	به نام خدا	

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

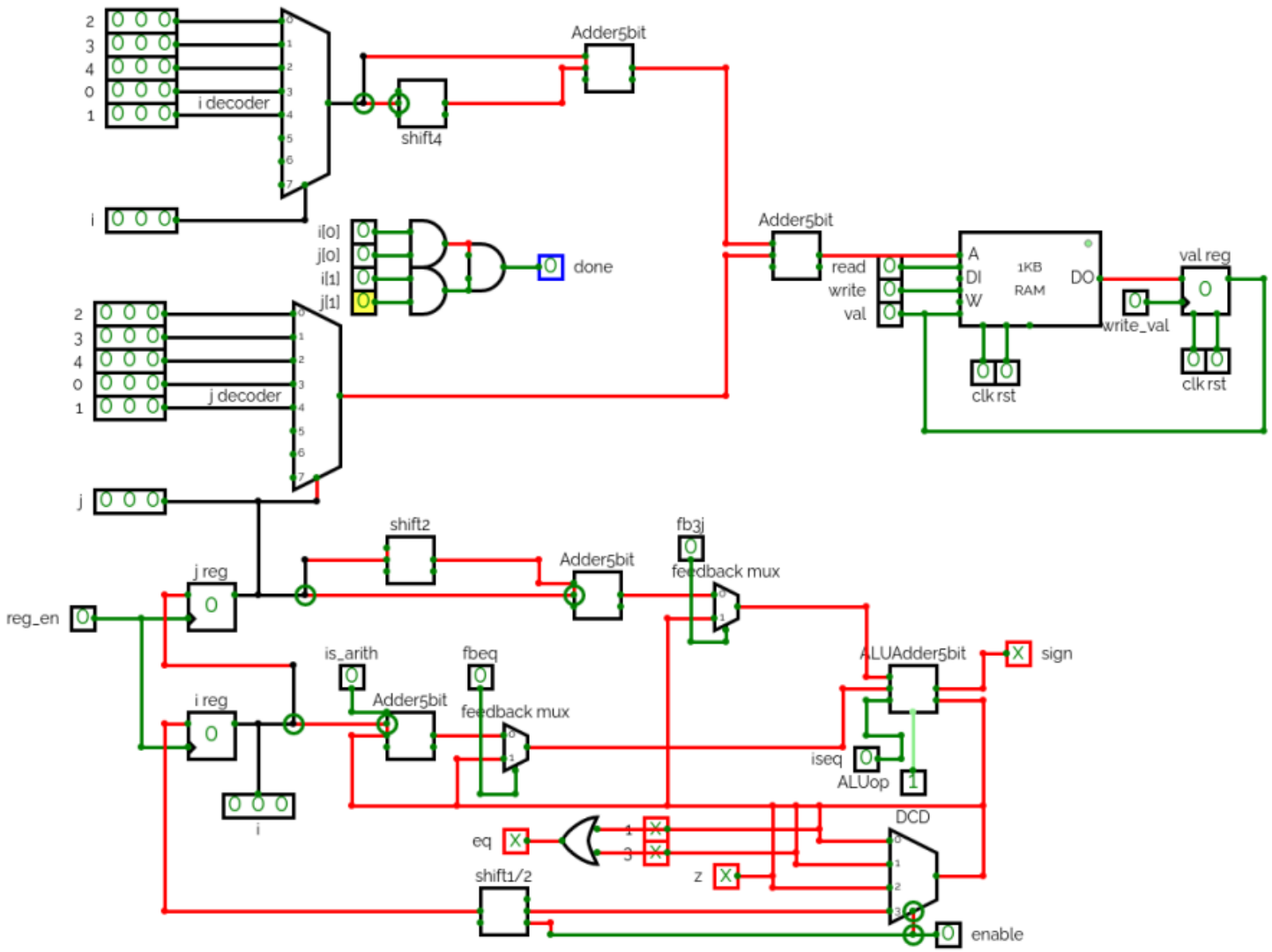
طراحی کامپیوتری سیستم دیجیتال

تمرین دستی1فاز2

نام و نام خانوادگی	آرین سلطانی-ریحانه احمدپور
شماره دانشجویی	810198494-810198558
تاریخ ارسال گزارش	1401/1/18 - پنجشنبه 18 فروردین

فهرست گزارش سؤالات

دیتاپث:



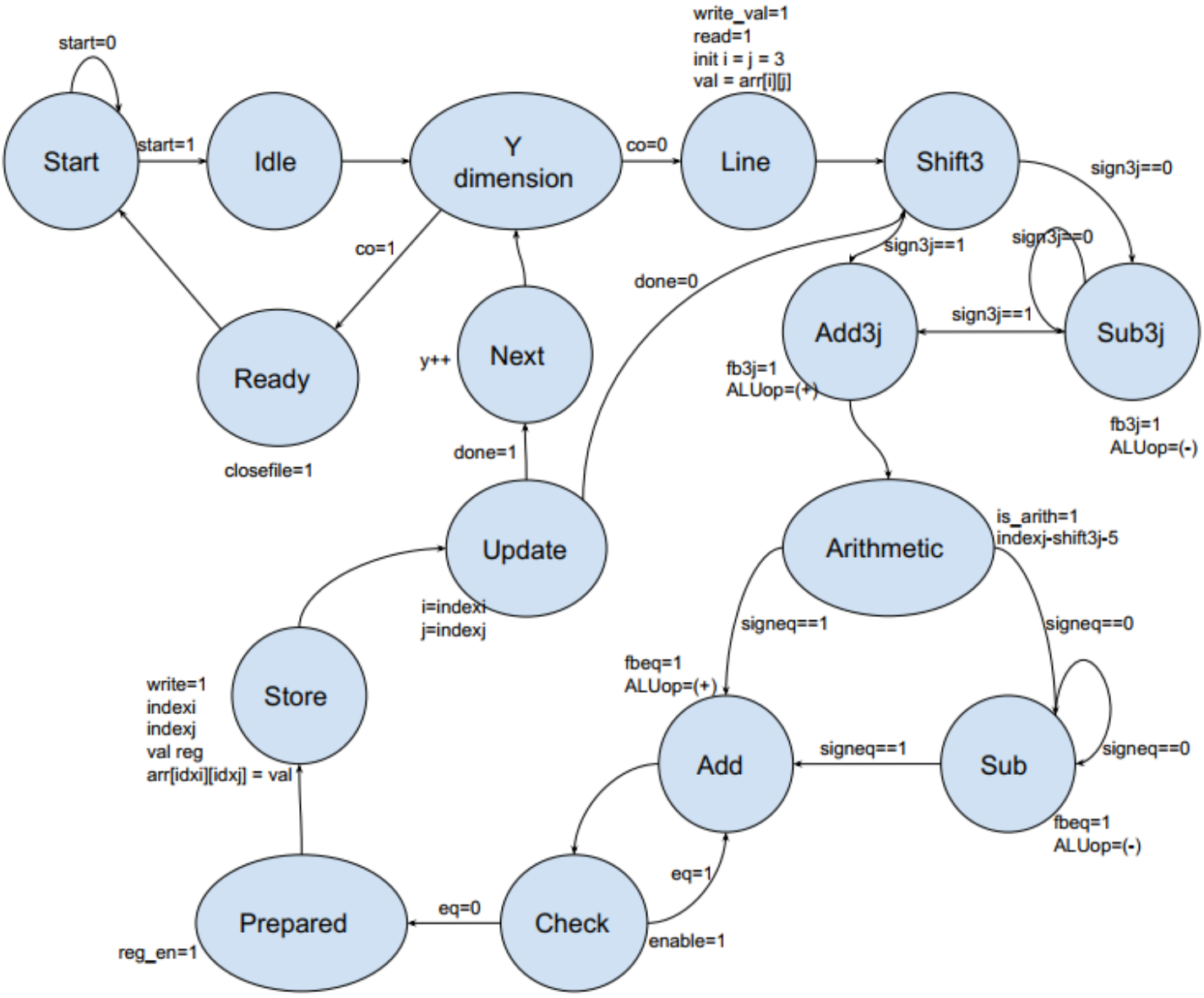

```
`timescale 1ns/1ns
module Datapath (
    clk,
    rst,
    IJen,
    ALUop,
    read,
    write,
    initLine,
    line,
    writeVal,
    IJregen,
    fb3j,
    fbeq,
    isArith,
    enable,
    waitCalNexti,
```

```
    writeMemReg,
    ldTillPositive,
    update,//enable for updating i,j after being checked and update "i"

    sign3j,
    signeq,
    done,
    sign,
    eq,
    mem,
    firstread,
    ok
);
    parameter size = 5;
    parameter memsize = 25;

    parameter initValIJ = 3;

    input clk, rst, firstread;
    input IJen, ALUop, read, write, initLine, writeMemReg;
    input writeVal, IJregen, fbeq, fb3j, isArith, enable, update, waitCalNexti,
ldTillPositive, ok;
    input [memsize-1:0]line;
    output [24:0]mem;

    output sign3j, signeq, done, sign, eq;

    wire [2:0]i;
    wire [2:0]j;
    wire [4:0]iMult4;
    wire [2:0]iReg;
    wire [2:0]jReg;
    wire [4:0]iMult5;
    wire [4:0]memIdx;
    wire [4:0]iMult2;
    wire [4:0]iMult3;
    wire [4:0]lastIndex;
    wire [4:0]iNextMult2;
    wire [4:0]iNextPos;
    wire [4:0]iNextPosAdd5;
    wire [2:0]iAtLast;
    wire [2:0]convertedI;
    wire [2:0]convertedJ;
    wire [4:0]memIdxOut;
    wire [4:0]memInp;


    wire newVal;
```

```
    wire regVal;

    IJMux newI(jReg, convertedI);
    IJMux newJ(iReg, convertedJ);

    Shifter #(5) multiplyI4(.data({2'b00, convertedI}), .coefficient({2'b01}),
.shifted(iMult4));

    Adder #(5) multiplyI5(.i1({2'b00, convertedI}), .i2(iMult4), .a(iMult5));

    Adder #(5) indexAdder(.i1(iMult5), .i2({2'b00, convertedJ}), .a(memIdx));

    Register #(5) indexMemReg(.clk(clk), .rst(rst), .ld(writeMemReg), .inputData(memIdx),
        .outputData(memIdxOut));

    assign memInp = write ? memIdxOut: memIdx;

    MemoryBlock #(5,25) MB(.clk(clk), .rst(rst), .init(initLine), .line(line),
        .index(memInp), .val(regVal), .write(write), .read(read), .out(newVal), .mem(mem),
.firstread(firstread), .ok(ok));

    Register #(1) valRegister(.clk(clk), .rst(rst), .ld(writeVal), .inputData(newVal),
        .outputData(regVal));

    Register #(3) JRegister(.clk(clk), .rst(rst), .ld(IJregen),
        .inputData(j), .outputData(jReg));

    Shifter #(5) multiplyI2(.data({2'b00, iReg}), .coefficient({2'b00}), .shifted(iMult2));

    Adder #(5) multiplyI3(.i1(iMult2), .i2({2'b00, iReg}), .a(iMult3));

    Register  #(5) regTillPositive(.clk(clk), .rst(rst), .ld(ldTillPositive),
        .inputData(iNextPosAdd5), .outputData(iNextPos));

    assign sign = iNextPosAdd5[4];

    assign iNextPosAdd5 = (waitCalNexti) ? (iNextPos + 5'b00101): iNextMult2;

    Register #(5) registerLastIndex(.clk(clk), .rst(rst), .ld(ld_index), .inputData(memIdx),
.outputData(lastIndex));

    Register #(3) IRegister(.clk(clk), .rst(rst), .ld(IJregen),
        .inputData(i), .outputData(iReg));

    assign iAtLast = (iNextPos == 5'b00000) ? 3'b000: (iNextPos == 5'b00001) ?
        3'b011: (iNextPos == 5'b00010) ? 3'b001: (iNextPos == 5'b00011) ? 3'b100:
        3'b010;
```

```
    Subtractor #(5) twiceNextI(.i1({2'b00, jReg}), .i2(iMult3), .en(isArith),
.out(iNextMult2));

    assign i = IJen ? 3'b011: update ? iAtLast : i;
    assign j = IJen ? 3'b011: update ? iReg : j;

    assign done = iReg[0] & iReg[1] & jReg[0] & jReg[1];

endmodule
```

کنترلر:




```
`timescale 1ns/1ns
module Controller (
```

```
    clk,
    rst,
    start,

    sign3j,
    signeq,
    done,
    sign,
    eq,

    waitCalNexti,
    writeMemReg,

    IJen,
    ALUop,
    read,
    write,
    initLine,
    line,
    writeVal,
    IJregen,
    fb3j,
    fbeq,
    isArith,
    enable,
    update,
    readLine,
    ldTillPositive,
    count,
    firstread,
    ok
);
    parameter size = 5;
    parameter memsize = 25;

    input clk, rst;
    input start, sign3j, signeq, done, sign, eq;
    reg [5:0] count2;
    input [5:0]count;

    output reg IJen, ALUop, read, write, initLine, firstread;
    output reg writeVal, IJregen, fbeq, fb3j, isArith, enable, update, waitCalNexti,
writeMemReg, ldTillPositive;
    input [memsize-1:0]line;
    output reg readLine, ok;

    parameter [3:0]
        Start = 4'd0,
```

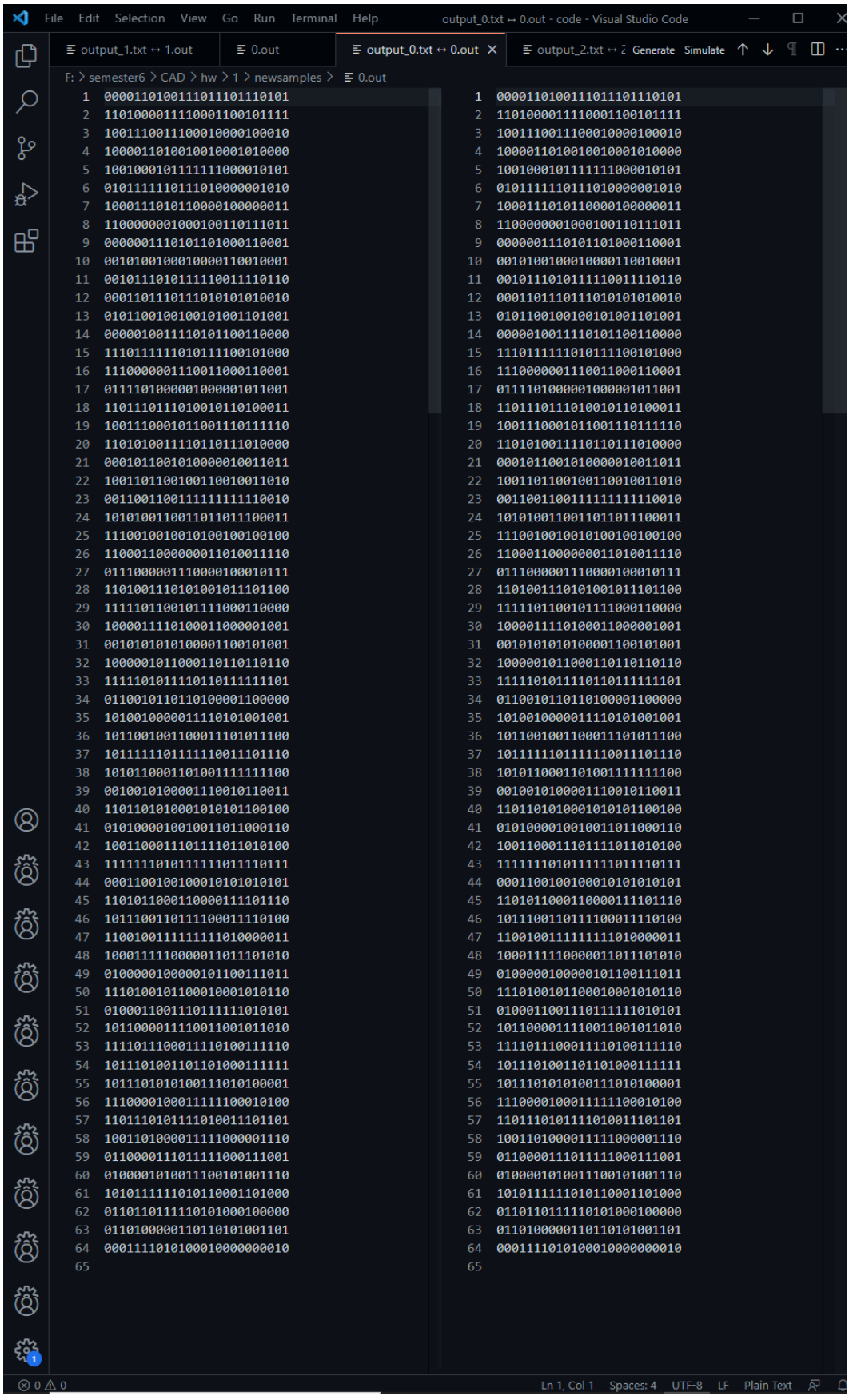
```
        Idle = 4'd1,
        Ydimension = 4'd2,
        Line = 4'd3,
        Shift3 = 4'd4,
        Sub3j = 4'd5,
        Add3j = 4'd6,
        Arithmetic = 4'd7,
        Sub = 4'd8,
        Add = 4'd9,
        Check = 4'd10,
        Prepared = 4'd11,
        Store = 4'd12,
        Updater = 4'd13,
        Next = 4'd14,
        Ready = 4'd15;

    reg enCount=0, loadCount=0, first = 0;
    reg [5:0]loadInit = 0; // n times count = 5
    wire coutCount;


    reg [3:0] ps, ns;

    Counter #6 cc(.clk(clk), .rst(rst), .en(enCount), .ld(loadCount), .initld(loadInit),
.co(coutCount));

    always @(posedge clk, posedge rst) begin
        if(rst)begin
            ps <= Start;
            count2 <= 6'b000000;
        end
        else
            ps <= ns;
    end

    wire [5:0] sig;
    wire tmp;
    assign sig = ~(count + ~count2 + 6'b000001);
    assign tmp = sig[5] & sig[4] & sig[3] & sig[2] & sig[1] & sig[0];

    always @(ps, start, sign3j, signeq, done, sign, eq, tmp) begin
        case (ps)
            Start:      ns = start ? Idle : Start;
            Idle:       ns = Ydimension;
            Ydimension: ns = coutCount ? Ready : Line;
            Line:       ns = Store;
            Store:      ns = Shift3;
            Shift3:     ns = Sub3j;
```

```
            Sub3j:      ns = ~sign ? Add3j : Sub3j;
            Add3j:      ns = Arithmetic;
            Arithmetic: ns = Sub;
            Sub:        ns = Add;
            Add:        ns = Check;
            Check:      ns = Updater;
            Prepared:   ns = Next;
            Updater:    ns = ~done ? Shift3 : Prepared;
            Next:       ns = (~tmp) ? Next: Ydimension;
            Ready:      ns = Start;
            default: ns = Start;
        endcase
    end

    always @(ps) begin
        {ok, firstread, ldTillPositive, writeMemReg, first, waitCalNexti, IJen, ALUop, read,
write, initLine, writeVal, IJregen, fbeq, fb3j, isArith, enable, update, readLine, loadCount,
enCount} = 0;
        case (ps)
            Start:      begin
                //nothing
            end
            Idle:       begin
                loadCount = 1'd1;
            end
            Ydimension: begin
                //nothing
                readLine = 1'b1;
            end
            Line:       begin
                IJen = 1'b1;
                initLine = 1'b1;
                IJregen = 1'b1;
                count2 = count2 + 1;
            end
            Store:     begin
                writeVal = 1'b1;
                firstread = 1'b1;
            end
            Shift3:     begin
                writeMemReg = 1'b1;
                ldTillPositive =1 'b1;
            end
            Sub3j:      begin
                waitCalNexti = 1'b1;
                ldTillPositive = sign;
                update = 1'b1;
            end
```

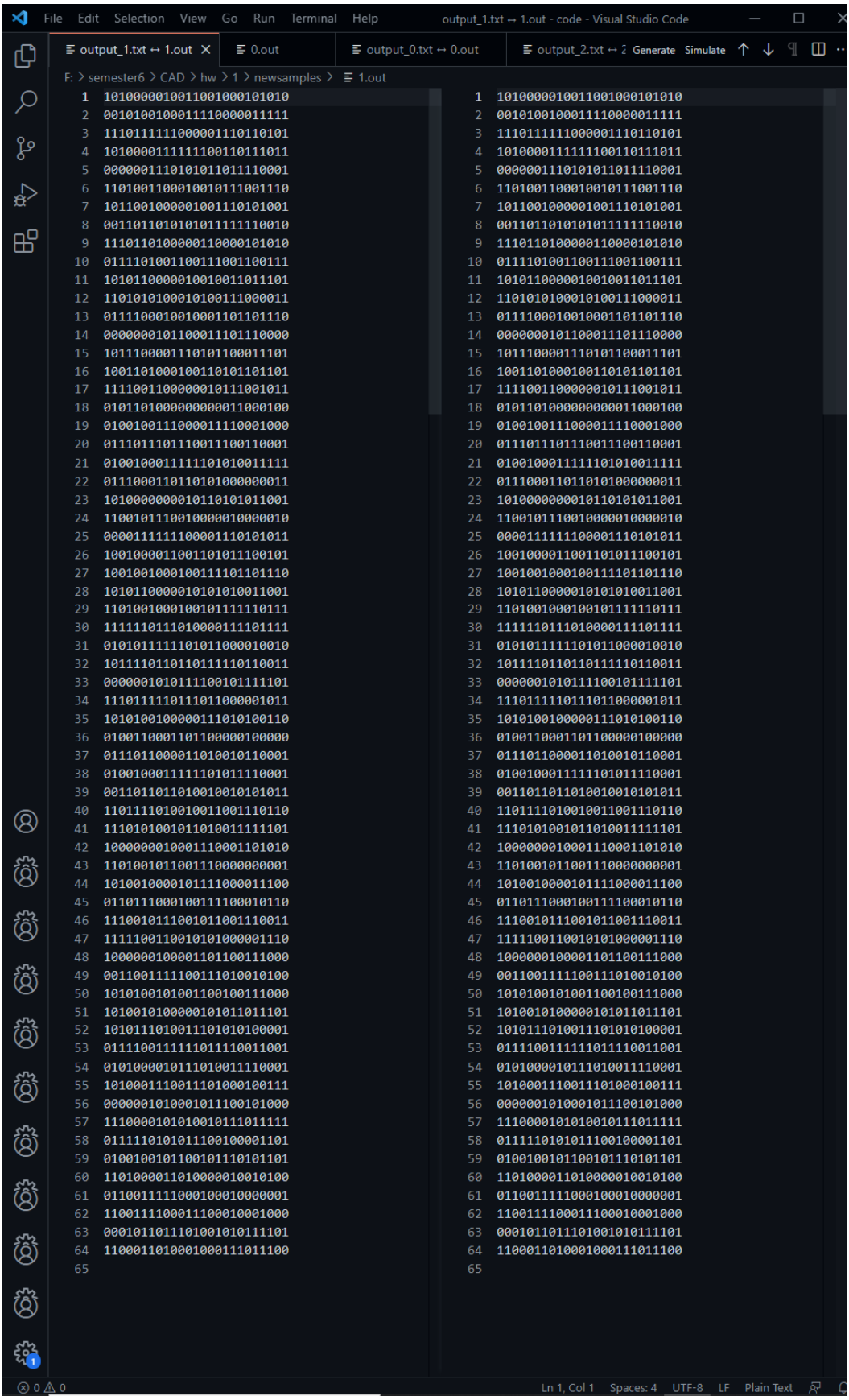
```
            Add3j:      begin
                IJregen = 1'b1;

            end
            Arithmetic: begin
                isArith = 1'b1;
            end
            Sub:        begin
                read = 1'b1;
                fbeq = 1'b1;
                ALUop =  1'b0;
                ok = 1'b1;
            end
            Add:        begin
                fbeq = 1'b1;
                ALUop =  1'b1;
                write = 1'b1;
                ok = 1'b1;
            end
            Check:      begin
                enable = 1'b1;
                ok = 1'b1;
            end
            Prepared:   begin
                ok = 1'b1;
                enCount = 1'b1;
            end
            Store:      begin // fix algo
            end
            Updater:    begin

            end
            Next:       begin
                ok = 1'b1;
            end
            Ready:     begin
                //nothing
            end
        endcase
    end


endmodule
```

نتایج خروجی:

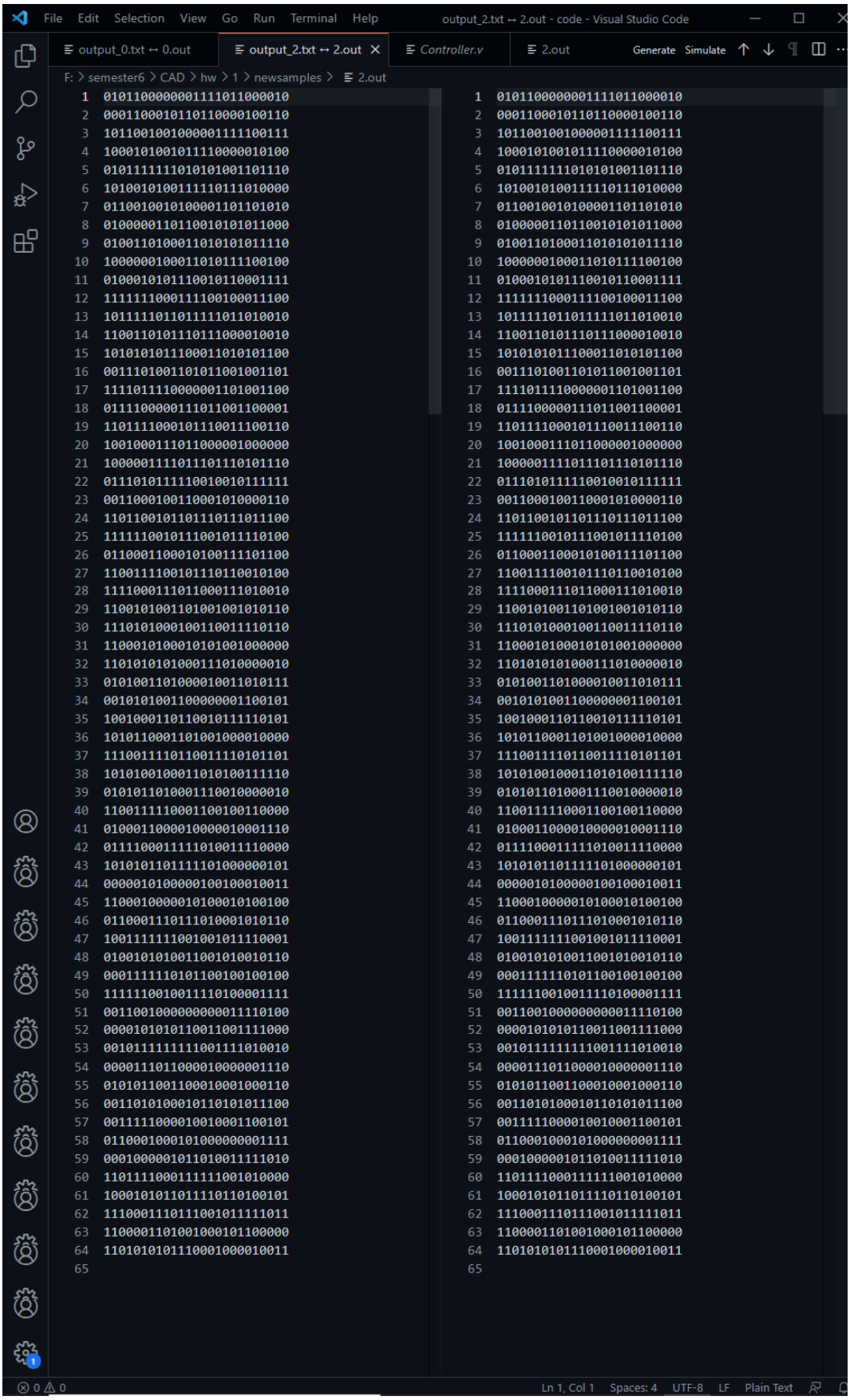
تست صفر:


File Edit Selection View Go Run Terminal Help
output_0.txt ↔ 0.out - code - Visual Studio Code
output_1.txt ↔ 1.out
0.out
output_0.txt ↔ 0.out
output_2.txt ↔ 2
Generate Simulate
F: > semester6 > CAD > hw > 1 > newsamples > 0.out
Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text

تست یک:


output_1.txt ↔ 1.out - code - Visual Studio Code
File Edit Selection View Go Run Terminal Help
output_1.txt ↔ 1.out
0.out
output_0.txt ↔ 0.out
output_2.txt ↔ 2
Generate Simulate
F: > semester6 > CAD > hw > 1 > newsamples > 1.out
Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text

تست دو:


File Edit Selection View Go Run Terminal Help
output_2.txt ↔ 2.out - code - Visual Studio Code
output_0.txt ↔ 0.out
output_2.txt ↔ 2.out
Controller.v
2.out
Generate Simulate
F: > semester6 > CAD > hw > 1 > newsamples > 2.out
Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text

Transcript

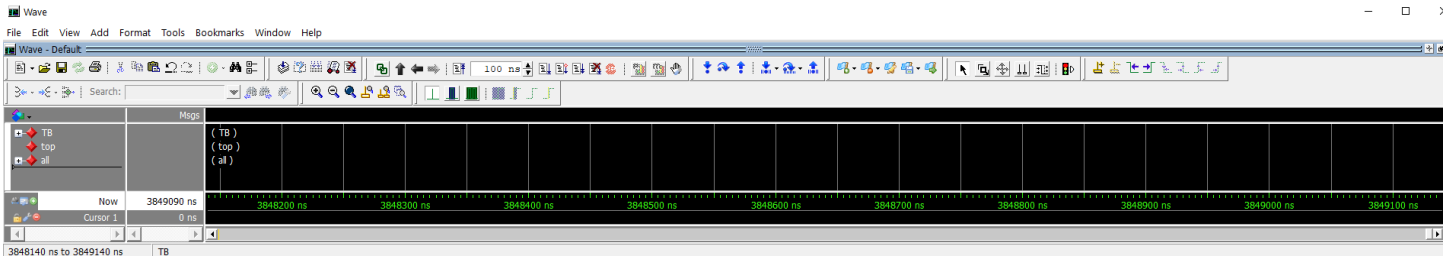
```
# ** Warning: (vsim-2685) [TFMPC] - Too few port connections for 'IRegister'.  Expected 6, found 5.
#    Time: 0 ns  Iteration: 0  Instance: /TB/dp/IRegister File: ../src/hdl/Datapath.v Line: 103
# ** Warning: (vsim-3015) [PCDPC] - Port size (6) does not match connection size (5) for port 'out'. The port defin
ition is at: ../src/hdl/Subtractor.v(6).
#    Time: 0 ns  Iteration: 0  Instance: /TB/dp/twiceNextI File: ../src/hdl/Datapath.v Line: 110
# ** Note: (vsim-8716) Reusing existing debug database vsim.dbg.
VSIM 141> run -all
# GetModuleFileName: The specified module could not be found.
#
#
# ** Note: $stop    : ./tb/TB.v(116)
#    Time: 3849090 ns  Iteration: 0  Instance: /TB
# Break in Module TB at ./tb/TB.v line 116
VSIM 142> quit -sim
# End time: 03:25:38 on Apr 08,2022, Elapsed time: 0:11:19
# Errors: 0, Warnings: 10
ModelSim> cd F:/semester6/CAD/hw/1/trunk/sim
# reading modelsim.ini

ModelSim>
```

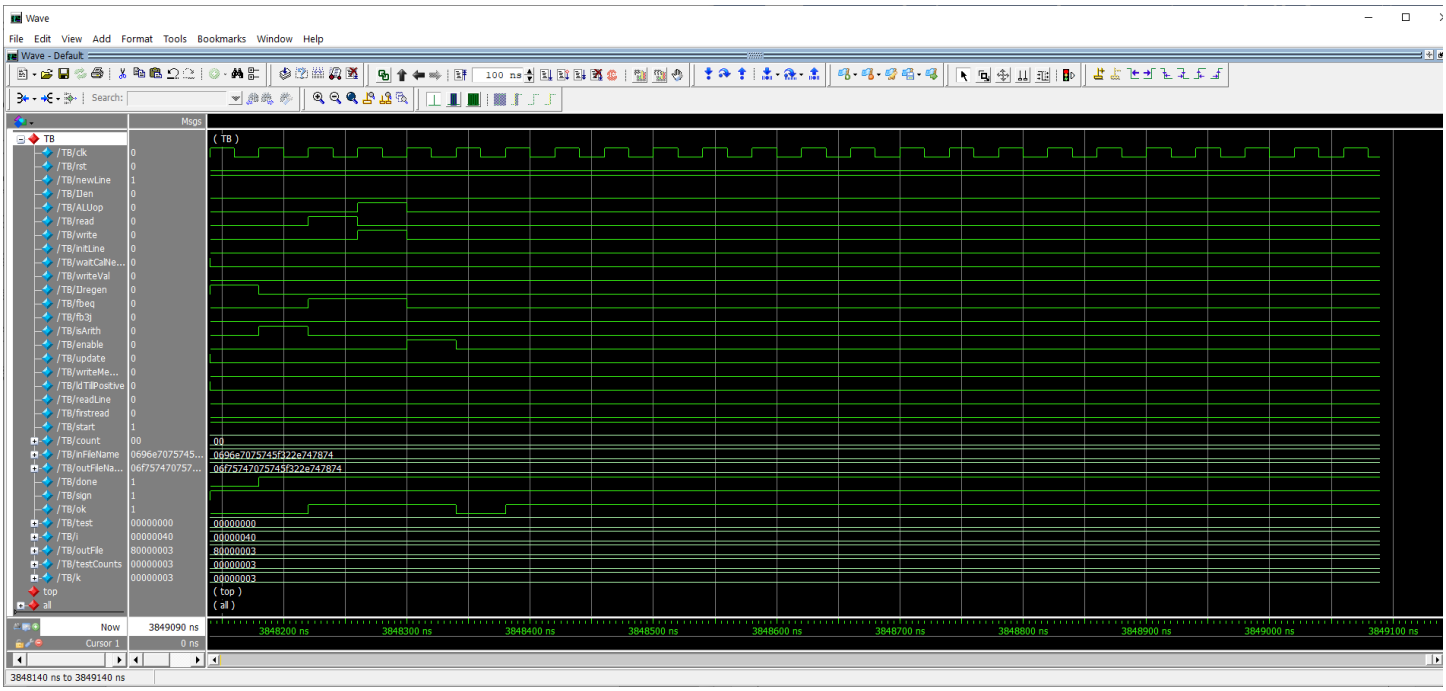
<No Design Loaded> <No Context>

Transcript

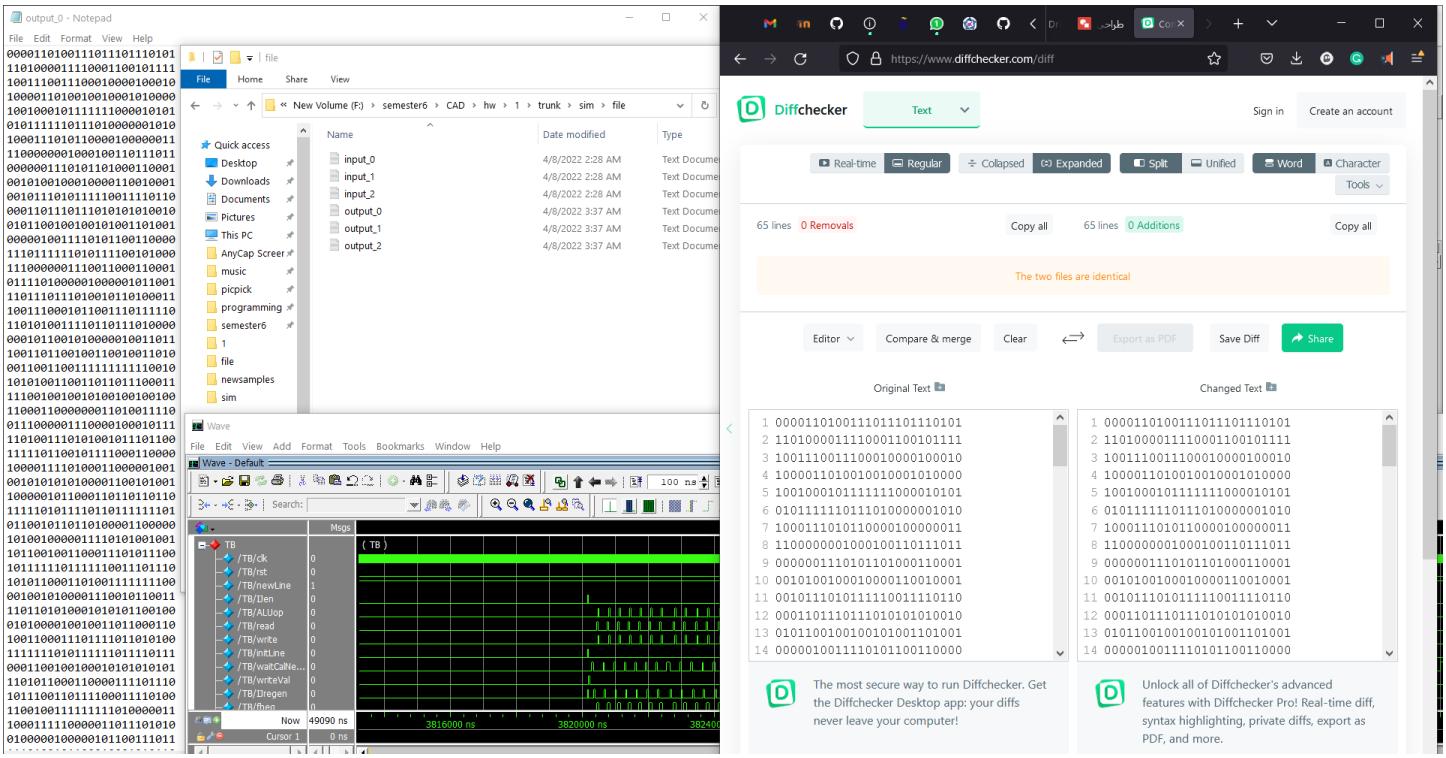
```
# ** Warning: (vsim-2685) [TFMPC] - Too few port connections for 'IRegister'.  Expected 6, found 5.
#    Time: 0 ns  Iteration: 0  Instance: /TB/dp/IRegister File: ../src/hdl/Datapath.v Line: 103
# ** Warning: (vsim-3015) [PCDPC] - Port size (6) does not match connection size (5) for port 'out'. The port defin
ition is at: ../src/hdl/Subtractor.v(6).
#    Time: 0 ns  Iteration: 0  Instance: /TB/dp/twiceNextI File: ../src/hdl/Datapath.v Line: 110
# ** Note: (vsim-8716) Reusing existing debug database vsim.dbg.
VSIM 141> run -all
# GetModuleFileName: The specified module could not be found.
#
#
# ** Note: $stop    : ./tb/TB.v(116)
#    Time: 3849090 ns  Iteration: 0  Instance: /TB
# Break in Module TB at ./tb/TB.v line 116
do sim_top.tcl

ModelSim> do sim_top.tcl
```

<No Design Loaded> <No Context>

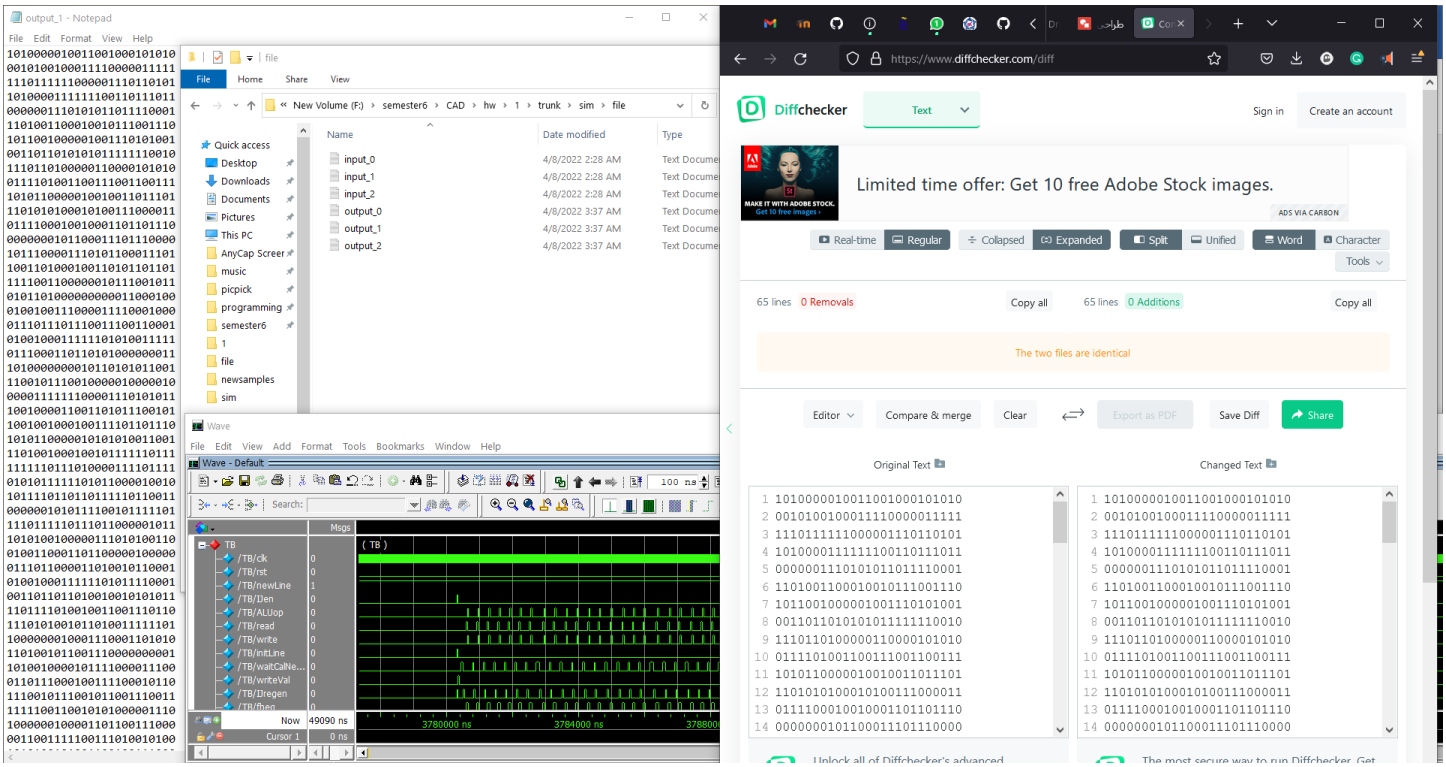




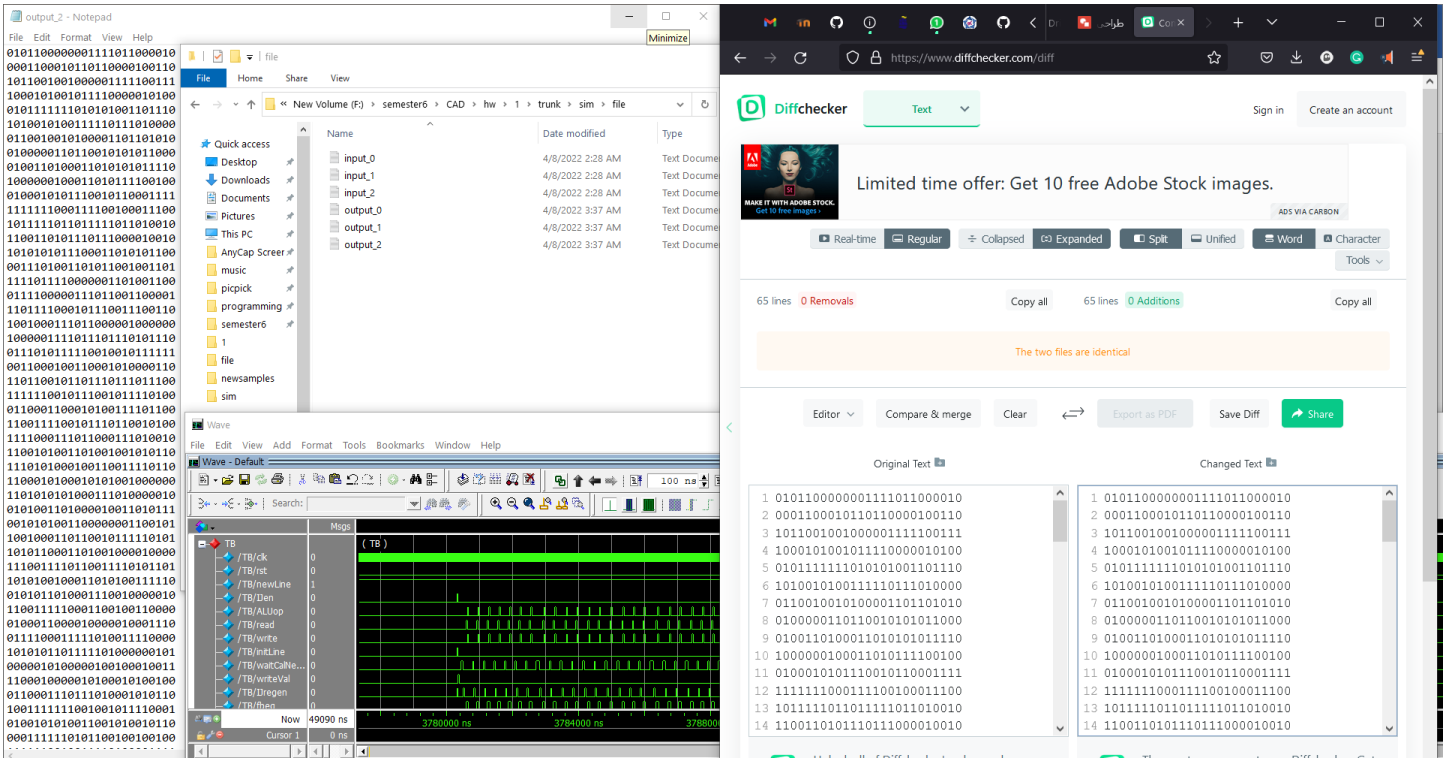
تست0:

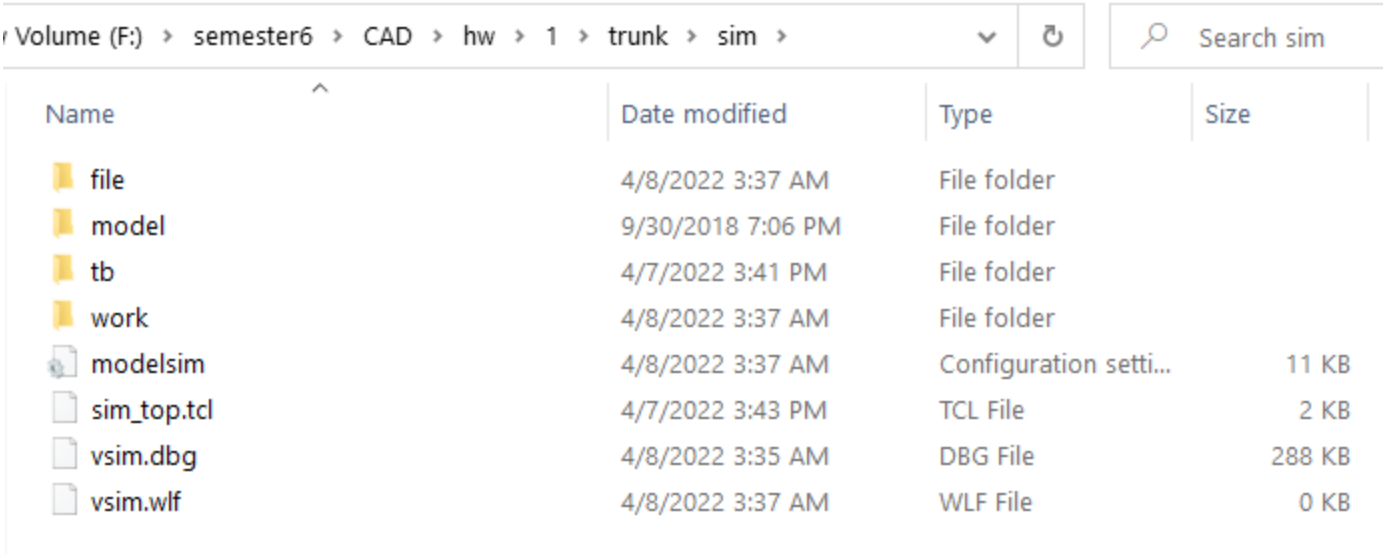


تست1:



تست2:




Volume (F:) › semester6 › CAD › hw › 1 › trunk › sim ›
Search sim
Name
Date modified
Type
Size
file
4/8/2022 3:37 AM
File folder
model
9/30/2018 7:06 PM
File folder
tb
4/7/2022 3:41 PM
File folder
work
4/8/2022 3:37 AM
File folder
modelsim
4/8/2022 3:37 AM
Configuration setti...
11 KB
sim_top.tcl
4/7/2022 3:43 PM
TCL File
2 KB
vsim.dbg
4/8/2022 3:35 AM
DBG File
288 KB
vsim.wlf
4/8/2022 3:37 AM
WLF File
0 KB