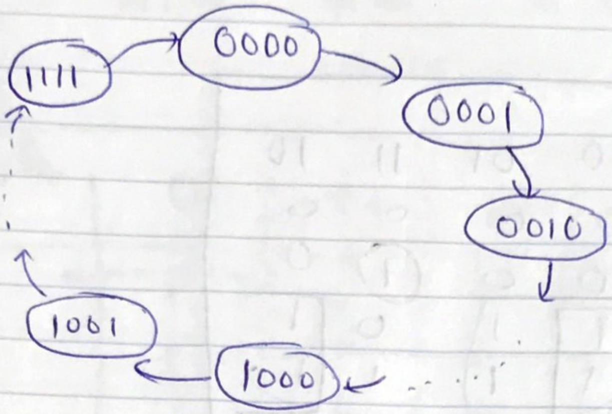
EECS700_HW3

Manjish Adhikari
KUID: 2870257

4-Bit Counter



State Diagram

Present State	Next State	D3	D2	D1	D0.
Q_3 Q_2 Q_1 Q_0	Q_3^+ Q_2^+ Q_1^+ Q_0°	0			
0 0 0 0	0 0 0 1	0	0	0	1
0 0 0 1	0 0 1 0	0	0	1	0
0 0 1 0	0 0 1 1	0	0	1	1
0 0 1 1	0 1 0 0	0	1	0	0
0 1 0 0	0 1 0 1	0	1	0	1
0 1 0 1	0 1 1 0	0	1	1	0
0 1 1 0	0 1 1 1	0	1	1	1
0 1 1 1	1 0 0 0	1	0	0	0
1 0 0 0	1 0 0 1	1	0	0	1
1 0 0 1	1 0 1 0	1	0	1	0
1 0 1 0	1 0 1 1	1	0	1	1
1 0 1 1	1 1 0 0	1	1	0	0
1 1 0 0	1 1 0 1	1	1	0	1
1 1 0 1	1 1 1 0	1	1	1	0
1 1 1 0	1 1 1 1	1	1	1	1
1 1 1 1	0 0 0 0	0	0	0	0

Constructing K-Map.

D3

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	0	1
10	1	1	1	1

$$D3 = Q_3\overline{Q_2} + Q_3\overline{Q_0} + Q_3\overline{Q_1} + \overline{Q_3}Q_2Q_1Q_0.$$

D2

$Q_3Q_2 \backslash Q_1Q_0$	00	01	10	11
00	0	0	1	0
01	1	1	0	1
10	1	1	0	1
11	0	0	1	0

$$D2 = Q_2\overline{Q_0} + Q_2\overline{Q_1} + \overline{Q_2}Q_1Q_0$$

D1

$Q_3Q_2 \backslash Q_1Q_0$	00	01	10	11
00	0	1	0	1
01	0	1	0	1
10	0	1	0	1
11	0	1	0	1

Scanned by CamScanner

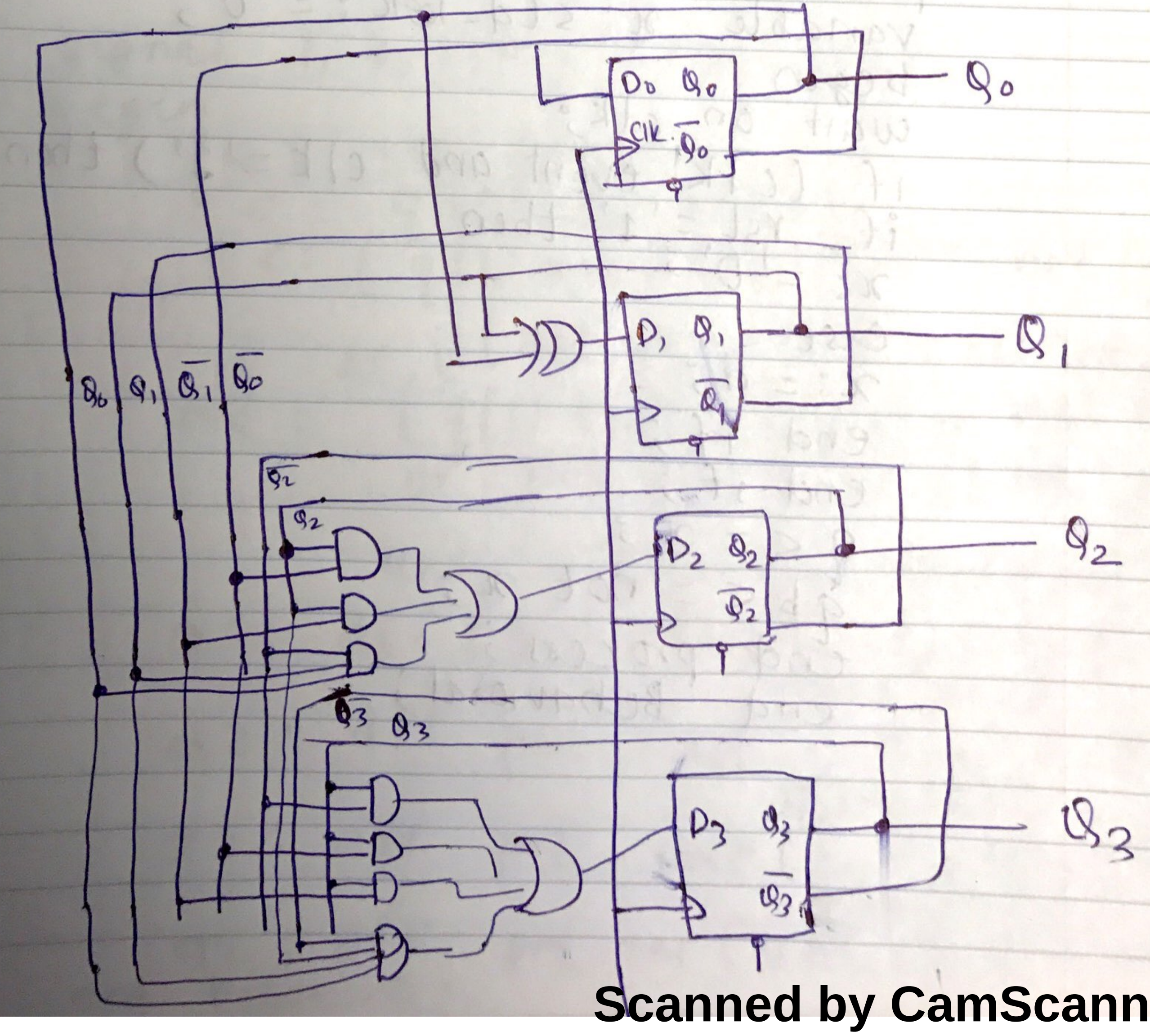
$$D1 = \overline{Q_1}Q_0 + Q_1\overline{Q_0} = Q_1 \oplus Q_0$$

For D_0

$Q_3Q_2 \backslash Q_1Q_0$	00	01	10	11
0 0	1	0	0	1
0 1	1	0	0	1
1 0	1	0	0	1
1 1	1	0	0	1

$$D_0 = \overline{Q_0}$$

Redrawing it as a circuit.



VHDL code:

Designing aDflip flop.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity dff is
 port ( d, rst, clk: IN STD_LOGIC;
          q, qb: OUT STD_LOGIC);

 end, dff

  architecture Behavioral of dff is
   begin
   process
   variable x: std_logic := '0';
   begin
   wait on clk;
   if (clk' event and clk='1') then.
   if rst='1' then
   x:='0';
   else
   x:=d;
    end if;
    end if;
    q<=x;
    qb<=not x;
    end process;
    end Behavioral;
```

Counter:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity dcounter is
port (clk, rst : IN STD_LOGIC;
      q, qbar : INOUT STD_LOGIC_VECTOR(3
                DOWNTO 0));
end dcounter;

architecture behavioral of dcounter is
component dff is
port (d, rst, clk : IN STD_LOGIC;
      q, qb : OUT STD_LOGIC);
end component;

signal i, k, l, m : STD_LOGIC;
begin
    i <= not (q(0));
    k <= q(0) xor q(1);
    l <= ( q(2) and ( not (q(1) or not (q(0)))
          or ( (not q(2)) and q(1) and q(0)));
    m <= ( q(3) and ( not (q(2) or not q(1)
          or not q(0))) or (( not (q(3))
          and q(2) and q(1) and q(0)));
   a1: dff port map( i, rst, clk, q(0),
                qbar(0));
   a2: dff port map( i, rst, clk, q(1),
                qbar(1));
   a3: dff port map( i, rst, clk, q(2),
                qbar(2));
   a4: dff port map ( i, rst, clk, q(3), qbar(3));
end behavioral;
```