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

ΑΠΑΝΤΗΣΕΙΣ 1ης Σειράς Ασκήσεων

Μέρος Ι

Άσκηση 1

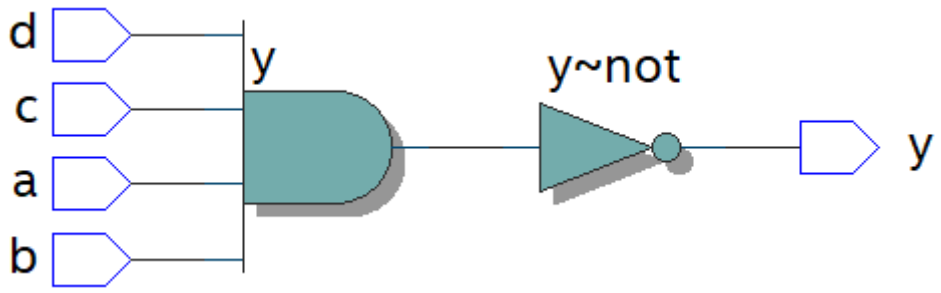
Ο κώδικας nand41.sv είναι:

```
module nand41(input logic a, b, c, d,
              output logic y);
    assign y = ~(a & b & c & d);
endmodule
```

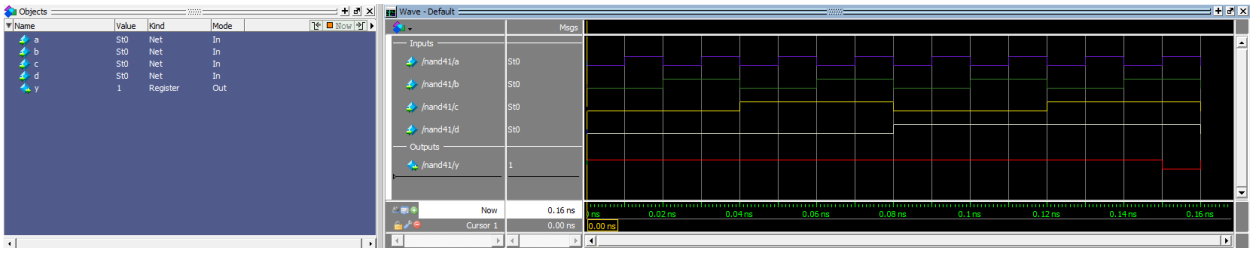
Ο κώδικας nand41.do είναι:

```
vsim nand41
add wave -divider Inputs
add wave -height 30 -color #6F1AD1 a -color #317523 b -color
#E6D706 c -color #DEE2CA d
force -repeat 20 a 0 0,1 10
force -repeat 40 b 0 0,1 20
force -repeat 80 c 0 0,1 40
force -repeat 160 d 0 0,1 80
add wave -divider Outputs
add wave -height 30 -color #FD0707 y
run 160
wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



Εικόνα του compilation report:

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sun Nov 07 17:55:01 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	nand41
Top-level Entity Name	nand41
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	1
Total registers	0
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

Άσκηση 2

Ο κώδικας CompFlags.sv είναι:

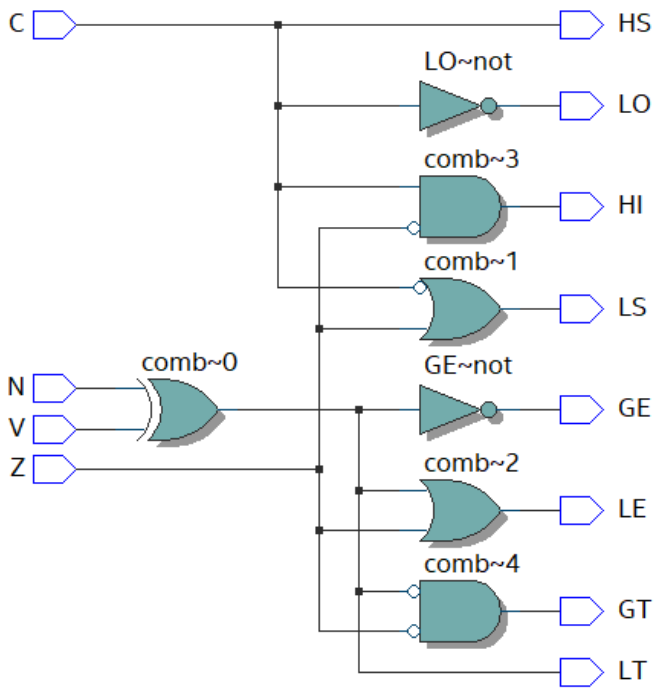
```
module CompFlags(input logic C, N, V, Z,
                 output logic HS, LS, HI, LO, GE, LE, GT, LT);

    logic NxorV;
    logic NxnorV;
    logic notC;
    logic notZ;

    xor(NxorV, N, V);
    not(NxnorV, NxorV);
    not(notC,C);
    not(notZ, Z);
```

```
    assign HS = C;              //HS
    not(LO, C);                 //LO
    and(HI, C, notZ);           //HI
    or(LS, notC, Z);            //LS
    not(GE, NxorV);             //GE
    or(LE, NxorV, Z);           //LE
    and(GT, NxnorV, notZ);      //GT
    assign LT = NxorV;          //LT

endmodule
```

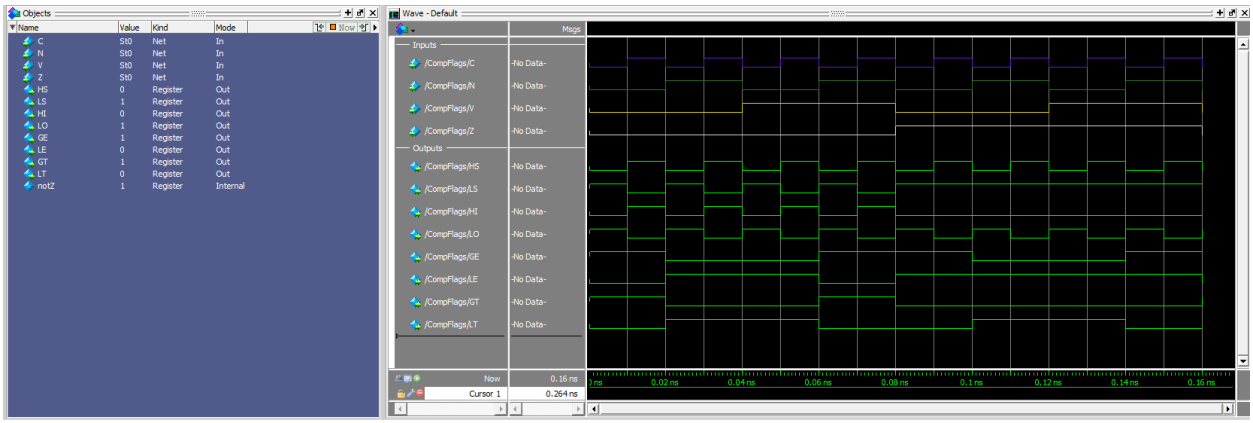
Ο κώδικας CompFlags.do είναι:

```
vsim CompFlags
add wave -divider Inputs
add wave -height 30 -color #6F1AD1 C -color #317523 N -color
#E6D706 V -color #DEE2CA Z
force -repeat 20 C 0 0,1 10
force -repeat 40 N 0 0,1 20
force -repeat 80 V 0 0,1 40
force -repeat 160 Z 0 0,1 80
add wave -divider Outputs
add wave -height 30 HS LS HI LO GE LE GT LT
run 160
wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



Εικόνα του compilation report:

Flow Summary

<<Filter>>

Flow Status	Successful - Sun Nov 07 19:55:45 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	CompFlags
Top-level Entity Name	CompFlags
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	3
Total registers	0
Total pins	12
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

Άσκηση 1

Περιγραφή με τον τελεστή υπό συνθήκη (?:) του πολυπλέκτη 2 σε 1 ενός bit (mux2.sv) με SystemVerilog:

```
module mux2(input logic x, y,
            input logic s,
            output logic m);
assign m = s ? y : x;
endmodule
```

Άσκηση 2

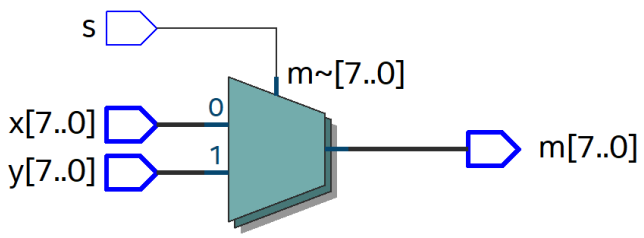
Ο κώδικας mux28bit.sv είναι:

```
module mux28bit(input logic [7:0] x, y,
                input logic s,
                output logic [7:0] m);

    assign m = s ? y : x;

endmodule
```

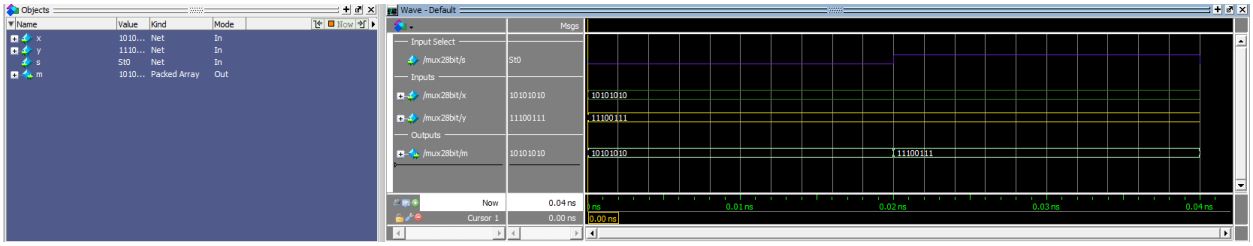
Ο κώδικας mux28bit.do είναι:

```
vsim mux28bit
add wave -divider "Input Select"
add wave -height 30 -color #6F1AD1 s
add wave -divider "Inputs"
add wave -height 30 -color #317523 x -color #E6D706 y
add wave -divider Outputs
add wave -height 30 m
force x 10101010
force y 11100111
force -repeat 40 s 0 0,1 20
run 40
wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



Εικόνα του compilation report:

Flow Summary

<<Filter>>

Flow Status	Successful - Thu Nov 11 18:08:02 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	mux28bit
Top-level Entity Name	mux28bit
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	8
Total registers	0
Total pins	25
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Μέρος ΙΙΙ

Ο κώδικας mux23.sv είναι:

```
module mux23(input logic [2:0] a, b,
             input logic s,
             output logic [2:0] m);

    assign m = s ? b : a;

endmodule
```

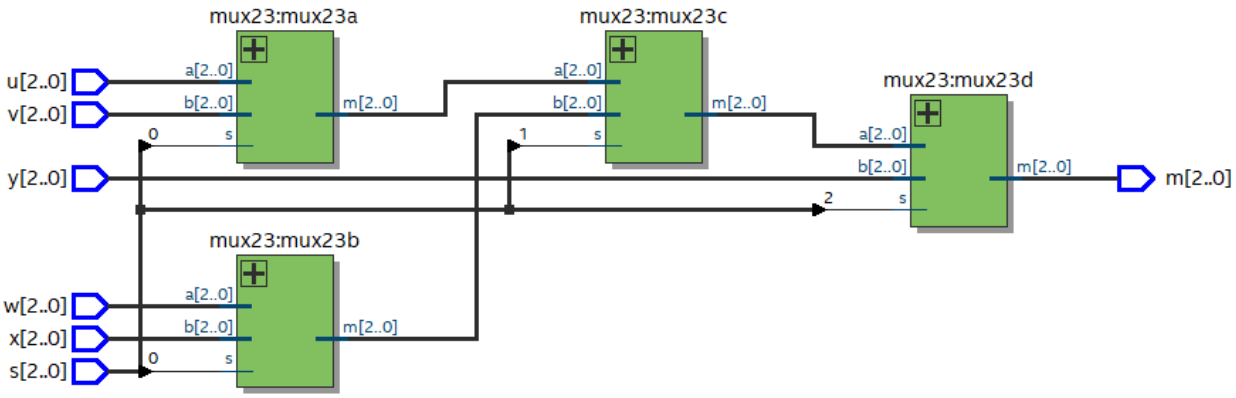
Ο κώδικας mux53.sv είναι:

```
module mux53(input logic [2:0] u, v, w, x, y,
             input logic [2:0] s,
             output logic [2:0] m);

    logic [2:0] w1;
    logic [2:0] w2;
    logic [2:0] w3;

    mux23 mux23a(u, v, s[0], w1);
    mux23 mux23b(w, x, s[0], w2);
    mux23 mux23c(w1, w2, s[1], w3);
    mux23 mux23d(w3, y, s[2], m);

endmodule
```

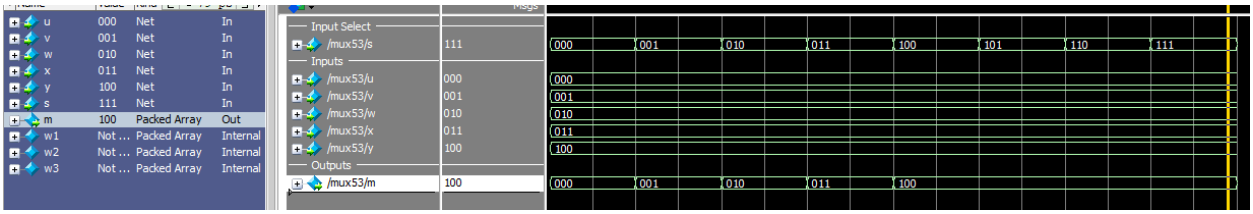
Ο κώδικας mux28bit.do είναι:

```
vsim mux53
add wave -divider "Input Select"
add wave s
add wave -divider "Inputs"
add wave u v w x y
add wave -divider "Outputs"
add wave m
force u 000
force v 001
force w 010
force x 011
force y 100
force -repeat 20 {s[0]} 0 0,1 10
force -repeat 40 {s[1]} 0 0,1 20
force -repeat 80 {s[2]} 0 0,1 40
run 80
wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



Εικόνα του compilation report:

Flow Status	Successful - Fri Nov 19 19:21:54 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	mux53
Top-level Entity Name	mux53
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	9
Total registers	0
Total pins	21
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

Άσκηση 1

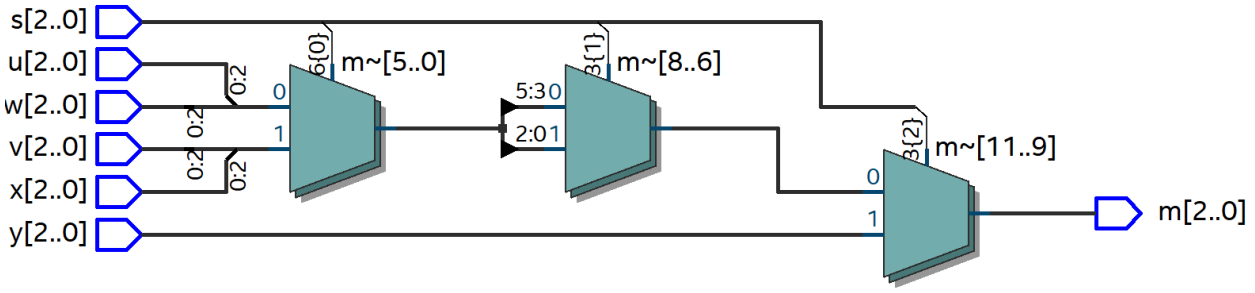
Ο κώδικας mux53beh.sv είναι:

```
module mux53beh(input logic [2:0] u, v, w, x, y,
                input logic [2:0] s,
                output logic [2:0] m);

always_comb

  if (s[2])
    m = y;
  else
    if (s[1])
      if (s[0])
        m = x;
      else
        m = w;
    else
      if (s[0])
        m = v;
      else
        m = u;

endmodule
```

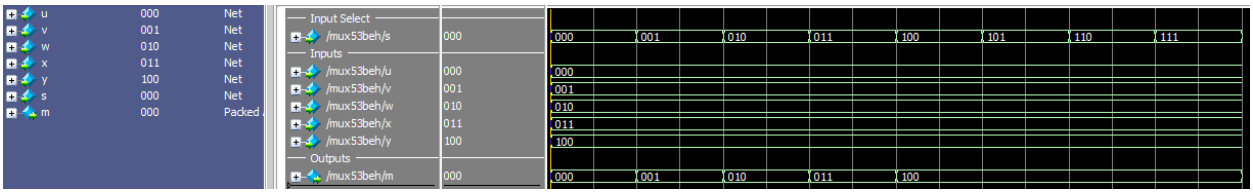
Ο κώδικας mux53beh.do είναι:

```
vsim mux53beh
add wave -divider "Input Select"
add wave s
add wave -divider "Inputs"
add wave u v w x y
add wave -divider "Outputs"
add wave m
force u 000
force v 001
force w 010
force x 011
force y 100
force -repeat 20 {s[0]} 0 0,1 10
force -repeat 40 {s[1]} 0 0,1 20
force -repeat 80 {s[2]} 0 0,1 40
run 80
wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



Εικόνα του compilation report:

Flow Status	Successful - Fri Nov 19 23:52:33 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	mux53beh
Top-level Entity Name	mux53beh
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	9
Total registers	0
Total pins	21
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 mux2.sv είναι:

```
module mux2(input logic x, y,
            input logic s,
            output logic m);
assign m = s ? y : x;
endmodule
```

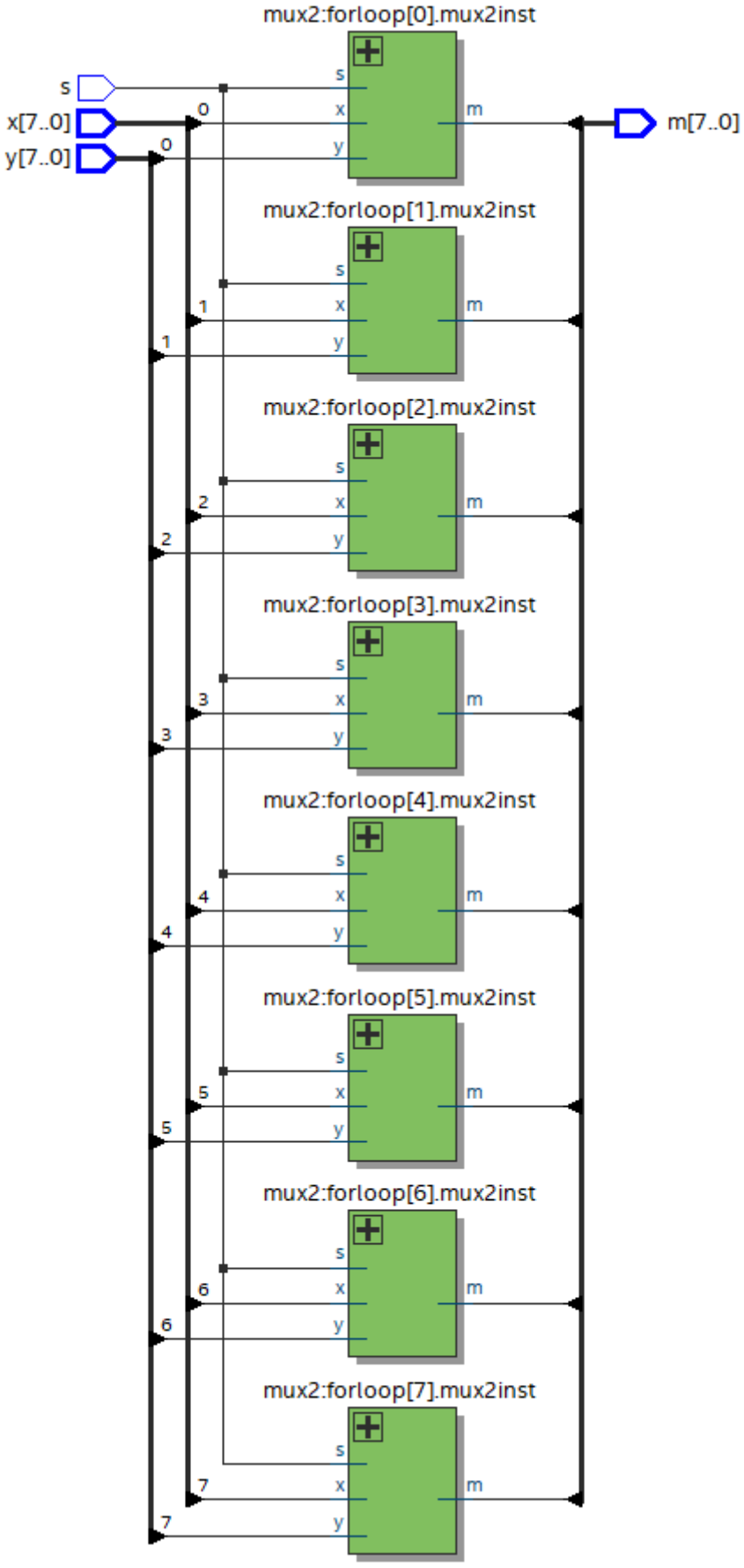
Ο κώδικας mux28fg.sv είναι:

```
module mux28fg
      # (parameter bits = 8)
        (input logic [bits-1:0] x, y,
         input s,
         output logic [bits-1:0] m);

  genvar i;
  generate
    for (i = 0; i < bits; i++) begin: forloop
      mux2 mux2inst(x[i], y[i], m[i]);
    end
  endgenerate

endmodule
```

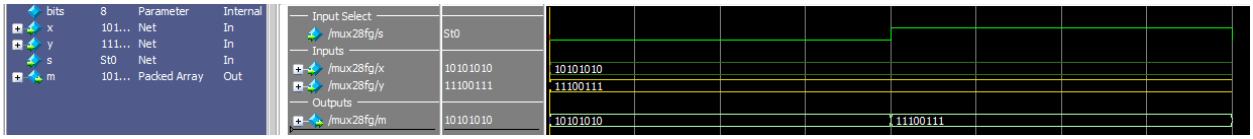
Ο κώδικας mux28fg.do είναι:

```
vsim mux28fg
add wave -divider "Input Select"
add wave s
add wave -divider "Inputs"
add wave -color #317523 x -color #E6D706 y
add wave -divider Outputs
add wave m
force x 10101010
force y 11100111
force -repeat 40 s 0 0,1 20
run 40
wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



Εικόνα του compilation report:

Flow Status	Successful - Fri Nov 19 22:37:09 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	mux28fg
Top-level Entity Name	mux28fg
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	8
Total registers	0
Total pins	25
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

Ο κώδικας prm.sv είναι:

```
module prm (input logic D, A, E, F,
            output logic [3:0] y);
  always_comb

    if      (D) y = 'b1000;
    else if (A) y = 'b0100;
    else if (E) y = 'b0010;
    else if (F) y = 'b0001;
    else        y = 'b0000; // Invalid

endmodule
```

Ο κώδικας sevensegprm.sv είναι:

```
module sevensegprm(input logic [3:0] y,
                   output logic [6:0] segments);
  always_comb
    case (y)
      //                            abc_defg
      4'b1000: segments =      7'b100_0010;  // d
      4'b0100: segments =      7'b000_1000;  // A
      4'b0010: segments =      7'b011_0000;  // E
      4'b0001: segments =      7'b011_1000;  // F
      default: segments =      7'b111_1111;  // Invalid
    endcase
endmodule
```

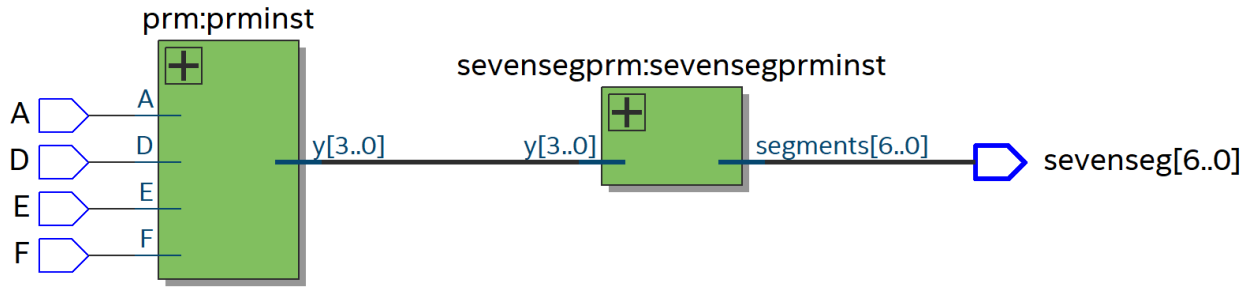
Ο κώδικας segprm.sv είναι:

```
module segprm(input logic D, A, E, F,
              output logic [6:0] sevenseg);
  logic [3:0] wy;

  prm prminst(D, A, E, F, wy);
  sevensegprm sevensegprminst(wy, sevenseg);

endmodule
```

Ο κώδικας segprm.do είναι:

```
vsim segprm
add wave -divider "Inputs"
add wave D A E F
add wave -divider "Segment Display"
add wave sevenseg
force -repeat 40 F 0 0,1 20
force -repeat 80 E 0 0,1 40
force -repeat 160 A 0 0,1 80
force -repeat 320 D 0 0,1 160
run 320
wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:

Εικόνα του compilation report:

Flow Status	Successful - Sat Nov 20 15:53:22 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	segprm
Top-level Entity Name	segprm
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	4
Total registers	0
Total pins	11
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

παραλλαγή της case για την ελαχιστοποίηση του συστήματος:

```
module sevensegprm(input logic [3:0] y,
                   output logic [6:0] segments);
  always_comb
    casez (y)
      //                              abc_defg
      4'b1???: segments =      7'b100_0010;  // d
      4'b?1??: segments =      7'b000_1000;  // A
      4'b??1?: segments =      7'b011_0000;  // E
      4'b???1: segments =      7'b011_1000;  // F
      default: segments =      7'b111_1111;  // Invalid

    endcase
endmodule
```

Μέρος VI

Ο κώδικας fa1bit.sv είναι:

```
module fa1bit(input logic a, b, cin,
              output logic s, cout);

  logic w;
  assign w = a ^ b;

  assign s = w ^ cin;
  assign cout = w ? cin : b;
endmodule
```

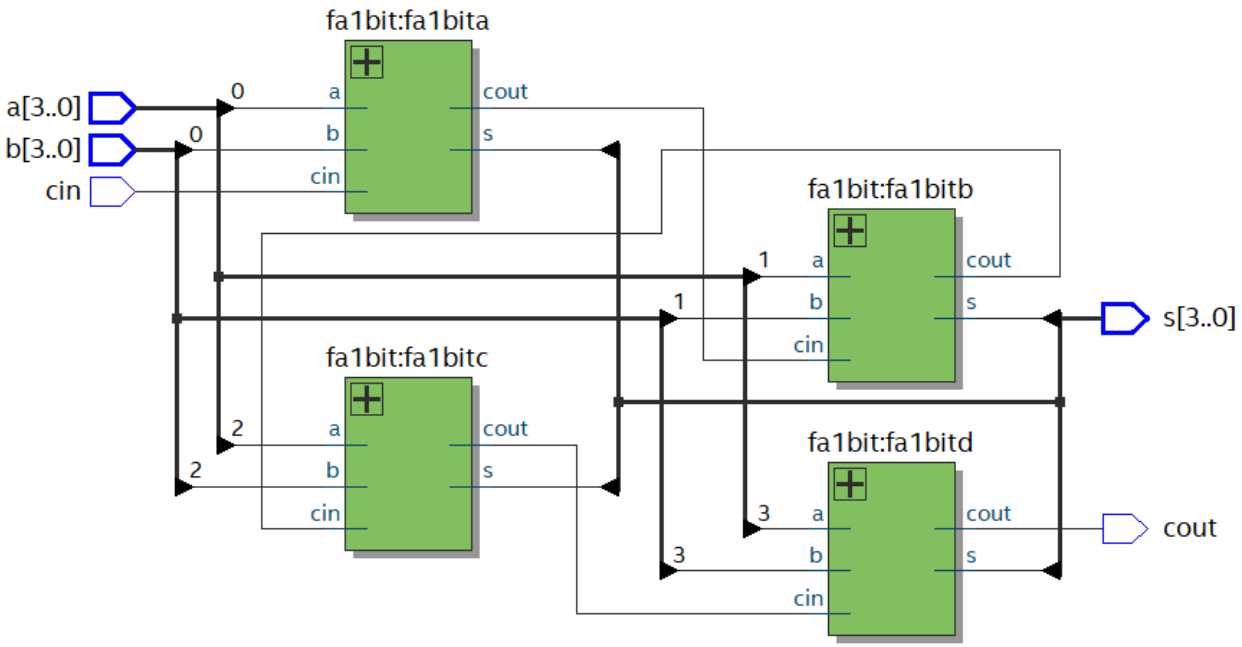
Ο κώδικας farc4bit.sv είναι:

```
module farc4bit(input logic [3:0] a, b,
                input logic cin,
                output logic [3:0] s,
                output logic cout);

  logic c1, c2, c3;

  fa1bit fa1bita(a[0], b[0], cin, s[0], c1);
  fa1bit fa1bitb(a[1], b[1], c1, s[1], c2);
  fa1bit fa1bitc(a[2], b[2], c2, s[2], c3);
  fa1bit fa1bitd(a[3], b[3], c3, s[3], cout);

endmodule
```

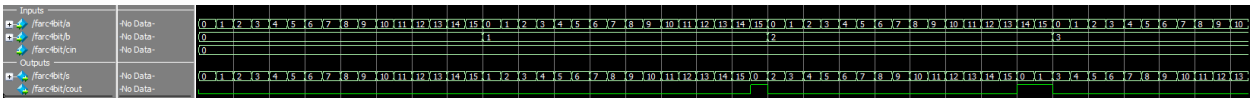
Ο κώδικας farc4bit.do είναι:

```
vsim farc4bit
add wave -divider "Inputs"
add wave a b cin
add wave -divider "Outputs"
add wave s cout
force -repeat 20 {a[0]} 0 0,1 10
force -repeat 40 {a[1]} 0 0,1 20
force -repeat 80 {a[2]} 0 0,1 40
force -repeat 160 {a[3]} 0 0,1 80
force -repeat 320 {b[0]} 0 0,1 160
force -repeat 640 {b[1]} 0 0,1 320
force -repeat 1280 {b[2]} 0 0,1 640
force -repeat 2560 {b[3]} 0 0,1 1280
force -repeat 5120 cin 0 0,1 2560
run 5120
radix signal a unsigned
radix signal b unsigned
radix signal s unsigned
```

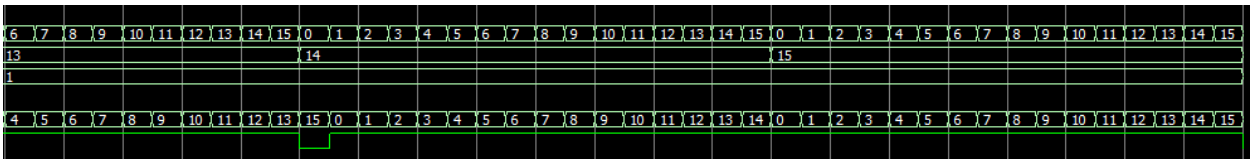
Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



στο ModelSim μπορούμε να δούμε ότι και οι υπόλοιπες πράξεις είναι σωστές...



Εικόνα του compilation report:

Flow Status	Successful - Sat Nov 20 19:43:31 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	farc4bit
Top-level Entity Name	farc4bit
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	9
Total registers	0
Total pins	14
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Μέρος VII

Ο κώδικας alunbit.sv είναι:

```
module alunbit
    #(parameter Nbits = 8)
     (input logic [Nbits-1:0] A, B,
      input logic [1:0] ALUControl,
      output logic [Nbits-1:0] Result,
      output logic [3:0] NZCV);

  logic [Nbits-1:0] NotResult;
  logic [Nbits-1:0] condinvb;
  logic [Nbits-1:0] sum;
  logic cout;
  logic wv;
  logic wc;
  logic wz;
  logic wn;
  assign condinvb = ALUControl[0] ? ~B : B;
  assign {cout, sum} = A + condinvb + ALUControl[0];
  always_comb begin
    casex (ALUControl[1:0])
      2'b0?: Result = sum;
      2'b10: Result = A & B;
      2'b11: Result = A | B;
    endcase

    NotResult = ~Result;
    wn = Result[Nbits-1];
    wz = &NotResult;
    wc = (cout & (~ALUControl[1]));
    wv = ~ALUControl[1] & (A[Nbits-1] ^ sum[Nbits-1]) & ~(ALUControl[0] ^ A[Nbits-1] ^ B[Nbits-1]);

    NZCV = {wn, wz, wc, wv};
  end

endmodule
```

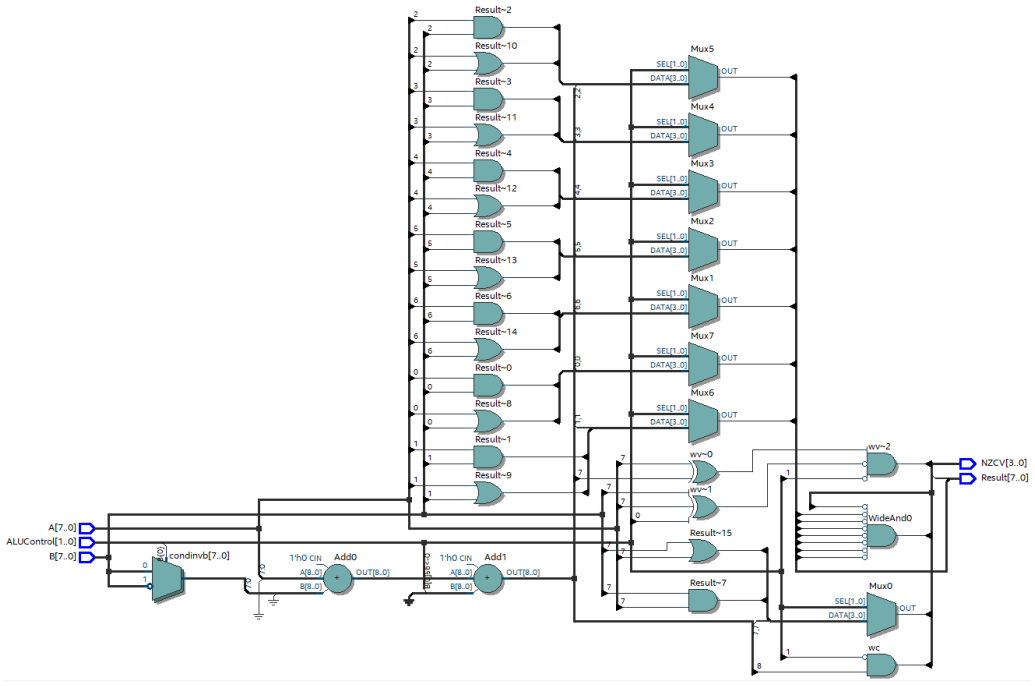
Ο κώδικας alunbit.do είναι:

```
vsim alunbit
add wave -divider "ALUControl"
add wave -hex ALUControl
add wave -divider "Inputs"
add wave -hex A -hex B
add wave -divider "Result"
add wave -hex Result
add wave -divider "ALUFlags"
add wave -hex NZCV
```

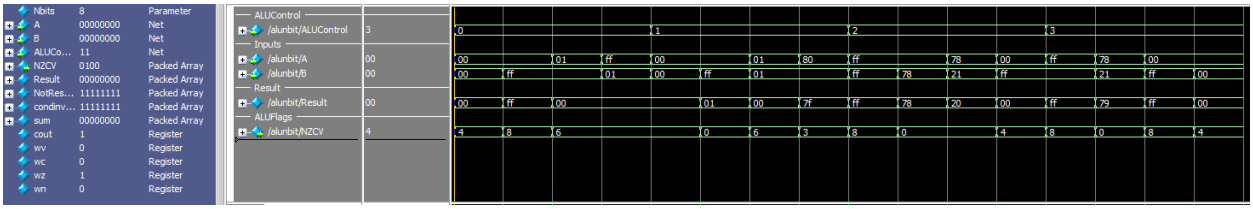
```
#   ADD 0+0
force ALUControl 'h0
force A 'h00
force B 'h00
run 10

#   ADD 0+(-1)
force B 'hFF
run 10

#   ADD 1+(-1)
force A 'h01
run 10

#   ADD FF+1
force A 'hFF
force B 'h01
run 10

#   SUB 0-0
force ALUControl 'h1
force A 'h00
force B 'h00
run 10

#   SUB 0-(-1)
force B 'hFF
run 10

#   SUB 1-1
force A 'h01
force B 'h01
run 10

#   SUB 80-1
force A 'h80
run 10

#   AND FF, FF
force A 'hFF
force B 'hFF
force ALUControl 'h2
run 10

#   AND FF, 78
force B 'h78
run 10

#   AND 78, 21
force A 'h78
force B 'h21
```

```
run 10

#   AND 00, FF
force A 'h00
force B 'hFF
run 10

#   OR FF, FF
force ALUControl 'h3
force A 'hFF
run 10

#   OR 78, 21
force A 'h78
force B 'h21
run 10

#   OR 00, FF
force B 'hFF
force A 'h00
run 10

#   OR 00, 00
force B 'h00
run 10

wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



Test	ALUControl[1:0]	A	B	Result	ALUFlags (NZCV)
ADD 0+0	0	00	00	00	4
ADD 0+(-1)	0	00	FF	FF	8
ADD 1+(-1)	0	01	FF	00	6
ADD FF+1	0	FF	01	**00**	**6**
SUB 0-0	1	00	00	00	6
SUB 0-(-1)	**1**	00	FF	01	0
SUB 1-1	**1**	01	**01**	**00**	**6**
SUB 80-1	**1**	80	**01**	**7F**	**3**
AND FF, FF	**2**	FF	**FF**	**FF**	**8**
AND FF, 78	**2**	FF	78	78	0
AND 78, 21	**2**	78	**21**	**20**	**0**
AND 00, FF	**2**	00	**FF**	**00**	**4**
OR FF, FF	**3**	FF	**FF**	**FF**	**8**
OR 78, 21	**3**	78	**21**	**79**	**0**
OR 00, FF	**3**	00	**FF**	**FF**	**8**
OR 00, 00	**3**	00	**00**	**00**	**4**

Εικόνα του compilation report:

Flow Status	Successful - Fri Dec 03 17:14:17 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	alunbit
Top-level Entity Name	alunbit
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	39
Total registers	0
Total pins	30
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Μέρος VIII

Ο κώδικας Barrel8bit.sv είναι:

```
module Barrel8bit(input [7:0] a,
                  input [2:0] amt,
                  input [1:0] opbarrel,
                  output [7:0] rmout);

  logic [7:0] LSL, LSR, ASR, ROR;

  shifter shifterinst(a, amt, LSL, LSR, ASR, ROR);
  mux4 mux4inst(LSL, LSR, ASR, ROR, opbarrel, rmout);

endmodule
```

Ο κώδικας shifter.sv είναι:

```
module shifter(input logic [7:0] a,
               input logic [2:0] amt,
               output logic [7:0] LSL, LSR, ASR, ROR);


  always_comb begin

    // Logical shift left LSL
    case(amt)
      0: LSL = a;
      1: LSL = {a[6], a[5], a[4], a[3], a[2], a[1], a[0], 1'b0};
      2: LSL = {a[5], a[4], a[3], a[2], a[1], a[0], 1'b0, 1'b0};
      3: LSL = {a[4], a[3], a[2], a[1], a[0], 1'b0, 1'b0, 1'b0};
      4: LSL = {a[3], a[2], a[1], a[0], 1'b0, 1'b0, 1'b0, 1'b0};
      5: LSL = {a[2], a[1], a[0], 1'b0, 1'b0, 1'b0, 1'b0, 1'b0};
      6: LSL = {a[1], a[0], 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0};
      7: LSL = {a[0], 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0};
      default: LSL = 8'bxxxxxxxx;
    endcase

    // Logical shift right LSR
    case(amt)
      0: LSR = a;
      1: LSR = {1'b0, a[7], a[6], a[5], a[4], a[3], a[2], a[1]};
      2: LSR = {1'b0, 1'b0, a[7], a[6], a[5], a[4], a[3], a[2]};
      3: LSR = {1'b0, 1'b0, 1'b0, a[7], a[6], a[5], a[4], a[3]};
      4: LSR = {1'b0, 1'b0, 1'b0, 1'b0, a[7], a[6], a[5], a[4]};
      5: LSR = {1'b0, 1'b0, 1'b0, 1'b0, 1'b0, a[7], a[6], a[5]};
      6: LSR = {1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, a[7], a[6]};
      7: LSR = {1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, a[7]};
      default: LSR = 8'bxxxxxxxx;
    endcase
```

```
    // Arithmetic Shift Right ASR
    case(amt)
      0: ASR = a;
      1: ASR = {a[7], a[7], a[6], a[5], a[4], a[3], a[2], a[1]};
      2: ASR = {a[7], a[7], a[7], a[6], a[5], a[4], a[3], a[2]};
      3: ASR = {a[7], a[7], a[7], a[7], a[6], a[5], a[4], a[3]};
      4: ASR = {a[7], a[7], a[7], a[7], a[7], a[6], a[5], a[4]};
      5: ASR = {a[7], a[7], a[7], a[7], a[7], a[7], a[6], a[5]};
      6: ASR = {a[7], a[7], a[7], a[7], a[7], a[7], a[7], a[6]};
      7: ASR = {a[7], a[7], a[7], a[7], a[7], a[7], a[7], a[7]};
      default: ASR = 8'bxxxxxxxx;
    endcase
    // Circulat rotation right ROR
    case(amt)
     0: ROR = a;
      1: ROR = {a[0], a[7], a[6], a[5], a[4], a[3], a[2], a[1]};
      2: ROR = {a[1], a[0], a[7], a[6], a[5], a[4], a[3], a[2]};
      3: ROR = {a[2], a[1], a[0], a[7], a[6], a[5], a[4], a[3]};
      4: ROR = {a[3], a[2], a[1], a[0], a[7], a[6], a[5], a[4]};
      5: ROR = {a[4], a[3], a[2], a[1], a[0], a[7], a[6], a[5]};
      6: ROR = {a[5], a[4], a[3], a[2], a[1], a[0], a[7], a[6]};
      7: ROR = {a[6], a[5], a[4], a[3], a[2], a[1], a[0], a[7]};
      default: ROR = 8'bxxxxxxxx;
    endcase
  end

endmodule
```

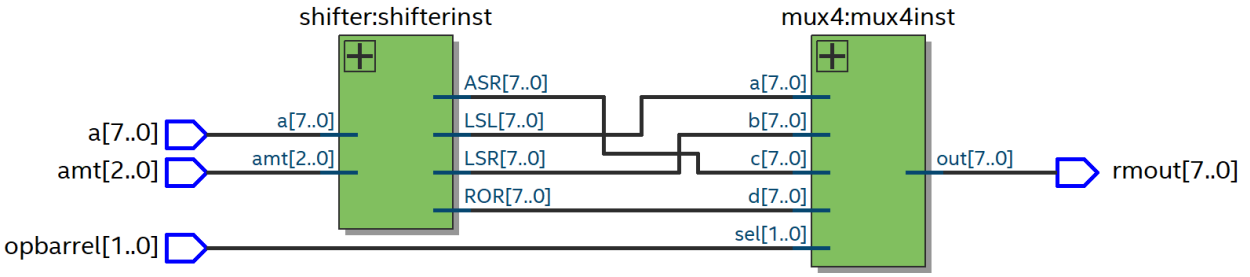
Ο κώδικας mux4.sv είναι:

```
module mux4(input logic [7:0] a, b, c, d,
            input logic [1:0] sel,
            output logic [7:0] out);

  assign out = sel[1] ? (sel[0] ? d : c ) : (sel[0] ? b : a);

endmodule
```

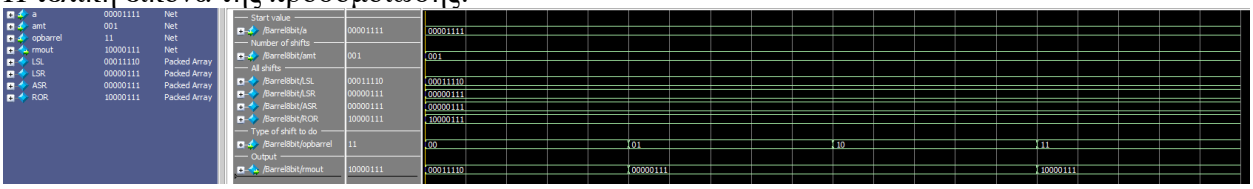
Ο κώδικας Barrel8bit.do είναι:

```
vsim Barrel8bit
force a 8'b00001111
add wave -divider "Start value"
add wave a
add wave -divider "Number of shifts"
add wave amt
add wave -divider "All shifts"
add wave LSL LSR ASR ROR
add wave -divider "Type of shift to do"
add wave opbarrel
add wave -divider "Output"
add wave rmout
force amt 3'b001
force opbarrel 00
run 50
force opbarrel 01
run 50
force opbarrel 10
run 50
force opbarrel 11
run 50
wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



Εικόνα του compilation report:

Flow Status	Successful - Sun Nov 21 20:16:34 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	Barrel8bit
Top-level Entity Name	Barrel8bit
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	82
Total registers	0
Total pins	21
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Μέρος IX

Ο κώδικας ALUBarrel.sv είναι:

```
module ALUBarrel(input logic [7:0] Rn, Rm,
                 input logic [2:0] amt,
                 input logic [1:0] opbarrel,
                 input logic [1:0] ALUControl,
                 output logic [7:0] Rd,
                 output logic N, Z, V, C,
                 output logic HS, LS, HI, LO, GE, LE, GT, LT);

  logic [7:0] w1;
  logic [3:0] NZCV;
  Barrel8bit Barrel8bitinst(Rm, amt, opbarrel, w1);
  alunbit alunbitinst(Rn, w1, ALUControl, Rd, NZCV);
  assign V = NZCV[0];
  assign C = NZCV[1];
  assign Z = NZCV[2];
  assign N = NZCV[3];
  CompFlags CompFlagsinst(C, N, V, Z, HS, LS, HI, LO, GE, LE, GT, LT);

endmodule
```

Ο κώδικας alunbit.sv είναι:
Ίδιος με τον alunbit στο Μέρος VII

Ο κώδικας Barrel8bit.sv είναι:
Ίδιος με τον Barrel8bit στο Μέρος VIII

Ο κώδικας shifter.sv είναι:
Ίδιος με τον shifter στο Μέρος VIII

Ο κώδικας mux4.sv είναι:
Ίδιος με τον mux4 στο Μέρος VIII

Ο κώδικας CompFlags.sv είναι:
Ίδιος με τον CompFlags στο Μέρος I Άσκηση_2

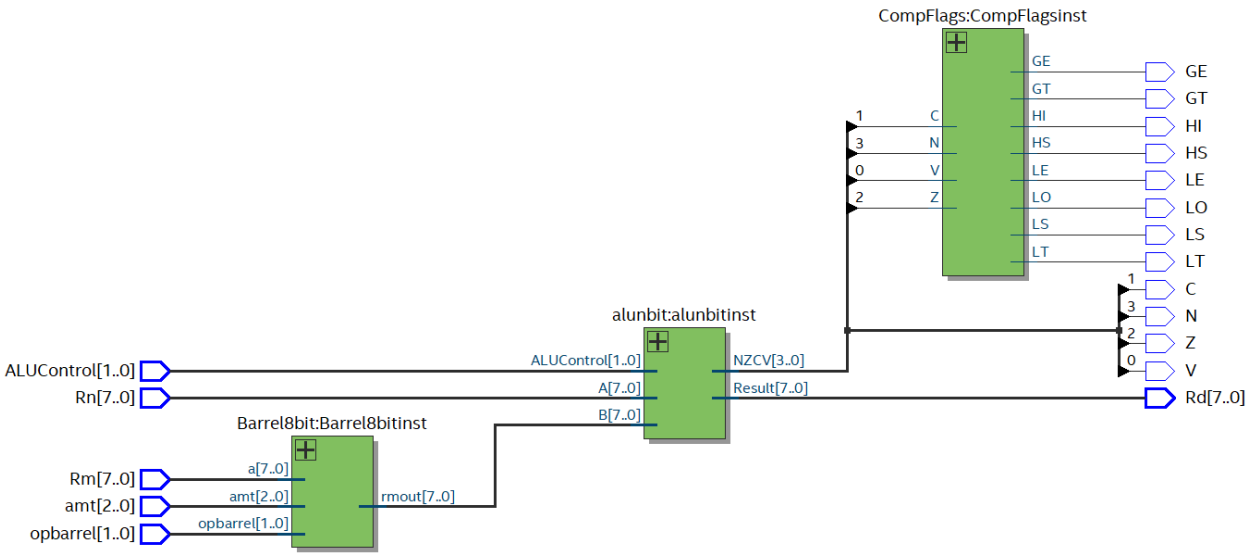
Ο κώδικας ALUBarrel.do είναι:

```
vsim ALUBarrel
add wave -divider "Barrel inputs"
add wave -radix hex opbarrel amt
add wave -divider "Inputs"
add wave -radix hex Rn Rm ALUControl
add wave -divider "Outputs"
add wave -radix hex Rd
virtual signal { N & Z & V & C } NZVC_Concatenated
add wave -radix hex NZVC_Concatenated
add wave -group "Condition Codes" -format literal HS LS HI LO GE LE GT LT
force opbarrel 'b00
force amt 'b000
```

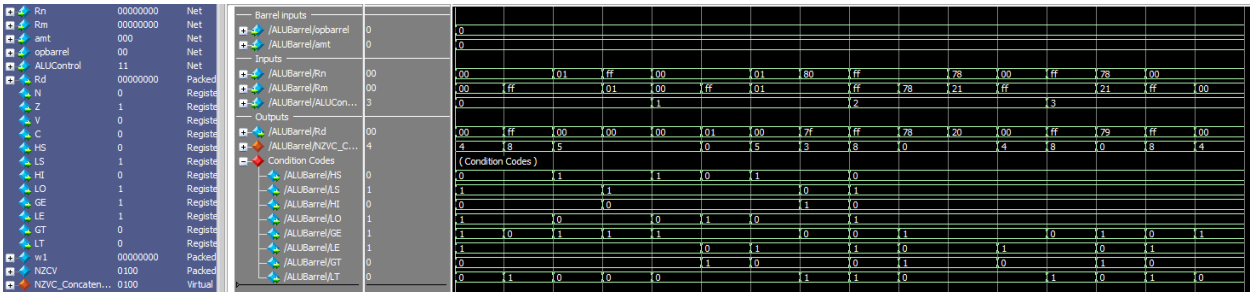
```
#   ADD 0+0
force ALUControl 'h0
force Rn 'h00
force Rm 'h00
run 10

#   ADD 0+(-1)
force Rm 'hFF
run 10

#   ADD 1+(-1)
force Rn 'h01
run 10

#   ADD FF+1
force Rn 'hFF
force Rm 'h01
run 10

#   SUB 0-0
force ALUControl 'h1
force Rn 'h00
force Rm 'h00
run 10

#   SUB 0-(-1)
force Rm 'hFF
run 10

#   SUB 1-1
force Rn 'h01
force Rm 'h01
run 10

#   SUB 80-1
force Rn 'h80
run 10

#   AND FF, FF
force Rn 'hFF
force Rm 'hFF
force ALUControl 'h2
run 10

#   AND FF, 78
force Rm 'h78
run 10

#   AND 78, 21
force Rn 'h78
force Rm 'h21
```

```
run 10

#   AND 00, FF
force Rn 'h00
force Rm 'hFF
run 10

#   OR FF, FF
force ALUControl 'h3
force Rn 'hFF
run 10

#   OR 78, 21
force Rn 'h78
force Rm 'h21
run 10

#   OR 00, FF
force Rm 'hFF
force Rn 'h00
run 10

#   OR 00, 00
force Rm 'h00
run 10

wave zoom full
```

Η εικόνα από τον netlist viewer:



Η τελική εικόνα της προσομοίωσης:



Test	ALUControl[1:0]	Rn	Rm	Result	ALUFlags (NZVC)*	HS	LS	HI	LO	GE	LE	GT	LT
ADD 0+0	0	00	00	00	4	0	1	0	1	1	1	0	0
ADD 0+(-1)	0	00	FF	FF	8	0	1	0	1	0	1	0	1
ADD 1+(-1)	0	01	FF	00	5	1	1	0	0	1	1	0	0
ADD FF+1	0	FF	01	00	5	1	1	0	0	1	1	0	0
SUB 0-0	1	00	00	00	5	1	1	0	0	1	1	0	0
SUB 0-(-1)	1	00	FF	01	0	0	1	0	1	1	0	1	0
SUB 1-1	1	01	01	00	5	1	1	0	0	1	1	0	0
SUB 80-1	1	80	01	7F	3	1	0	1	0	0	1	0	1
AND FF, FF	2	FF	FF	FF	8	0	1	0	1	0	1	0	1
AND FF, 78	2	FF	78	78	0	0	1	0	1	1	0	1	0
AND 78, 21	2	78	21	20	0	0	1	0	1	1	0	1	0
AND 00, FF	2	00	FF	00	4	0	1	0	1	1	1	0	0
OR FF, FF	3	FF	FF	FF	8	0	1	0	1	0	1	0	1
OR 78, 21	3	78	21	79	0	0	1	0	1	1	0	1	0
OR 00, FF	3	00	FF	FF	8	0	1	0	1	0	1	0	1
OR 00, 00	3	00	00	00	4	0	1	0	1	1	1	0	0

*(αυτός ο πίνακας έχει με άλλη σειρά τα flags σε σύγκριση με τον πινάκα 1 από το Μέρος VII)

Εικόνα του compilation report:

Flow Status	Successful - Fri Dec 03 18:11:38 2021
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	ALUBarrel
Top-level Entity Name	ALUBarrel
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	124
Total registers	0
Total pins	43
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0