BBQM

Under Supervision of:

DR\ Ahmed Shalaby && ENG\ Abdallah El Ghamry

➢ **Team members.**

1. Mohamed Sabry Mahmoud Shehab Eldean (Team Leader).
2. Mohamed Salama Elhalawane.
3. Mohamed Tarek Amin.

April 26, 2022

BFCAI

Team

➢The Project Description.

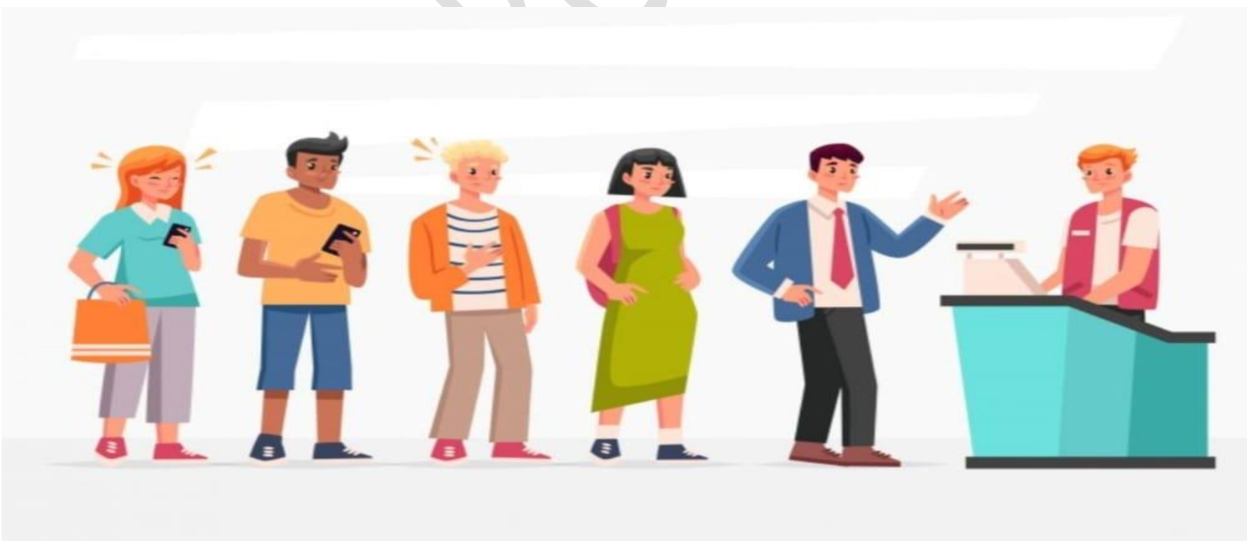
The manager of the Banha Bank branch at BFCAI suggested the installation of an embedded system to monitor the client queue in front of the tellers.

The proposed system Called BBQMTM, it is designed to display various information about the status of the queue.

The number of the tellers is [3/0], and the client level is 0 to 7 clients.

And it's got three Led (Full _flag, Empty _flag, Alarm).

The Alarm light when the number of client exceeds 7 or the number of client is less than 0, and The Full _flag light when the number of client is 7, and Empty _flag light when the number of client is 0.



➢The project consists of several module and each module has a certain function.

- Up down Counter
- Clock _divider
- Rom
- Counter _to_ decoder
- Decoder _7seg
- Flip flop

➢Up Down Counter

In this module is used to find out the inputs and outputs used within the project and it has the starting point.

This module consists of inputs (Clkup, Switch, reset) and outputs (Pcount, Full _ flag, Empty _ flag, Alarm).

Pcount: is the unit of the number of people standing in line and represents the number of bits [0-3].

Reset: is the point at which the project begins and the switch it is Step complete it.

When we start the project if the reset=0

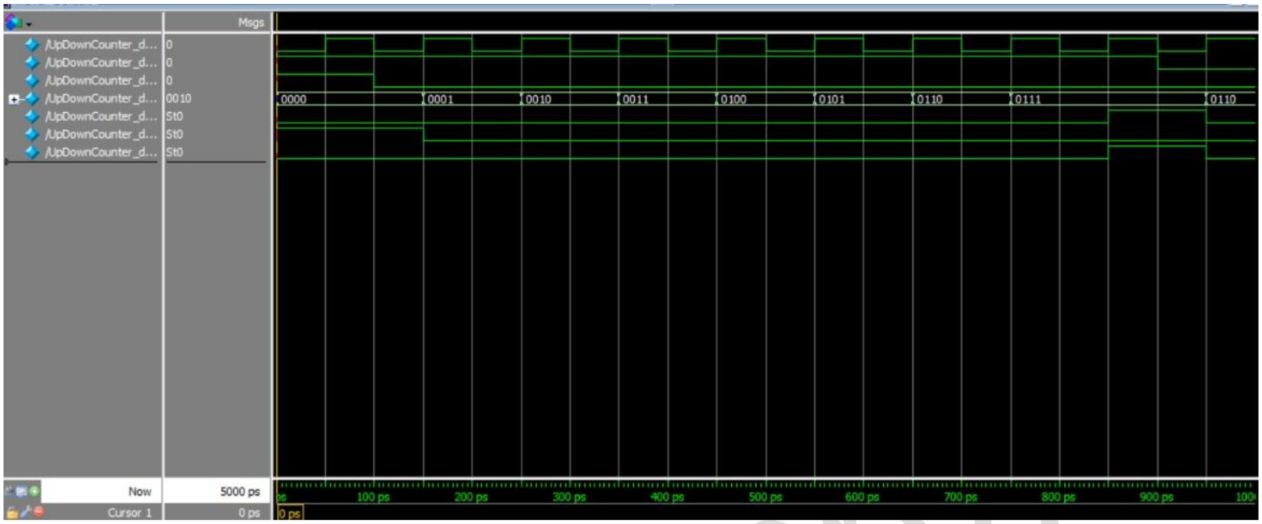
The led (Full _ flag and Alarm) is not light and the led Empty _ flag is light and, if the reset =1 we have two Cases.

First if Pcount=7 the led (Alarm) is light and if Pcount is another value Bigger than 0 the led (Alarm) is not light or when the Pcount value increases and then becomes =7 the led (Full _ flag) is light.

Secondly Pcount=0 the led (Alarm) is light and if Pcount is another value less than 7 the led (Full _ flag and Alarm) is not light or when the Pcount value increases and then becomes=0the led (Empty _ flag) is light.

These are the possibilities when running the system and the Led light up.

This is a schematic diagram of the test benches.



➢Clock divider

Clock divider works to divide the FPGA Clock to 1 Hz.

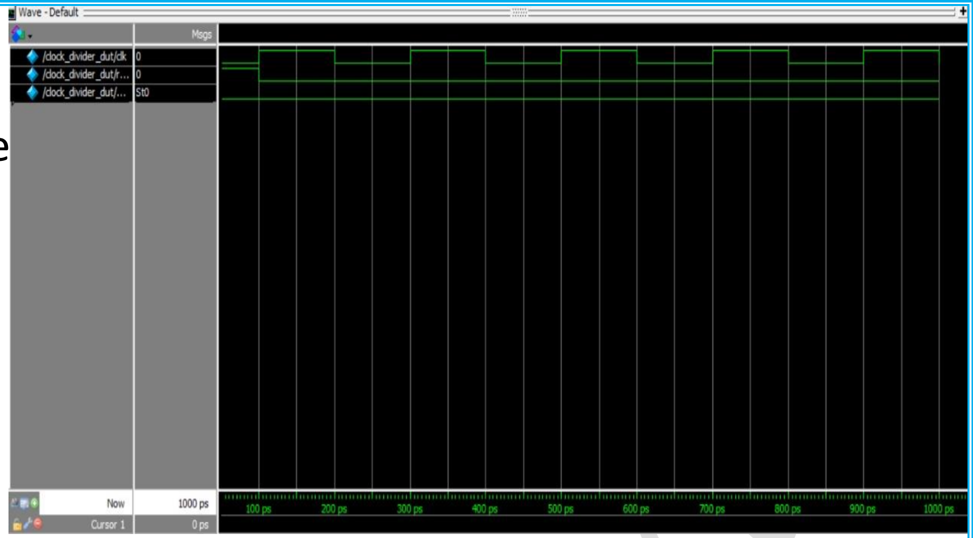
Taking into account the note as data, the n-bit counter will be used from top to bottom to create Pcount. Maximum Pcount value It will be (2n-1), with a virtual value of 7, where n is a general value, with a default value

Of 3.

Since FPGA clock is 25 MHz >>>>>>>>>>>1000 MS

SO X MHz >>>>>>>>>>>10 MS

So x=250000.



In short, this Module

Makes the FBGA

Change the count

Every 250000 if

Reset

So count =0 and slow _CLK<= 0 and when Count < 250000

So count <= count + 1 else slow _CLK = ~slow _CLK

And count <= 0.

This is a schematic Diagram of the test benches.

➢Rom

The Rom is used to store wtime and consists of pcoun, tcount , wtime.

And calculate the time in all possible cases and the pcount consists of 3 bits and a count consisting of 2 bits.

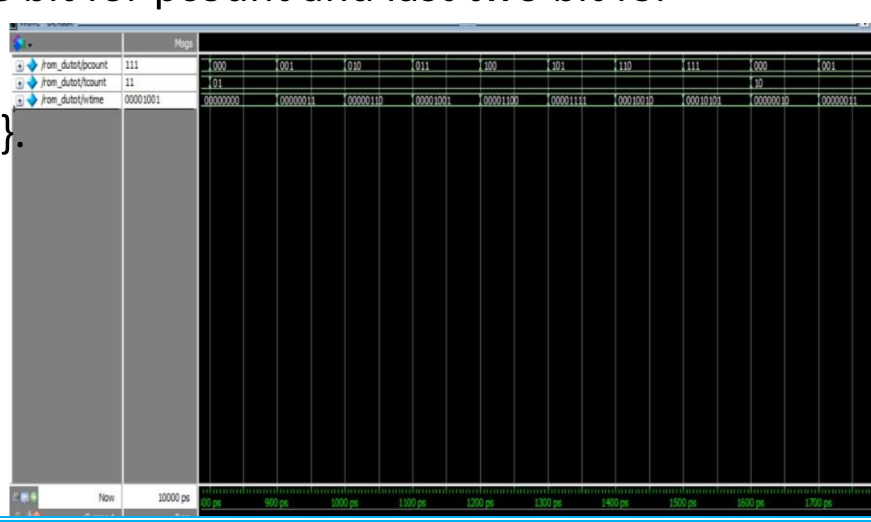
And the wtime is the proceeds of the integration of pcount, tcount and the first three bit for pcount and last two bit for tcount.

Wtime = {pcount, tcount}.

This is a schematic

Diagram of the test

Benches.



➢Decoder_7seg

Seven segment decoder using logic equations

A seven-segment display decoder takes a 4-bit data input A, B, C, D.

And produces seven outputs to control light-emitting diodes to display a digit from 0 to 9.

And each led has its own equation, for example:

Led _a = ~ (A | C | B&D | ~B&~D).

Led _b = ~ (~B | ~C&~D | C&D).

Led _c = ~ (B | ~C | D).

Led _d = ~ (~B&~D | C&~D | B&~C&D | ~B&C |A).

Led _e = ~ (~B&~D | C&~D).

Led _f = ~ (A | ~C&~D | B&~C | B&~D).

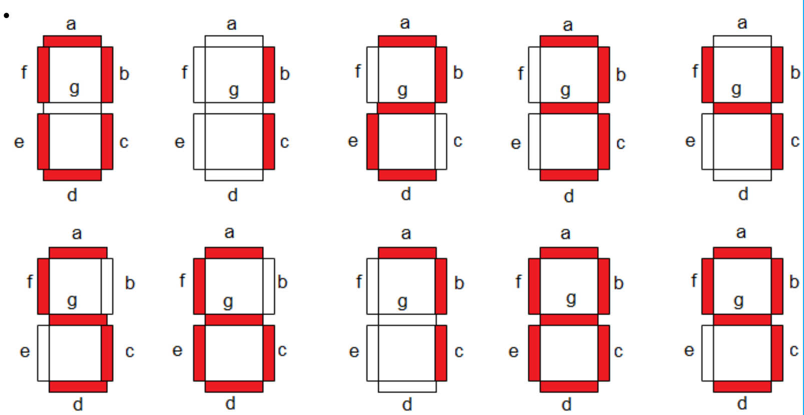
Led _g = ~ (A | B&~C | ~B&C | C&~D).

We use this Mark (~) because the seven segment decoder using by (common Anode).

And this is a picture

Showing how the led

Works.



This is a schematic Diagram of The test Benches.

➢flip flop

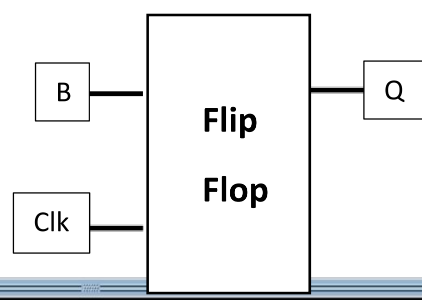
This module (flip flop) solving debouncing problem

This input (clk, b) and output (q).

And q<=b.

And this is a picture showing how

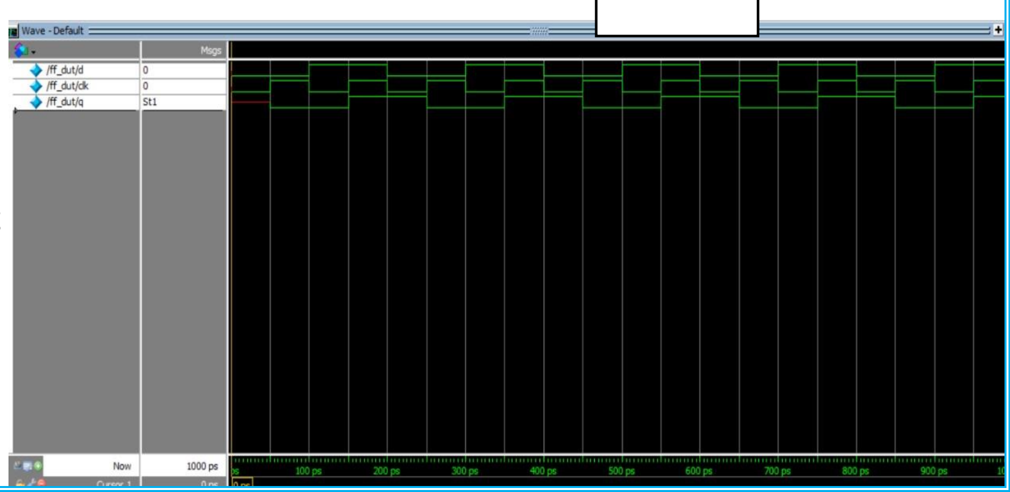
Flip flop the works.



This is a schematic

Diagram of

The test Benches.



➢counter _to_ decoder

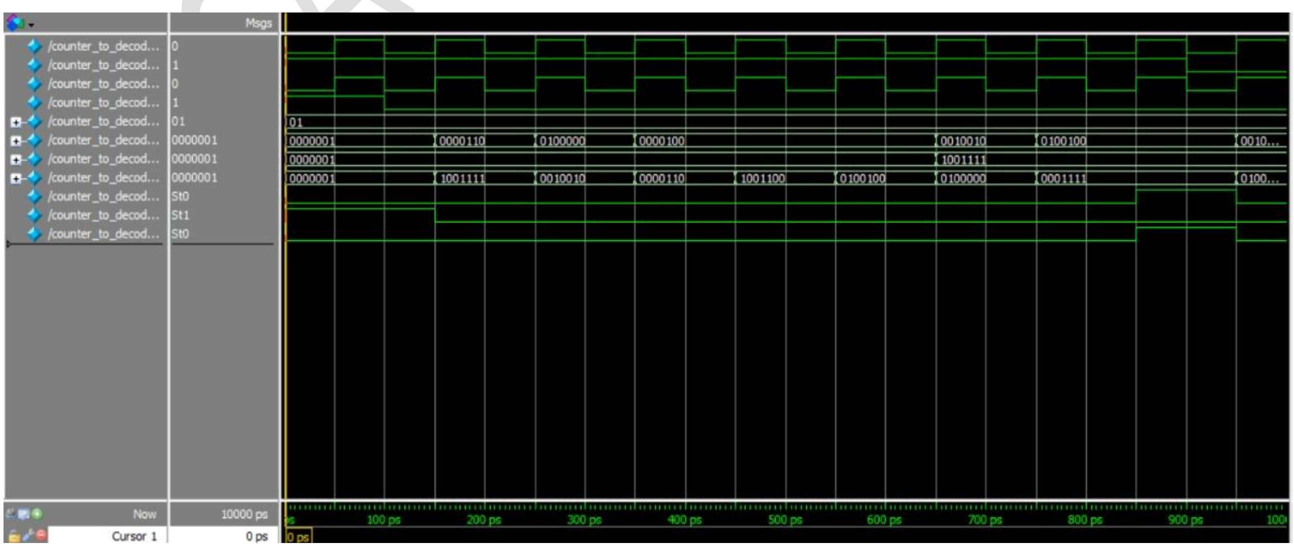
Counter _to_ decoder connect all modules to each other in an orderly manner.

The module contain all inputs of the project and output that will appear on FBGA.

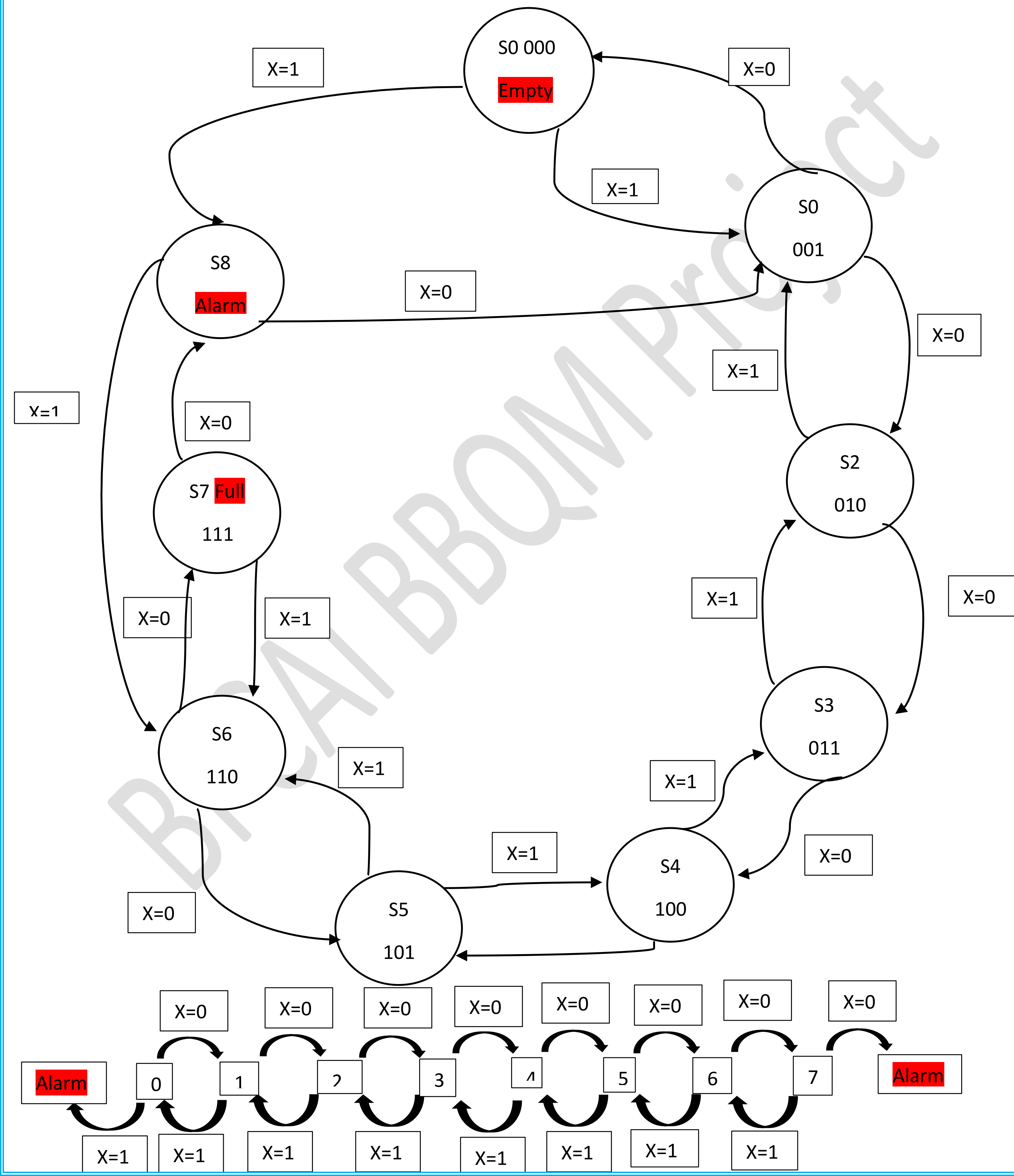
We take instant from each module and identifiy inputs and outputs and output of there.

The module connect to input in the next module each.

This is a schematic Diagram of The test Benches.



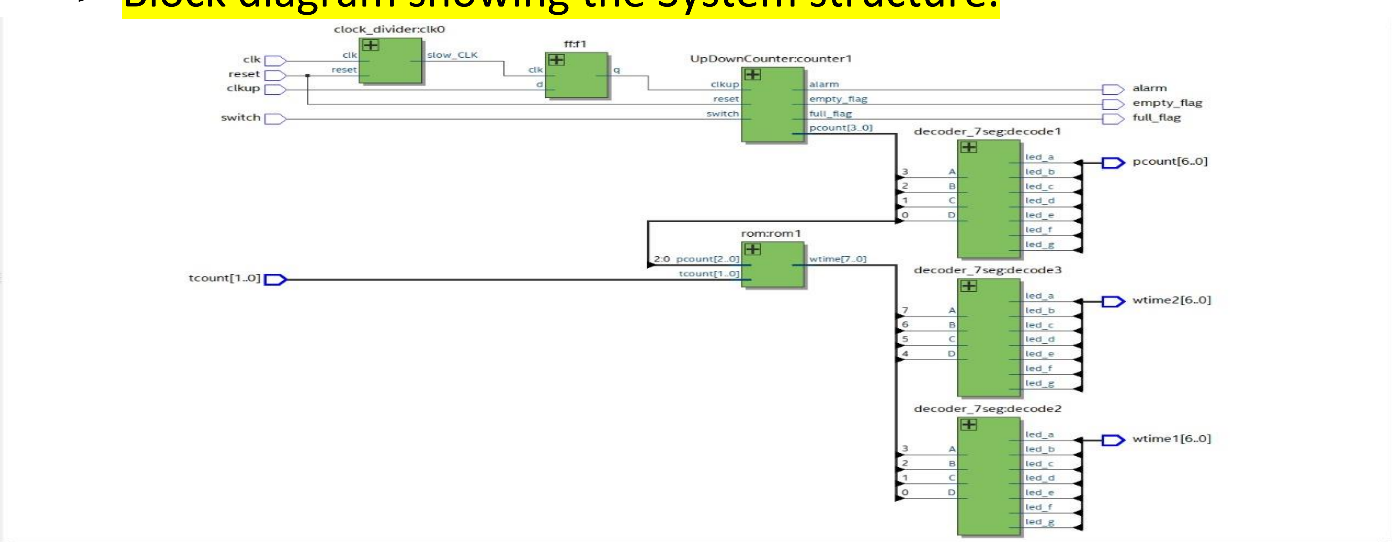
FSM diagram


S0 000
Empty
X=1
X=0
X=1
S0
001
S8
Alarm
X=0
X=0
X=1
x=1
X=0
S7 Full
111
S2
010
X=0
X=1
X=0
X=1
S3
011
S6
110
X=1
X=1
X=0
X=1
S4
100
S5
101
X=0
X=0
X=0
X=0
X=0
X=0
X=0
X=0
X=0
Alarm
0
1
2
3
4
5
6
7
Alarm
X=1
X=1
X=1
X=1
X=1
X=1
X=1
X=1

➢ Table identifies inputs and outputs and briefly describes their meaning and possible values.

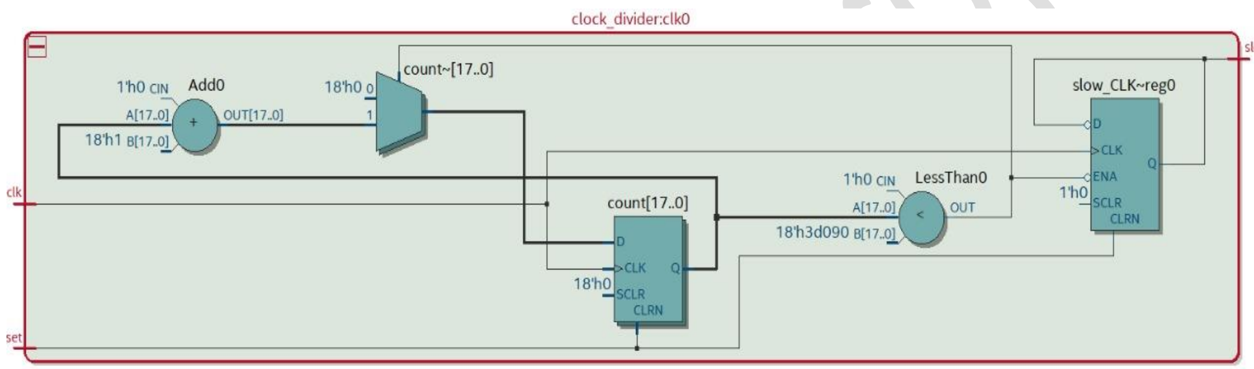
Reset	switch	pb	tcount1	tcount0	pcount2	pcount1	pcount0	w4	w3	w2	w1	w0	full	empty	alarm
1	x	x	0	0	0	0	0	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	1	1	0	1	0	0	1	0	0	0
0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	0
0	1	1	0	1	1	0	1	0	1	1	1	1	0	0	0
0	1	1	0	1	1	1	0	1	0	0	1	0	0	0	0
0	1	1	1	0	1	1	1	0	1	1	0	0	1	0	0
0	1	1	1	0	1	1	1	0	1	1	0	0	1	0	1
0	0	1	1	1	1	1	0	0	1	0	0	0	0	0	0
0	0	1	1	1	1	0	1	0	0	1	1	1	0	0	0
0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0
0	0	1	1	1	0	1	1	0	0	1	0	1	0	0	0
0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	1	0	0	0	1	1	0	0	0
0	0	1	1	1	0	0	0	0	0	0	1	0	0	1	0
0	0	1	1	1	0	0	0	0	0	0	1	0	0	1	1
1	x	x	1	1	0	0	0	0	0	0	1	0	0	1	0

➢ Block diagram showing the System structure.

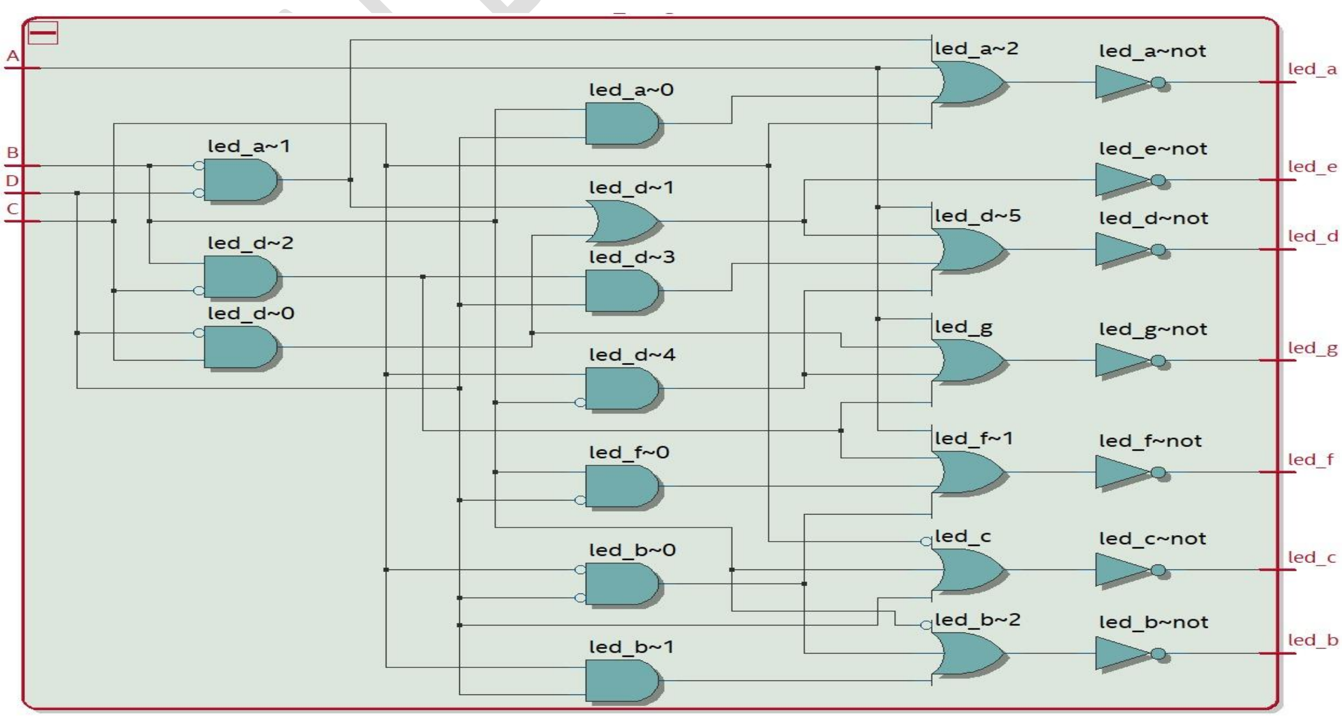


➢icon for the system showing inputs and outputs.

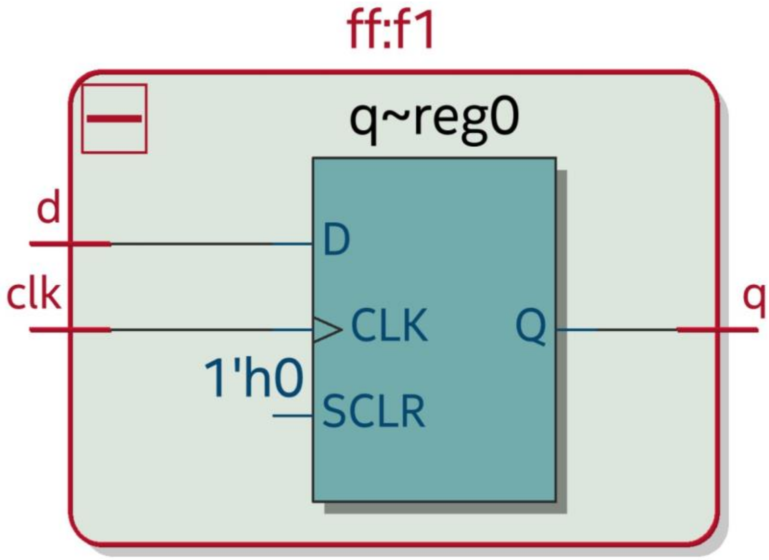
➢the block of clock divider



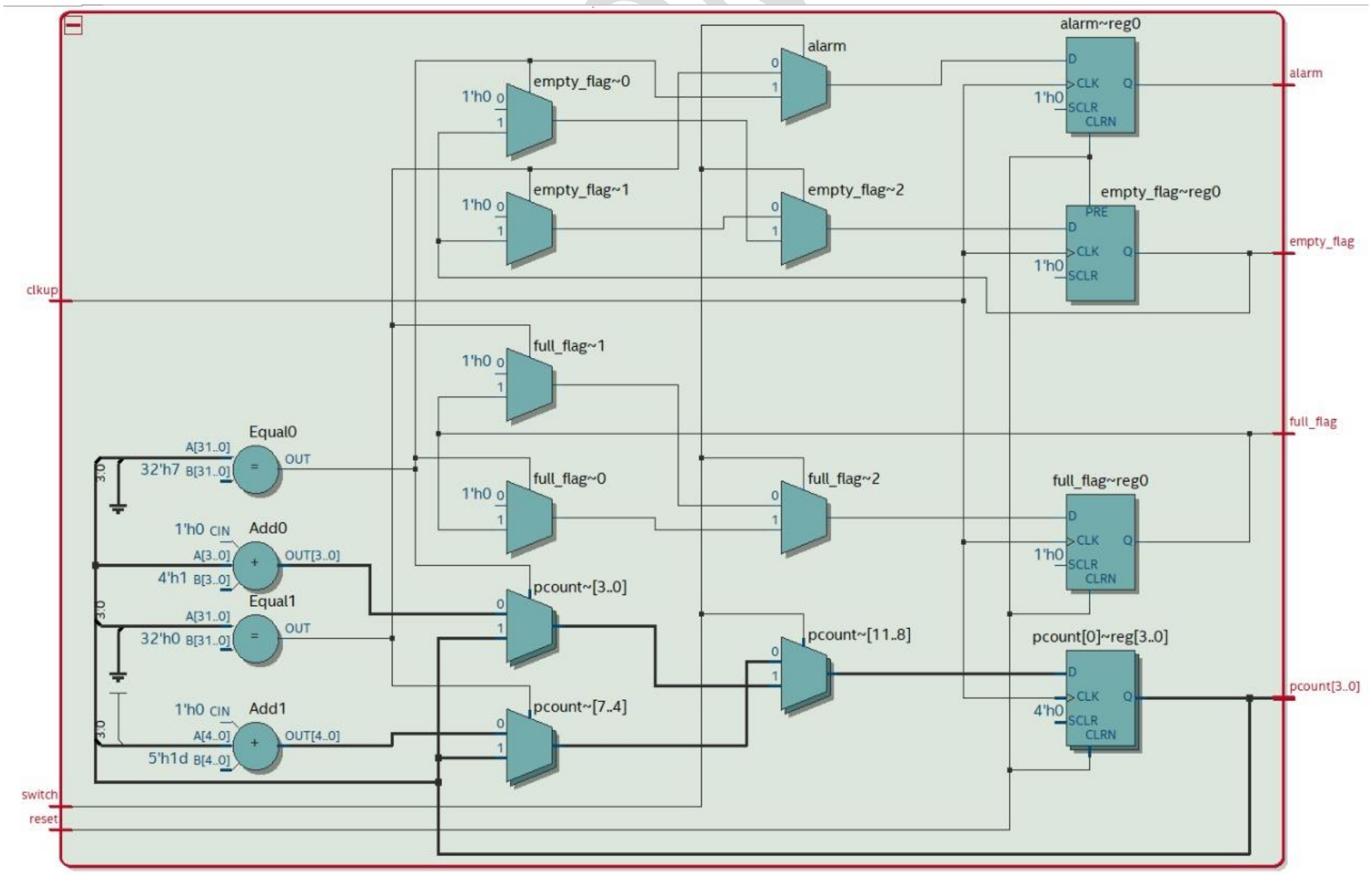
➢block of decoder



➢block of flip flop



➢block of up _down counter



➢At the end of this report we can calculate the wtime and all the probabilities of the pcount and we knew how to show it on seven segment and we knew solving debouncing problem and we know how organization this in counter _to_ decoder .and we use hierarchy in this project This project taught us how to use programming in practical life and how to divide project into modules and submodules and how to link all modules in onemodule

Merci beaucoup Dr:Ahmed Shalaby.

Eng:Abdullah Elghamry