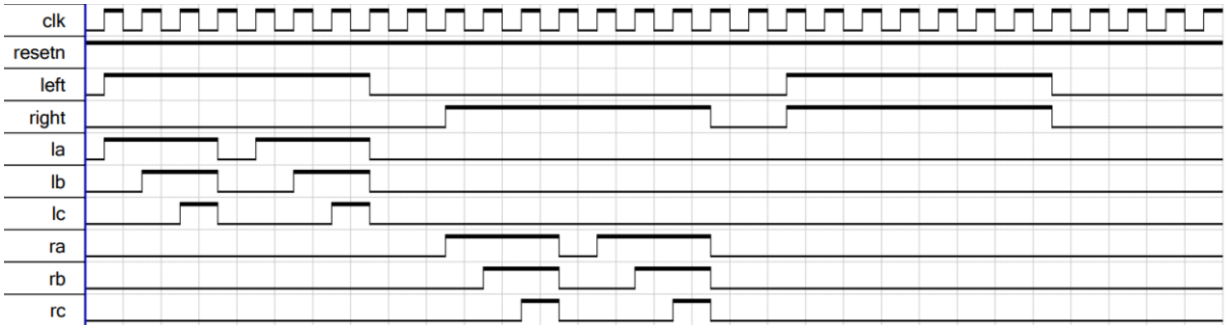
Task1

Verilog Code:

```
module eee304_5_14 (clk, resetn, left, right, la, lb, lc, ra,
rb, rc);

     input clk,resetn,left,right;

     output la,lb,lc,ra,rb,rc;

     reg [3:1] y,Y;

     parameter [3:1] LROFF=3'b000, LA=3'b001, LB=3'b010,
LC=3'b011, RA=3'b100, RB=3'b101, RC=3'b110;

 // Define the next state combinational circuit

     always @(left,right,y)

          case(y)

               LROFF: if(left & ~right) Y=LA;

                      else if(right & ~left) Y=RA;

                      else Y=LROFF;

               LA: Y=LB;

               LB: Y=LC;

               LC: Y=LROFF;

               RA: Y=RB;

               RB: Y=RC;

               RC: Y=LROFF;

               default: Y=LROFF;

          endcase

// Define the sequential block

     always @( negedge resetn, posedge clk)

          if(resetn == 0) y<=LROFF;

          else y<=Y;

// Define output combinational circuit

     assign la = (y==LA)|(y==LB)|(y==LC);

     assign lb = (y==LB)|(y==LC);

     assign lc = (y==LC);
```

```
    assign ra = (y==RA)|(y==RB)|(y==RC);
    assign rb = (y==RB)|(y==RC);
    assign rc = (y==RC);
endmodule
```

Output Waveform:



Task2:

Verilog Code:

```
module stompstompclap(clk, resetn, stomp, clap, out);
     input clk,resetn,stomp,clap;
     output out;
     reg [3:1] y,Y;
     parameter [3:1] A=3'b000, B=3'b001, C=3'b010, D=3'b011,
E=3'b100, F=3'b101;
 // Define the next state combinational circuit
     always @(stomp,clap,y)
          case(y)
               A: if(stomp & ~clap) Y=B;
                  else if (~stomp & clap) Y=A;
                  else if (~stomp & ~clap) Y=A;
                  else Y=A;//Stomp and clap are pressed
together
               B: if(stomp & ~clap) Y=C;
                  else if (~stomp & clap) Y=A;
                  else if (~stomp & ~clap) Y=A;
                  else Y=A;
               C: if(stomp & ~clap) Y=C;
                  else if (~stomp & clap) Y=D;
                  else if (~stomp & ~clap) Y=A;
                  else Y=A;
               D: if(stomp & ~clap) Y=E;
                  else if (~stomp & clap) Y=A;
                  else if (~stomp & ~clap) Y=A;
                  else Y=A;
               E: if(stomp & ~clap) Y=F;
                  else if (~stomp & clap) Y=A;
                  else if (~stomp & ~clap) Y=A;
```

```
                    else Y=A;
                F: if(stomp & ~clap) Y=C;
                    else if (~stomp & clap) Y=D;
                    else if (~stomp & ~clap) Y=A;
                    else Y=A;
                default: Y=A;
          endcase
// Define the sequential block
     always @(negedge resetn, posedge clk)
          if(resetn==0) y<=A;
          else y<=Y;
 // Define output combinational block
     assign out= (y==D)|(y==E)|(y==F);

endmodule
```

Output Waveform:

Figure 1: stomp stomp clap stomp stomp clap

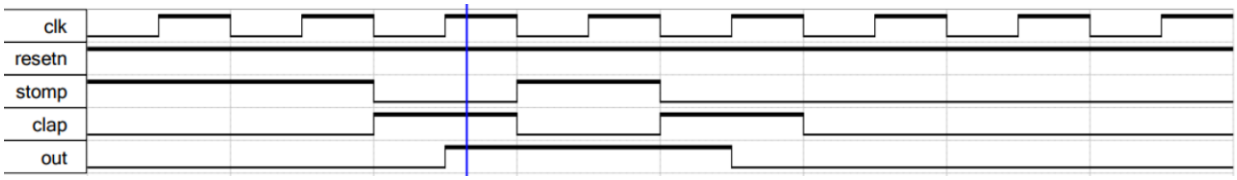


Figure 2: stomp stomp clap stomp clap

Figure 3: stomp stomp stomp clap clap stomp