Τμήμα Πληροφορικής
Μάθημα: Προηγμένες Εφαρμογές Ψηφιακής Σχεδίασης
Ονοματεπώνυμο : Μάριος Πλεγχίδης
ΑΕΜ: 5299
Εξάμηνο: 3ο

ΑΠΑΝΤΗΣΕΙΣ 2ης Σειράς Ασκήσεων

Μέρος Ι

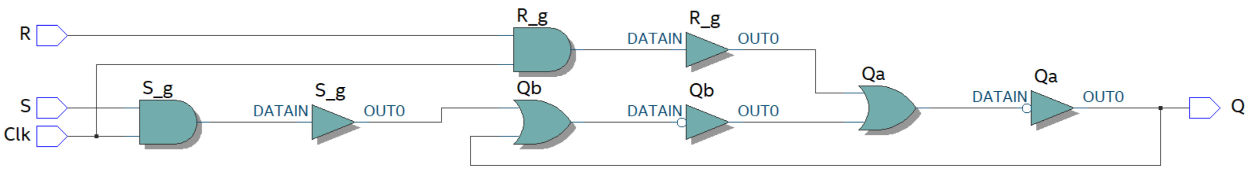
Ο κώδικας rslatch41.sv είναι:

```
module rslatch41(input logic R, S, Clk,
                 output logic Q);

  logic R_g, S_g, Qa, Qb /* synthesis keep */;

  assign R_g = R && Clk;
  assign S_g = S && Clk;
  assign Qa = ~(R_g || Qb);
  assign Qb = ~ (S_g || Qa);
  assign Q = Qa;
endmodule
```

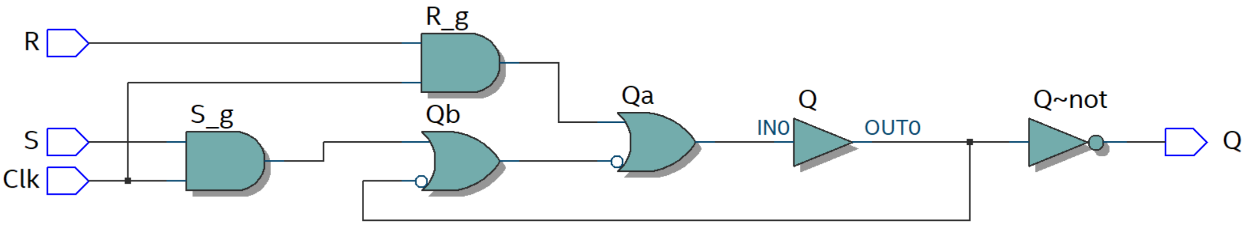
Ο κώδικας rslatch41.do είναι:

```
vsim rslatch41
add wave Clk
add wave R
add wave S
add wave Q
force Clk 1 0, 1 40, 0 60, 1 80, 0 100
force S 1 0, 0 60, 1 100
force R 0 0, 1 60, 0 100
run 100
```

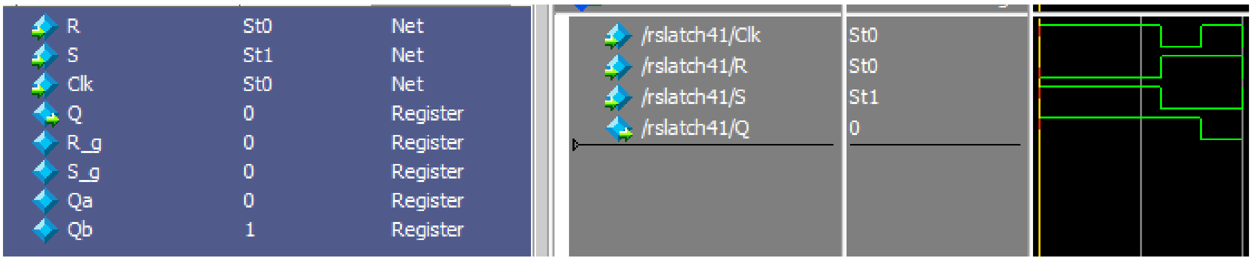
Η εικόνα από τον netlist viewer με /* synthesis keep */:



Η εικόνα από τον netlist viewer χωρίς /* synthesis keep */:



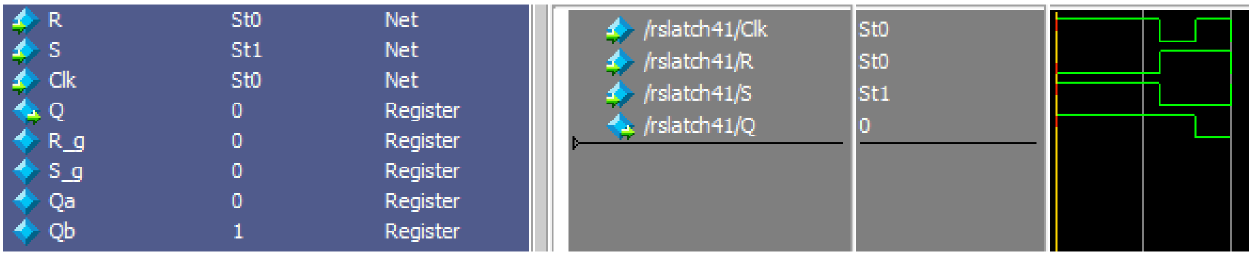
Η τελική εικόνα της προσομοίωσης με /* synthesis keep */:



R	St0	Net
S	St1	Net
Clk	St0	Net
Q	0	Register
R_g	0	Register
S_g	0	Register
Qa	0	Register
Qb	1	Register

/rslatch41/Clk	St0
/rslatch41/R	St0
/rslatch41/S	St1
/rslatch41/Q	0

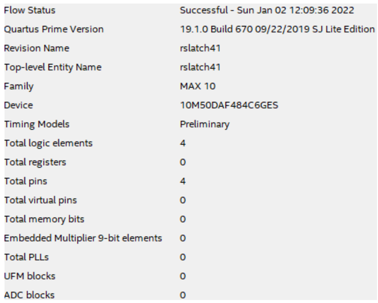
Η τελική εικόνα της προσομοίωσης χωρίς /* synthesis keep */:



R	St0	Net
S	St1	Net
Clk	St0	Net
Q	0	Register
R_g	0	Register
S_g	0	Register
Qa	0	Register
Qb	1	Register

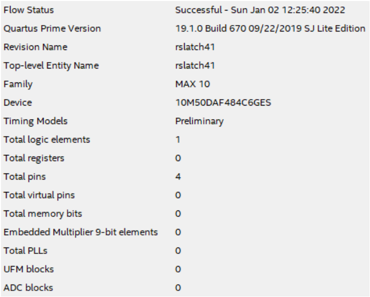
/rslatch41/Clk	St0
/rslatch41/R	St0
/rslatch41/S	St1
/rslatch41/Q	0

Εικόνα του compilation report με /* synthesis keep */:



Flow Status	Successful - Sun Jan 02 12:09:36 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	rslatch41
Top-level Entity Name	rslatch41
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	4
Total registers	0
Total pins	4
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Εικόνα του compilation report με /* synthesis keep */:



Flow Status	Successful - Sun Jan 02 12:25:40 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	rslatch41
Top-level Entity Name	rslatch41
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	1
Total registers	0
Total pins	4
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Μέρος ΙΙ

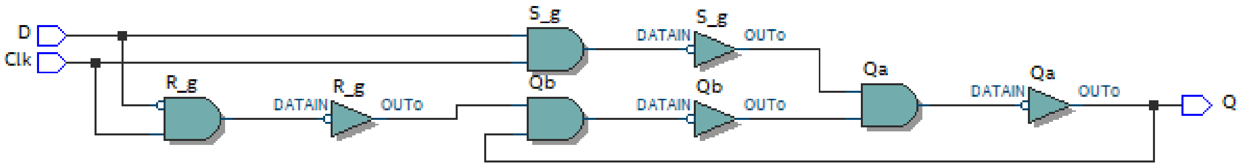
Ο κώδικας gdlatch.sv είναι:

```
module gdlatch(input logic D, Clk,
               output logic Q);
  logic S_g, R_g, Qa, Qb /* synthesis keep */;

  assign S_g = ~(D && Clk);
  assign R_g = ~(~D && Clk);
  assign Qa = ~(S_g && Qb);
  assign Qb = ~(R_g && Qa);
  assign Q = Qa;
endmodule
```

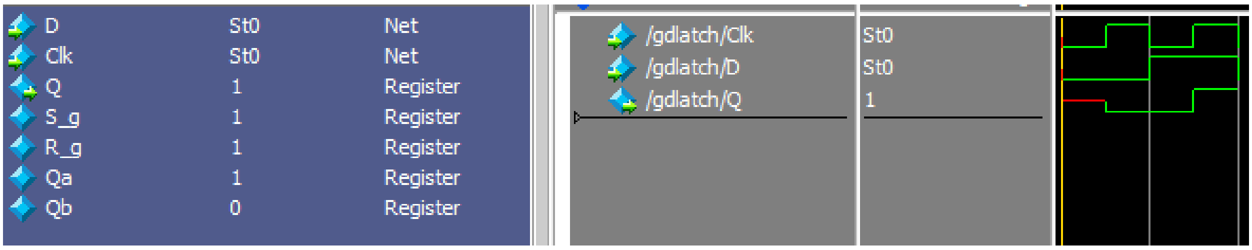
Ο κώδικας gdlatch.do είναι:

```
vsim gdlatch
add wave Clk D Q
force -repeat 100 D 0 0, 1 50
force -repeat 50 Clk 0 0, 1 25
run 100
```

Η εικόνα από τον netlist viewer:



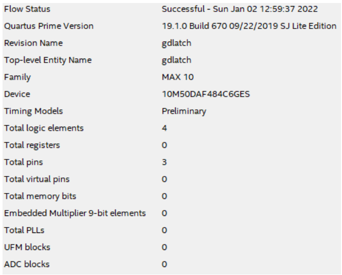
Η τελική εικόνα της προσομοίωσης:



D	St0	Net
Clk	St0	Net
Q	1	Register
S_g	1	Register
R_g	1	Register
Qa	1	Register
Qb	0	Register

/gdlatch/Clk	St0
/gdlatch/D	St0
/gdlatch/Q	1

Εικόνα του compilation report:



Flow Status	Successful - Sun Jan 02 12:59:37 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	gdlatch
Top-level Entity Name	gdlatch
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	4
Total registers	0
Total pins	3
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Μέρος III

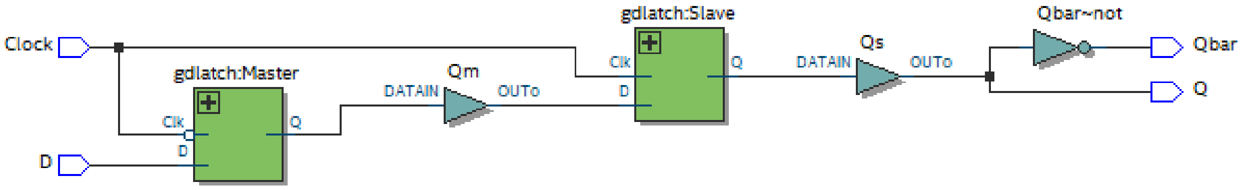
Ο κώδικας msdff.sv είναι:

```
module msdff(input logic D, Clock,
             output logic Q, Qbar);
  logic Qm, Qs /* synthesis keep */;

  gdlatch Master(D, ~Clock, Qm);
  gdlatch Slave(Qm, Clock, Qs);
  assign Q = Qs;
  assign Qbar = ~Q;
endmodule
```

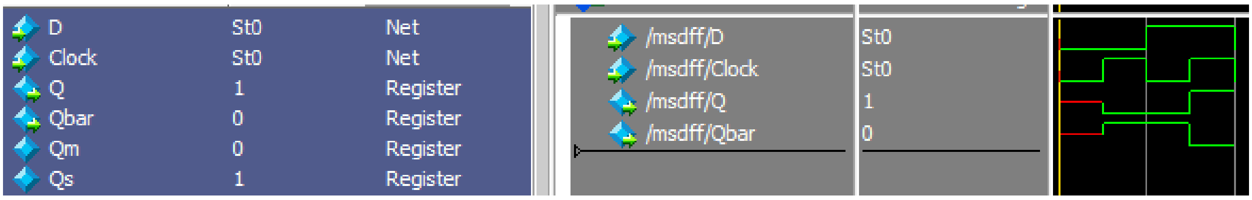
Ο κώδικας msdff.do είναι:

```
vsim msdff
add wave D Clock Q Qbar
force -repeat 50 Clock 0 0, 1 25
force -repeat 100 D 0 0, 1 50
run 100
```

Η εικόνα από τον netlist viewer:



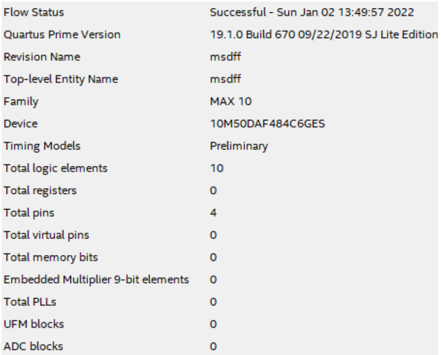
Η τελική εικόνα της προσομοίωσης:



D	St0	Net
Clock	St0	Net
Q	1	Register
Qbar	0	Register
Qm	0	Register
Qs	1	Register

/msdff/D	St0
/msdff/Clock	St0
/msdff/Q	1
/msdff/Qbar	0

Εικόνα του compilation report:



Flow Status	Successful - Sun Jan 02 13:49:57 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	msdff
Top-level Entity Name	msdff
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	10
Total registers	0
Total pins	4
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Μέρος IV

dalwlatch.sv

```
module dalwlatch(input logic D, Clk,
                 output Q);

  always_latch
    if (Clk) Q <= D;

endmodule
```

pedff.sv

```
module pedff(input logic D, Clk,
             output logic Q);

  always_ff@(posedge Clk)
    Q <= D;

endmodule
```

nedff.sv

```
module nedff(input logic D, Clk,
             output logic Q);

  always_ff@(negedge Clk)
    Q <= D;

endmodule
```

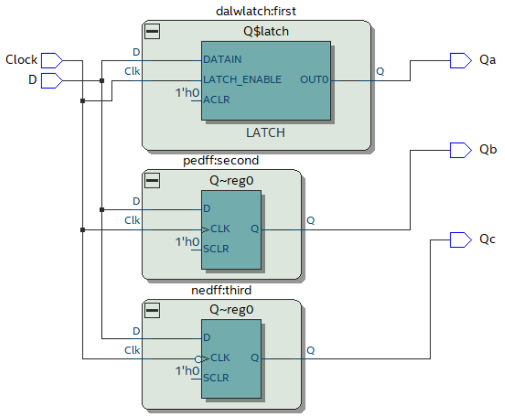
Ο κώδικας lab2p4.sv είναι:

```
module lab2p4(input logic D, Clock,
              output logic Qa, Qb, Qc);

dalwlatch first(D, Clock, Qa);
pedff second(D, Clock, Qb);
nedff third(D, Clock, Qc);

endmodule
```

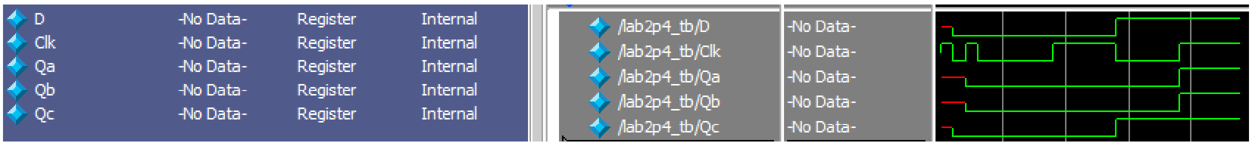
Ο κώδικας lab2p4_tb.sv είναι:

```
module lab2p4_tb();

  logic D, Clk;
  logic Qa,Qb,Qc;

  lab2p4 dut(D, Clk, Qa, Qb, Qc);

  initial begin
    Clk = 1;         #10;
    Clk = 0; D = 0; #10;
    Clk = 1; D = 0; #10;
    Clk = 0;         #10;    // setting 0 as initial Values


    Clk = 0; D = 0; #50;
    Clk = 1; D = 0; #50;
    Clk = 0; D = 1; #50;
    Clk = 1; D = 1; #50;

  end
endmodule
```

Η εικόνα από τον netlist viewer:



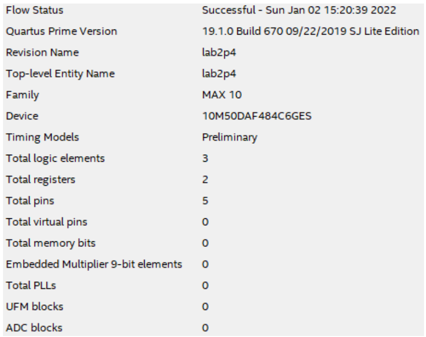
Η τελική εικόνα της προσομοίωσης:



D	-No Data-	Register	Internal
Clk	-No Data-	Register	Internal
Qa	-No Data-	Register	Internal
Qb	-No Data-	Register	Internal
Qc	-No Data-	Register	Internal

/lab2p4_tb/D	-No Data-
/lab2p4_tb/Clk	-No Data-
/lab2p4_tb/Qa	-No Data-
/lab2p4_tb/Qb	-No Data-
/lab2p4_tb/Qc	-No Data-

Εικόνα του compilation report:



Flow Status	Successful - Sun Jan 02 15:20:39 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	lab2p4
Top-level Entity Name	lab2p4
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	3
Total registers	2
Total pins	5
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Μέρος V

Άσκηση 1

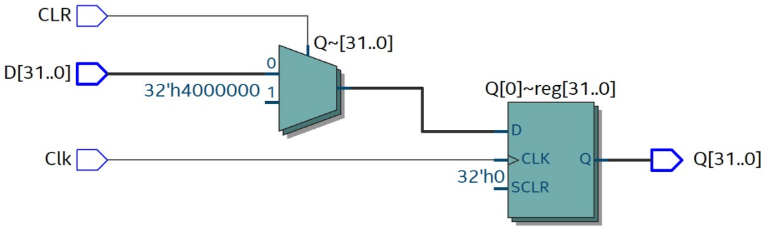
Ο κώδικας regN.sv είναι:

```
module regN
      # (parameter N = 32)
        (input logic [N-1:0] D,
         input logic Clk, CLR,
         output logic [N - 1:0] Q);

  always_ff@(posedge Clk)

    if(CLR) Q <= N-1'b0;
    else    Q <= D;
endmodule
```

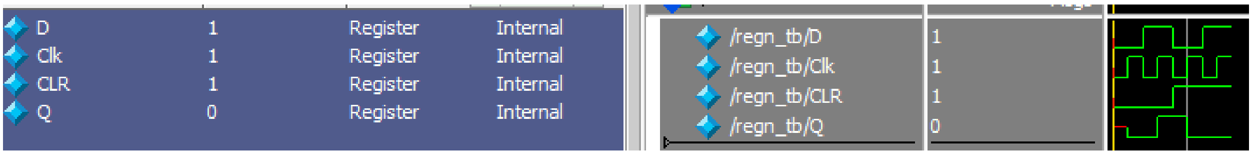
Ο κώδικας regN_tb.sv είναι:

```
module regN_tb();

  logic D, Clk, CLR;
  logic Q;

  regN dut(D, Clk, CLR, Q);

  initial begin


    CLR = 0; D = 0; Clk =0; #10;
    CLR = 0; D = 0; Clk =1; #10;
    CLR = 0; D = 1; Clk =0; #10;
    CLR = 0; D = 1; Clk =1; #10;
    CLR = 1; D = 0; Clk =0; #10;
    CLR = 1; D = 0; Clk =1; #10;
    CLR = 1; D = 1; Clk =0; #10;
    CLR = 1; D = 1; Clk =1; #10;

  end
endmodule
```

Η εικόνα από τον netlist viewer:



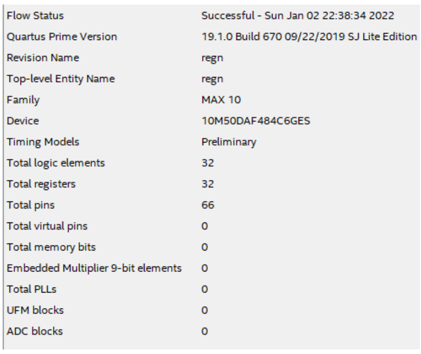
Η τελική εικόνα της προσομοίωσης:



D	1	Register	Internal
Clk	1	Register	Internal
CLR	1	Register	Internal
Q	0	Register	Internal

/regn_tb/D	1
/regn_tb/Clk	1
/regn_tb/CLR	1
/regn_tb/Q	0

Εικόνα του compilation report:



Flow Status	Successful - Sun Jan 02 22:38:34 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	regn
Top-level Entity Name	regn
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	32
Total registers	32
Total pins	66
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Άσκηση 2

Ο κώδικας SregFile.sv είναι:

```
module SregFile(input logic [31:0] DataIn,
                input logic [2:0] Addr,
                input logic regWE, CLR, clk,
                output logic [31:0] DataOut);

  logic DWo0, DWo1, DWo2, DWo3, DWo4, DWo5, DWo6, DWo7;
  logic [31:0] Ro0, Ro1, Ro2, Ro3, Ro4, Ro5, Ro6, Ro7;

    WriteDecoder WDinst(Addr, regWE, DWo0, DWo1, DWo2, DWo3, DWo4, DWo5, DWo6, DWo7);
    regN R0(DataIn, clk && DWo0, CLR, Ro0);
    regN R1(DataIn, clk && DWo1, CLR, Ro1);
    regN R2(DataIn, clk && DWo2, CLR, Ro2);
    regN R3(DataIn, clk && DWo3, CLR, Ro3);
    regN R4(DataIn, clk && DWo4, CLR, Ro4);
    regN R5(DataIn, clk && DWo5, CLR, Ro5);
    regN R6(DataIn, clk && DWo6, CLR, Ro6);
    regN R7(DataIn, clk && DWo7, CLR, Ro7);
    mux3281 muxinst(Ro0, Ro1, Ro2, Ro3, Ro4, Ro5, Ro6, Ro7, Addr, DataOut);


endmodule
```

Ο κώδικας WriteDecoder.sv είναι:

```
module WriteDecoder(input logic [2:0] Addr,
                    input logic regWE,
                    output logic o0, o1, o2, o3, o4, o5, o6, o7);

  always_comb begin
    if (regWE) begin
      o0 = 0; o1 = 0; o2 = 0; o3 = 0; o4 = 0; o5 = 0; o6 = 0; o7 = 0;
      case (Addr)
        0:          o0 = 1;
        1:          o1 = 1;
        2:          o2 = 1;
        3:          o3 = 1;
        4:          o4 = 1;
        5:          o5 = 1;
        6:          o6 = 1;
        7:          o7 = 1;
        default: begin  o0 = 0; o1 = 0; o2 = 0; o3 = 0; o4 = 0; o5 = 0; o6 = 0; o7 = 0; end
      endcase
    end
    else        begin      o0 = 0; o1 = 0; o2 = 0; o3 = 0; o4 = 0; o5 = 0; o6 = 0; o7 = 0; end
  end
endmodule
```

Ο κώδικας mux3281.sv είναι:

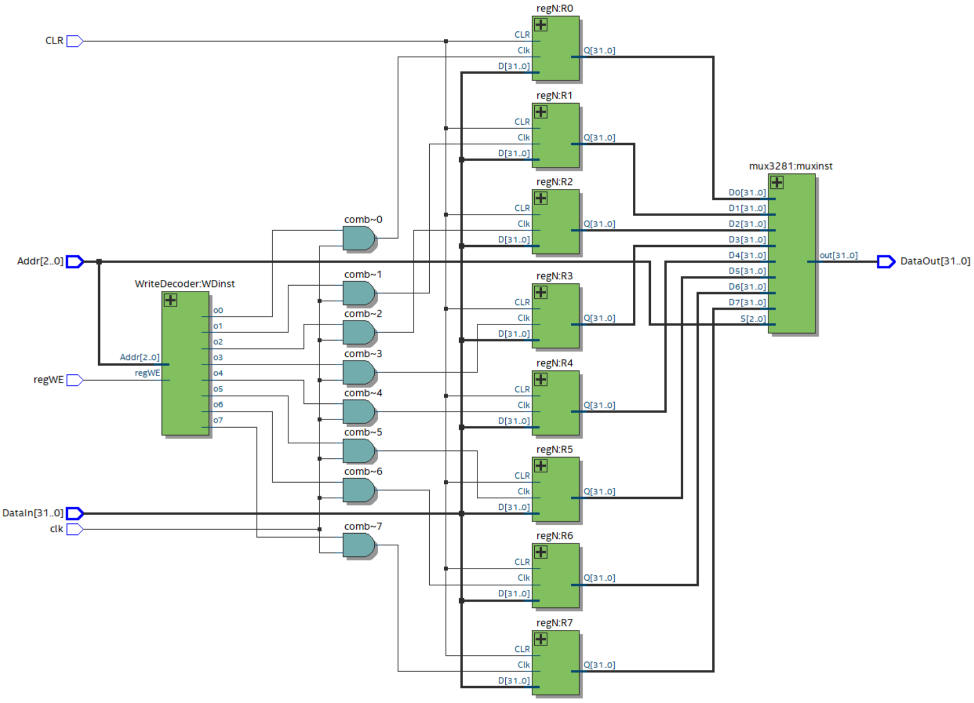
```
module mux3281(input logic [31:0] D0, D1, D2, D3, D4, D5, D6, D7,
               input logic [2:0] S,
               output logic [31:0] out);

  always_comb
    case(S)
      0: out = D0;
      1: out = D1;
      2: out = D2;
      3: out = D3;
      4: out = D4;
      5: out = D5;
      6: out = D6;
      7: out = D7;
      default: out = 2'h00;
    endcase
endmodule
```

Ο κώδικας RegN.sv είναι:
Ίδιος με τον RegN στο **<u>Μέρος V</u>** Άσκηση 1

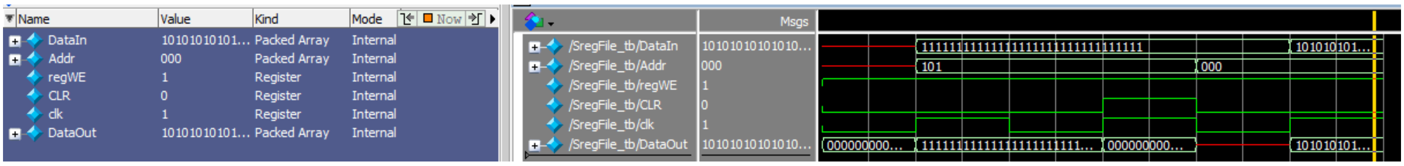
Ο κώδικας SregFile_tb.sv είναι:

```
module SregFile_tb();

  logic [31:0] DataIn;
  logic [2:0] Addr;
  logic regWE, CLR, clk;
  logic [31:0] DataOut;

  SregFile dut(DataIn, Addr, regWE, CLR, clk, DataOut);

  initial begin

    // write 0xFFFFFFFF to 6th register
    CLR = 0; clk = 0; regWE = 1; #10;
    Addr = 5; DataIn ='hFFFFFFFF; clk = 1; #10;
    clk = 0; #10;
    // clear 5th register and write 0xAAAAAAAA to first register
    CLR = 1; clk = 1; #10;
    Addr = 0; clk =0 ; CLR =0 ;#10;
    DataIn = 'hAAAAAAAA; clk = 1; #10;

  end
endmodule
```

Η εικόνα από τον netlist viewer:



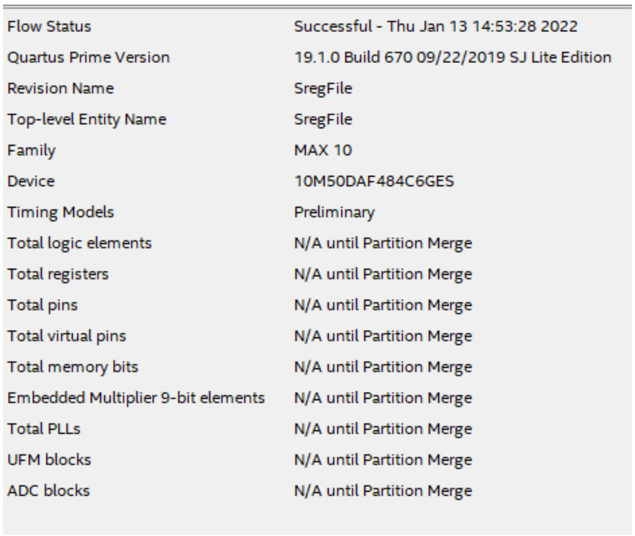
Η τελική εικόνα της προσομοίωσης:



Name	Value	Kind	Mode
DataIn	10101010101...	Packed Array	Internal
Addr	000	Packed Array	Internal
regWE	1	Register	Internal
CLR	0	Register	Internal
clk	1	Register	Internal
DataOut	10101010101...	Packed Array	Internal

	Msgs
/SregFile_tb/DataIn	10101010101010...
/SregFile_tb/Addr	000
/SregFile_tb/regWE	1
/SregFile_tb/CLR	0
/SregFile_tb/clk	1
/SregFile_tb/DataOut	10101010101010...

Εικόνα του compilation report:



Flow Status	Successful - Thu Jan 13 14:53:28 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	SregFile
Top-level Entity Name	SregFile
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	N/A until Partition Merge
Total registers	N/A until Partition Merge
Total pins	N/A until Partition Merge
Total virtual pins	N/A until Partition Merge
Total memory bits	N/A until Partition Merge
Embedded Multiplier 9-bit elements	N/A until Partition Merge
Total PLLs	N/A until Partition Merge
UFM blocks	N/A until Partition Merge
ADC blocks	N/A until Partition Merge

Άσκηση 3 (με Behavioral)

δεν μπόρεσα να διορθώσω ένα error

Μέρος VI

Άσκηση 1

Ο κώδικας ROM.sv είναι:

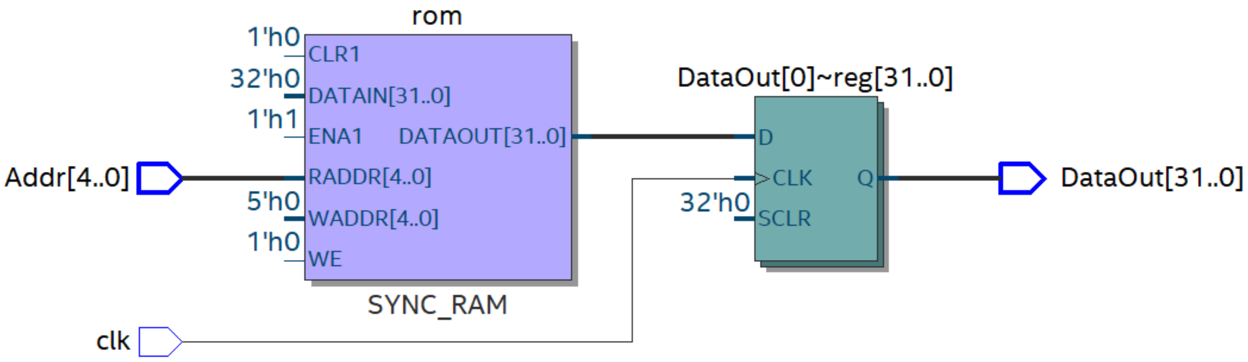
```
module ROM (input logic[4:0] Addr,
            input logic clk,
            output logic[31:0] DataOut);
  logic [31:0] rom [32];

  initial

  begin
    $readmemh("rom_init.mem", rom);
  end

  always @(posedge clk) begin
    DataOut = rom[Addr];
  end

endmodule
```

Ο κώδικας ROMFulltestbench.sv είναι:

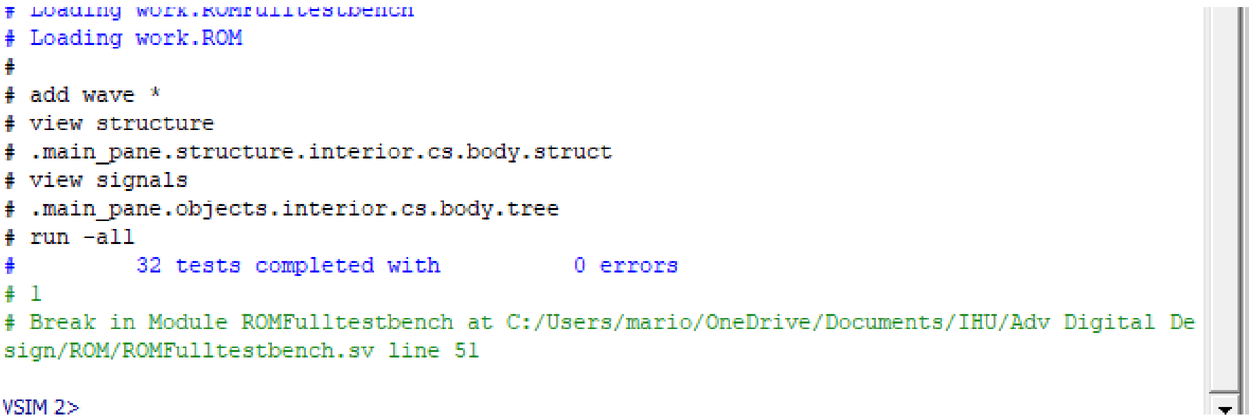
```
module ROMFulltestbench();

  logic [4:0]  Addr;
  logic        clk, reset;
  logic [31:0] DataOut, DataOutExpected;
  logic [31:0] vectornum, errors;
  logic [36:0] testvectors[10000:0];

  // instantiate device under test
 ROM dut(Addr, clk, DataOut);

  // generate clock
  always
    begin
      clk = 1; #5; clk = 0; #5;
    end

  // at start of test, load vectors
  // and pulse reset
  initial
    begin

      $readmemb("../../stimulus.tv", testvectors);
      vectornum = 0; errors = 0;
      reset = 0; #27; reset = 0; //optional Reset functionality
    end

  // apply test vectors on rising edge of clk
  always @(posedge clk)
    begin

          #10; {Addr, DataOutExpected} = testvectors[vectornum];

      end

  // check results on falling edge of clk
  always @(negedge clk)
    if (~reset) begin // skip during reset -
      if (DataOut !== DataOutExpected) begin  // check result
        $display("Error: inputs = %b", {Addr});
        $display("  outputs = %b (%b expected)",DataOut, DataOutExpected);
        errors = errors + 1;
      end

          vectornum = vectornum + 1;

          if (testvectors[vectornum] === 'hxxxxxxxx)
          begin
        $display("%d tests completed with %d errors", vectornum, errors);
            //$stop;
        $finish(0); //important to finish simulation
      end
    end
endmodule
```

Το stimulus.tv είναι:	Το rom_init.mem είναι:
00000_11100000010011110000000000001111	e04f000f
00001_11100010100000000010000000000101	e2802005
00010_11100010100000000011000000001100	e280300c
00011_11100010010000110111000000001001	e2437009
00100_11100001100001110100000000000010	e1874002
00101_11100000000000110101000000000100	e0035004
00110_11100000100001010101000000000100	e0855004
00111_11100000010101011000000000000111	e0558007
01000_00001010000000000000000000001100	a00000c
01001_11100000010100111000000000000100	e0538004
01010_10101010000000000000000000000000	aa000000
01011_11100010100000000101000000000000	e2805000
01100_11100000010101111000000000000010	e0578002
01101_10110010100001010111000000000001	b2857001
01110_11100000010001110111000000000010	e0477002
01111_11100101100000110111000001010100	e5837054
10000_11100101100100000010000001100000	e5902060
10001_11100000100011111111000000000000	e08ff000
10010_11100010100000000010000000000001	e2802001
10011_11101010000000000000000000000001	ea000001
10100_11100010100000000010000000000001	e2802001
10101_11100010100000000010000000000001	e2802001
10110_11100101100000000010000001010100	e5802054
10111_00000000000000000000000000000000	00000000
11000_00000000000000000000000000000000	00000000
11001_00000000000000000000000000000000	00000000
11010_00000000000000000000000000000000	00000000
11011_00000000000000000000000000000000	00000000
11100_00000000000000000000000000000000	00000000
11101_00000000000000000000000000000000	00000000
11110_00000000000000000000000000000000	00000000
11111_00000000000000000000000000000000	00000000

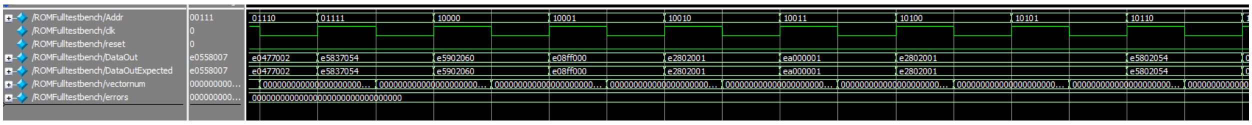
Η εικόνα από τον netlist viewer:



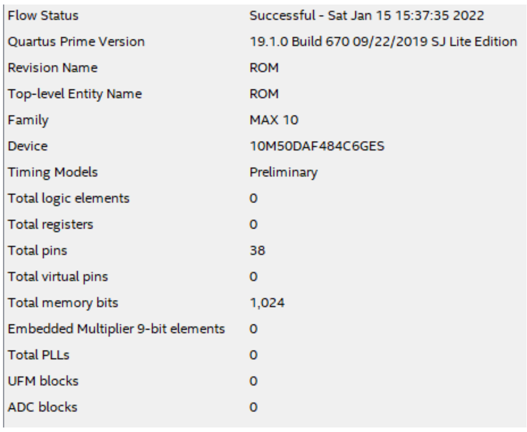
Η τελική εικόνα της προσομοίωσης:



```
# Loading work.ROMFulltestbench
# Loading work.ROM
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
#          32 tests completed with          0 errors
# 1
# Break in Module ROMFulltestbench at C:/Users/mario/OneDrive/Documents/IHU/Adv Digital De
sign/ROM/ROMFulltestbench.sv line 51

VSIM 2>
```



Εικόνα του compilation report:



Flow Status	Successful - Sat Jan 15 15:37:35 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	ROM
Top-level Entity Name	ROM
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	0
Total registers	0
Total pins	38
Total virtual pins	0
Total memory bits	1,024
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Άσκηση 2

Ο κώδικας PCROM.sv είναι:

```
module PCROM(input logic CLR, clk,
             output logic [31:0] DataOut);

  logic [4:0] counter;
  PC PCinst(CLR, clk, counter);
  ROM ROMinst(counter, clk, DataOut);

endmodule
```

Ο κώδικας PC.sv είναι:

```
module PC
           #(parameter width = 5)
            (input logic CLR, clk,
             output logic [width-1:0] Q);
  always_ff @(posedge clk) begin
  if (CLR)
    Q <= 0;
  else
    Q <= Q + 1;

  end
endmodule
```

Ο κώδικας ROM.sv είναι:

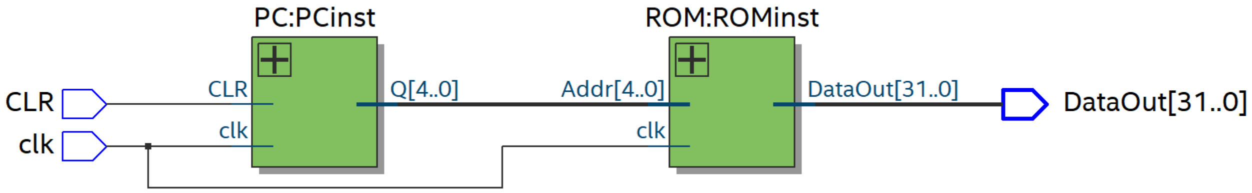
```
module ROM (input logic[4:0] Addr,
            input logic clk,
            output logic[31:0] DataOut);
  logic [31:0] rom [32];

  initial begin
    $readmemh("./command_mem.mem", rom);
  end

  always @(posedge clk) begin
    DataOut = rom[Addr];
  end

endmodule
```

Ο κώδικας PCROMFulltestbench.sv είναι:

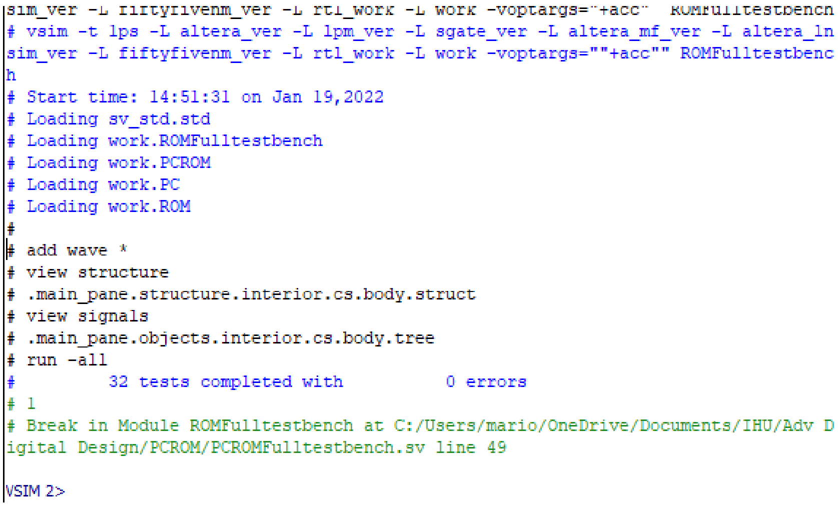
```
module PCROMFulltestbench();
  logic        clk, CLR;
  logic [31:0] DataOut, DataOutExpected;
  logic [31:0] vectornum, errors;
  logic [31:0] testvectors[10000:0];



  PCROM dut(CLR, clk, DataOut);



  always begin
    begin
      clk = 1; #5; clk = 0; #5;
    end
  end


  initial begin
    CLR = 1;
    $readmemb("../../stimulus.tv", testvectors);
    vectornum = 0; errors = 0;
  end
  always @(posedge clk) begin
    begin

      CLR = 0; {DataOutExpected} = testvectors[vectornum];
     end
  end



  always @(negedge clk) begin
    #10; begin // this delay is needed because in the first clock pulse we are clearning the PC
      if (DataOut !== DataOutExpected) begin
        $display("  outputs = %b (%b expected)",DataOut, DataOutExpected);
        errors = errors + 1;
      end

      vectornum = vectornum + 1;

       if (testvectors[vectornum] === 'hxxxxxxxx) begin
        $display("%d tests completed with %d errors", vectornum, errors);
        $finish(0);
      end
    end
  end

endmodule
```

stimulus.tv και command_mem.mem είναι ίδια με την προηγούμενη άσκηση.

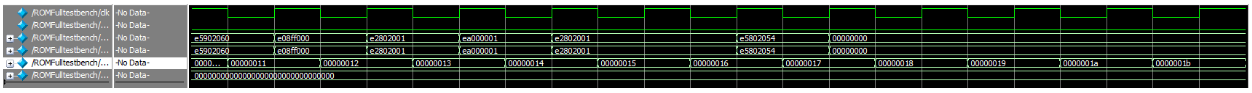
Η εικόνα από τον netlist viewer:



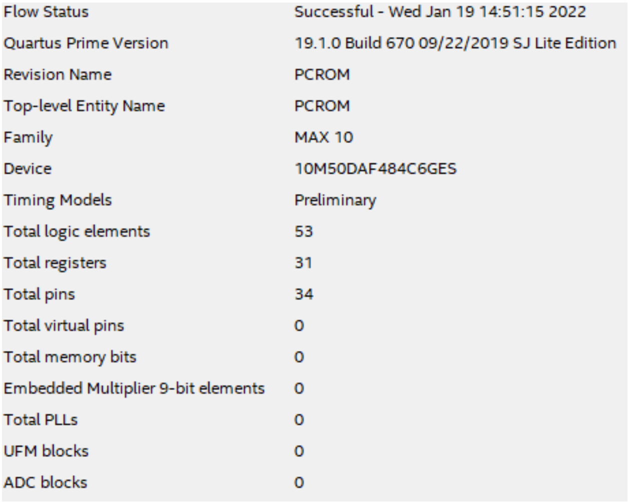
Η τελική εικόνα της προσομοίωσης:



```
sim_ver -L fiftyfivenm_ver -L rtl_work -L work -voptargs="+acc"  ROMFulltestbench
# vsim -t 1ps -L altera_ver -L lpm_ver -L sgate_ver -L altera_mf_ver -L altera_ln
sim_ver -L fiftyfivenm_ver -L rtl_work -L work -voptargs=""+acc"" ROMFulltestbenc
h
# Start time: 14:51:31 on Jan 19,2022
# Loading sv_std.std
# Loading work.ROMFulltestbench
# Loading work.PCROM
# Loading work.PC
# Loading work.ROM
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
#         32 tests completed with          0 errors
# 1
# Break in Module ROMFulltestbench at C:/Users/mario/OneDrive/Documents/IHU/Adv D
igital Design/PCROM/PCROMFulltestbench.sv line 49

VSIM 2>
```

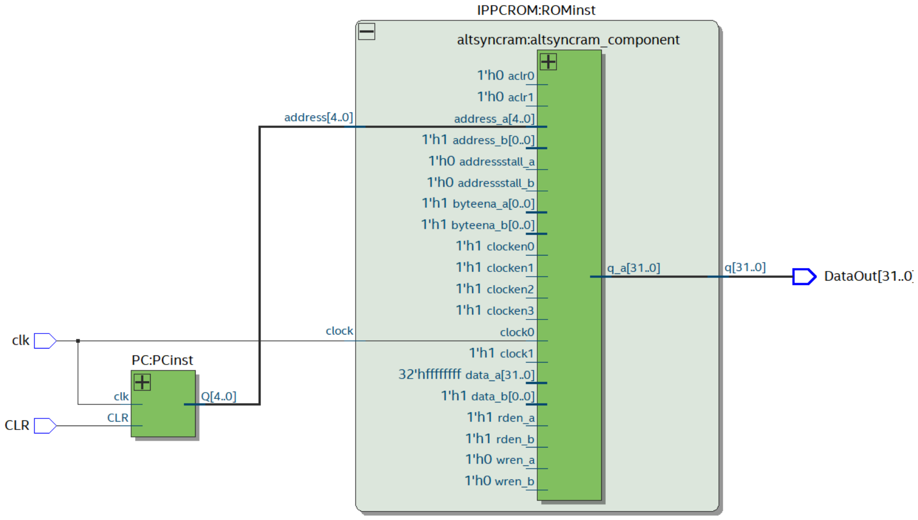


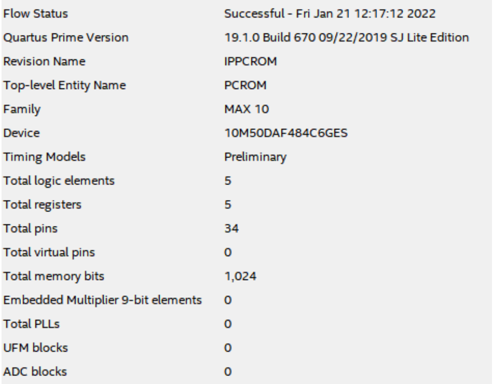
Εικόνα του compilation report:



Flow Status	Successful - Wed Jan 19 14:51:15 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	PCROM
Top-level Entity Name	PCROM
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	53
Total registers	31
Total pins	34
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

<u>Άσκηση 3 (</u>με Quartus IP Catalog (Altera Megafunctions)<u>)</u>



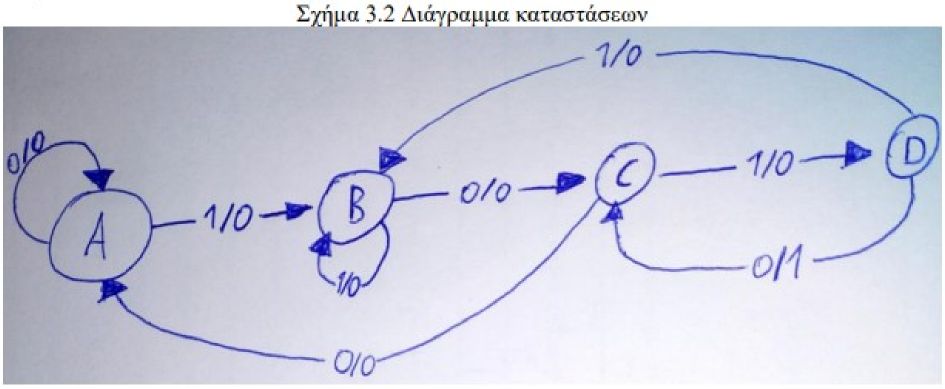


Flow Status	Successful - Fri Jan 21 12:17:12 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	IPPCROM
Top-level Entity Name	PCROM
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	5
Total registers	5
Total pins	34
Total virtual pins	0
Total memory bits	1,024
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Μέρος VII

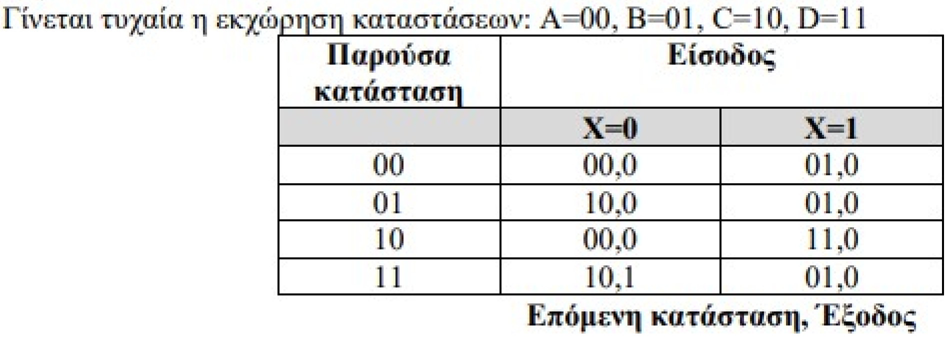
σύμφωνα με την ανάλυση που έχετε διδαχθεί στο μάθημα Ψηφιακή Σχεδίαση :

Σχήμα 3.2 Διάγραμμα καταστάσεων



Παρούσα κατάσταση	Είσοδος	
	X=0	X=1
A	A,0	B,0
B	C,0	B,0
C	A,0	D,0
D	C,1	B,0

Επόμενη κατάσταση, Έξοδος

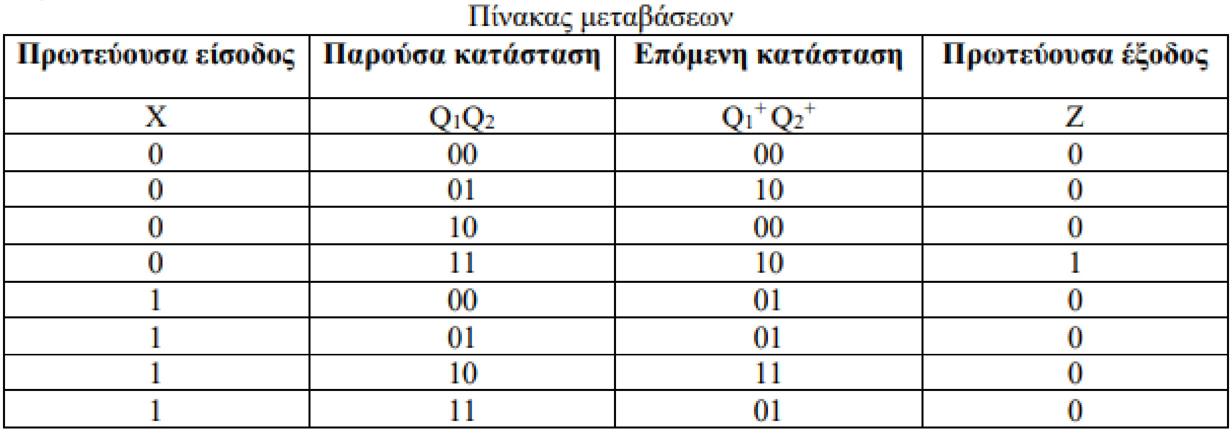


Γίνεται τυχαία η εκχώρηση καταστάσεων: A=00, B=01, C=10, D=11

Παρούσα κατάσταση	Είσοδος	
	X=0	X=1
00	00,0	01,0
01	10,0	01,0
10	00,0	11,0
11	10,1	01,0

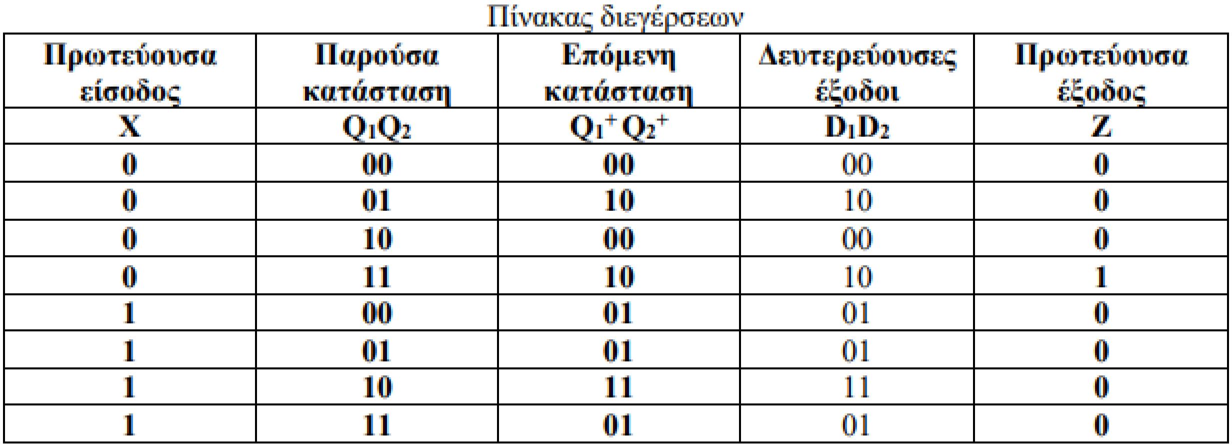
Επόμενη κατάσταση, Έξοδος

Πίνακας μεταβάσεων

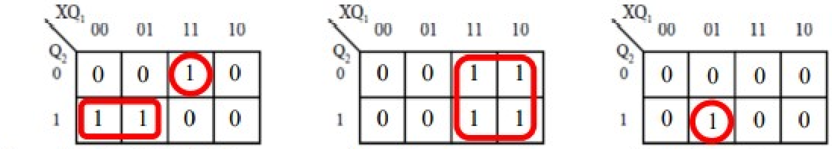


Πρωτεύουσα είσοδος	Παρούσα κατάσταση	Επόμενη κατάσταση	Πρωτεύουσα έξοδος
X	Q_1Q_2	$Q_1^+Q_2^+$	Z
0	00	00	0
0	01	10	0
0	10	00	0
0	11	10	1
1	00	01	0
1	01	01	0
1	10	11	0
1	11	01	0

Πίνακας διεγέρσεων

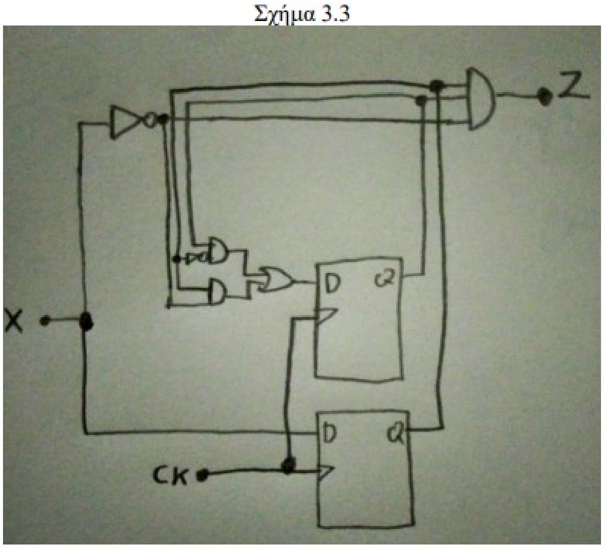


Πρωτεύουσα είσοδος	Παρούσα κατάσταση	Επόμενη κατάσταση	Δευτερεύουσες έξοδοι	Πρωτεύουσα έξοδος
X	**Q_1Q_2**	**$Q_1^+Q_2^+$**	**D_1D_2**	**Z**
0	**00**	**00**	00	**0**
0	**01**	**10**	10	**0**
0	**10**	**00**	00	**0**
0	**11**	**10**	10	**1**
1	**00**	**01**	01	**0**
1	**01**	**01**	01	**0**
1	**10**	**11**	11	**0**
1	**11**	**01**	01	**0**



$D_1 = X \cdot Q_1 \cdot \overline{Q_2} + \overline{X} \cdot Q_2$ $D_2 = X$ $Z = \overline{X} \cdot Q_1 \cdot Q_2$

Σχήμα 3.3



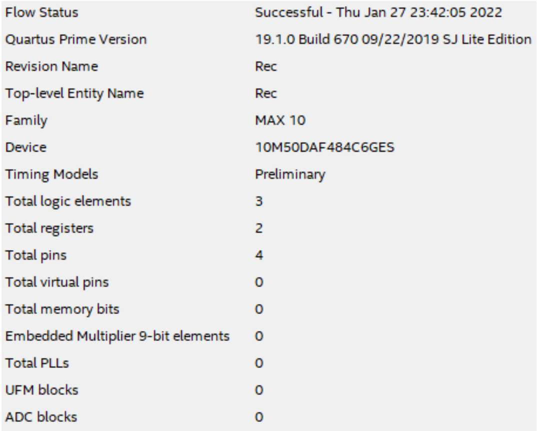
Ο κώδικας Rec.sv είναι:

```
module Rec(input logic CK,
           input logic RESET,
           input logic X,
           output logic Z);

  logic dff1out;
  logic dff2out;
  logic dff1in;

  assign dff1in =(dff1out && X && ~dff2out)||(dff2out && ~X);

  dffr dffr1(CK, RESET, dff1in, dff1out);
  dffr dffr2(CK, RESET, X, dff2out);

  assign Z = (dff2out && dff1out && ~X);
endmodule
```

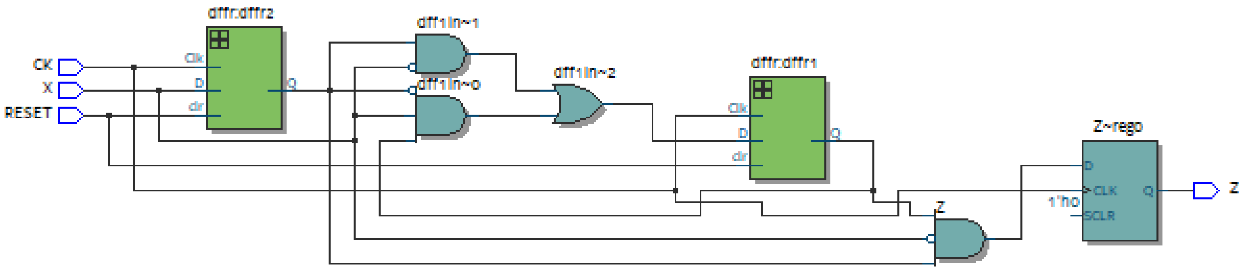
Ο κώδικας Rec.do είναι:

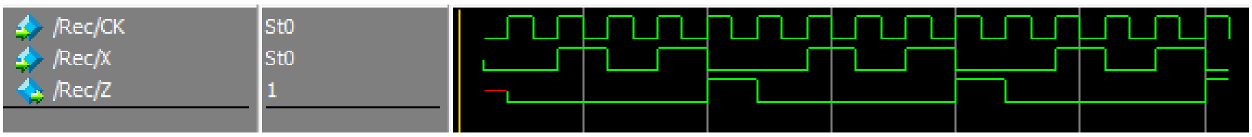
```
vsim Rec
force CK 0
force RESET 1
run 5
force CK 1
```

```
run 5
force CK 0
force RESET 0
add wave CK X Z
force -repeat 20 CK 0 0, 1 10
force -repeat 100 X 0 0, 1 30, 0 50, 1 70, 0 90
run 300
wave zoom full
```



Flow Status	Successful - Thu Jan 27 23:42:05 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	Rec
Top-level Entity Name	Rec
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	3
Total registers	2
Total pins	4
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

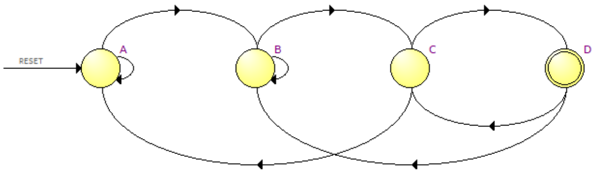


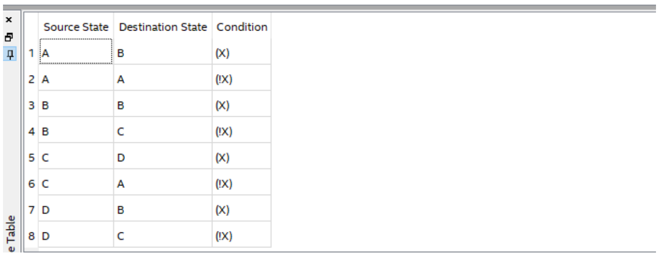


με Behavioral SystemVerilog:

Ο κώδικας Recbeh.sv είναι:

```
module Recbeh(input logic CK,
              input logic RESET,
              input logic X,
              output logic Z);


  typedef enum logic [1:0] {A, B, C, D} statetype;

  statetype state, nextstate;

  always_ff@(posedge CK, posedge RESET)
    if(RESET) state <= A;
    else      state <= nextstate;

  always_comb
    case(state)
      A : nextstate =  X ? B : state;
      B : nextstate = ~X ? C : state;
      C : nextstate =  X ? D : A;
      D : nextstate = ~X ? C : B;
    endcase

  always_ff @(posedge CK)
    Z = (dff2out && dff1out && ~X);

endmodule
```

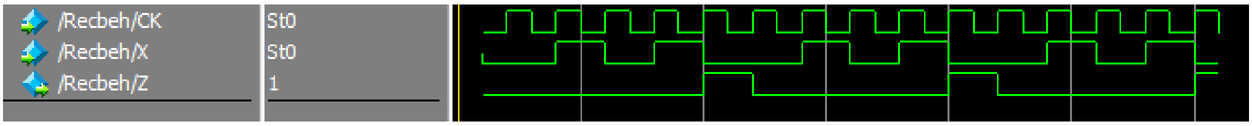


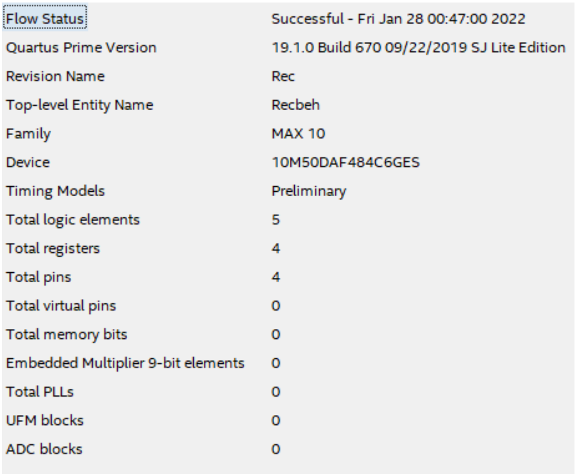


	Source State	Destination State	Condition
1	A	B	(X)
2	A	A	(!X)
3	B	B	(X)
4	B	C	(!X)
5	C	D	(X)
6	C	A	(!X)
7	D	B	(X)
8	D	C	(!X)

Ο κώδικας Recbeh.do είναι:

```
vsim Recbeh
force CK 0
force RESET 1
run 5
force CK 1
run 5
force CK 0
force RESET 0
add wave CK X Z
force -repeat 20 CK 0 0, 1 10
force -repeat 100 X 0 0, 1 30, 0 50, 1 70, 0 90
run 300
wave zoom full
```





Flow Status	Successful - Fri Jan 28 00:47:00 2022
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	Rec
Top-level Entity Name	Recbeh
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	5
Total registers	4
Total pins	4
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0