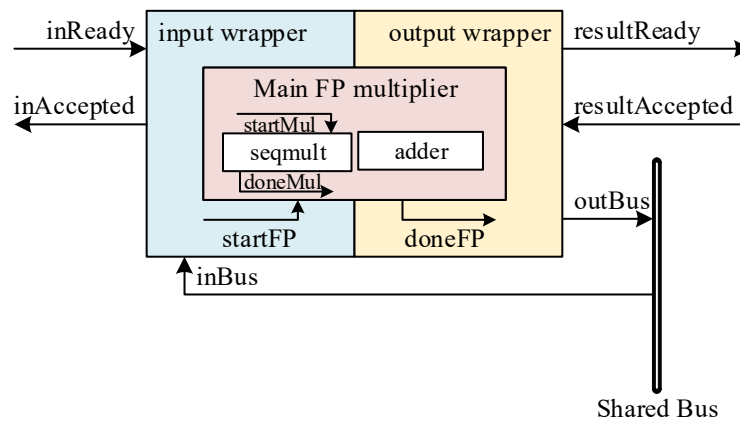
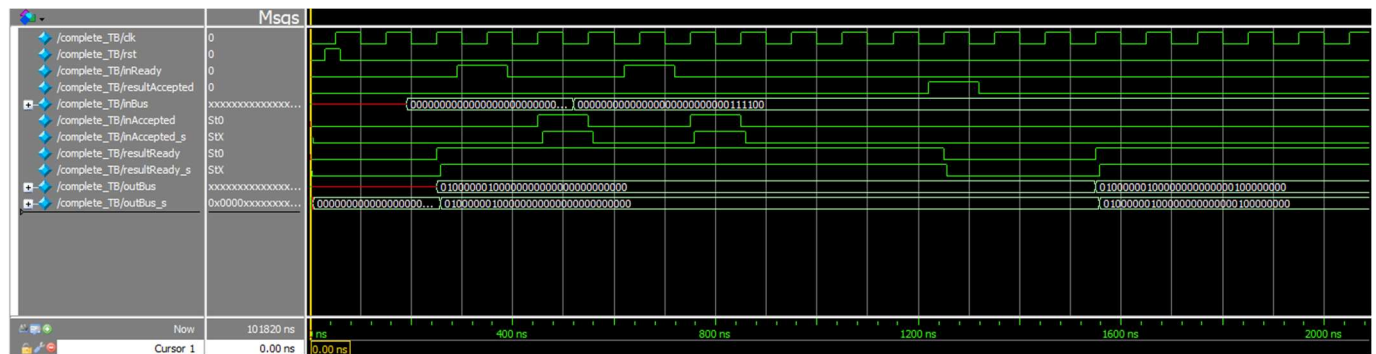


Block diagram of the entire circuit (complete) including the three parts, Wrappers, Floating Point, and the Sequential Multiplier:

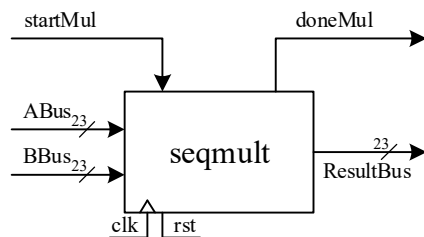


After doing parts b, c and d, results of pre-syntheses and post-syntheses of this component is shown below.

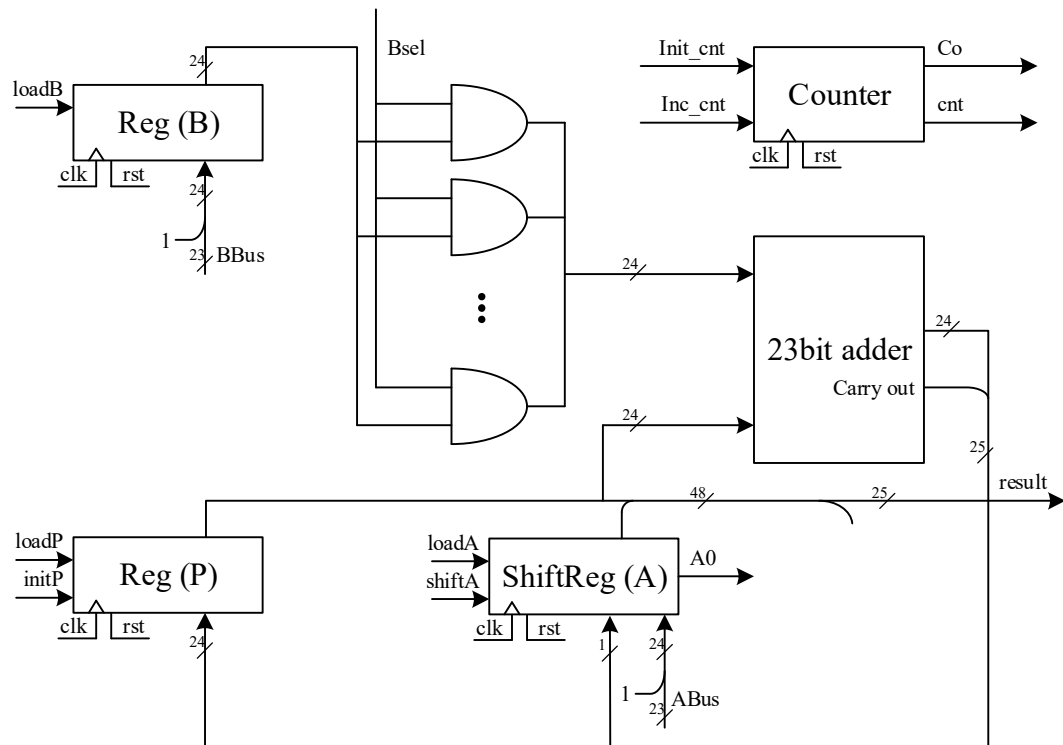


۲/۲۵	تاریخ تحویل: ۱۴۰۰/۴/۶	سیستم‌های دیجیتال CA6	محمد مهدی معینی منش ۸۱۰۱۹۸۴۷۵	شماره سوال
------	--------------------------	--------------------------	----------------------------------	---------------

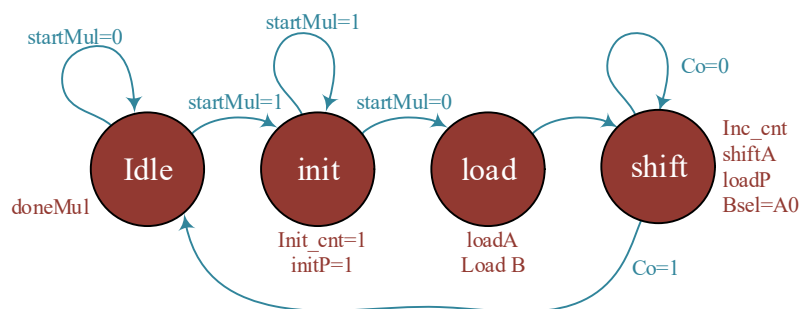
Block diagram of sequential multiplier (seqmult):



Datapath of sequential multiplier (seqmult):



State diagram of controller of sequential multiplier (seqmult):



شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:	۳/۲۵
	۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	

System Verilog code of Datapath of seqmult:

b

```

3  module MULTDP ( input clk, rst, loadA, loadB, loadP, shiftA, initP, BSel,
4      input [22:0] ABus, BBus, output [24:0] result, output A0);
5
6      reg [23:0] Areg, Breg, Preg;
7      wire [23:0] B_AND;
8      wire [24:0] AddBus;
9      //B Register
10     always @(posedge clk, posedge rst) begin
11         if (rst)
12             Breg <= 0;
13         else
14             if (loadB)
15                 Breg <= {1'b1, BBus};
16     end
17     //P Register
18     always @(posedge clk, posedge rst) begin
19         if (rst)
20             Preg <= 0;
21         else begin
22             if (initP)
23                 Preg <= 0;
24             else
25                 if (loadP)
26                     Preg <= AddBus [24:1];
27         end
28     end
29     //A Shift Register
30     always @(posedge clk, posedge rst) begin
31         if (rst)
32             Areg <= 0;
33         else begin
34             if (loadA)
35                 Areg <= {1'b1, ABus};
36             else
37                 if (shiftA)
38                     Areg <= {AddBus[0], Areg [23:1]};
39         end
40     end
41
42     assign B_AND = BSel ? Breg : 0;
43     assign AddBus = B_AND + Preg;
44     assign result = {Preg[15:0], Areg[23:15]};
45     assign A0 = Areg[0];
46
47 endmodule

```

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:	۴/۲۵
	۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	

System Verilog code of **controller** of seqmult:

b

```

49 module MULTCU ( input clk, rst, startMul, A0,
50                 output reg loadA, shiftA, loadB, loadP, initP, BSel, doneMul);
51
52     wire Co;
53     reg Init_cnt, Inc_cnt;
54     reg [1:0] pstate, nstate;
55     reg [4:0] count;
56     parameter [1:0] Idle = 0,
57                     init = 1,
58                     load = 2,
59                     shift = 3;
60
61     always @(pstate, startMul, A0, Co) begin
62         nstate = 0;
63         {loadA, shiftA, loadB, loadP, initP, BSel, doneMul} = 7'b0;
64         {Init_cnt, Inc_cnt} = 2'b0;
65
66         case (pstate)
67             Idle: begin nstate = startMul ? init : Idle; doneMul = 1'b1; end
68             init: begin nstate = startMul ? init : load; Init_cnt = 1'b1; initP = 1'b1; end
69             load: begin nstate = shift; loadA = 1'b1; loadB = 1'b1; end
70             shift: begin nstate = Co ? Idle : shift; loadP = 1'b1;
71                     shiftA = 1'b1; Inc_cnt = 1'b1; BSel = A0; end
72         endcase
73     end
74
75     always @(posedge clk, posedge rst) begin
76         if (rst)
77             pstate <= Idle;
78         else
79             pstate <= nstate;
80     end
81
82     always @(posedge clk, posedge rst) begin
83         if (rst) count <= 0;
84         else
85             if (Init_cnt) count <= 0;
86             else
87                 if (Inc_cnt) count <= count + 1;
88     end
89
90     assign Co = & count;
91 endmodule

```

۵/۲۵	تاریخ تحویل: ۱۴۰۰/۴/۶	سیستم‌های دیجیتال CA6	محمد مهدی معینی منش ۸۱۰۱۹۸۴۷۵	شماره سوال
	<p>System Verilog code of <u>TOP level</u> of seqmult:</p> <pre> 93 module MULT_TOP (input clk, rst, startMul, input [22:0] A, B, 94 output [24:0] result, output doneMul); 95 96 wire A0; 97 wire loadA, shiftA, loadB, loadP, initP, BSel; 98 MULTDP dp(clk, rst, loadA, loadB, loadP, shiftA, initP, BSel, A, B, result, A0); 99 MULTCU cu(clk, rst, startMul, A0, loadA, shiftA, loadB, loadP, initP, BSel, doneMul); 100 101 endmodule </pre>			b

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:
۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	۶/۲۵

System Verilog code of [Testbench](#) for seqmult:

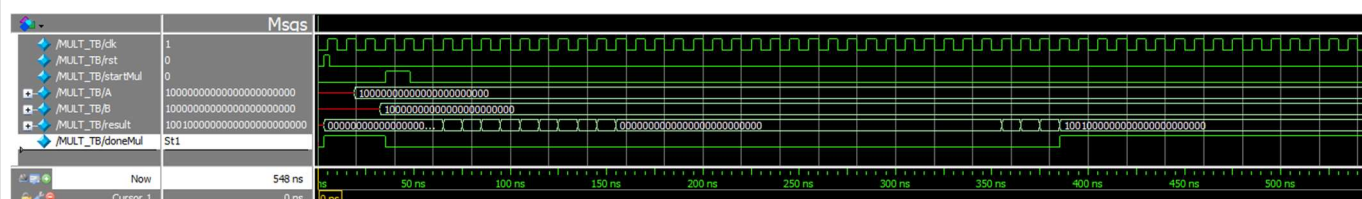
b

```

103 module MULT_TB ();
104     reg clk = 0;
105     reg rst = 0;
106     reg startMul = 0;
107     reg [22:0] A;
108     reg [22:0] B;
109     wire[24:0] result;
110     wire doneMul;

112     MULT_TOP my_ic(clk, rst, startMul, A, B, result, doneMul);
113
114     always #5 clk <= ~clk;
115     initial begin
116         #3 rst = 1;
117         #3 rst = 0;
118         #13 A = 23'b10000000000000000000000;
119         #13 B = 23'b10000000000000000000000;
120         #3 startMul = 1;
121         #13 startMul = 0;
122         #500 $stop;
123     end
124 endmodule

```



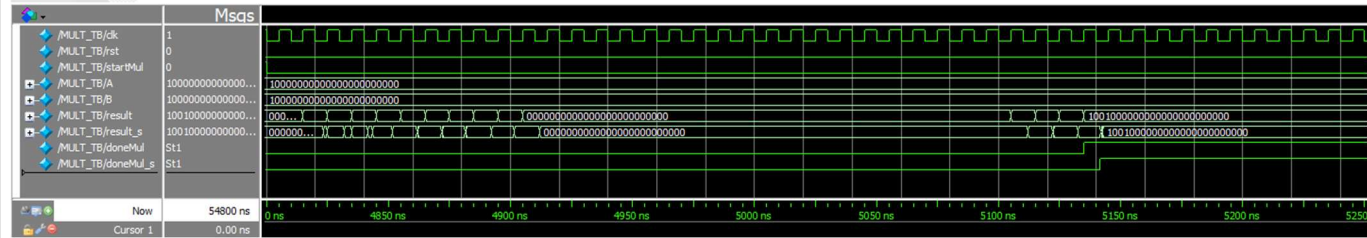
شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:
۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	۷/۲۵

Syntheses of seqmult:

```

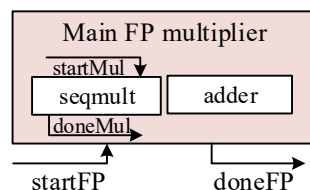
103 module MULT_TB ();
104     reg clk = 0;
105     reg rst = 0;
106     reg startMul = 0;
107     reg [22:0] A;
108     reg [22:0] B;
109     wire[24:0] result, result_s;
110     wire doneMul, doneMul_s;
111
112     MULT_TOP my_ic(clk, rst, startMul, A, B, result, doneMul);
113     seqmult synth(clk, rst, startMul, A, B, result_s, doneMul_s);
114
115     always #5 clk <= ~clk;
116     initial begin
117         #300 rst = 1;
118         #300 rst = 0;
119         #1300 A = 23'b1000000000000000000000;
120         #1300 B = 23'b1000000000000000000000;
121         #300 startMul = 1;
122         #1300 startMul = 0;
123         #50000 $stop;
124     end
125 endmodule

```

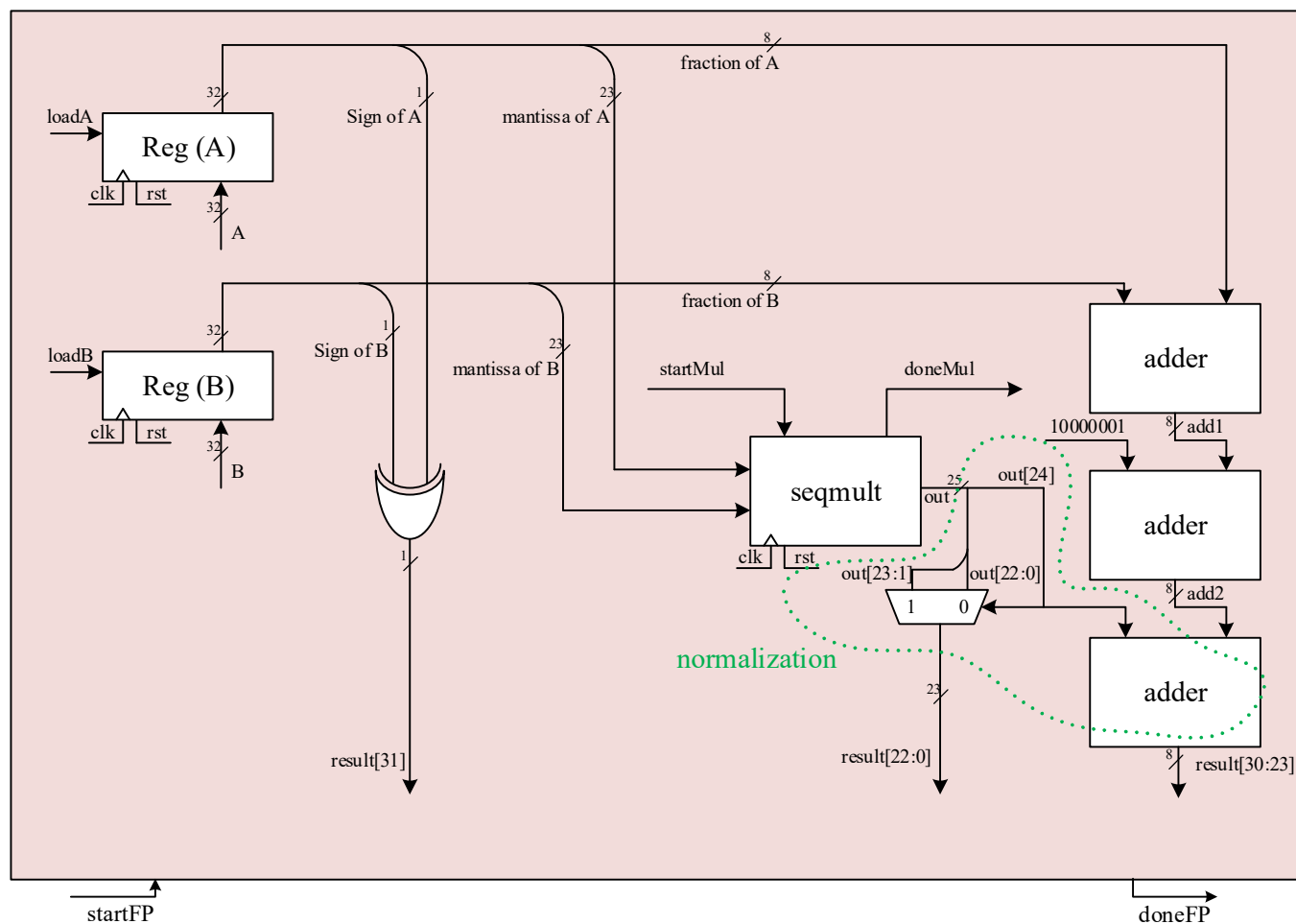


۸/۲۵	تاریخ تحویل: ۱۴۰۰/۴/۶	سیستم‌های دیجیتال CA6	محمد مهدی معینی منش ۸۱۰۱۹۸۴۷۵	شماره سوال
------	-----------------------	--------------------------	----------------------------------	------------

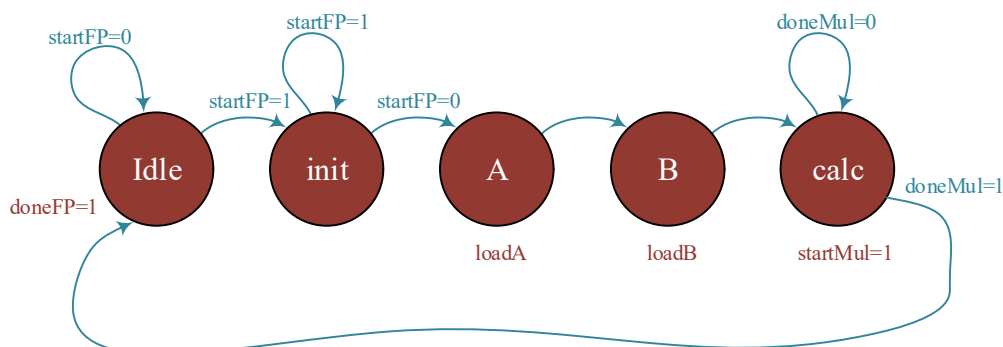
Block diagram of Main FP multiplier (MainFP):



Datapath of Main FP multiplier (MainFP):



State diagram of controller of Main FP multiplier (MainFP):



۹/۲۵	تاریخ تحویل: ۱۴۰۰/۴/۶	سیستم‌های دیجیتال CA6	محمد مهدی معینی منش ۸۱۰۱۹۸۴۷۵	شماره سوال
	<p>System Verilog code of <u>Datapath</u> of MainFP:</p> <pre> 1 `timescale 1ns/1ns 2 3 module mainFPDP (input clk, rst, loadA, loadB, startMul, 4 input [31:0] A, B, output [31:0] result, output doneMul); 5 6 reg [31:0] Areg, Breg; 7 reg [24:0] out; 8 wire [7:0] add1, add2; 9 //B Register 10 always @(posedge clk, posedge rst) begin 11 if (rst) 12 Breg <= 0; 13 else 14 if (loadB) 15 Breg <= B; 16 end 17 //A Register 18 always @(posedge clk, posedge rst) begin 19 if (rst) 20 Areg <= 0; 21 else 22 if (loadA) 23 Areg <= A; 24 end 25 26 xor xor_sign(result[31], Areg[31], Breg[31]); 27 28 MULT_TOP seqmult(clk, rst, startMul, Areg[22:0], Breg[22:0], out, doneMul); 29 30 assign add1 = Areg[30:23] + Breg[30:23]; 31 assign add2 = add1 + 8'b10000001; 32 assign result[30:23] = add2 + out[24]; 33 assign result[22:0] = out[24] ? out[23:1] : out[22:0]; 34 endmodule </pre>			c

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:
۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	۱۰/۲۵
c	System Verilog code of controller of MainFP: <pre>36 module mainFPCU (input clk, rst, startFP, doneMul, 37 output reg loadA, loadB, startMul, doneFP); 38 39 reg [2:0] pstate, nstate; 40 parameter [2:0] Idle = 0, 41 init = 1, 42 A = 2, 43 B = 3, 44 calc = 4; 45 46 always @(pstate, startFP) begin 47 nstate = 0; 48 {loadA, loadB, startMul, doneFP} = 4'b0; 49 50 case (pstate) 51 Idle: begin nstate = startFP ? init : Idle; doneFP = 1'b1; end 52 init: begin nstate = startFP ? init : A; end 53 A: begin nstate = B; loadA = 1'b1; end 54 B: begin nstate = calc; loadB = 1'b1; end 55 calc: begin nstate = doneMul ? Idle : calc; startMul = 1'b1; end 56 endcase 57 end 58 59 always @(posedge clk, posedge rst) begin 60 if (rst) 61 pstate <= Idle; 62 else 63 pstate <= nstate; 64 end 65 endmodule</pre>		

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:	۱۱/۲۵
	۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	
c	<p>System Verilog code of <u>TOP level</u> of MainFP:</p> <pre> 67 module mainFP_TOP (input clk, rst, startFP, input [31:0] A, B, 68 output [31:0] result, output doneFP); 69 70 wire loadA, loadB, startMul, doneMul; 71 mainFPDP dp(clk, rst, loadA, loadB, startMul, A, B, result, doneMul); 72 mainFPCU cu(clk, rst, startFP, doneMul, loadA, loadB, startMul, doneFP); 73 74 endmodule </pre>			

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:
۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	۱۲/۲۵

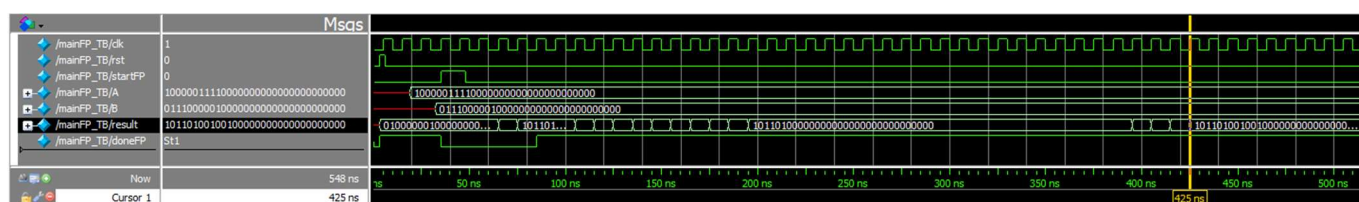
System Verilog code of [Testbench](#) for MainFP:

c

```

76 module mainFP_TB ();
77     reg clk = 0;
78     reg rst = 0;
79     reg startFP = 0;
80     reg [31:0] A;
81     reg [31:0] B;
82     wire[31:0] result;
83     wire doneFP;
84
85     mainFP_TOP my_ic(clk, rst, startFP, A, B, result, doneFP);
86
87     always #5 clk <= ~clk;
88     initial begin
89         #3 rst = 1;
90         #3 rst = 0;
91         #13 A = 32'b10000011110000000000000000000000;
92         #13 B = 32'b01110000010000000000000000000000;
93         #3 startFP = 1;
94         #13 startFP = 0;
95         #500 $stop;
96     end
97 endmodule

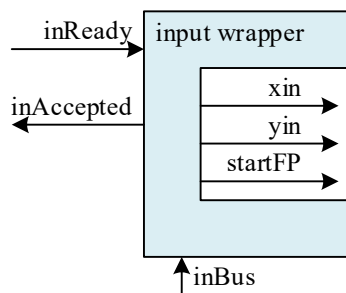
```



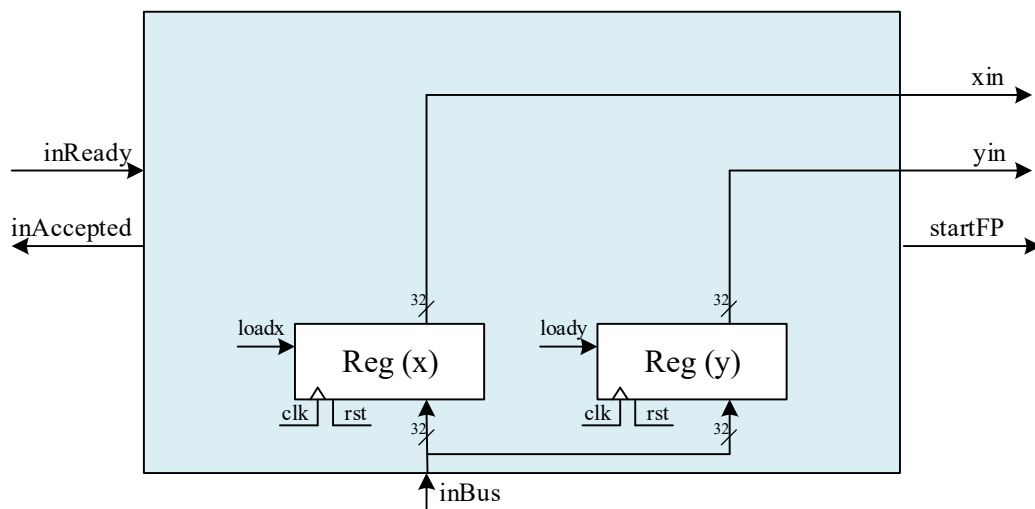
شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:
۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	۱۳/۲۵
c	<p><u>Syntheses</u> of MainFP: Unfortunately occurs an error!</p> <pre> # Loading work.mainfP # Loading work.hard_block # Loading cycloneive_ver.cycloneive_io_obuf # Loading cycloneive_ver.cycloneive_io_ibuf # Loading cycloneive_ver.cycloneive_clkctrl # Loading cycloneive_ver.cycloneive_mux41 # Loading cycloneive_ver.cycloneive_ena_reg # Loading cycloneive_ver.cycloneive_lcell_comb # Loading altera_ver.dffeas # Error loading design # End time: 17:30:59 on Jun 27,2021, Elapsed time: 0:00:01 # Errors: 1, Warnings: 1 </pre>		

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:
۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	۱۴/۲۵

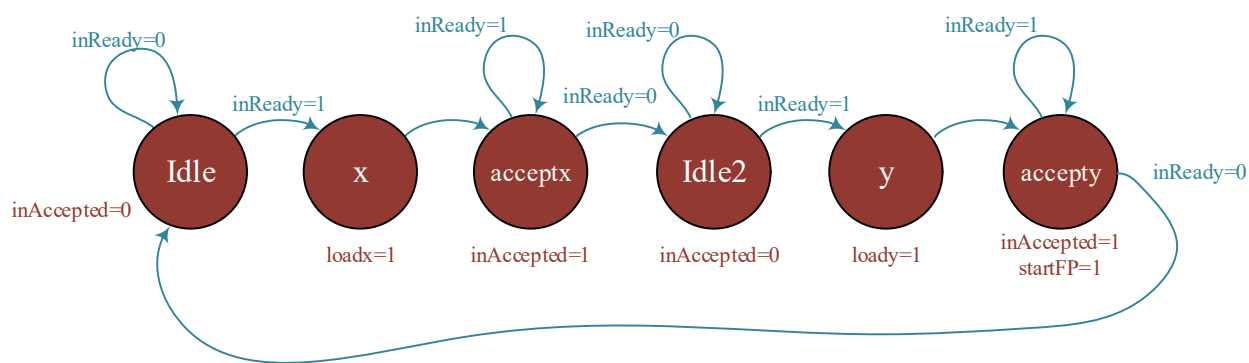
Block diagram of input wrapper (in_wrapper):



Datapath of input wrapper (in_wrapper):



State diagram of controller of input wrapper (in_wrapper):



۱۵/۲۵	تاریخ تحویل: ۱۴۰۰/۴/۶	سیستم‌های دیجیتال CA6	محمد مهدی معینی منش ۸۱۰۱۹۸۴۷۵	شماره سوال
	<p>System Verilog code of <u>Datapath</u> of in_wrapper:</p> <pre> 1 `timescale 1ns/1ns 2 3 module in_wrapperDP (input clk, rst, loadx, loady, 4 input [31:0] inBus, output [31:0] xin, yin); 5 6 reg [23:0] xreg, yreg; 7 //x Register 8 always @(posedge clk, posedge rst) begin 9 if (rst) 10 xreg <= 0; 11 else 12 if (loadx) 13 xreg <= inBus; 14 end 15 //y Register 16 always @(posedge clk, posedge rst) begin 17 if (rst) 18 yreg <= 0; 19 else 20 if (loady) 21 yreg <= inBus; 22 end 23 24 assign xin = xreg; 25 assign yin = yreg; 26 27 endmodule </pre>			d

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:
۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	۱۶/۲۵

System Verilog code of **controller** of in_wrapper:

```

29 module in_wrapperCU ( input clk, rst, inReady,
30                       output reg loadx, loady, inAccepted, startFP);
31
32     reg [2:0] pstate, nstate;
33     parameter [2:0] Idle    = 0,
34                     x       = 1,
35                     acceptx = 2,
36                     Idle2   = 3,
37                     y       = 4,
38                     accepty = 5;
39
40     always @(pstate, inReady) begin
41         nstate = 0;
42         {loadx, loady, inAccepted, startFP} = 4'b0;
43
44         case (pstate)
45             Idle:    begin nstate = inReady ? x : Idle; inAccepted = 1'b0; end
46             x:      begin nstate = acceptx; loadx = 1'b1; end
47             acceptx:begin nstate = inReady ? acceptx : Idle2; inAccepted = 1'b1; end
48             Idle2:  begin nstate = inReady ? y : Idle2; inAccepted = 1'b0; end
49             y:      begin nstate = accepty; loady = 1'b1; end
50             accepty:begin nstate = inReady ? accepty : Idle; inAccepted = 1'b1; startFP = 1'b1; end
51         endcase
52     end
53
54     always @(posedge clk, posedge rst) begin
55         if (rst)
56             pstate <= Idle;
57         else
58             pstate <= nstate;
59     end
60
61 endmodule

```

d

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:	۱۷/۲۵
	۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	
d	<p>System Verilog code of <u>TOP level</u> of in_wrapper:</p> <pre> 63 module in_wrapper_TOP (input clk, rst, inReady, input [31:0] inBus, 64 output [31:0] xin, yin, output inAccepted, startFP); 65 66 wire loadx, loady; 67 in_wrapperDP dp(clk, rst, loadx, loady, inBus, xin, yin); 68 in_wrapperCU cu(clk, rst, inReady, loadx, loady, inAccepted, startFP); 69 70 endmodule </pre>			

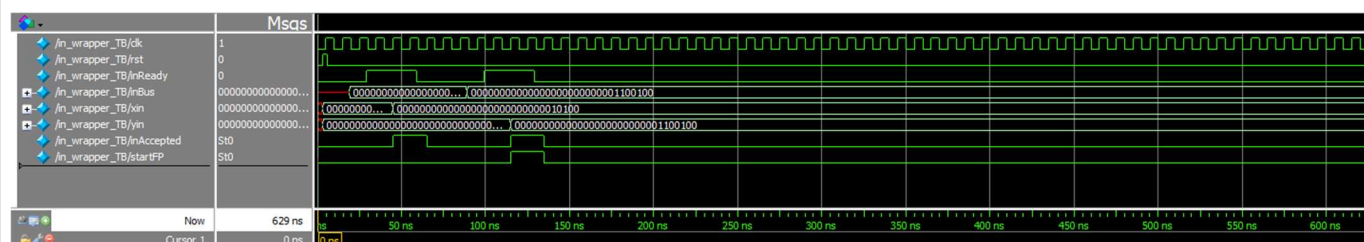
شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:
۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	۱۸/۲۵

System Verilog code of [Testbench](#) for in_wrapper:

```

72 module in_wrapper_TB ();
73     reg clk = 0;
74     reg rst = 0;
75     reg inReady = 0;
76     reg [31:0] inBus;
77     wire[31:0] xin, yin;
78     wire inAccepted;
79     wire startFP;
80
81     in_wrapper_TOP my_ic(clk, rst, inReady, inBus, xin, yin, inAccepted, startFP);
82
83     always #5 clk <= ~clk;
84     initial begin
85         #3 rst = 1;
86         #3 rst = 0;
87         #13 inBus = 20;
88         #10 inReady = 1;
89         #30 inReady = 0;
90         #30 inBus = 100;
91         #10 inReady = 1;
92         #30 inReady = 0;
93         #500 $stop;
94     end
95 endmodule

```



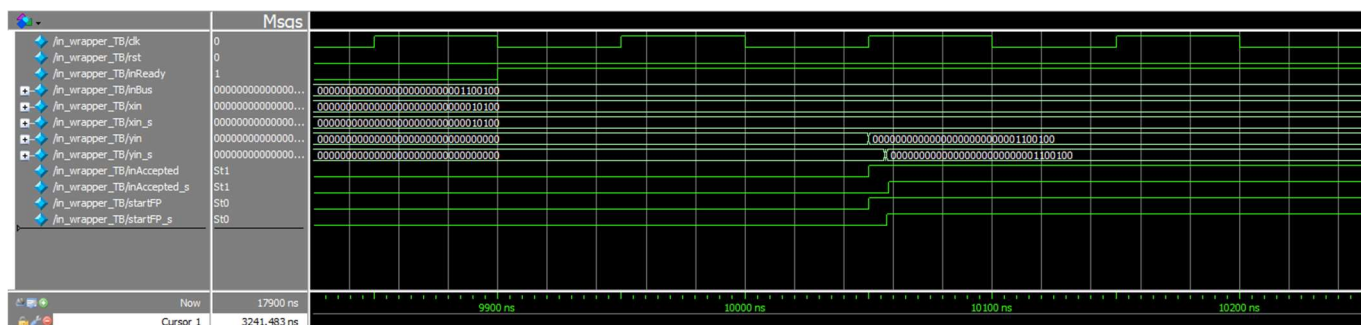
Syntheses of in_wrapper:

d

```

72 module in_wrapper_TB ();
73     reg clk = 0;
74     reg rst = 0;
75     reg inReady = 0;
76     reg [31:0] inBus;
77     wire[31:0] xin, xin_s, yin, yin_s;
78     wire inAccepted, inAccepted_s;
79     wire startFP, startFP_s;
80
81     in_wrapper_TOP my_ic(clk, rst, inReady, inBus, xin, yin, inAccepted, startFP);
82     in_wrapper      synth(clk, rst, inReady, inBus, xin_s, yin_s, inAccepted_s, startFP_s);
83     always #50 clk <= ~clk;
84     initial begin
85         #300 rst = 1;
86         #300 rst = 0;
87         #1300 inBus = 20;
88         #1000 inReady = 1;
89         #3000 inReady = 0;
90         #3000 inBus = 100;
91         #1000 inReady = 1;
92         #3000 inReady = 0;
93         #5000 $stop;
94     end
95 endmodule

```



شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:
۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	۲۰/۲۵

d

Block diagram of output wrapper (out_wrapper):

```
graph LR
    result --> out_wrapper
    doneFP --> out_wrapper
    out_wrapper --> resultReady
    out_wrapper --> resultAccepted
    out_wrapper --> outBus
```

Datapath of output wrapper (out_wrapper):

```
graph LR
    result --> datapath
    doneFP --> datapath
    datapath --> resultReady
    datapath --> resultAccepted
    datapath --> outBus
```

State diagram of controller of output wrapper (out_wrapper):

```
graph LR
    Idle((Idle)) -- "doneFP=0" --> Idle
    Idle -- "doneFP=1" --> load((load))
    load -- "resultAccepted=0" --> accept((accept))
    load -- "resultAccepted=1" --> accept
    accept -- "resultAccepted=0" --> accept
    accept -- "resultAccepted=1" --> finish((finish))
    finish -- "resultAccepted=0" --> Idle
    finish -- "resultAccepted=1" --> finish
```

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:	۲۱/۲۵
	۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	
d	<p>System Verilog code of <u>Datapath</u> of out_wrapper:</p> <pre> 1 `timescale 1ns/1ns 2 3 module out_wrapperDP (input clk, rst, loadout, 4 input [31:0] result, output [31:0] outBus); 5 6 reg [31:0] outreg; 7 //out Register 8 always @(posedge clk, posedge rst) begin 9 if (loadout) 10 outreg <= result; 11 end 12 assign outBus = outreg; 13 14 endmodule </pre>			

۲۲/۲۵	تاریخ تحویل: ۱۴۰۰/۴/۶	سیستم‌های دیجیتال CA6	محمد مهدی معینی منش ۸۱۰۱۹۸۴۷۵	شماره سوال
	<p>System Verilog code of <u>controller</u> of out_wrapper:</p> <pre> 16 module out_wrapperCU (input clk, rst, doneFP, resultAccepted, 17 output reg loadout, resultReady); 18 19 reg [1:0] pstate, nstate; 20 parameter [1:0] Idle = 0, 21 load = 1, 22 accept = 2, 23 finish = 3; 24 25 always @(pstate, doneFP, resultAccepted) begin 26 nstate = 0; 27 {loadout, resultReady} = 2'b0; 28 29 case (pstate) 30 Idle: begin nstate = doneFP ? load : Idle; end 31 load: begin nstate = accept; loadout = 1'b1; end 32 accept: begin nstate = resultAccepted ? finish : accept; resultReady = 1'b1; end 33 finish: begin nstate = resultAccepted ? finish : Idle; resultReady = 1'b0; end 34 endcase 35 end 36 37 always @(posedge clk, posedge rst) begin 38 if (rst) 39 pstate <= Idle; 40 else 41 pstate <= nstate; 42 end </pre>			d

شماره سوال	محمد مهدی معینی منش	سیستم‌های دیجیتال	تاریخ تحویل:	۲۳/۲۵
	۸۱۰۱۹۸۴۷۵	CA6	۱۴۰۰/۴/۶	
d	<p>System Verilog code of <u>TOP level</u> of out_wrapper:</p> <pre> 46 module out_wrapper_TOP (input clk, rst, doneFP, resultAccepted, input [31:0] result, 47 output [31:0] outBus, output resultReady); 48 49 wire loadout; 50 out_wrapperDP dp(clk, rst, loadout, result, outBus); 51 out_wrapperCU cu(clk, rst, doneFP, resultAccepted, loadout, resultReady); 52 53 endmodule </pre>			

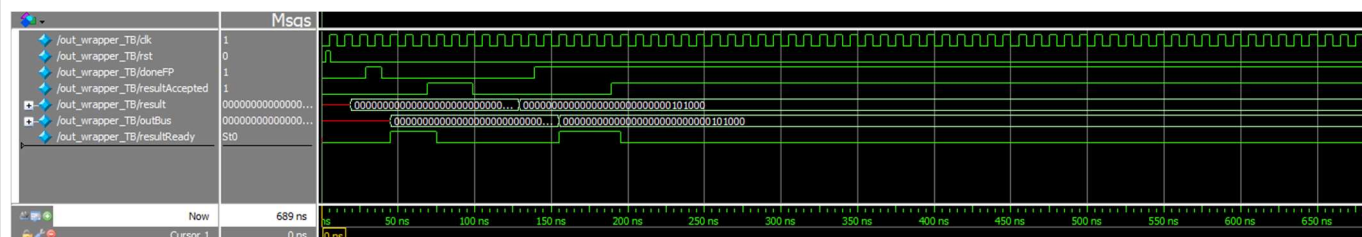
System Verilog code of Testbench for out_wrapper:

d

```

55 module out_wrapper_TB ();
56     reg clk = 0;
57     reg rst = 0;
58     reg doneFP = 0;
59     reg resultAccepted = 0;
60     reg [31:0] result;
61     wire[31:0] outBus;
62     wire resultReady;
63
64     out_wrapper_TOP my_ic(clk, rst, doneFP, resultAccepted, result, outBus, resultReady);
65
66     always #5 clk <= ~clk;
67     initial begin
68         #3 rst = 1;
69         #3 rst = 0;
70         #13 result = 20;
71         #10 doneFP = 1;
72         #10 doneFP = 0;
73         #30 resultAccepted = 1'b1;
74         #30 resultAccepted = 1'b0;
75         #30 result = 40;
76         #10 doneFP = 1;
77         #50 resultAccepted = 1'b1;
78         #500 $stop;
79     end
80 endmodule

```



Syntheses of out_wrapper: