"MINI PROJECT REPORT"
ON
AUTOMATIC WASHER CONTROL

A report submitted in partial fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS ENGINEERING DEPARTMENT

by

NAND KISHOR VYAS(1908410300025)
SAKSHI TRIPATHI(1908410300041)
UJJWAL PANDEY(1908410300053)
CHANDAN YADAV(1908410300016)

Under Supervision of
Prof/Dr.Ahinav gupta sir
Designation
(Duration: 28th November, 2021 to 23rd December, 2021)

ELECTRONICS ENGINEERING DEPARTMENT
RAJKIYA ENGINEERING COLLEGE
(An AICTE Approved Government Engineering College, Affiliated to AKTU Lucknow)
CHURK SONBHADRA, UATTAR PRADESH – 231206

Month & 2022

Department of Electronics Engineering
Rajkiya Engineering College
Churk, Sonbhadra – 231206, Uttar Pradesh, India

Email: contact@recsonbhadra.ac.in website: recsonbhadra.ac.in Fax: 05444-252002 Tel: 05444-252003

CERTIFICATE

This is to certify that this industrial training report entitled **"Title of industrial training"** submitted by **Nand Kishor vyas, sakshi Tripathi, Ujjwal pandey, Chandan kumar**, Roll No.: **1908410300025, 1908410300041, 1908410300053, 1908410300016**,in partial fulfilment of the requirements for the degree of Bachelor of Technology in Electronics Engineering of Rajkiya Engineering College, Sonbhadra,UP (An AICTE Approved Government Engineering College, Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow) during the academic year 2020-21, is a bonafide record of work carried out under my guidance and supervision. This is her own industrial training and the report is fit for submission.

Dr. Abhinav gupta sir
(Coordinator)
Assistant Professor
ELD, REC Sonbhadra

Dr. Himanshu katiyar sir
(Head of Department)
Associate Professor
ELD, REC Sonbhadra

DELCARATION

I, (**Nand Kishor Vyas, Sakshi Tripathi, Ujjwal Pandey, Chandan Yadav),** hereby declare that the Internship Training and Summer Project Report, entitled **" AUTOMATIC WASHER CONTROL"** , submitted to the **RAJKIYA ENGINEERING COLLEGE, SONBHADRA Uttar Pradesh** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology is a record of original training undergone by me during the period (**December 2021**) under the supervision and guidance of (**Dr Himanshu Katiyar , Dr Abhinav Gupta),** Department of Electronics Engineering, Rajkiya Engineering College, Sonbhadra and it has not formed the basis for the award of any Degree/Fellowship or other similar title to any candidate of any University.

Place:
SONBHADRA U.P (231206)

Nand Kishor Vyas
Sakshi Tripathi
Ujjwal Pandey
Chandan Yadav

Date: 11/01/2022

Table of Contents

1. **INTRODUCTION**

2. **DESIGN PROCESS AND OPERATION**

3. **BLOCK DIAGRAM**

4. **RTL SCHEMETIC VIEW**

5. **OUTPUT WAVEFORM**

6. **REFRENCE**

7. **APPENDIX**

   A) INTRODUCTION TO VERILOG HDL

   B) VERILOG CODE OF AUTOMATIC WASHER CONTROL

   C) TEST BENCH

1. INTRODUCTION

This project is based on principle of working a washing machine in real era using fsm machine.

A finite state machine (FSM) or finite state automation (pural: automata),or simply a state machine ,is a mathematical model of computation used to design both computer programs and sequential logic circuits. There are two types : Mealy machine & Moore machine .Mealy machine is a finite state whose output values are determined both by its current state and current inputs.

This is a deterministic finite state transducer : for each state and input , at most one transition is possible. Moore machine is a finite state machine whose output values are determined solely by its current state .In this paper , Mealy state machine is used to implement the control system of washing machine.The washing machine control system generates all the control signal is required for the operation of washing machine and is designed using Verilog HDL .The digital design is implemented on spartan 6 FPGA .Use of FPGAs facilitates the reduction in development cycle.

2.DESIGN PRPOCESS AND OPERATION

PRINCIPLE OF AUTOMATIC WASHING MACHINE

The inside bucket of fully-automatic washing machine has many small holes, through them the water bucket between the inside and outside is interlinked, electromagnetic valve pumps the water in and out. When the water fills in the control system the electromagnetic valve opens up, this allows the water to be fed into outer barrel. During drainage, the control system lets the drain electromagnetic valve open up, so the water by outer barrel drains out. When dehydrating, the control system will close valve and by turning on washing motor driving internal vats to dry. High, middle and low water level control switches are used to detect the high, middle and low water level. Start button activates the washing machine; stop button is used to manually stop water, drainage, dehydration and alarm. Drainage button is to achieve manual drainage [3]

Most of the washing machines today involve application of microcontrollers in order to perform various functions. The objective of this paper is to design an Application Specific IC (ASIC) for a washing machine. This ASIC will be used in the washing machines. The Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) has been used as the programming language. An ASIC is a customized integrated circuit. Implementation of analog circuit and mixed signal designs are possible in ASIC. It reduces the system cost, area, power consumption and also reduces the assembly and testing costs. It avoids the effects of components no longer being available. It gives design flexibility which helps in speed optimizations. ASIC requires fewer component hence reduces service/maintenance costs. It is more reliabl

WORKING OF AUTOMATIC WASHER MACHINE

As discussed earlier there are two types of washing machines, the top loader and the front loader. With either machine it is filled with clothing or other linens, some detergent is added and it turns on. They are directly hooked to water lines which bring water into the drum and mixes with the detergent. Then they agitate or bounce the clothing through the soapy water, thereby cleaning out the dirt and soil. The machines go into a spin cycle and pull all the water back out of the clothing. Once more, water fills the tub and rinses out the remaining soap. Again, with high speed spinning it spins out the water and leaves the clothes dried [1].

The functional block diagram of a washing machine is shown below in fig 1. Fig. 1. Block diagram of automatic washing machine In the above figure there are units which are controlled by a controller. These units are described below: 1. Valve control unit: This unit control the water inlet valve and drain valve. It contains all the information about the water valve, such as when it open and close. Similarily, it contains all the information about the opening and closing of drain valve. 2. Sensor unit: Sensor unit contains all the information of the sensors which are used in washing machine, such as, sensor for load check(clothes weight) gives the information that how much load is present inside the tub accordingly water fills into the tub. Similarily, it contains information for all other sensors required in washing machine, such as, water availability check, detergent availability, door open/close, balance check and trap check. 3. Motor control unit: This unit controls the functioning of motor. It contains all the information about the rotation of motor, such as when motor rotates in clockwise direction and when in counter clockwise. It also contains the information when motor will on and off in all three cases i.e., washing, rinsing and drying. 4. Display unit: The display unit consist of LEDs

to indicate the completion of process, occurrence of some problem while washing, set or reset of buttons,

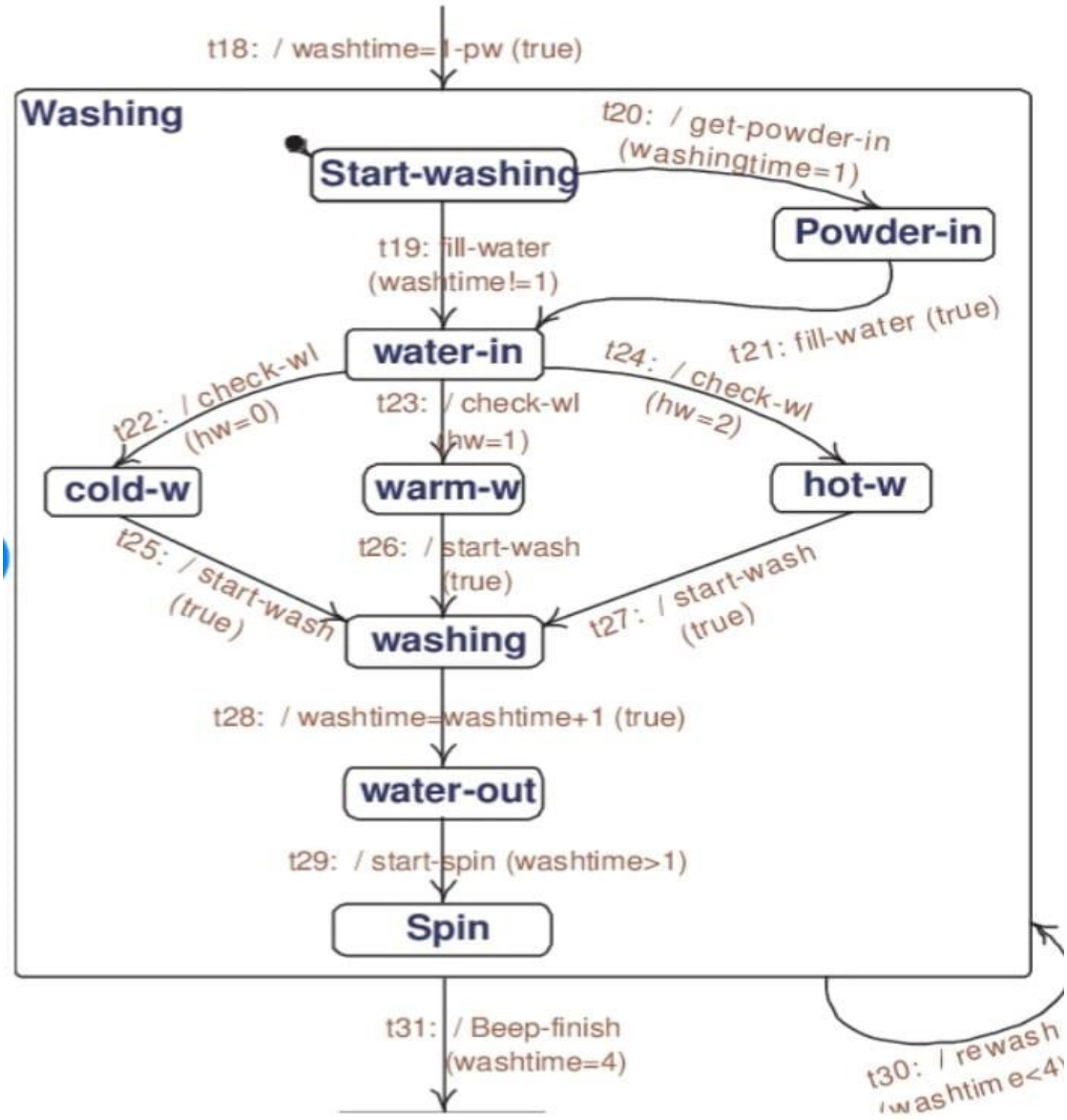
WASHING TECHNIQUE

There are two washing techniques used in washing machine. One of the technique is widely used in top loaded washing machine and the other is used in front loaded washing machine or say fully automatic washing machine. These techniques are explained below: 1. Agitator wash technique: In this technique, a rod with fin is used at the centre of the washing machine. A rubbing action of an agitator squeezes the dirt out of clothes. But it restricts the space and the clothes tend to get entangled. This technique is used in top loaded washing machine [5], [6]. The spinning requirements on vertical axis washers are much less so this part of the control is reasonably simple. However the agitation phase requirements are more complicated to achieve [7]. 2.

Tumble wash technique: This technique is used in front loading washing machine. This type of machine contains steel drum. This drum rotates along a horizontal axis and the clothes present inside the tub rub against its metal surface due to centrifugal action. The cleaning done by this technique is superior but there is a risk of ruining gentle fabrics [5], [6]. During the tumble action, the motor runs at low rpm clockwise and counter clockwise with a very high torque requirement. During the spin the motor is always in field weakening, so the motor works always very far from its nominal speed.

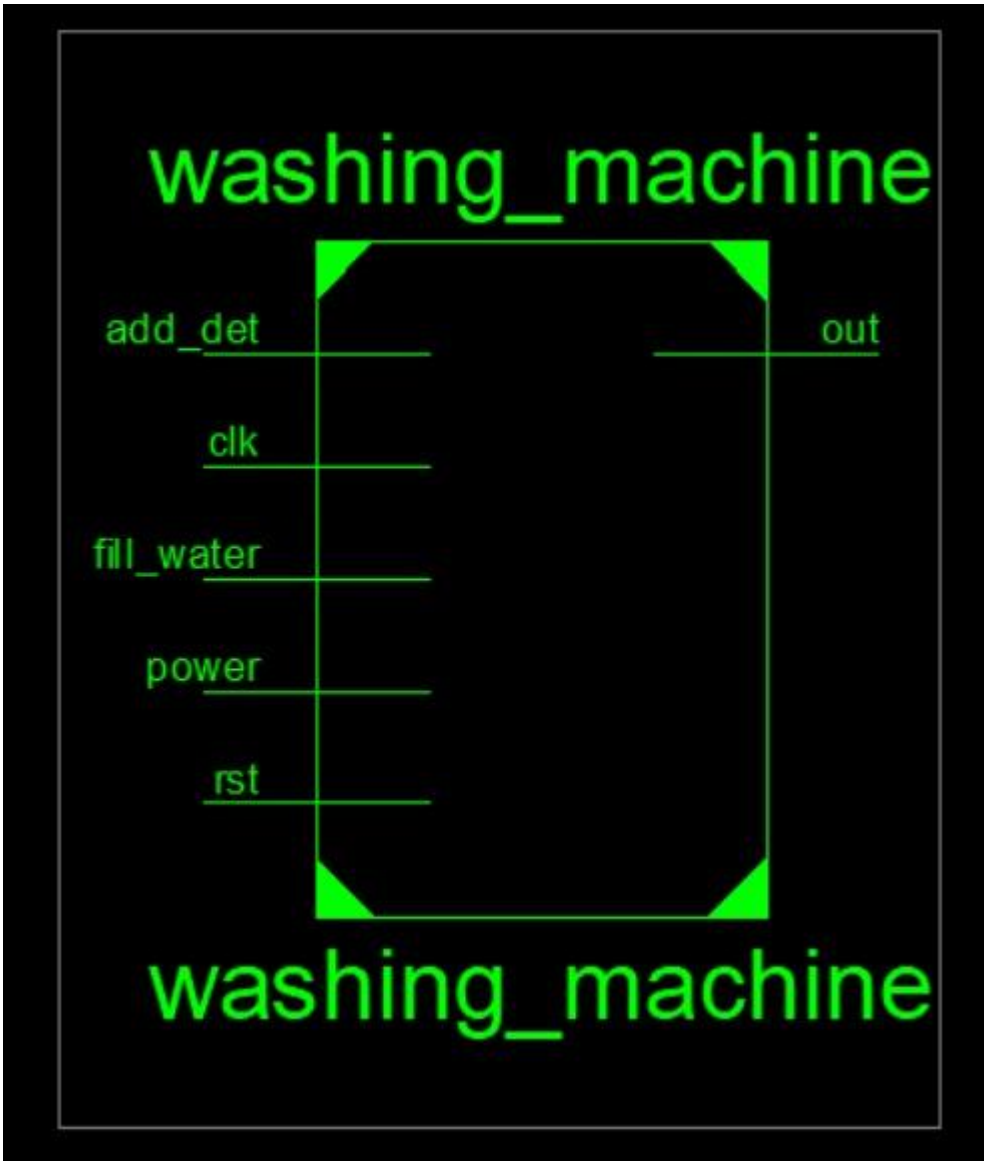
BLOCK DIAGRAM

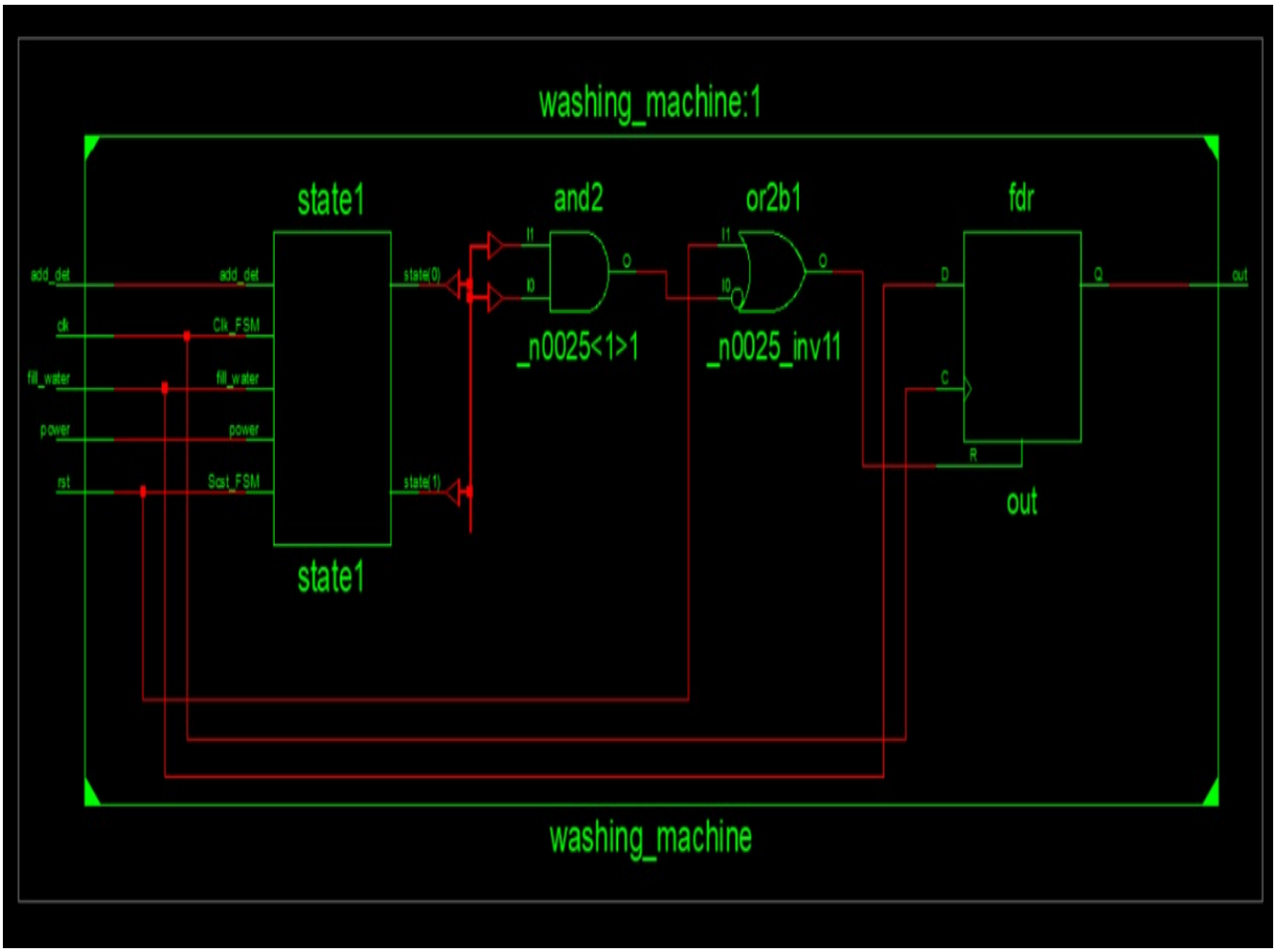
The block diagram of automatic washer control system is based on the finite state machine (FSM).

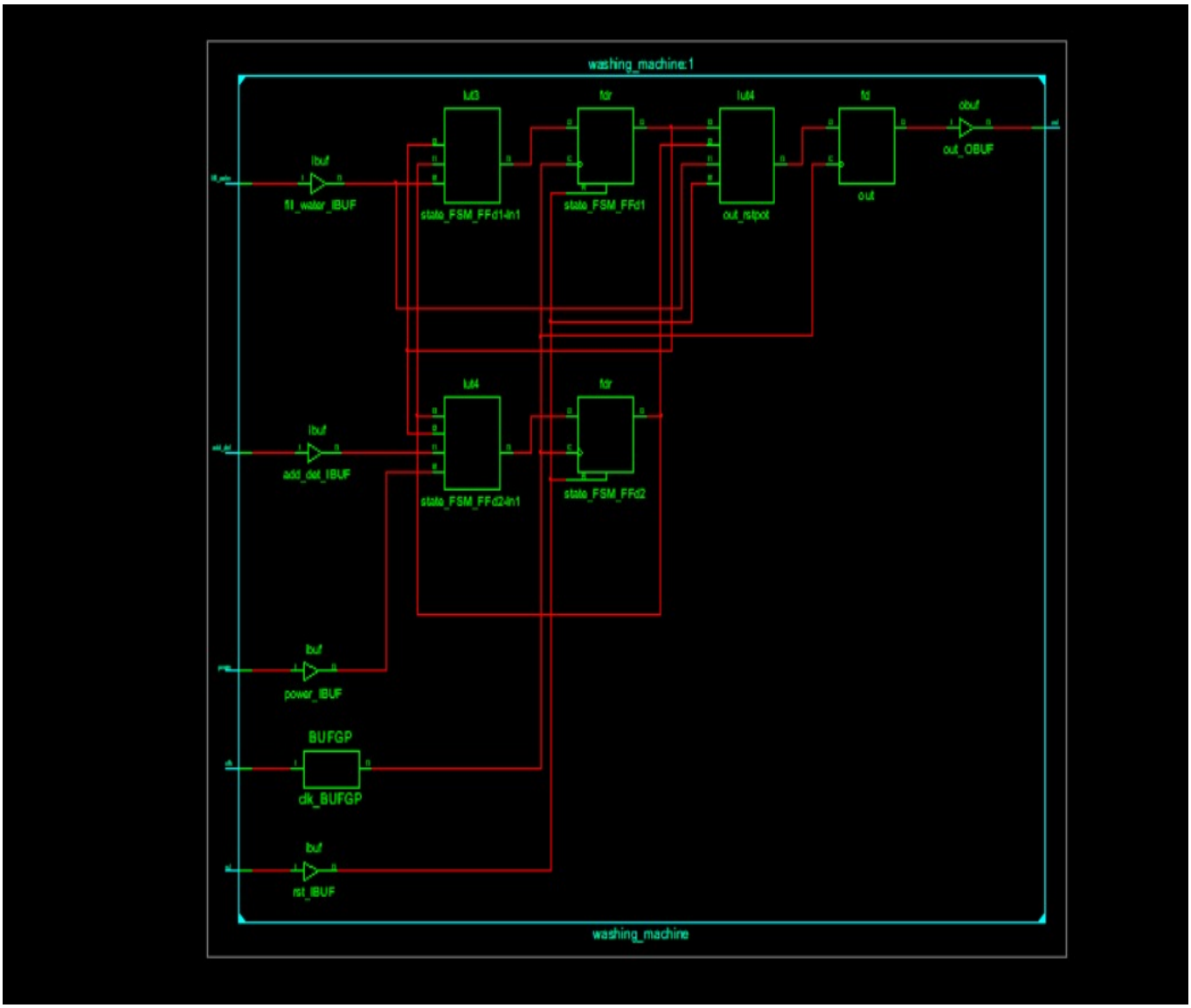


RTL SCEMATIC VIEW

The RTL SCHEMATIC view of automatic washer control system is given below :

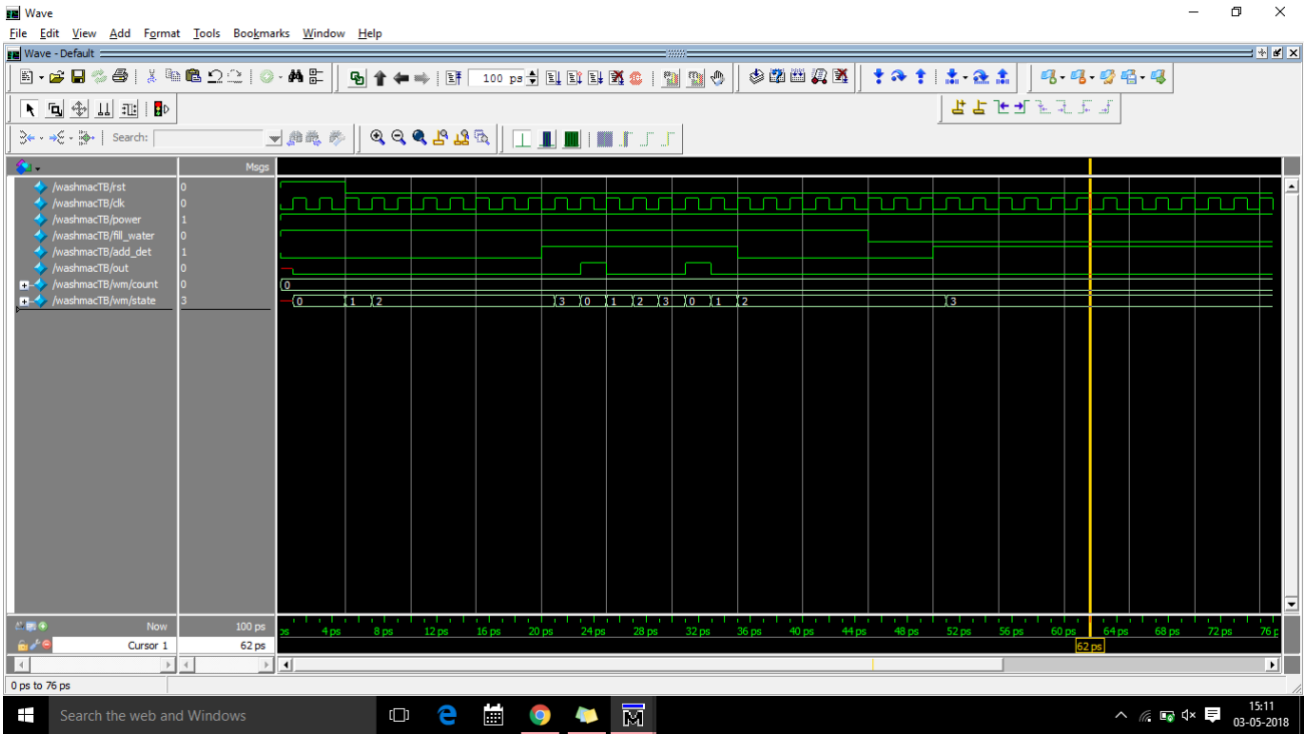


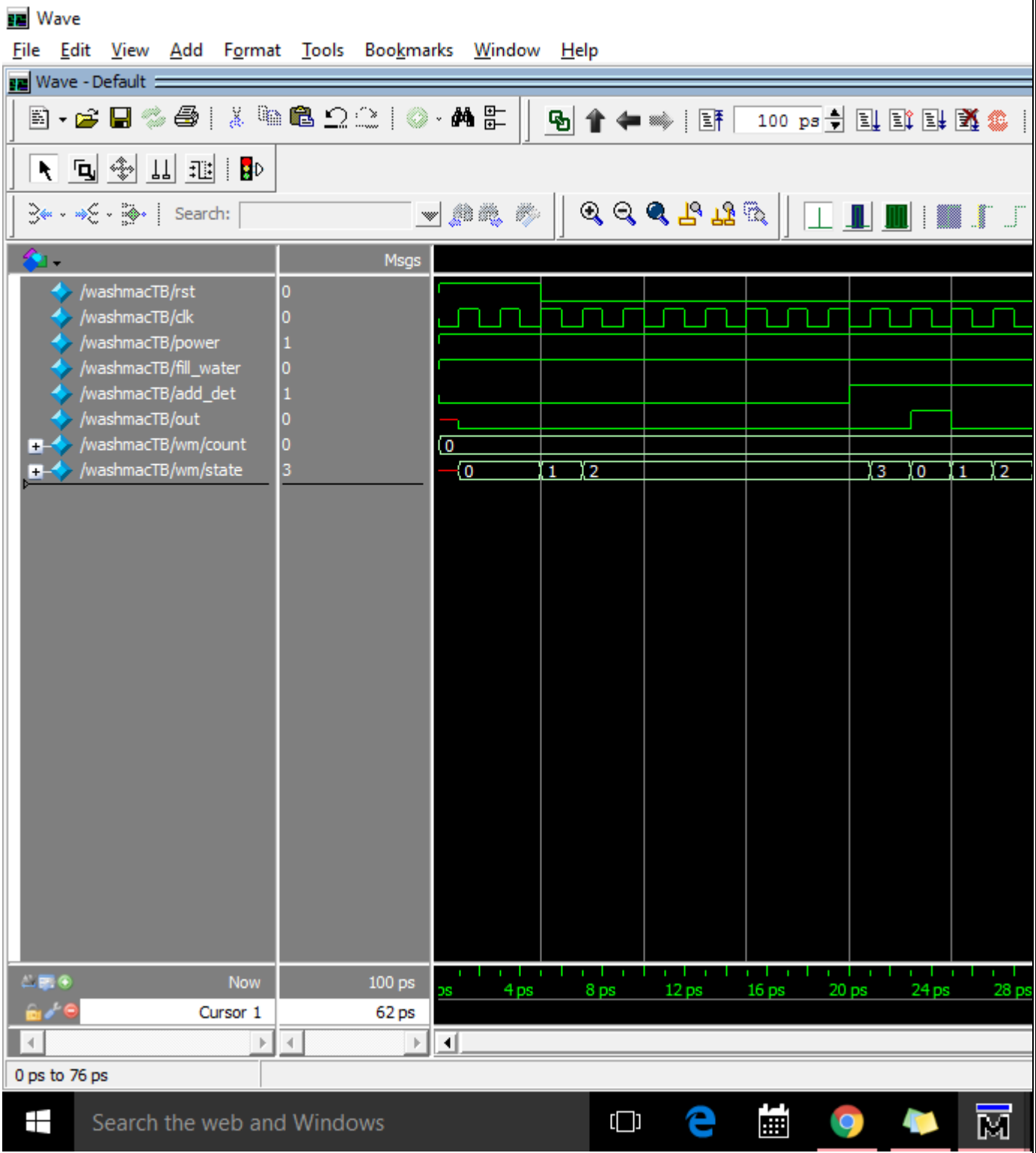

washing_machine:1
state1
and2
or2b1
fdr
add_det
clk
fill_water
power
rst
add_det
Clk_FSM
fill_water
power
Scst_FSM
state(0)
state(1)
I1
I0
O
_n0025<1>1
_n0025_inv11
D
C
R
Q
out
out
state1
washing_machine


washing_machine:1
lut3
fdr
lut4
fd
obuf
ibuf
fll_water_IBUF
state_FSM_FFd1-In1
state_FSM_FFd1
out_rstpot
out
out_OBUF
lut4
fdr
ibuf
add_det_IBUF
state_FSM_FFd2-In1
state_FSM_FFd2
power_IBUF
BUFGP
clk_BUFGP
rst_IBUF
washing_machine

OUTPUT WAVEFORM

The output waveform of the automatic washer control system is as given below.




Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
100 ps
Search:
Msgs
/washmacTB/rst 0
/washmacTB/clk 0
/washmacTB/power 1
/washmacTB/fill_water 0
/washmacTB/add_det 1
/washmacTB/out 0
/washmacTB/wm/count 0
/washmacTB/wm/state 3
0
0 1 2 3 0 1 2 3
Now 100 ps
Cursor 1 62 ps
ps 4 ps 8 ps 12 ps 16 ps 20 ps 24 ps 28 ps
0 ps to 76 ps
Search the web and Windows

INTRODUCTION TO VERILOG HDL

Verilog HDL is one of the two most common Hardware Description Languages (HDL) used by integrated circuit (IC) designers. The other one is VHDL. HDL's allows the design to be simulated earlier in the design cycle in order to more readable than schematics, particularly for large circuits. Verilog can be used to describe designs at four levels of abstraction:

(i) Algorithmic level (much like c code with if, case and loop statements).
(ii) Register transfer level (RTL uses registers connected by Boolean equations).
(iii) Gate level (interconnected AND, NOR etc.).
(iv) Switch level (the switches are MOS transistors inside gates).

The language also defines constructs that can be used to control the input and output of simulation. More recently Verilog is used as an input for synthesis programs which will

generate a gate-level description (a netlist) for the circuit. Some Verilog constructs are not synthesizable. Also the way the code is written will greatly effect the size and speed of the synthesized circuit. Most readers will want to synthesize their circuits, so nonsynthesizable constructs should be used only for test benches. These are program modules used to generate I/O needed to simulate the rest of the design. The words “not synthesizable” will be used for examples and constructs as needed that do not synthesize.

There are two types of code in most HDLs: Structural, which is a verbal wiring diagram without storage. assign a=b & c | d; /* “|” is a OR */ assign d = e & (~c); Here the order of the statements does not matter. Changing e will change a. Procedural which is used for circuits with storage, or as a convenient way to write conditional logic. always @(posedge clk) // Execute the next statement on every rising clock edge. count <= count+1; Procedural code is written like c code and assumes every assignment is stored in memory until over written. For synthesis, with flip-flop storage, this type of thinking generates too much storage. However people prefer procedural code because it is usually much easier to write, for example, if and case statements are only allowed in procedural code. As a result, the synthesizers have been constructed which can recognize certain styles of procedural code as actually combinational.

They generate a flip-flop only for left-hand variables which truly need to be stored. However if you stray from this style, beware. Your synthesis will start to fill with superfluous latches. This manual introduces the basic and most common Verilog behavioral and gate-level modelling constructs, as well as Verilog compiler directives and system functions. Full description of the language can be found in Cadence Verilog-XL Reference Manual and Synopsys HDL Compiler for Verilog Reference Manual. The latter emphasizes only those Verilog constructs that are supported for synthesis by the Synopsys Design Compiler synthesis tool. In all examples, Verilog keyword are shown in boldface. Comments are shown in italics

VERILOG CODE FOR AUTOMATIC WASHER CONTROL SYSTEM

```
module washing_machine(rst,clk,power,fill_water,add_det,out);

 input rst,clk,power,fill_water,add_det;

 output out;

 reg out;
```

```
reg [3:0] count=0;
reg [1:0] state;

parameter idle=0,a=1,b=2,c=3;

always @(posedge clk) begin

 if (rst==1) begin

 state <= idle;
 out <= 0;

 end

 else

case (state)

idle: if (power==1) begin

 state <= a;
 out <= 0;
```

```
 end

else
 begin

 state <= idle;
 out <= 0;

 end

a: if (fill_water==1) begin

 state <= b;
 out <= 0;

 end

else
 begin

 state <= a;
 out <= 0;

 end
```

```
b: if (add_det==1) begin

 state <= c;
 out <= 0;

 end

else
 begin

 state <= b;
 out <= 0;

 end

c: if (fill_water==1) begin

 state <= idle;
 out <= 1;

 end

else
```

```
 begin

 state <= c;
 out <= 0;

 end

endcase

   end

endmodule
```

TEST BENCH FOR AUTOMATIC CONTROL WASHER

/Testbench:

```
module washmacTB;

 reg rst,clk,power,fill_water,add_det;
 wire out;
```

```
 washing_machine
wm(.rst(rst),.clk(clk),.power(power),.fill_water(fill_water),.add_det(add_d
et),.out(out));


 initial begin


 $monitor($time,"rst %b clk %b power %b fill_water %b add_det %b out
%b",rst,clk,power,fill_water,add_det,out);


 clk=0;

 rst=1;

 power=1;

 fill_water=1;

 add_det=0;


#5 rst=0;

#5 power=1;

#5 fill_water=1;

#5 add_det=1;


#5 power=1;

#5 fill_water=1;

#5 add_det=0;


#5 power=1;
```

```
#5 fill_water=0;
#5 add_det=1;

 end
```