Sequence Generator

EE214 Autumn 2021

September 28, 2021

1 Introduction:

Consider the following Sequence generator which generates a certain sequence upon every input clock pulse and on reset the sequence will go into default sequence i.e 2 in this case.

Note: The reset is asynchronous in nature i.e reset effects the output sequence irrespective of the input clock arrival.

Inputs: Reset, clock
Output(3 bit):Y2Y1Y0

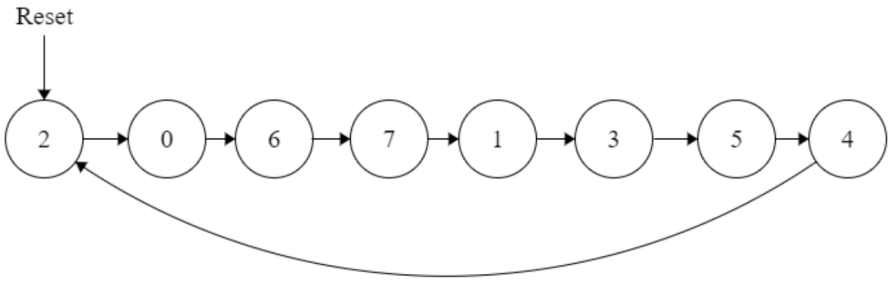

Figure 1: Sequence generator

2 Design Procedure:

2.1 State Table:

Present State(Q2Q1Q0)	Next state(nQ2nQ1nQ0)	D2D1D0
000(0)	110(6)	110
001(1)	011(3)	011
010(2)	000(0)	000
011(3)	101(5)	101
100(4)	010(2)	010
101(5)	100(4)	100
110(6)	111(7)	111
111(7)	001(1)	001

Here next state and outputs are same

2.2 Next state and output Equations:

nQ=f (Q_2, Q_1, Q_0,Reset), Y=f(Q_2, Q_1, Q_0,Reset)
nQ_2**:**

Q_2 \ Q_1Q_0	00	01	11	10
0	1	0	1	0
1	0	1	0	1

D_2=Q_0 xnor (Q_1 xor Q_0)

nQ_1**:**

Q_2 \ Q_1Q_0	00	01	11	10
0	1	1	0	0
1	1	0	0	1

D_1=$Q_2.\overline{Q_0} + \overline{Q_2}.\overline{Q_1}$

nQ_0:

Q_2 \ Q_1Q_0	00	01	11	10
0	0	1	1	0
1	0	0	1	1

D_0=$\overline{Q_2}.Q_0 + Q_2.Q_1$

$Y_2 = nQ_2 = \overline{reset}.D_2$
$Y_1 = nQ_1 = reset + D_1$
$Y_0 = nQ_0 = \overline{reset}.D_0$

2.3 Circuit Diagram:

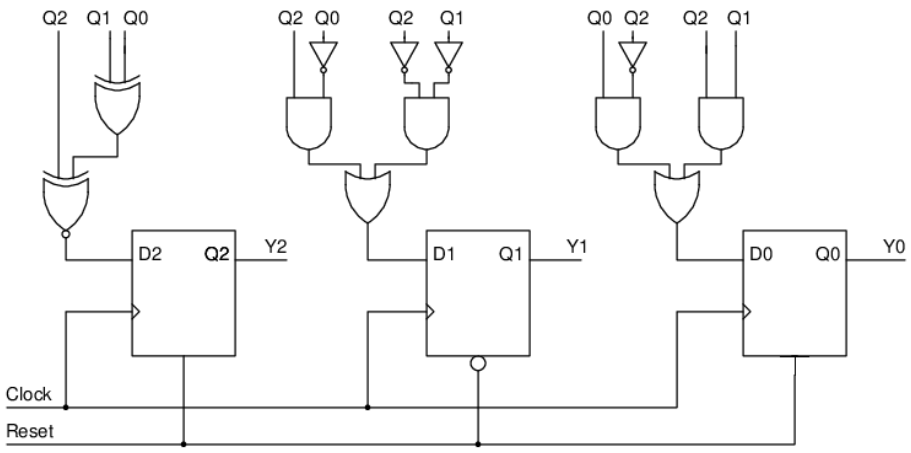

Figure 2: Circuit diagram of the sequence generator

3 VHDL Code:

3.1 Structural Style:

Instantiate 3 D flip flops and interconnect with required combinational logic so that the code should depict above circuit.

D flip flop:

```
--D flip flop
library ieee;
use ieee.std_logic_1164.all;
entity dff is port(D,clk,reset:in std_logic;Q:out std_logic);
end entity dff;
architecture behav of dff is
begin
dff_proc: process (clk,reset)
begin
if(reset='1')then
Q <= '0';
elsif (CLK'event and (CLK='1')) then
Q <= D;
end if ;
end process dff;
end behav;
```

Template code

```
library ieee;
use ieee.std_logic_1164.all;
library work;
use work.<user_defined_packege_name>.all;

entity Sequence_ex is
port (reset,clock: in std_logic;
y:out std_logic_vector(2 downto 0));
end entity Sequence_ex;

architecture struct of Sequence_ex is
signal D2,D1,D0 :std_logic;
signal Q:std_logic_vector(2 downto 0);
begin
-- write the equations here
D2<=
D1<=
D0<=
y(2)<=
y(1)<=
y(0)<=
-- Do the port mapping
--Q0
dff_0  : dff0 port map();
--Q1
dff_1  : dff1 port map();
--Q2
dff_2  : dff2 port map();
end struct;
```

3.2 Behavioral Style:

```
library ieee;
use ieee.std_logic_1164.all;

entity Sequence_behavior is
port (reset,clock: in std_logic;
y:out std_logic_vector(2 downto 0));
end entity Sequence_behavior;

architecture behav  of Sequence_behavior is
signal state:std_logic_vector(2 downto 0);     --state binary encoding
constant s_2:std_logic_vector(2 downto 0) : ="010";
-- write the remaining constant declarations
begin
-- process for next state & flip flop logic
reg_process: process(clock,reset) --sensitivity list
begin
if(reset='1')then
state<=;  --write reset state
elsif(clock'event and clock='1')then
case state is
     --reset
     when s_2=>
    state<=s_0;
    -- write remaining choices
     --DEFAULT choice
     when others=>
       state<=; --write the reset state
     end case;
end if;
end process reg_process;
-- output logic concurrent statemet or one more process
y<=state;
end behav;
```

4 LAB TASK:

Complete the VHDL description of sequence generator and Perform RTL and Gate level simulation and scan chain with the given **TRACEFILE**

Tracefile format $< reset\ clock >< y_2y_1y_0 > 111$

5 Home Work(to be submitted by next lab:)

Design and implement both structural (7 marks) and behavioral (3 marks) below sequence generator and Perform RTL and Gate level simulation and scan chain with the given **TRACEFILE**
Hint: Unused sequence should be mapped to one of the known sequence i.e reset sequence

Tracefile format $< reset\ clock >< y_3y_2y_1y_0 > 1111$