Special study report

Shi Cheng

1. Project motivation

We are going to develop a tendon-driven neurosurgery robot to remove subdural hemorrhage. The main body of the robot is composed of multiple parts, and each part is made of ultra-relaxed nickel-titanium alloy spring. Each part is actuated by superelastic Nitinol wire. Taige leads the design of the main body of this robot. And I lead the design of the tendon driven actuation system. This drive device can also drive 6 tendons, respectively.

2. Background

Brain tumor, be it primary or metastatic, is usually life threatening for a person of any age. Primary surgical resection is one of the most effective ways of treating brain tumors and can have a tremendously increased success rate if the appropriate imaging modality is used for complete tumor resection. Magnetic resonance imaging (MRI) is the imaging modality of choice for brain tumor imaging because of its excellent soft-tissue contrast. MRI combined with continuum soft robotics has immense potential to be the next major technological breakthrough in the field of brain cancer diagnosis and therapy. A potential solution to lower the difficulty of this procedure is to house the flexible camera in a multisection continuum robot. By using multiple bending sections, the endoscopists can achieve an optimal tip position corresponding to topology of the tumor and the surrounding portions and can control other degree-of-freedom (DOF) motions to adjust the tip position in a confined space in the ventricles.

Moreover, planning of accessing postures of the multisection continuum robot before the endoscopic procedure for each patient may provide effective and minimally invasive access to the ventricles as suitable for the individual. An estimation of the robot posture is a key to success in the precision control and planning. Among different actuation principals of the multisection continuum robot, in particular, a tendon-driven continuum robot is an ideal choice for neuroendoscopy, as the tendon-driven continuum robot can provide adequate power through narrow, tortuous pathways and allow the actuators to be located at a safe distance from the patient.

3. Research status

After reading some related documents, I chose to choose the drive system from the following 4 directions: 1.SMA spring actuators drive 2. Motor drive 3. Hydraulic drive 4.Pneumatic drive. After considering the advantages and disadvantages of these four methods, I finally chose a hydraulically driven tendon. Not only can it be controlled linearly, the pulling force is also greater than pneumatic drive. Two syringes are connected to form a simple hydraulic system, as figure 1 shown. Only one syringe push rod is connected to the tendon, and the other syringe push rod is connected to a mechanical device, which can drive the syringe piston to reciprocate.

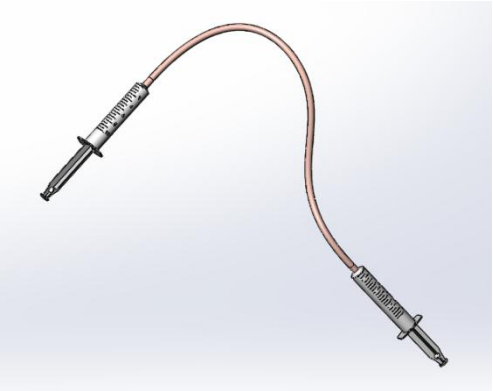


Fig1 Simple hydraulic system

4. Solution

Initial design: In the initial design, I used the mechanical mechanism of the cam pusher to drive the stretching and pressing of the syringe. As shown in Figure 2, the push rod slides on the cam to complete the reciprocating motion of forward and backward. Finally, Figure 3 shows the complete design of the motor-driven cam that finally drives the syringe piston movement.

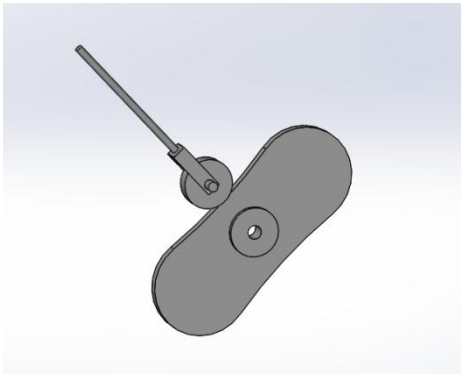


Fig.2 cam pusher

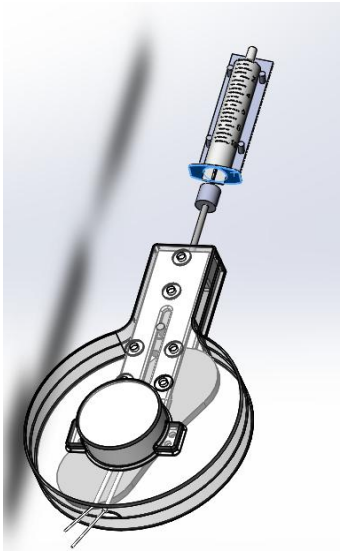


Fig.3 Initial design

Final design: In the initial design, when driving a single tendon, this mechanical structure requires a lot of space. And the mechanical structure is too complicated, it is not suitable for driving 6 tendons. We ended up using bolts and threaded rods to control the reciprocating movement of the syringe piston, as shown in Figure 4. By controlling the forward and reverse rotation of the threaded rod, the reciprocating movement of the bolt on the threaded rod is controlled. Figure 5 shows the complete design, which uses a specific structure to drive the six syringe pistons to reciprocate. Compared with the original design, the cost is lower, the structure is simpler, it is easier to manufacture, and the use space is smaller.

Fig.4 threaded rods and nuts

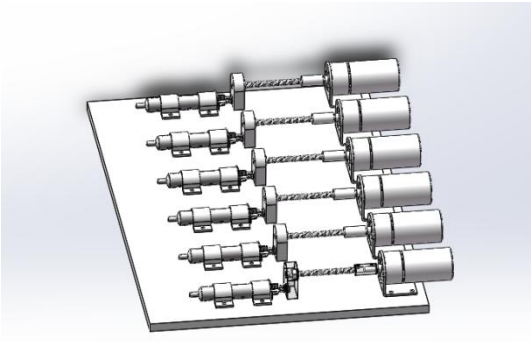


Fig.5 Final design

Directed research report

Shi Cheng

1. Project motivation

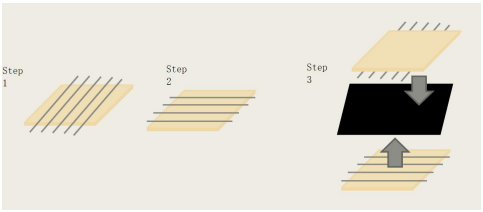
Project1:Retractable gloves that study how to simulate human palm touch
Project2:Investigate how to make vascular robots function properly in complex blood vessels without getting stuck.

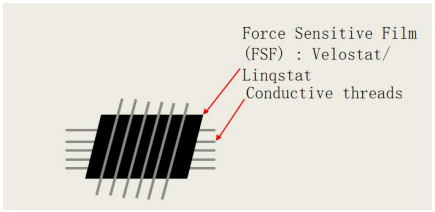
2. Background

Project1:Humans can feel, weigh, and hold various objects while applying the right amount of force—a difficult task for modern robots. A network of mechanoreceptors that provides sensory feedback and makes human-controlled flexibility still difficult to replicate in robots. Although computer vision-based robotic grabbing strategies have made great strides with large amounts of visual data and emerging machine learning tools, there are currently no equivalent sensing platforms and large-scale data sets available to detect human interactions. Use of haptic information. Here, we use a retractable array of gloves with tactile sensors to identify a single object, estimate its weight and explore the typical tactile pattern that occurs when grasping an object.
Project2:In recent years, vascular robot technology has been rapidly developed and is gradually applied to medical fields such as disease diagnosis, information collection, vascular dredging, and drug delivery. Due to the large number of branches of the blood vessels and the complicated blood environment in the blood vessels, the requirements for the vascular robot are high and the design is difficult. Many key technologies need to be further solved and the functions of the vascular robot need to be further improved. Here we use Ni-Ti Shape Memory Alloy Tubes for simulation experiments.

3. Research status and data collation

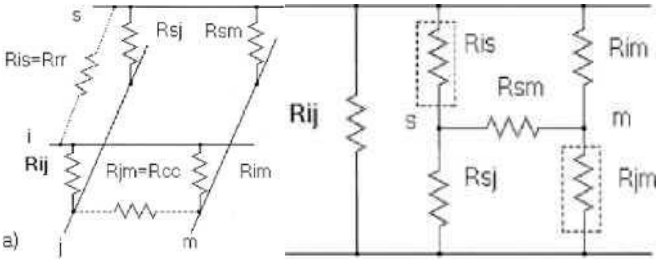
Project1:1.Tactile sensor selection and fabrication of sensor circuits
We can choose flexible resistance sensor structure or flexible capacitive sensor.They have advantages and disadvantages. The resistance pressure sensor has a high frequency response, high sensitivity, and good accuracy, but has a large nonlinearity for large strains and a weak output signal. However, certain compensation measures can be taken.The capacitive pressure sensor has good temperature stability, simple structure, good dynamic response, non-contact measurement and high sensitivity but high output impedance and poor load capacity.After discussion, two schemes are formulated, one is to make a resistive pressure sensor and a capacitive sensor. However, it is preferred to make a resistance sensor and its circuit.
After 3d printing 2 circuits, the Force Sensitive Film (FSF) was sandwiched in the middle, and the resistive pressure sensor and sensor circuit were also completed.



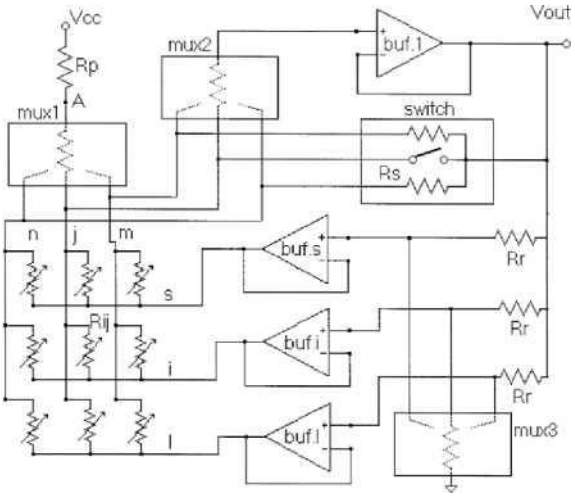


2. Design of the sensor circuit

Simplified fabrication of circuits is shown in the figure above after the preparation is completed.



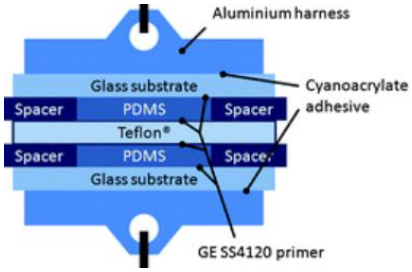
This is a measuring circuit. The scanning rows and columns are controlled by a computer to accurately measure the resistance change at each point.



$I_i = \frac{V_{out}}{R_{ij}} + (N-1)\frac{V_{out}}{R_{ik}}$ $\Delta I_i = \frac{V_{out}}{R_{ris}} + \frac{V_{out}}{R_{rik}}$ then we get $V_{out} = -V_i \frac{R_r}{R_{ij}}$

3. Print flat PDMS film

PDMS film is prepared by using a flat glass plate or a silicon plate during preparation. Do not separate.printing the circuit directly on a 3d printer .Separating the PDMS film after printing. And the circuit is just on the PDMS film.In addition,applying a teflon film to the glass,then we can easily to separate the pdms from the glass.



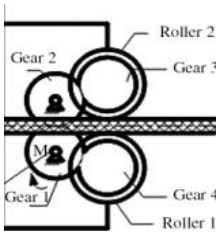
Project2:1.Pipe fixing device

Considering the elasticity of Ni-Ti pipes, silicone bolts can be used to fix the pipe clamp.

2.Thrust selection and device design

Silicone edge roller1 and roller2 clamp to fix.

Silicone edge gear 1and gear 2 provide thrust.



Pull force selection and device design

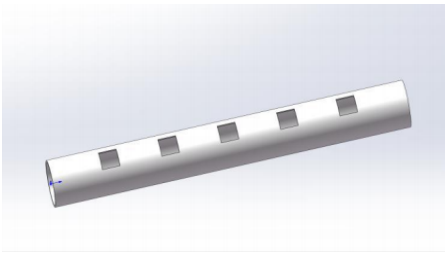
1. Consider the use of pressure difference to generate pressure as tension in the case of intravascular fluid.

2. Because of the Ni-Ti material, it is not a good solution to use magnetic field and magnetism to generate tensio.

4. Test Results

Project1:After changing a series of materials, a relatively complete circuit was eventually printed, but the resistance was too large and the stability could continue to improve. We need to optimize the structure of the circuit. 1. Reduce the printing speed of 3D printer 2. Increase the cross-sectional area of the printed circuit.

Project2:Limited by the length of the material, it can be easily interspersed with only thrust. So change the experimental direction and consider making a small hole in the tube and adding silica gel. Control the deformation and recovery of the tube by controlling the deformation of the silica gel by temperature and so on.



5. Summary

Project1:After the sensor array is printed, we must consider the construction of the measurement circuit and how to control the fixed-point measurement through the computer. Select from the FPGA development board or microcontroller and write the code.

Project2:Now we need to start thinking about simplifying and analyzing the mathematical model of the tube under various stress conditions.

Final project

Student：Shi Cheng

Student ID：X673894

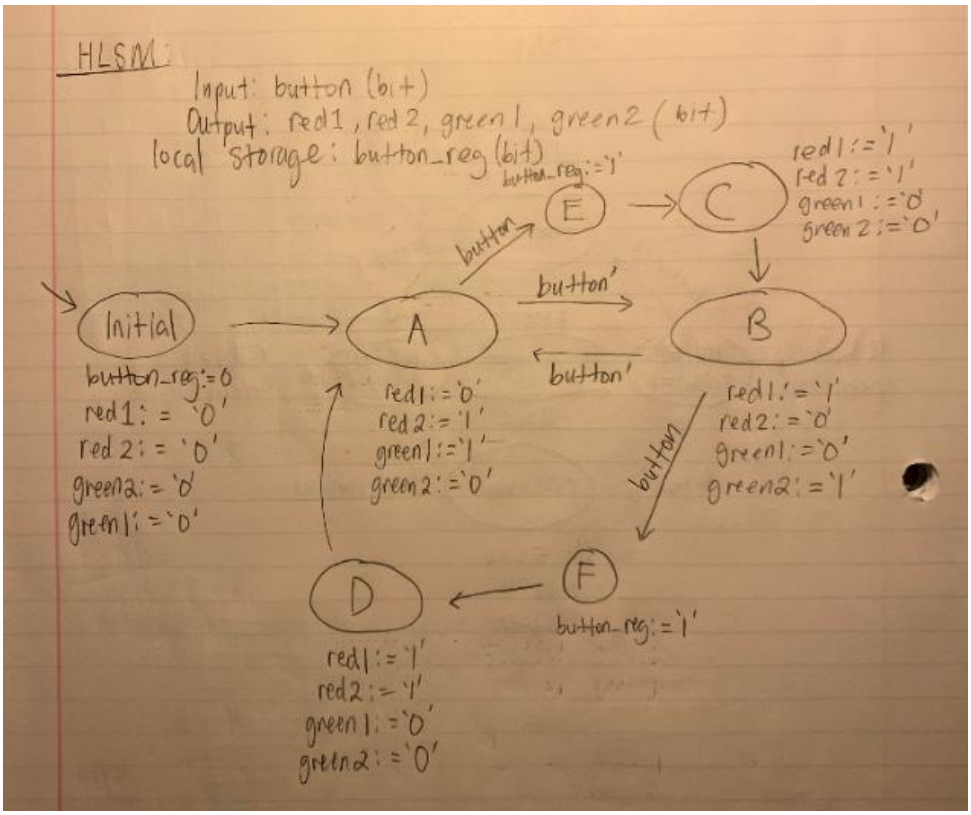
Course and section：EE 120A Section21

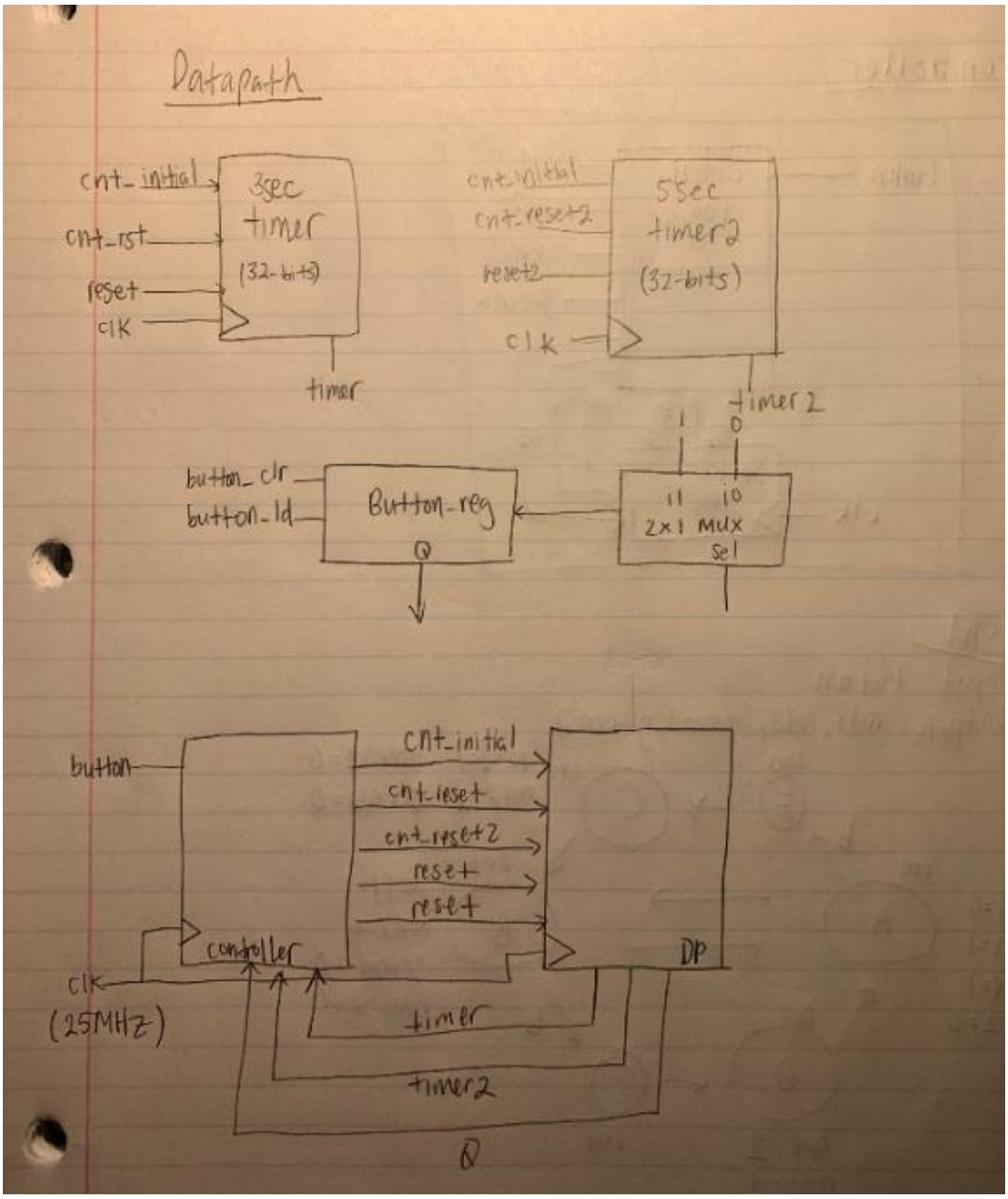
Partner: Yilun Liang

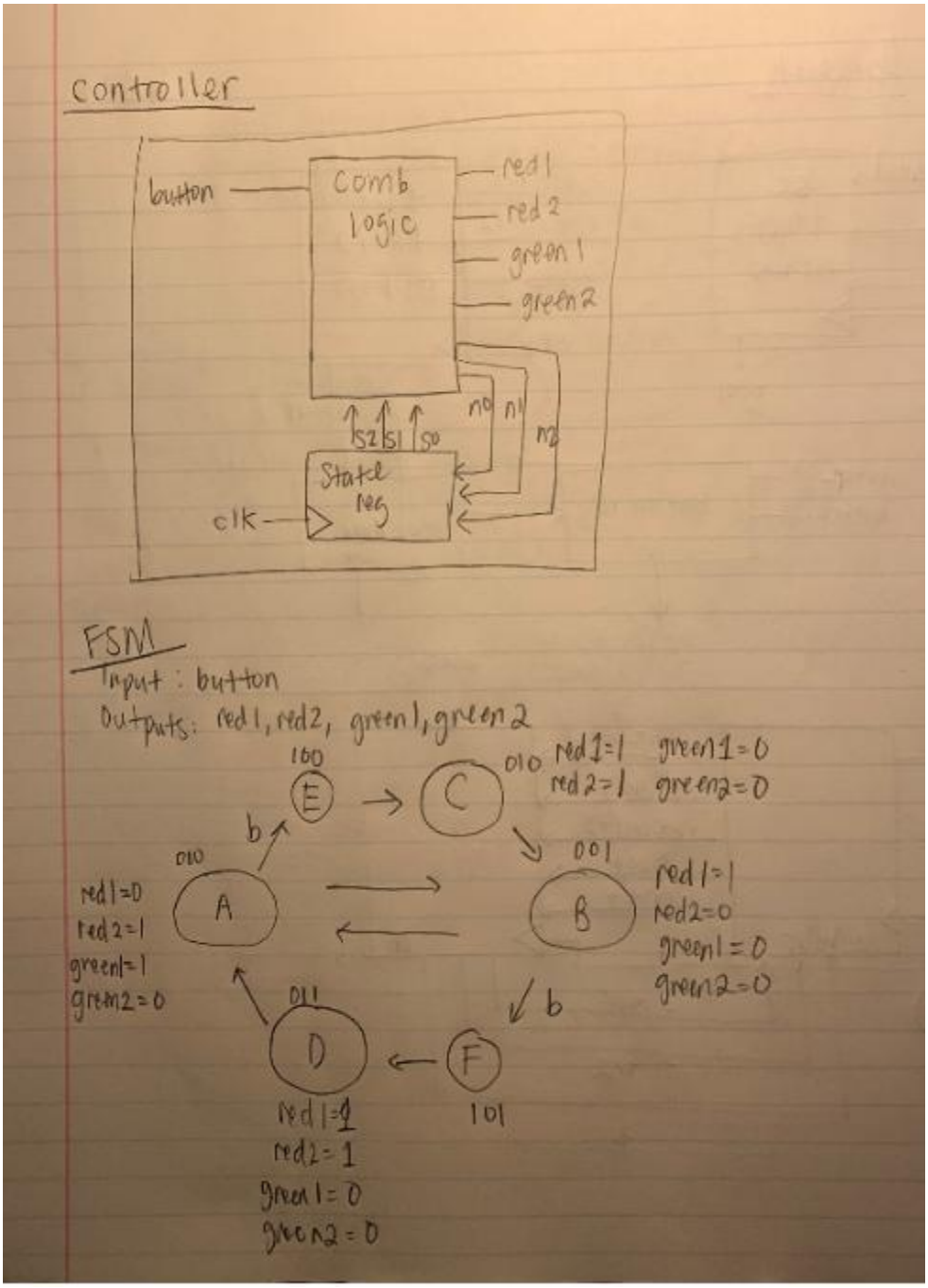
Purpose

The purpose is to make a four-way intersection traffic light using RTL design and Verilog. Some design constraint were that the traffic light only have two light for each way. The constraint is no yellow light and turn signals. The traffic lights is ON for 3 seconds. We need to also add a crosswalk button for pedestrians to cross the road. When you push the crosswalk button, all the traffic light must turn red after three seconds then it allows the pedestrians to cross. The crosswalk is ON for 5 seconds. We used 25MHz to calculate the ticks for the timer.

Circuit Design




Datapath
cnt_initial
cnt_rst
reset
clk
3sec timer
(32-bits)
timer
cnt_initial
cnt_reset2
reset2
clk
5sec timer2
(32-bits)
timer2
1
0
button_clr
button_ld
Button_reg
Q
I1
I0
2x1 MUX
Sel
button
cnt_initial
cnt_reset
cnt_reset2
reset
reset
controller
DP
clk
(25MHz)
timer
timer2
Q



https://youtu.be/ImMyQ0gpg90

```
module traffic#(
   parameter NBITS=32)
   (
   input clk,
   input button,
   input button2,
   output reg red1,
   output reg red2,
   output reg green1,
   output reg green2
    );

   wire timer;
   wire timer2;
   reg reset=1;
   reg reset2=1;
   wire [NBITS-1:0] cnt_ini;
   wire [NBITS-1:0] cnt_rst1;
   wire [NBITS-1:0] cnt_rst2;
   reg [2:0]current_state= 3'b000;
   reg [2:0]next_state=3'b000;
   reg button_reg = 1'b0;
   reg button2_reg = 1'b0;

always @(posedge clk) begin
current_state = next_state;
end
```

```
assign cnt_ini = 32'h0000;
assign cnt_rst1 = 32'h47868C0;
assign cnt_rst2 = 32'h7735940;

localparam A = 3'b000;
localparam B = 3'b001;
localparam C = 3'b010;
localparam D = 3'b011;
localparam E = 3'b100;
localparam F = 3'b101;
localparam G = 3'b110;
localparam H = 3'b111;

always @(posedge clk) begin
case (current_state)
A: begin
   reset= 0;
   red1 = 0;
   green1 = 1;
   red2 = 1;
   green2 =0;
   if (button == 1)
   begin
   button_reg = 1;
   reset=0;
   next_state = E;
   end
   else if (timer == 1)
   begin
```

```
    reset=1;
    next_state = B;
    end
end
    //else
    //next_state = A;
    //end

B: begin
    reset=0;
    red1 = 1;
    green1 = 0;
    red2 = 0;
    green2 = 1;
    if (button2 == 1)
    begin
    button2_reg = 1;
    reset=0;
    next_state = F;
    end
    else if (timer == 1)
    begin
    reset=1;
    next_state = A;
    end
end

C:begin
    reset2 = 0;
    red1 = 1;
    green1 = 0;
    red2 = 1;
    green2 = 0;
    if (timer2 == 1)
    begin
    reset = 1;
    reset2 = 1;
    next_state = B;
    end
    //else
    //begin
    //next_state=C;
    //end
end

D:begin
    reset2 = 0;
    red1 = 1;
    green1 = 0;
    red2 = 1;
    green2 = 0;
    if (timer2 ==1)
    begin
    reset = 1;
    reset2 = 1;
```

```
    next_state = A;
    end
end

E:begin
    reset = 0;
    red1 = 0;
    green1 = 1;
    red2 = 1;
    green2 = 0;
    if (timer == 1 && button_reg == 1)
    begin
    reset = 1;
    reset2 = 1;
    next_state = C;
    end
end

F:begin
    reset = 0;
    red1 = 1;
    green1 = 0;
    red2 = 0;
    green2 = 1;
    if (timer == 1 && button2_reg == 1)
    begin
    reset = 1;
    reset2 = 1;
    next_state = D;

    end
end

G:begin
    reset = 0;
    red1=1;
    green1=0;
    red2 = 0;
    green2=1;
    if (button == 1)
    begin
    button_reg = 1;
    reset=0;
    next_state = E;
    end
end

H: begin
    reset = 0;
    red1 = 0;
    green1 = 1;
    red2 = 1;
    green2 = 0;
    if (button2 == 1)
    begin
    button2_reg = 1;
    reset=0;
    next_state = F;
    end
```

```
end

   default: begin
   red1 = 0;
   green1 = 1;
   red2 =1;
   green2 =0;
   next_state = B ;
   end

   endcase
end

TIMER_ST #( .NBITS(NBITS) ) timerst (
.timer(timer),
.clk(clk),
.reset(reset) ,
.cnt_ini(cnt_ini),
.cnt_rst1(cnt_rst1)
);

TIMER_ST #( .NBITS(NBITS) ) timerst2 (
.timer(timer2),
.clk(clk),
.reset(reset2) ,
.cnt_ini(cnt_ini),
.cnt_rst1(cnt_rst2)
);
endmodule

module TIMER_ST #(
    parameter NBITS = 32
)
(
 output wire timer ,
 input wire clk ,
 input wire reset,
 input [NBITS-1:0] cnt_ini ,
 input [NBITS-1:0] cnt_rst1
);
wire [NBITS-1:0] q ;
wire [NBITS-1:0] qnext ;
// Compute the next value
adder #( .NBITS(NBITS ) )
 c1 (.q(q),
.cnt_ini(cnt_ini),
.cnt_rst1(cnt_rst1),
.nextq(qnext),
.tick(timer) );
// Save the next state
floprl #( .NBITS(NBITS ) )
 c2 (.clk(clk),
.reset(reset),
.cnt_ini(cnt_ini),
.nextq(qnext),
.q(q) );
endmodule
```

```
module adder#( parameter NBITS = 16 )(
input [NBITS-2:0] q ,
input [NBITS-2:0] cnt_ini ,
input [NBITS-2:0] cnt_rst1 ,
output[NBITS-2:0] nextq,
output tick
);
wire same ;
wire[NBITS-2:0] inextq;

addergen_st #(.NBITS(NBITS))
nextval ( .r(inextq), // Next value
 .cout(), // Carry out - Don't use
 .a(q), // Current value
 .b(16'b0000_0001), // Plus One
 .cin(16'b0000_0000) ) ; // No carry in

comparatorgen_st #(.NBITS(NBITS))
comparator (
.r(same) ,
.a(inextq),
.b(cnt_rst1) );

assign tick = (same) ? 'd1 : 'd0 ;
assign nextq = (same) ? cnt_ini : inextq ;
endmodule
```

```
module addergen_st #( parameter NBITS = 16 )(
output wire[NBITS-1:0] r ,
output wire cout ,
input wire[NBITS-1:0] a ,
input wire[NBITS-1:0] b ,
input wire cin ) ;
wire [NBITS:0] carry;
assign carry[0]= cin ;
genvar k ;
generate
for (k=0; k < NBITS; k = k+1)
begin : blk
 fulladder_st FA (
 .r(r[k]),
 .cout(carry[k+1]),
 .a(a[k]),
 .b(b[k]),
 .cin(carry[k]) ) ;
end
endgenerate
assign cout = carry[NBITS] ;
endmodule
```

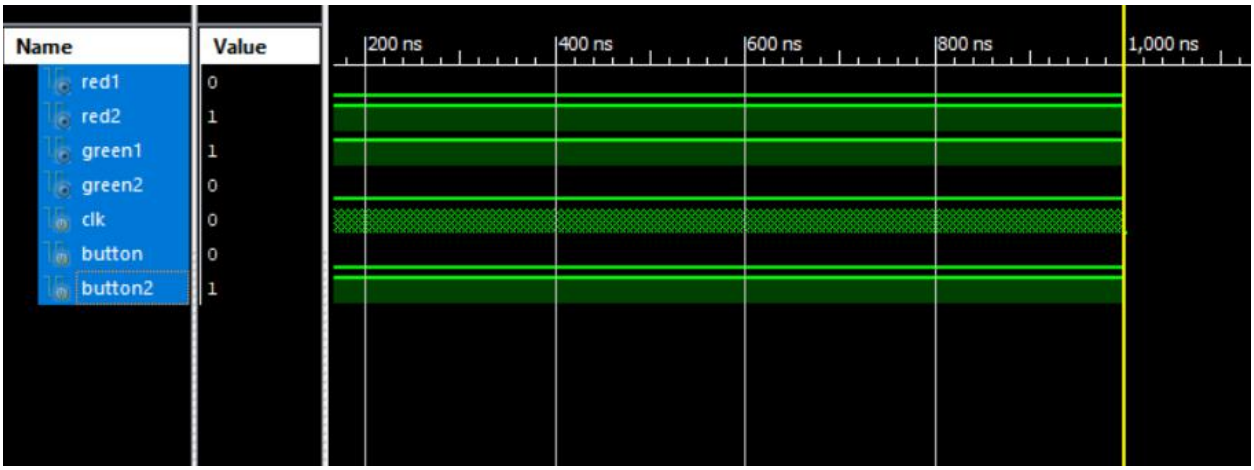
```
module fulladder_st(
output wire r,
output wire cout,
input wire a,
input wire b,
input wire cin
) ;
assign r = (a ^ b) ^ (cin) ;
assign cout = (a & b) | ( a & cin ) | ( b & cin ) ;
endmodule


module comparatorgen_st #( parameter NBITS = 16 )(
output wire r ,
input wire[NBITS-1:0] a ,
input wire[NBITS-1:0] b );
wire [NBITS-1:0] iresult ;
genvar k ;
generate
for (k=0; k < NBITS; k = k+1)
begin : blk
 xor cl (iresult[k], a[k], b[k] ) ;
end
endgenerate
// Reduction plus negation
assign r = ~(|iresult);
endmodule

module floprl#( parameter NBITS = 16 )(
 input clk,
input reset,
input [NBITS-2:0] cnt_ini,
input [NBITS-2:0] nextq,
output[NBITS-2:0] q
);
reg [NBITS-2:0] iq=0 ;
always @(posedge clk) begin
 if (reset) begin
iq <= cnt_ini ;
 end
 else begin
iq <= nextq;
 end
end
assign q = iq ;
endmodule
```

UCF:

```
// Inputs
NET "clk" LOC = "B8" ;
NET "button" LOC = "G12" ;
NET "button2" LOC = "M4";
// Outputs
NET "red1" LOC = "G1" ;
NET "green1" LOC = "P4" ;
NET "red2" LOC = "N4" ;
NET "green2" LOC = "N5" ;
```

Testbench:

```
module testbench;

   // Inputs
   reg clk;
   reg button;
   reg button2;

   // Outputs
   wire red1;
   wire red2;
   wire green1;
   wire green2;

   // Instantiate the Unit Under Test (UUT)
   traffic uut (
      .clk(clk),
      .button(button),
      .button2(button2),
      .red1(red1),
      .red2(red2),
      .green1(green1),
      .green2(green2)
   );

      always begin
      #2 clk = ~clk;
      end
   initial begin
      clk = 0;

       // Initialize Inputs
       button = 1;
       #3;

       button2 = 1;
       button = 0;
       #3;
    end

endmodule
```



Problems and Technical Issues

Problems and technical issues encountered were we try to add something where if someone or a pedestrian continues to push the crosswalk button, it will stay in that state. We couldn't get it to work, so we decided to scrap it. Another problem was that we wanted to add the extra button where the traffic lights become red whichever buttons is pressed, but instead we added the extra crosswalk button for their specific traffic light. We had a problem with the waveform from the test bench that looks incorrect because of the timer module.

Conclusion

Our project worked according to our specifications we designed our traffic light system. We were able to design the traffic light using our HLSM design and controller. The HLSM had six states and the data path design had two timer, one for the crosswalk and the traffic light. Some ways to improve the system is to add maybe yellow light and traffic light for the four way and not just two traffic lights. The most difficult part of the project is the coding.

TA problem

How many timers do we use in the project?

Ans:We use two timers.One is for timing 3s,the other one is for timing 5s.