University of Tehran

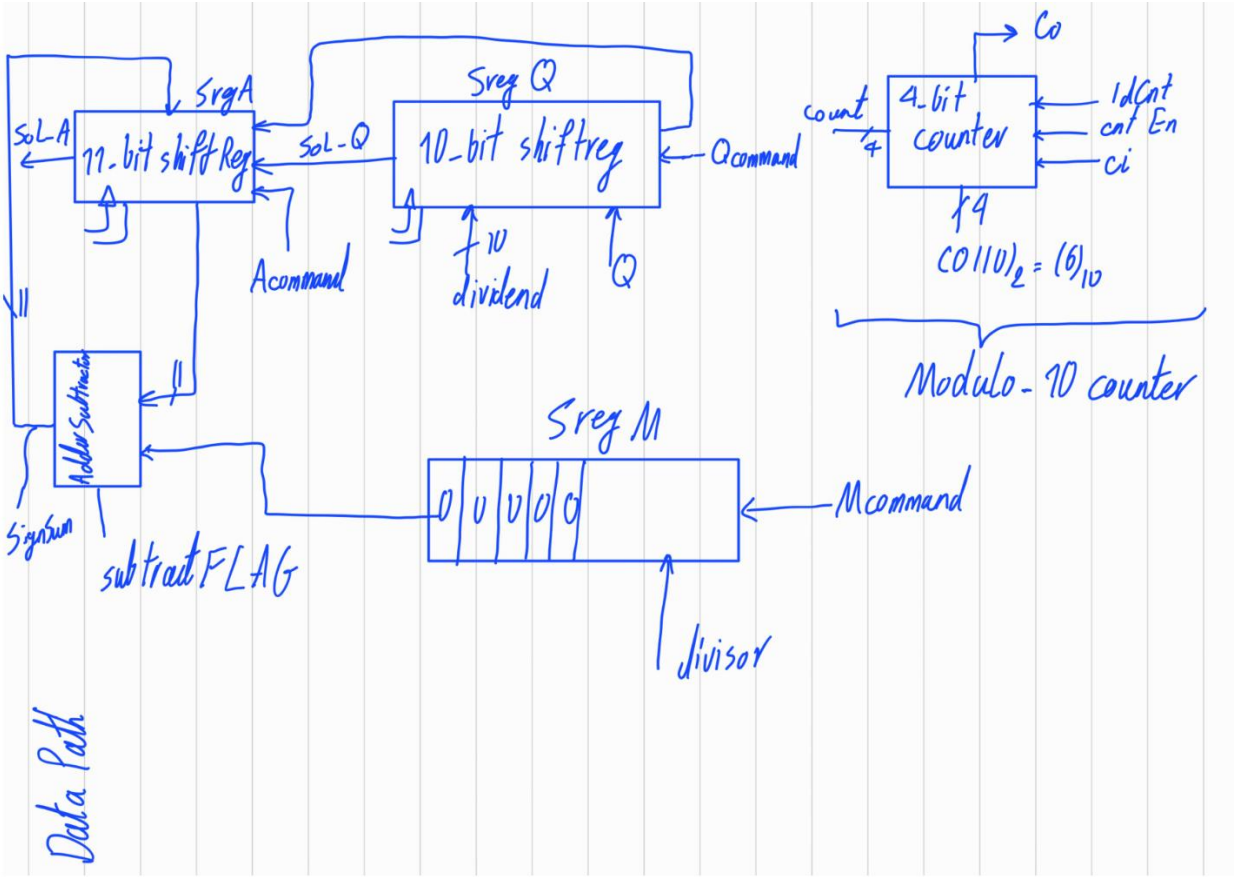
Electrical and Computer Engineering Department

Computer Architecture

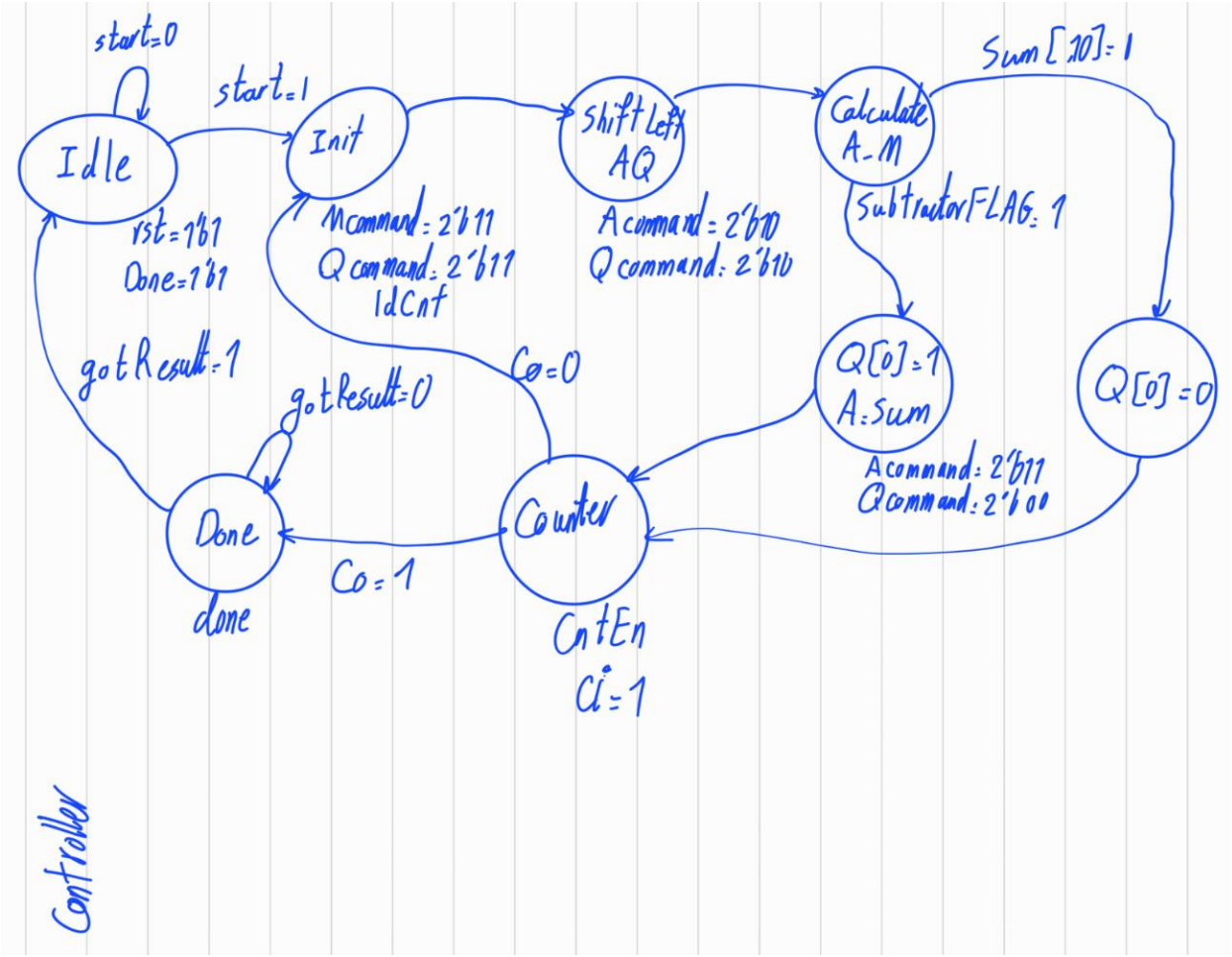
Computer Assignment #1

Name	Danial Saeedi Mohammad Ghare Hasanloo
Student No	810198571 810198461
Date	November 07, 2021

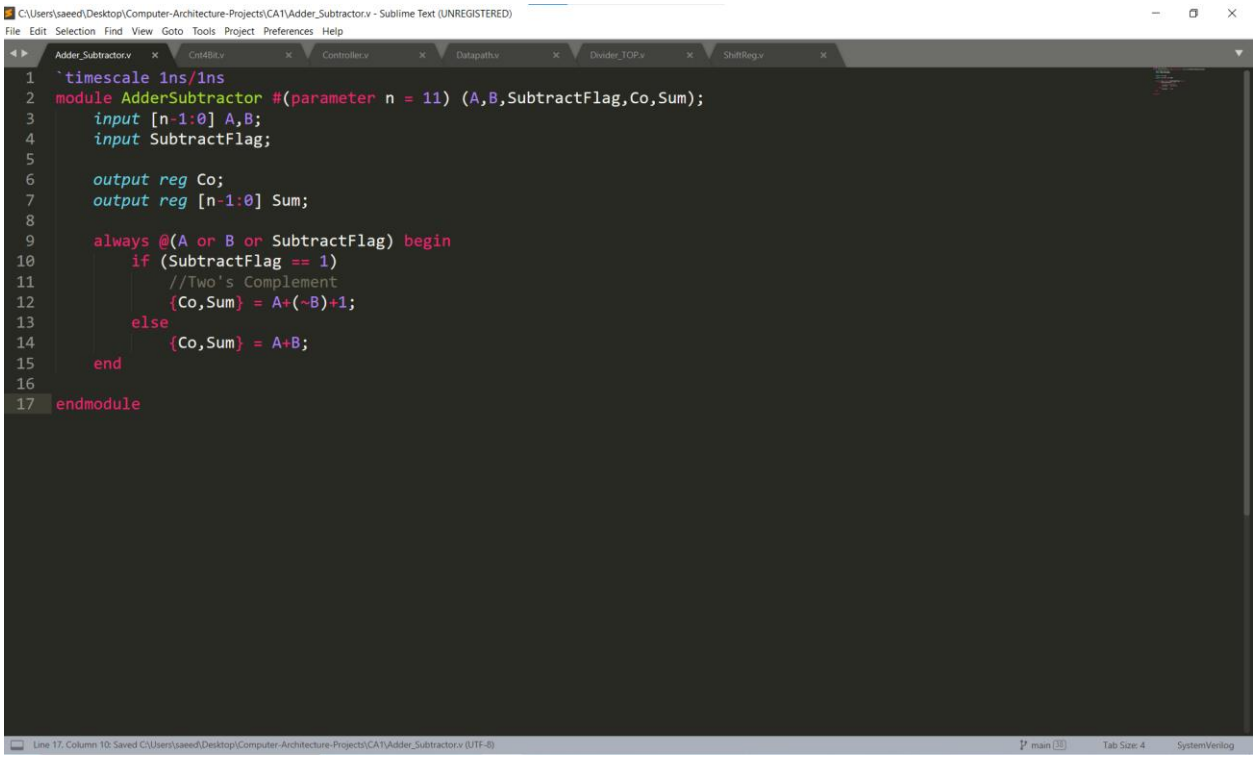
1. Datapath


SregA
Sreg Q
SoL-A
11-bit shift Reg
SoL-Q
10-bit shift reg
Qcommand
Co
count
4-bit counter
ldCnt
cnt En
ci
4
Acommand
10
dividend
Q
$(0110)_2 = (6)_{10}$
Modulo-10 counter
Adder Subtractor
Sreg M
0 0 0 0 0
Mcommand
SignSum
subtract FLAG
divisor
Data Path

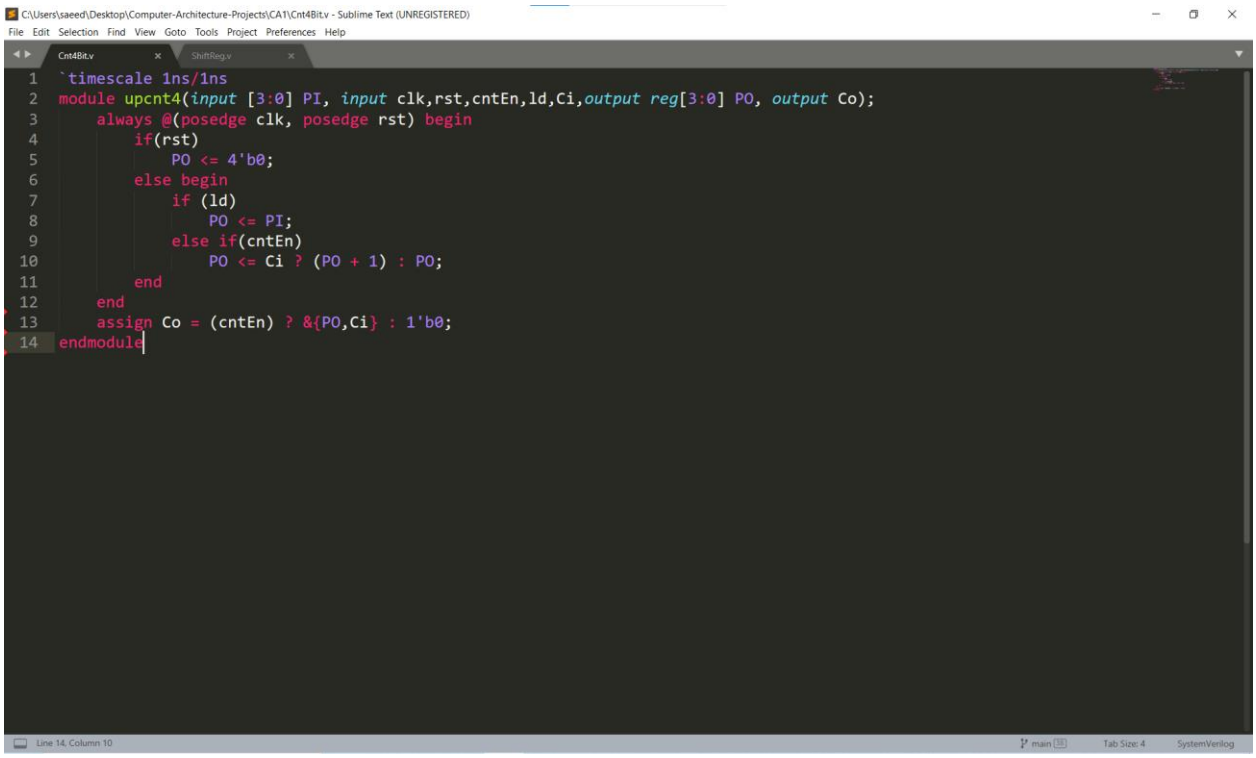
2. Control Unit



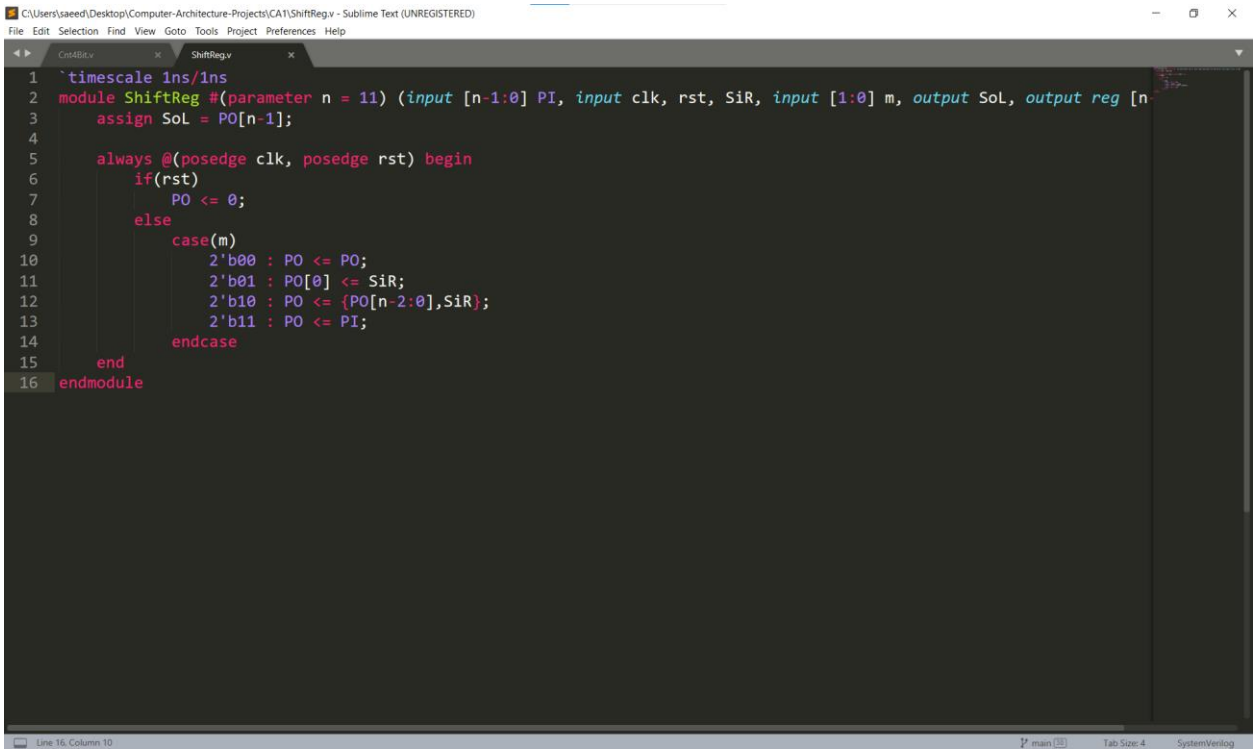
Adder / Subtractor Module



Modulo-10 Counter

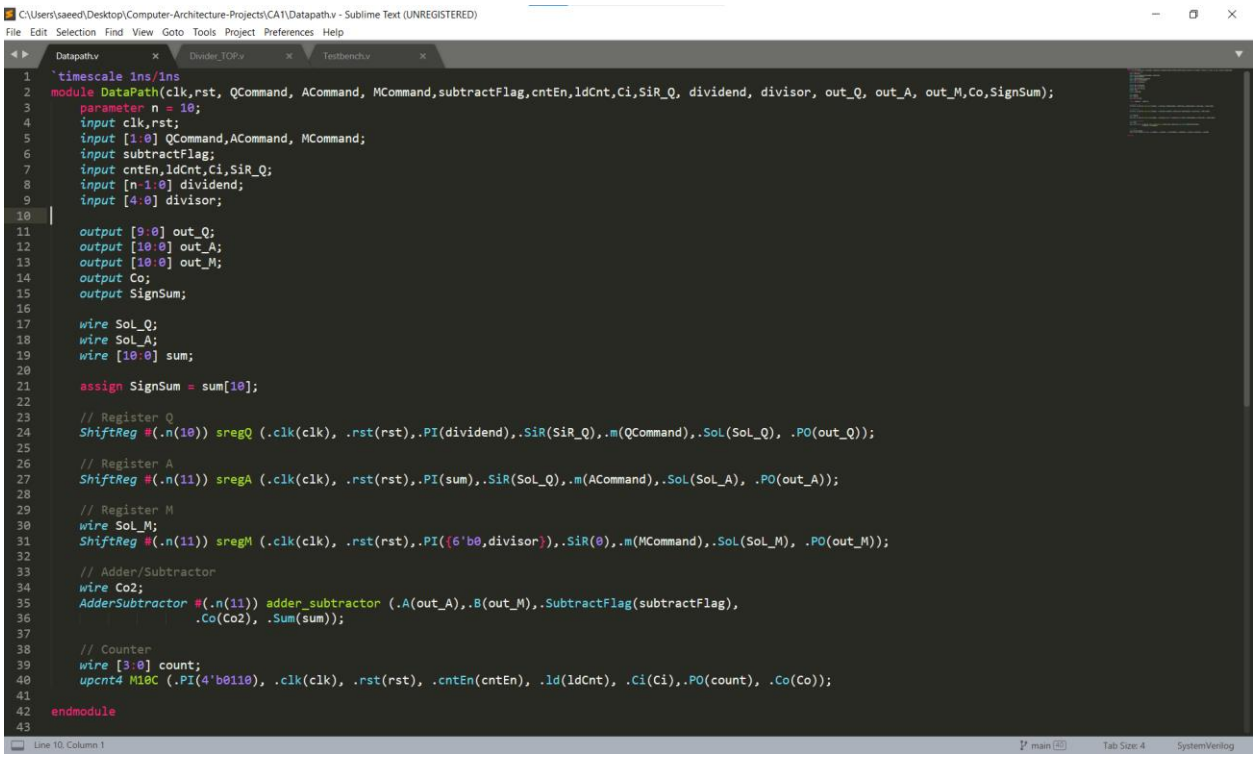


Universal Shift Register

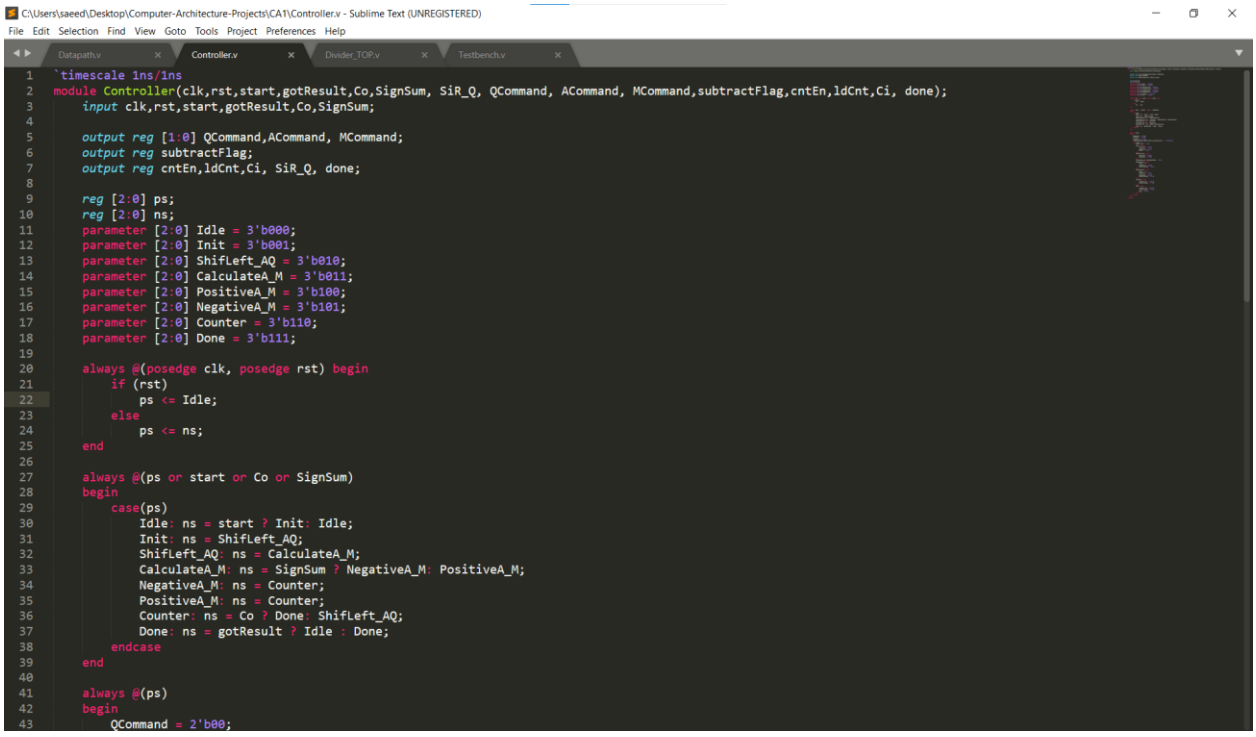


Data Path Module

Quotient is stored at Q and the remainder is stored at A.



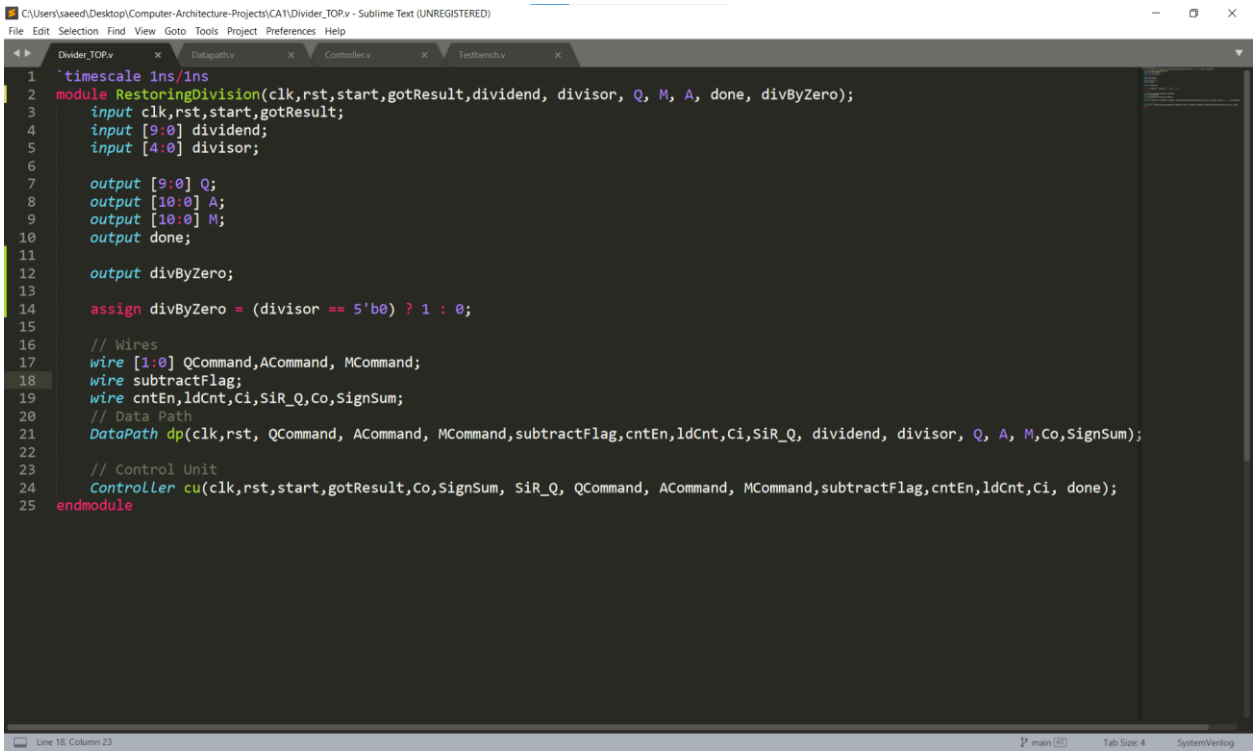
Control Unit Module




```
`timescale 1ns/1ns
module Controller(clk,rst,start,gotResult,Co,SignSum, SiR_Q, QCommand, ACommand, MCommand,subtractFlag,cntEn,ldCnt,Ci, done);
    input clk,rst,start,gotResult,Co,SignSum;

    output reg [1:0] QCommand,ACommand, MCommand;
    output reg subtractFlag;
    output reg cntEn,ldCnt,Ci, SiR_Q, done;

    reg [2:0] ps;
    reg [2:0] ns;
    parameter [2:0] Idle = 3'b000;
    parameter [2:0] Init = 3'b001;
    parameter [2:0] ShifLeft_AQ = 3'b010;
    parameter [2:0] CalculateA_M = 3'b011;
    parameter [2:0] PositiveA_M = 3'b100;
    parameter [2:0] NegativeA_M = 3'b101;
    parameter [2:0] Counter = 3'b110;
    parameter [2:0] Done = 3'b111;

    always @(posedge clk, posedge rst) begin
        if (rst)
            ps <= Idle;
        else
            ps <= ns;
    end

    always @(ps or start or Co or SignSum)
    begin
        case(ps)
            Idle: ns = start ? Init: Idle;
            Init: ns = ShifLeft_AQ;
            ShifLeft_AQ: ns = CalculateA_M;
            CalculateA_M: ns = SignSum ? NegativeA_M: PositiveA_M;
            NegativeA_M: ns = Counter;
            PositiveA_M: ns = Counter;
            Counter: ns = Co ? Done: ShifLeft_AQ;
            Done: ns = gotResult ? Idle : Done;
        endcase
    end

    always @(ps)
    begin
        QCommand = 2'b00;
```

```
    always @(ps)
    begin
        QCommand = 2'b00;
        ACommand = 2'b00;
        MCommand = 2'b00;
        {subtractFlag,cntEn,ldCnt,Ci,SiR_Q,done} = 6'b000000;
        case (ps)
            Idle: done = 1'b1;
            Init: begin
                MCommand = 2'b11;
                QCommand = 2'b11;
                ldCnt = 1'b1;
            end
            ShifLeft_AQ: begin
                QCommand = 2'b10;
                ACommand = 2'b10;
            end
            CalculateA_M: subtractFlag = 1'b1;
            NegativeA_M: begin
                SiR_Q = 0;
                QCommand = 2'b01;
                subtractFlag = 1'b1;
            end
            PositiveA_M: begin
                SiR_Q = 1;
                QCommand = 2'b01;
                ACommand = 2'b11;
                subtractFlag = 1'b1;
            end
            Counter: begin
                {cntEn, Ci} = 2'b11;
                subtractFlag = 1'b1;
            end
            Done: begin
                {cntEn, Ci} = 2'b11;
                subtractFlag = 1'b1;
                done = 1'b1;
            end
        endcase
    end
endmodule
```

Top Module


C:\Users\saeed\Desktop\Computer-Architecture-Projects\CA1\Divider_TOP.v - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Divider_TOP.v
Datapath.v
Controller.v
Testbench.v
`timescale 1ns/1ns
module RestoringDivision(clk,rst,start,gotResult,dividend, divisor, Q, M, A, done, divByZero);
input clk,rst,start,gotResult;
input [9:0] dividend;
input [4:0] divisor;
output [9:0] Q;
output [10:0] A;
output [10:0] M;
output done;
output divByZero;
assign divByZero = (divisor == 5'b0) ? 1 : 0;
// Wires
wire [1:0] QCommand,ACommand, MCommand;
wire subtractFlag;
wire cntEn,ldCnt,Ci,SiR_Q,Co,SignSum;
// Data Path
DataPath dp(clk,rst, QCommand, ACommand, MCommand,subtractFlag,cntEn,ldCnt,Ci,SiR_Q, dividend, divisor, Q, A, M,Co,SignSum);
// Control Unit
Controller cu(clk,rst,start,gotResult,Co,SignSum, SiR_Q, QCommand, ACommand, MCommand,subtractFlag,cntEn,ldCnt,Ci, done);
endmodule
Line 18, Column 23
main
Tab Size: 4
SystemVerilog

Testbench

```
`timescale 1ns/1ns
module Testbench();
    parameter n = 10;
    reg [9:0] dividend;
    reg [4:0] divisor;

    reg rst, start,gotResult, clk;

    wire [9:0] Q;
    wire [10:0] A;
    wire [10:0] M;
    wire done;

    RestoringDivision RD(clk,rst,start,gotResult,dividend, divisor, Q, M, A, done);

    initial begin
        start = 1'b0;
        rst = 1'b0;
        clk = 1'b0;
        gotResult = 1'b0;
        #13 rst = 1'b1;
        #20 rst = 1'b0;
        #13 start = 1'b1;dividend = 10'b1001000011; divisor = 5'b01111;
        #1000 start = 1'b0; gotResult = 1'b1;
        #200 rst = 1'b1; gotResult = 1'b0;
        #20 rst = 1'b0; start = 1'b1; dividend = 10'b0001000011; divisor = 5'b00101;
        #1000 $stop;
    end

    always
    begin
    #10 clk = ~clk;
    end
```

Figure 1 Testbench.v