Experiment No. 08

**Design of RTL using Verilog HDL**

**OBJECTIVE:** To implement RTL design : Soda Dispenser Machine.

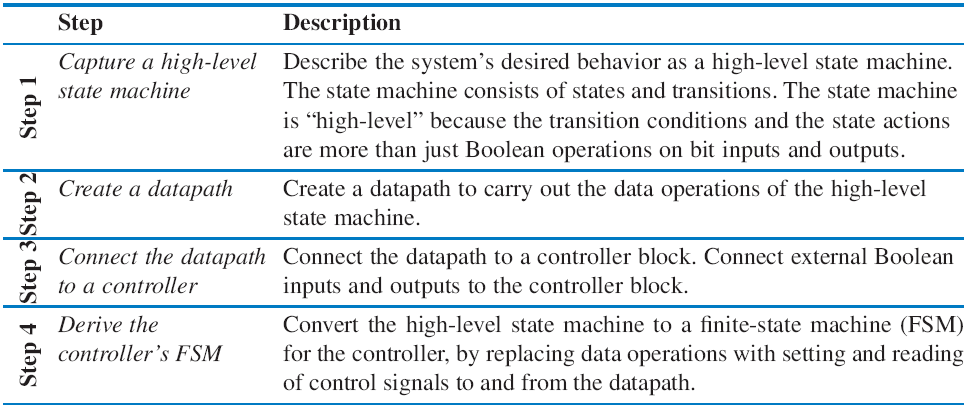
**SOFTWARE**: EDA playground

**THEORY:**

In [digital circuit design](https://en.wikipedia.org/wiki/Integrated_circuit_design#Digital_design), **register-transfer level** (**RTL**) is a design abstraction which models a [synchronous](https://en.wikipedia.org/wiki/Synchronous_circuit) [digital circuit](https://en.wikipedia.org/wiki/Digital_circuit) in terms of the flow of digital signals ([data](https://en.wikipedia.org/wiki/Data)) between [hardware registers](https://en.wikipedia.org/wiki/Hardware_register), and the [logical operations](https://en.wikipedia.org/wiki/Boolean_logic) performed on those signals.

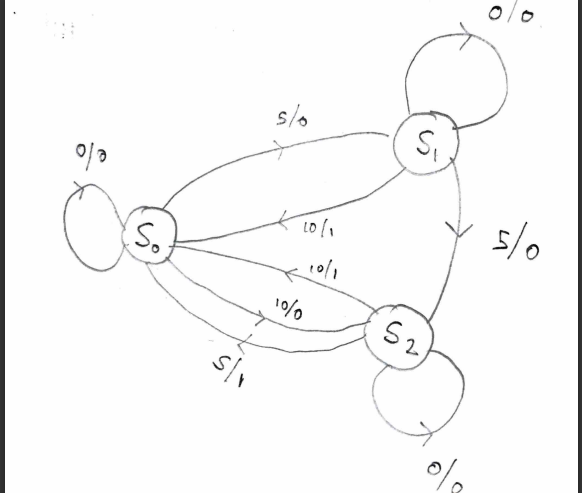
Register-transfer-level abstraction is used in [hardware description languages](https://en.wikipedia.org/wiki/Hardware_description_language) (HDLs) like [Verilog](https://en.wikipedia.org/wiki/Verilog) and [VHDL](https://en.wikipedia.org/wiki/VHDL) to create high-level representations of a circuit, from which lower-level representations and ultimately actual wiring can be derived.

RTL design Method :



**PROCEDURE:**

1. Open EDA playground
2. Write Verilog code and test bench
3. Select tool and click on open EV waveform
4. Generate Waveform.

STATE DIAGRAM :

**OUTPUT:**

**Code :**

// Code your testbench here

// or browse Examples

module vending\_machine\_tb;

//input

reg clk;

reg rst;

reg [1:0] in;

//output

wire out;

//instantiate the unit under test(UUt)

vending\_machine\_12 uut(

.clk(clk),

.rst(rst),

.in(in),

.out(out)

);

initial

begin

$dumpvars(1,vending\_machine\_tb);

//initialize inputs

rst =1;

clk =0;

#6 rst = 0;

in = 1;

#19 in = 2;

#10 rst = 1;

end

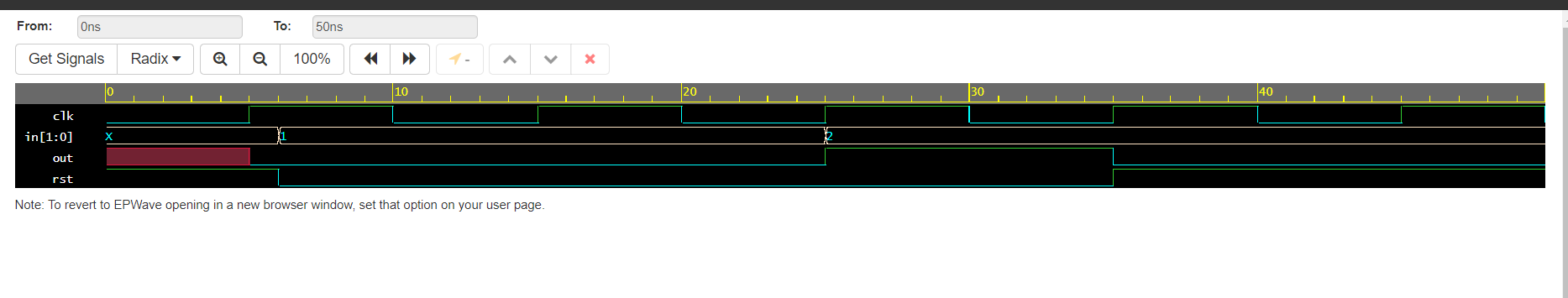
always #5 clk=~clk;

endmodule

**Testbench :**

module vending\_machine\_12(  
  input clk,  
  input rst,  
  input[1:0] in,  
  output reg out  
);  
  parameter s0=2'b00; // reset  
  parameter s1=2'b01; //5rs state  
  parameter s2=2'b10;  
  reg[1:0] c\_state,n\_state;  
  
  always @(posedge clk)  
begin  
      if (rst==1)  
        begin  
          c\_state = 0;  
          n\_state = 0;  
          out=0;  
        end  
      else  
        begin  
          c\_state =n\_state;  
            case (c\_state)  
            s0: if (in==0)  
              begin  
                n\_state =s0;  
                out=0;  
              end  
            else if(in==2'b01)  
              begin  
                n\_state =s1;  
                out =0;  
              end  
            else if(in==2'b10)  
              begin  
                n\_state =s2;  
                out =0;  
              end  
             s1: if (in==0)  
              begin  
                n\_state =s0;  
                out=0;  
  
              end  
            else if(in==2'b01)  
              begin  
                n\_state =s2;  
                out =0;  
              end  
            else if(in==2'b10)  
              begin  
                n\_state =s0;  
                out =1;  
  
              end  
             s2: if (in==0)  
              begin  
                n\_state =s0;  
                out=0;  
  
              end  
            else if(in==2'b01)  
              begin  
                n\_state =s0;  
                out =1;  
  
              end  
            else if(in==2'b10)  
              begin  
                n\_state =s0;  
                out =1;  
  
              end  
        endcase  
        end  
  
end  
endmodule

**Waveform:**

****

**CONCLUSION:**

**Implementation of RTL design : Soda Dispenser Machine in eda playground using Verilog was performed successfully. The concept of RTL and use of eda playground was studied. Basic of Verilog coding was learned. The output was generated and was verified.**