Experiment 7:
Designing Counters

Lab Project

Tanuj Chahar – 2023EE10036

Chetan Sai Morla – 2023EE10176

Yashwant Kasotiya – 2023EE10013

Table number- 14(Monday batch)

TA name: Pramsu Shrivastava Sir

Date of Submission: 28/10/2024

Submission Deadline: 11/11/2024

Contents:

1.Experiment Description:

1.1 Objective

To design a Synchronous 4-bit Gray-Code Counter using SR Flip-Flops and write a VHDL code.

1.2 Apparatus:

1. CPLD (Complex Programmable Logic Device)
2. Connecting cable
3. PC with compiler installed

1.3 Procedure:

The sr_flip_flop module implements an SR Flip Flop. Inside the always block, conditions for set and reset are specified, while the other two conditions are not defined. One undefined condition represents the memory state, which does not need to be explicitly handled, and the other is an invalid state that won't occur in this design.

The gcount module is used to instantiate a Gray code counter, which consists of 4 SR Flip Flops. The S and R values for each flip-flop are determined using Karnaugh maps. This is a synchronous counter, so all flip-flops are driven by the same clock signal. The output pins 28, 29, 33, and 31 are connected to out[0], out[1], out[3], and out[2], respectively, and the observations were recorded in a table.

```
module srff( input s, input r, input clk, output reg out);
    always@(posedge clk)
    begin
    if(~s && r)
        out <= 0;
    else if(s && ~r)
        out <= 1;
    end
endmodule


module main(input clk, output [3:0] out);

    srff sr1(~(out[1]^out[2]^out[3]),out[1]^out[2]^out[3],clk,out[0]);

    srff sr2((out[0])&(~(out[2]^out[3])),out[0]&(out[2]^out[3]),clk,out[1]);

    srff sr3((~out[0])&(out[1])&(~out[3]),(~out[0])&(out[1])&(out[3]),clk,out[2]);

    srff sr4((~out[0])&(~out[1])&(out[2]),(~out[0])&(~out[1])&(~out[2]),clk,out[3]);

endmodule
```

1.4 Theory :

1.Counter:

A counter is a digital circuit or device that records and counts the occurrences of specific events or conditions. In computing and electronics, counters are used to monitor sequences, measure time intervals, or count discrete events. Typically implemented using flip-flops, counters in digital electronics are designed to increment or decrement their stored value by a fixed amount (often by 1) with each clock pulse or input signal.

2. Synchronous Counter:

In a synchronous counter, all flip-flops receive the clock signal simultaneously. This simultaneous updating ensures that all bits change state at the same time, making synchronous counters faster and more reliable than asynchronous ones, as they avoid propagation delays.

3. Grey Code:

Gray code is a binary numbering system in which consecutive values differ by only one bit. This single-bit change property is particularly valuable in digital circuits where it helps to minimize errors and improve precision by avoiding multiple simultaneous bit transitions.

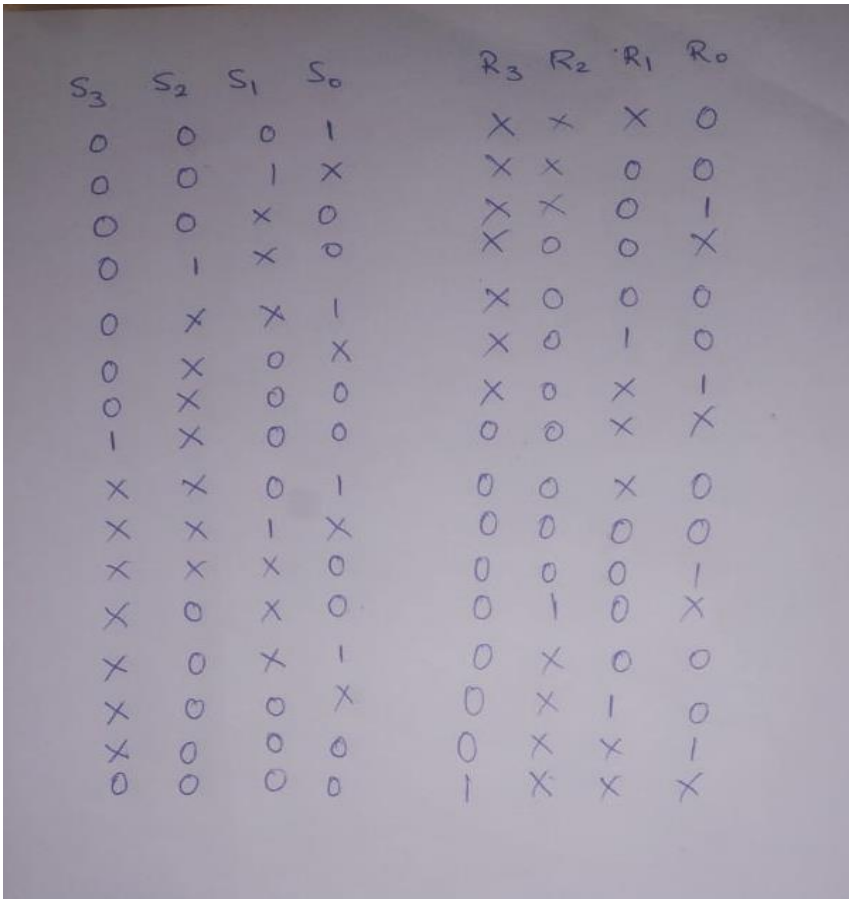
4. SR flip-flops :

An SR Flip-Flop, or SR latch, is a fundamental bistable circuit with two stable states that can store a single bit of information. Widely used in digital electronics, it serves as a memory element for storing binary data.

5.Synchronous 4-bit Gray-Code Counter :

A Gray code counter advances so that only one bit changes between any two consecutive states. This type of counter is especially useful in applications where reducing resource-intensive state transitions is critical, such as in high-precision or error-sensitive digital systems.

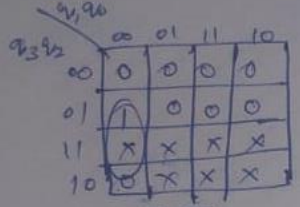
1.4 Observations:



S_3	S_2	S_1	S_0	R_3	R_2	R_1	R_0
0	0	0	1	X	X	X	0
0	0	1	X	X	X	0	0
0	0	X	0	X	X	0	1
0	1	X	0	X	0	0	X
0	X	X	1	X	0	0	0
0	X	0	X	X	0	1	0
0	X	0	0	X	0	X	1
1	X	0	0	0	0	X	X
X	X	0	1	0	0	X	0
X	X	1	X	0	0	0	0
X	X	X	0	0	0	0	1
X	0	X	0	0	1	0	X
X	0	X	1	0	X	0	0
X	0	0	X	0	X	1	0
X	0	0	0	0	X	X	1
0	0	0	0	1	X	X	X

K-maps of S and R flip flops

For S_3

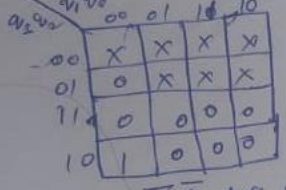


$S_3 = \bar{q}_0 \bar{q}_1 q_2$

for S_2

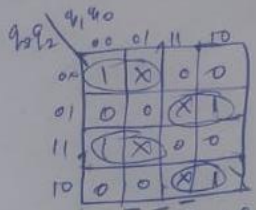
$S_2 = \bar{q}_0 q_1 \bar{q}_3$

for S_1



$S_1 = q_0 \bar{q}_2 \bar{q}_3 + q_0 q_2 q_3$

for S_0



$S_0 = \bar{q}_1 \bar{q}_2 q_3 + q_1 q_2 \bar{q}_3 + \bar{q}_1 q_2 q_3 + q_1 \bar{q}_2 q_3$

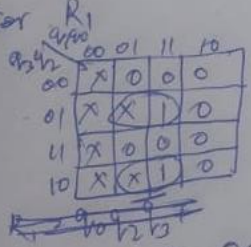
For R_3

$R_3 = \bar{q}_0 \bar{q}_1 \bar{q}_2$

for R_2

$R_2 = \bar{q}_0 q_1 q_3$

for R_1



~~$R_1 = q_0 q_2 q_3$~~

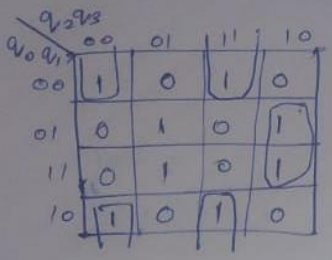
$R_1 = q_0 (q_2 \oplus q_3)$

for R_0

$R_0 = q_1 \oplus q_2 \oplus q_3$

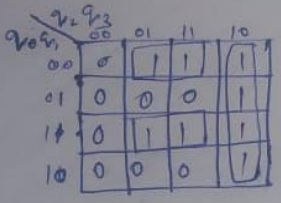
kmaps

For d_0



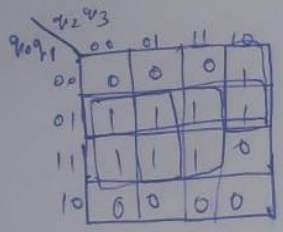
$d_0 = \bar{q}_1\bar{q}_2\bar{q}_3 + q_1\bar{q}_2q_3 + \bar{q}_1q_2q_3 + q_1q_2\bar{q}_3$

For d_1



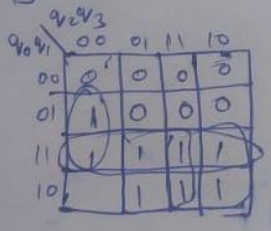
$d_1 = q_1\bar{q}_0 + q_0\bar{q}_2\bar{q}_3 + q_0q_2q_3$

For d_2



$d_2 = q_2\bar{q}_3 + \bar{q}_1q_2 + q_0q_2 + \bar{q}_0q_1\bar{q}_3$

For d_3



$d_3 = q_2q_3 + q_0q_3 + q_1q_3 + \bar{q}_0\bar{q}_1q_2$

State	gray code	binary
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0011
6	0101	0100
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111
16	0000	0000

Bromsy
28/10/24

Verified

Clock Diagram

Sources of error:

- Connections: Loose or incorrect connections can lead to wrong results.
- Errors in clock signal or wiring may cause incorrect counting.
- Malfunction of components: Improper working of any components can lead to incorrect results..
- Unstable power or clock signals on CPLD.
- LED's: Improper working of LED's and other components of circuit may cause incorrect results.

2. Precautions:

- Ensure connections are secure to avoid floating inputs or outputs.
- Double-check Verilog code syntax and connections before uploading to the CPLD.
- Check all the components beforehand if they are working Supply correct voltage.
- Don't work on the circuit with the voltage source ON.