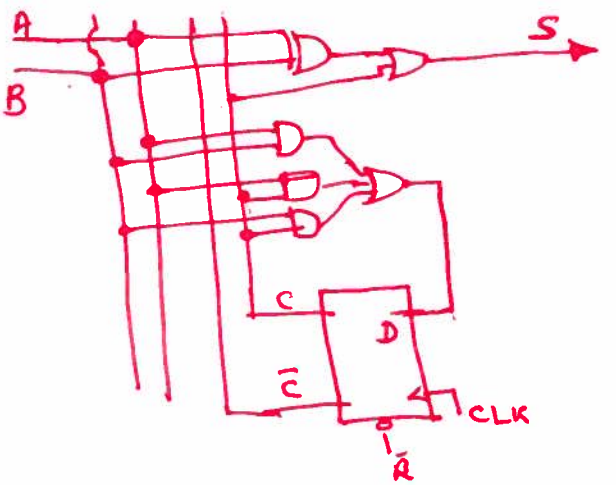
Transition Table

$S = A \oplus B \oplus C$

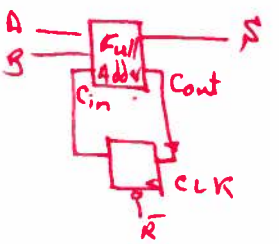


$C = AB + AC + BC$

Using D FlipFlop



or



Q4 Dec 2016

Paths to be considered

P1 $U_0 \rightarrow U_{10} \rightarrow U_{11}$
P2 $U_0 \rightarrow U_4 \rightarrow U_5 \rightarrow U_7 \rightarrow U_8$
P3 $U_0 \rightarrow U_4 \rightarrow U_1 - U_2$
P4 $U_2 \rightarrow U_{10} - U_{11}$
P5 $U_2 \rightarrow U_3 - U_6 - U_7 - U_8$
P6 $U_8 \rightarrow U_9 \rightarrow U_{10} \rightarrow U_{11}$
P7 $U_8 \rightarrow U_6 \rightarrow U_7 \rightarrow U_8$
P8 $U_8 \rightarrow U_5 - U_7 \rightarrow U_8$
P9 $U_8 \rightarrow U_1 \rightarrow U_2$

* } Critical Path with fan out only

We will Consider Fan-Out Only

Combinational Path only

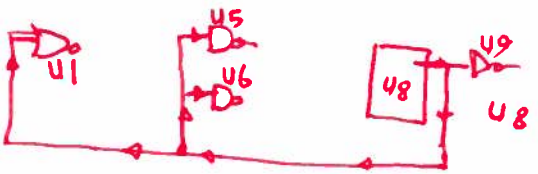
$C_{P1} = 2*0.25 + 0.25 = 0.75$ ns
$C_{P2} = 2*0.25 + 0.15 + 2*0.1 + 0.25 + 0.25 = 1.4$ ns
$C_{P3} = 2*0.25 + 0.15 + 2\times0.1 + 0.25 = 1.15$ ns
$C_{P4} = 2*0.25 + 0.25 = 0.75$ ns
$C_{P5} = 2*0.25 + 0.15 + 0.25 + 0.25 = 1.15$ ns
$C_{P6} = 5*0.25 + 0.15 + 0.25 = 1.65$ ns
$C_{P7} = 5*0.25 + 0.25 + 0.25 = 1.75$
$C_{P8} = 5*0.25 + 0.25 + 0.25 = 1.75$ } Critical Path
$C_{P9} = 5*0.25 + 0.25 = 1.5$

Path Delay $= \overset{C\rightarrow Q}{1.5} + \overset{C_t}{1.75} + \overset{S_U}{1} = 4.25$ ns

Frequency of operation $\leq \frac{109}{4.25} \leq 235$ MHz

$K_T = \frac{273 + (40 + 2*30)}{273 + 40} = \left(\frac{373}{313}\right)^{1.5} = 1.3$

New frequency of operation $\leq \frac{235}{1.3} \leq 180$ MHz



U8 has the highest fan-out

Q5 Dec 2016

There are 3 paths in this circuit.

Path 1 $D_0 \rightarrow XOR \rightarrow D_0$ = 3.65 ns ✓ Critical Path

Path 2 $D_1 \rightarrow XOR \rightarrow D_0$ = 2.75 ns

Path 3 $D1 \rightarrow D1$ = 2.2 ns

a) Arrival at Point A = $0.7 + 0.3 * 2 + 0.1(1.5+2) = 1.65$ ns

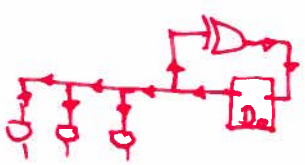
b) Time period = $\frac{10^9}{200 \times 10^6} = 5$ ns

D_0:
- Time required for both D_0 & D_1 = $T - t_{su} = 5 - 0.5 = 4.5$ ns
- Arrival at point D_0 = 1.65 + 2 = 3.65 ns
- Set-up slack = 4.5 − 3.65 = 0.85 ns
- $T_h \leq t_{cQ} + t_{\ell} - t_{cs} = t_{cQ} + t_{\ell} = 1.65 + 0.3 + 0.2*3 + 0.2 = 2.75$ ns
- Hold time slack = 2.75 − 0.2 = 2.55

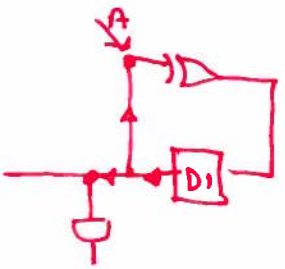
D_1:
- Arrival at point D_1 = $0.7 + \underbrace{3*0.3}_{f_0} + \underbrace{(2*1.5*3)0.1}_{\text{load}} = 0.7 + 0.9 + 0.6 = 2.2$ ns
- Tset up slack = 4.5 − 2.2 = 2.3 ns
- $T_h \leq t_{cQ_{min}} + t_{\ell_{min}} - t_{cs} = 0.7 + (3*0.3) + (2*1.5*3)0.1 = 2.2$ ns
- Thold slack = 2.2 − 0.2 = 2 ns

Path 1 = $0.7 + 4*0.3 + (4.5+2)0.1 + 0.3 + 0.2 + 3*0.2 = 3.65$

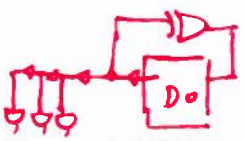
Maximum frequency = $\frac{}{(3.65+0.5)\text{ns}} = 240$ MHz



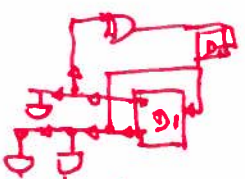
Critical path



Arrival @ A



Slack for hold & setup time D_0



Slack for hold & setup time D_1

Q6 Dec 2016

"This is only sample of possible answer" refer to Alan notes for more detail

MUX

```
library ieee
use ieee.std_logic_1164.all;
entity Mux is
Port (D1, D2, C : in std_logic;
      Z : out std_logic);
end Mux;

Architecture RTL of Mux is
  Component AND2 is
  Port (A, B : in std_logic ; C : out std_logic); end component AND2;
  Component OR2 is
  Port (A, B : in std_logic; C : out std_logic ); end Component OR2;
  Component INV is
  Port ( A : in std_logic; B : out std_logic); end Component INV;
  Signal S1, S2, S3 : std_logic;
→ For statements ⟶ Work . . .   Configuration Statements ←
  Port map statement →
end RTL;
```

Generator

```
library ieee;
use ieee.std_logic_1164.all;
entity Generator is
Port (A, B, C : out std_logic); end Generator;
Architecture of Behavioral of Generator is
P1 : Process
Begin
  A <= '0'; B <= '0'; C <= '0'; wait for T1 ns;
  → other values of A, B, C with wait statements ←
end Process P1; end Behavioral;
Use ieee.Std_logic_1164.all
entity Test Bench is    end Test Bench;
Architecture of TB is
Component Generator is Port (A, B, C : out std_logic); end Component;
Component MUX is Port (D1, D2, C : in std_logic; Z : out std_logic; end Component;
Signal S1, S2, S3 : Std logic
Begin  Generator Portmap (S1, S2, S3); Portmap (S1, S2, S3 Z1);
end TA;
```