

4^η Εργαστηριακή Άσκηση Σχεδιασμός Συστημάτων VLSI

Κόλλιας Ιωάννης 1084578

Ποδηματά Θεοδώρα Παναγιώτα 1090031

4^ο Έτος 2023-2024

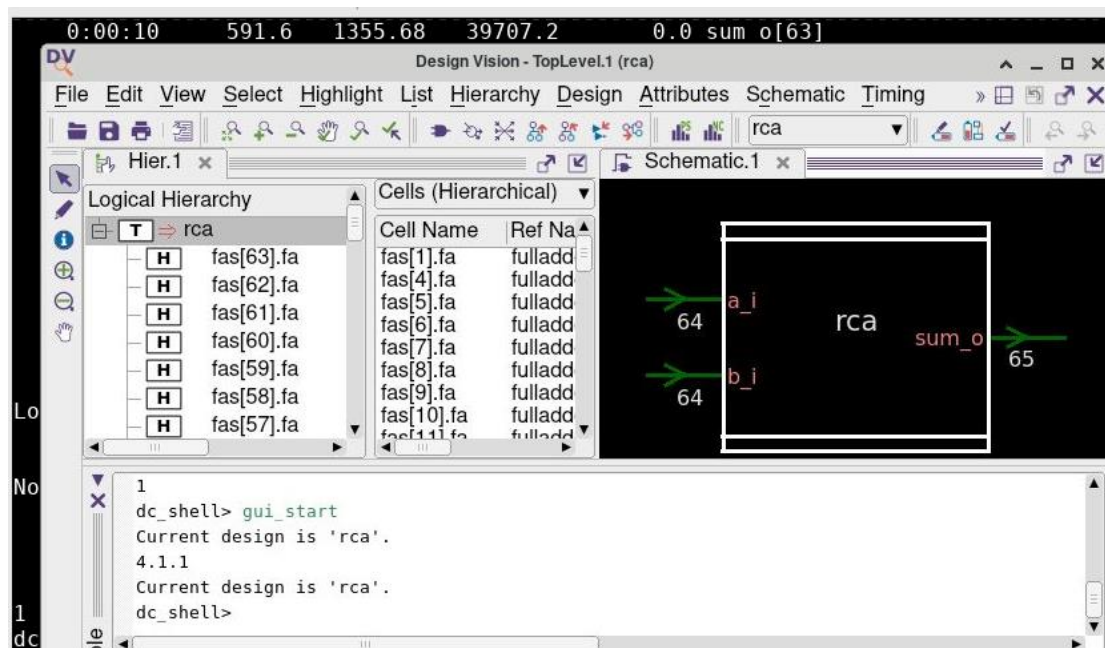
Ερώτημα 1.1

Στο παράδειγμα 1 πειραματιστήκαμε με διάφορες τιμές για την καθυστέρηση. Δυστυχώς με την εκτέλεση των εντολών μια προς μια (απλό compile) το κύκλωμα με καμία καθυστέρηση δεν μπορούσε να υλοποιηθεί σωστά (violated). Αυτό που παρατηρούμε είναι, πως εφόσον δεν μπορούσε να εκτελέσει αυτό που επιθυμούσαμε με τις συγκεκριμένες καθυστερήσεις, έκανε drop out πριν ολοκληρώσει κάτι.

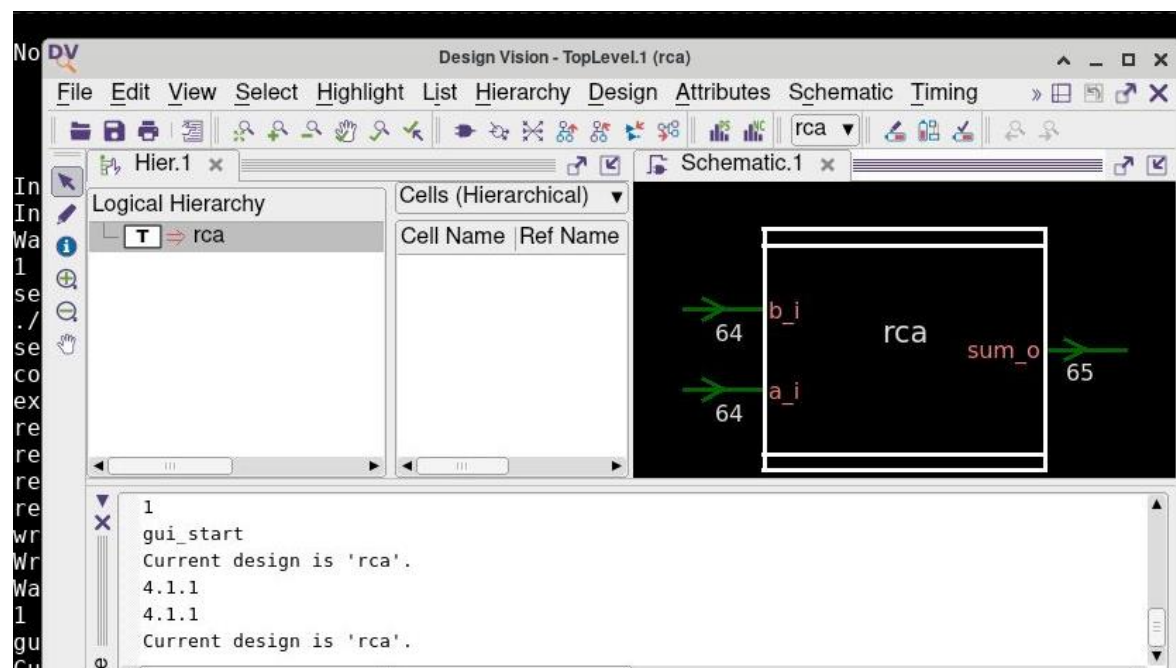
Ερώτημα 1.2

Εκτελούμε για το συγκεκριμένο ερώτημα και το απλό compile και το compile ultra, για καθυστέρηση 180. Οι βασικές διαφορές είναι πως στο απλό compile εμφανίζονται τα αποτελέσματα σε επίπεδο fa, ενώ με το compile ultra τα βλέπουμε σε πύλες (σπάμε το επίπεδο και το βλέπουμε σε άλλη μορφή) και με το compile ultra δεν υπάρχει καμία ιεραρχία. Επιπροσθέτως, παρατηρούμε πως όσο μικραίνει η καθυστέρηση τόσο αυξάνεται το area και το power, ενώ μειώνονται οι αριθμοί στο αρχείο timing. Όσον αφορά το αρχείο qor, εντοπίσαμε πως όταν εκτελέσαμε το compile ultra μειωνόταν το cell count αλλά πρόσθετε περισσότερη «λογική» στο κύκλωμα.

compile:



compile ultra:



timing_report.compile.100.rpt x	timing_report.compile_ultra.100.rpt x
results_rca > timing_report.compile.100.rpt	results_rca > timing_report.compile_ultra.100.rpt
1 Information: Updating design information... (UID-85)	1 Information: Updating design information... (UID-85)
2	2
3 *****	3 *****
4 Report : timing	4 Report : timing
5 -path full	5 -path full
6 -delay max	6 -delay max
7 -max_paths 1	7 -max_paths 1
8 Design : rca	8 Design : rca
9 Version: T-2022.03-SP5	9 Version: T-2022.03-SP5
10 Date : Wed May 15 12:20:14 2024	10 Date : Mon May 13 15:59:33 2024
11 *****	11 *****
12	12
13 Operating Conditions: PVT_0P7V_25C Library: asap7	13 Operating Conditions: PVT_0P7V_25C Library: asap7
14 Wire Load Model Mode: top	14 Wire Load Model Mode: top
15	15
16 Startpoint: b_i[0] (input port)	16 Startpoint: a_i[1] (input port)
17 Endpoint: sum_o[64] (output port)	17 Endpoint: sum_o[30] (output port)
18 Path Group: default	18 Path Group: default
19 Path Type: max	19 Path Type: max
20	20
21	21
22	22
23 input external delay	23 input external delay
24 b_i[0] (in)	24 a_i[1] (in)
25 fas[0].fa/b (fulladder_0)	25 U543/Y (INVx8_ASAP7_75t_R)
26 fas[0].fa/U2/Y (NAND2x2_ASAP7_75t_R)	26 U633/Y (AOI21x1_ASAP7_75t_R)
27 fas[0].fa/U2/Y (AOI21x1_ASAP7_75t_R)	27 U632/Y (AOI21x1_ASAP7_75t_R)
28 fas[0].fa/cout (fulladder_0)	28 U471/Y (AOI21x1_ASAP7_75t_R)
29 fas[1].fa/cin (fulladder_63)	29 U733/Y (AOI31xp67_ASAP7_75t_R)
30 fas[1].fa/U2/Y (INVx2_ASAP7_75t_R)	30 U485/Y (AOI21x1_ASAP7_75t_R)
31 fas[1].fa/U2/Y (AOI21x1_ASAP7_75t_R)	31 U533/Y (BUFx3_ASAP7_75t_R)
32 fas[1].fa/cout (fulladder_63)	32 U463/Y (AOI21x1_ASAP7_75t_R)
33 fas[2].fa/cin (fulladder_62)	33 U462/Y (O2A101Ixp33_ASAP7_75t_R)
34 fas[2].fa/U1/Y (INVx2_ASAP7_75t_R)	34 U696/Y (XNOR2xp5_ASAP7_75t_R)
35 fas[2].fa/U7/Y (AOI22x1_ASAP7_75t_R)	35 sum_o[30] (out)
36 fas[2].fa/cout (fulladder_62)	36 data arrival time
37 fas[3].fa/cin (fulladder_61)	37
38 fas[3].fa/U5/Y (INVx2_ASAP7_75t_R)	38
39 fas[3].fa/U7/Y (AOI22x1_ASAP7_75t_R)	38 max_delay
40 fas[3].fa/cout (fulladder_61)	39 output external delay
41 fas[4].fa/cin (fulladder_60)	40 data required time
42 fas[4].fa/U1/Y (INVx2_ASAP7_75t_R)	41
43 fas[4].fa/U5/Y (AOI22x1_ASAP7_75t_R)	42 data required time
44 fas[4].fa/cout (fulladder_60)	43 data arrival time
45 fas[5].fa/cin (fulladder_59)	44
46 fas[5].fa/U3/Y (AO22x2_ASAP7_75t_R)	45 slack (VIOLATED)
47 fas[5].fa/cout (fulladder_59)	46

Ερώτημα 2.1

Σχεδιάσαμε έναν 16-bit rca του οποίου οι είσοδοι και έξοδοι πάνε σε registers. Η μέγιστη συχνότητα που θα μπορούσε να λειτουργήσει το κύκλωμά μας είναι $1/43\text{ns}=0,023\text{ns}$. Παρακάτω παραθέτουμε πως βρήκαμε την συχνότητα και ότι επιπλέον το 43 αποτελεί το critical path, όπου δηλώνει το πιο αργό μονοπάτι. Χωρίς τους καταχωρητές η περίοδος μικραίνει.

```
≡ rca3stages.v  ≡ rca_syn.tcl  ≡ rca.v  ≡ serial.v  ≡ timing_report.compile.40.rpt X
results_rca > ≡ timing_report.compile.40.rpt
1  Information: Updating design information... (UID-85)
2
3  *****
4  Report : timing
5  | | | | -path full
6  | | | | -delay max
7  | | | | -max_paths 1
8  Design : rca_16
9  Version: T-2022.03-SP5
10 Date   : Wed May 15 12:50:15 2024
11 *****
12
13 Operating Conditions: PVT_0P7V_25C   Library: asap7
14 Wire Load Model Mode: top
15
16 Startpoint: out_reg_reg[16]
17 | | | | | (rising edge-triggered flip-flop)
18 Endpoint: sum_o[16] (output port clocked by clk)
19 Path Group: (none)
20 Path Type: max
21
22 Point                                Incr      Path
23 -----
24 out_reg_reg[16]/CLK (ASYNC_DFFHx1_ASAP7_75t_R)    0.00      0.00 r
25 out_reg_reg[16]/QN (ASYNC_DFFHx1_ASAP7_75t_R)   43.73     43.73 r
26 sum_o[16] (out)                                0.00     43.73 r
27 data arrival time                                43.73
28 -----
29 (Path is unconstrained)
30
31
32 1
33
```

```
module rca_16
```

```
#(
```

```
    parameter width = 16
```

```
)(
```

```
    input [width-1:0] a_i,
```

```
    input [width-1:0] b_i,
```

```
    input clk,
```

```
    input rstn,
```

```
    output [width:0] sum_o
```

```
);
```

```

wire [width-1:0] temp_sum;

wire [width:0] temp_c;

wire [width:0] result;


reg [width-1:0] a_reg;

reg [width-1:0] b_reg;

reg [width:0] out_reg;


always @(posedge clk or negedge rstn)begin

    if(rstn==0)begin

        a_reg<=0;

        b_reg<=0;

    end

    else begin

        a_reg<=a_i;

        b_reg<=b_i;

    end

end

end


assign temp_c[0] = 1'b0;

genvar gi;

generate

    for (gi=0; gi<width; gi=gi+1) begin: fas

        fulladder fa (

            .a(a_reg[gi]),

            .b(b_reg[gi]),

            .cin(temp_c[gi]),

            .s(temp_sum[gi]),

            .cout(temp_c[gi+1])

        );

    end

endgenerate


assign result= {temp_c[width],temp_sum};

```

```
always @(posedge clk or negedge rstn)begin
```

```
    if(rstn==0)
```

```
        out_reg<=0;
```

```
    else
```

```
        out_reg<=result;
```

```
end
```

```
assign sum_o = out_reg;
```

```
endmodule
```

Ερώτημα 2.2

Το pipelining είναι χρήσιμο για την αύξηση της συνολικής ταχύτητας επεξεργασίας και απόδοσης ενός συστήματος. Περιλαμβάνει τον διαχωρισμό μιας εργασίας σε πολλαπλά στάδια, όπου κάθε στάδιο εκτελεί ένα μέρος της συνολικής εργασίας. Αυτά τα στάδια συνδέονται σε μια σειρά, επιτρέποντας την επεξεργασία πολλαπλών εργασιών ταυτόχρονα αλλά σε διαφορετικά στάδια ολοκλήρωσης.

Για να μετατρέψουμε το Ripple Carry Adder (RCA), θα χωρίσουμε τη διαδικασία προσθήκης σε τρία στάδια. Κάθε στάδιο θα χειριστεί ένα τμήμα της προσθήκης και θα περάσει τα αποτελέσματα στο επόμενο στάδιο μέσω των καταχωρητών.



Στάδιο 1: Επεξεργασία των λιγότερο σημαντικών bits

Στάδιο 2: Επεξεργασία των μεσαίων bits.

Στάδιο 3: Επεξεργασία των πιο σημαντικών bits.

Remaining Width: Το πλάτος που απομένει είναι $\text{width} - 2 * \text{stage_width}$ για να ληφθούν υπόψη τυχόν εναπομείναντα bit.

Αρχικά παρατηρούμε, πως όταν εκτελούμε απλό compile για τον rca 3 stages pipeline, ο χρόνος είναι πολύ μικρότερος και δεν γίνεται violated το κύκλωμα.

results_rca3stages >  qor_report.compile.200.rpt	results_rca >  qor_report.compile.200.rpt
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	

Report : qor	Report : qor
Design : rca_pipeline	Design : rca
Version: T-2022.03-SP5	Version: T-2022.03-SP5
Date : Sun May 19 20:13:04 2024	Date : Sun May 12 20:44:25 2024

Timing Path Group (none)	Timing Path Group 'default'
Levels of Logic: 0.00	Levels of Logic: 92.00
Critical Path Length: 38.96	Critical Path Length: 1563.10
Critical Path Slack: uninit	Critical Path Slack: -1363.10
Critical Path Clk Period: n/a	Critical Path Clk Period: n/a
Total Negative Slack: 0.00	Total Negative Slack: -39920.52
No. of Violating Paths: 0.00	No. of Violating Paths: 56.00
Worst Hold Violation: 0.00	Worst Hold Violation: 0.00
Total Hold Violation: 0.00	Total Hold Violation: 0.00
No. of Hold Violations: 0.00	No. of Hold Violations: 0.00

Ωστόσο, στο area παρατηρείτε μια μικρή αύξηση.

```

results_rca3stages > area_report.compile.200.rpt
1
2 *****
3 Report : area
4 Design : rca_pipeline
5 Version: T-2022.03-SP5
6 Date   : Sun May 19 20:13:22 2024
7 *****
8
9 Library(s) Used:
10
11 |      asap7 (File: /usr/local/eda/synLibs/asap7/7nm/db/asap7.db)
12
13 Number of ports:                514
14 Number of nets:                 832
15 Number of cells:               469
16 Number of combinational cells: 299
17 Number of sequential cells:    106
18 Number of macros/black boxes:  0
19 Number of buf/inv:             106
20 Number of references:           67
21
22 Combinational area:             477.990715
23 Buf/Inv area:                  74.183041
24 Noncombinational area:         494.553581
25 Macro/Black Box area:          0.000000
26 Net Interconnect area:         undefined (No wire load specified)
27
28 Total cell area:                972.544296

results_rca > area_report.compile.200.rpt
1
2 *****
3 Report : area
4 Design : rca
5 Version: T-2022.03-SP5
6 Date   : Sun May 12 20:44:33 2024
7 *****
8
9 Library(s) Used:
10
11 |      asap7 (File: /usr/local/eda/synLibs/asap7/7nm/db/asap7.db)
12
13 Number of ports:                513
14 Number of nets:                 787
15 Number of cells:               403
16 Number of combinational cells: 339
17 Number of sequential cells:    0
18 Number of macros/black boxes:  0
19 Number of buf/inv:             82
20 Number of references:           65
21
22 Combinational area:             603.495351
23 Buf/Inv area:                  76.049281
24 Noncombinational area:         0.000000
25 Macro/Black Box area:          0.000000
26 Net Interconnect area:         undefined (No wire load specified)
27
28 Total cell area:                603.495351

```


Τέλος, όσον αφορά το `power_report`, η ανάλυση στο `rca 3 stages pipeline` γίνεται για κάθε `fa` κάθε επιπέδου, όπως και στον `rca` του Παραδείγματος 1, όπου αναλύονταν σε επίπεδο `fa`.

```

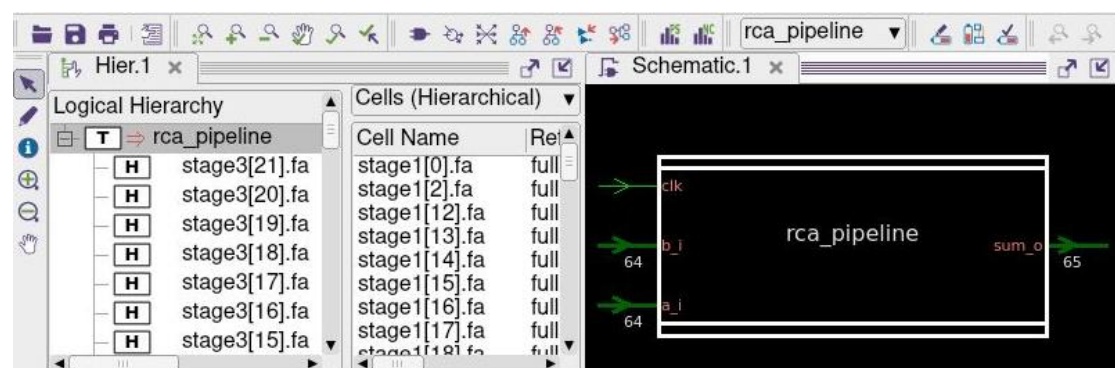
results_rca3stages > E power_report.compile.200.rpt
27 Power Spectral Density Introduction ;
33
34
35 -----
36 | | | | | | | | | | | | | | Switch Int Leak Total
37 Hierarchy Power Power Power Power %
38 -----
39 rca_pipeline 11.868 25.287 5.40e+04 37.155 100.0
40 stage3[21].fa (fulladder_1) 0.112 0.158 378.115 0.271 0.7
41 stage3[20].fa (fulladder_2) 9.71e-02 0.161 378.135 0.258 0.7
42 stage3[19].fa (fulladder_3) 9.58e-02 0.159 378.124 0.255 0.7
43 stage3[18].fa (fulladder_4) 9.55e-02 0.159 378.211 0.254 0.7
44 stage3[17].fa (fulladder_5) 9.77e-02 0.160 378.082 0.257 0.7
45 stage3[16].fa (fulladder_6) 9.60e-02 0.158 378.106 0.254 0.7
46 stage3[15].fa (fulladder_7) 9.75e-02 0.160 378.115 0.258 0.7

```

```

32 results_rca >  power_report.compile.200.rpt
33
34 -----
35 | | | | | | | | | | | | | | | | | | | | Switch  Int  Leak  Total
36 Hierarchy      Power    Power    Power    Power    %
37 -----
38 rca              16.042    21.492  5.01e+04    37.534 100.0
39 | fas[63].fa (fulladder_1)      0.435    0.458  1.08e+03    0.892  2.4
40 | fas[62].fa (fulladder_2)      0.265    0.267  599.440    0.532  1.4
41 | fas[61].fa (fulladder_3)      0.220    0.329  669.708    0.549  1.5
42 | fas[60].fa (fulladder_4)      0.177    0.284  609.398    0.462  1.2
43 | fas[59].fa (fulladder_5)      0.184    0.292  609.299    0.476  1.3
44 | fas[58].fa (fulladder_6)      0.178    0.285  609.710    0.462  1.2
45 | fas[57].fa (fulladder_7)      0.265    0.254  547.897    0.520  1.4
46 | fas[56].fa (fulladder_8)      0.247    0.239  534.269    0.486  1.1

```



```

module rca_pipeline

#(
    parameter width = 64
)(
    input clk,
    input [width-1:0] a_i,
    input [width-1:0] b_i,
    output [width:0] sum_o
);

    localparam stage_width = width / 3;
    localparam remaining_width = width - 2 * stage_width;

    reg [stage_width-1:0] sum_stage1, sum_stage2;
    reg [remaining_width-1:0] sum_stage3;
    reg carry_stage1, carry_stage2, carry_stage3;

    wire [stage_width-1:0] sum_temp1, sum_temp2;
    wire [remaining_width-1:0] sum_temp3;
    wire [stage_width:0] carry_temp1, carry_temp2;
    wire [remaining_width:0] carry_temp3;

    assign carry_temp1[0] = 1'b0;

    // stage 1
    genvar i;
    generate
        for (i = 0; i < stage_width; i = i + 1) begin: stage1
            fulladder fa (
                .a(a_i[i]),
                .b(b_i[i]),
                .cin(carry_temp1[i]),
                .s(sum_temp1[i]),
                .cout(carry_temp1[i+1])
            );
        end
    endgenerate

```

```
end  
endgenerate
```

```
//for pipeline registers  
always @(posedge clk) begin  
    sum_stage1 <= sum_temp1;  
    carry_stage1 <= carry_temp1[stage_width];  
end
```

```
// stage 2  
generate  
    for (i = 0; i < stage_width; i = i + 1) begin: stage2  
        fulladder fa (  
            .a(a_i[stage_width + i]),  
            .b(b_i[stage_width + i]),  
            .cin((i == 0) ? carry_stage1 : carry_temp2[i]),  
            .s(sum_temp2[i]),  
            .cout(carry_temp2[i+1])  
        );  
    end  
endgenerate
```

```
always @(posedge clk) begin  
    sum_stage2 <= sum_temp2;  
    carry_stage2 <= carry_temp2[stage_width];  
end
```

```
// stage 3  
generate  
    for (i = 0; i < remaining_width; i = i + 1) begin: stage3  
        fulladder fa (  
            .a(a_i[2*stage_width + i]),  
            .b(b_i[2*stage_width + i]),  
            .cin((i == 0) ? carry_stage2 : carry_temp3[i]),  
            .s(sum_temp3[i]),
```



```

        .cout(carry_temp3[i+1])
    );
end
endgenerate

always @(posedge clk) begin
    sum_stage3 <= sum_temp3;
    carry_stage3 <= carry_temp3[remaining_width];
end

assign sum_o = {carry_stage3, sum_stage3, sum_stage2, sum_stage1};

endmodule

```

Επιπροσθέτως, μπορούμε να πετύχουμε παρόμοια αποτελέσματα χωρίς να κάνουμε manual pipelining στο κύκλωμά μας χρησιμοποιώντας flip-flops για να δημιουργήσουμε το pipeline στα ενδιάμεσα στάδια. Ουσιαστικά, αυτό σημαίνει ότι θα προσθέσουμε καταχωρητές μεταξύ των εισόδων και των εξόδων των full adders για να επιτύχουμε το ίδιο αποτέλεσμα όπως πριν, αλλά με πιο αυτοματοποιημένο τρόπο. Ένας πιθανός τρόπος ίσως είναι ο παρακάτω:

```

module rca_pipeline
#(
    parameter width = 64
)(
    input clk,
    input [width-1:0] a_i,
    input [width-1:0] b_i,
    output reg [width:0] sum_o
);

    reg [width-1:0] sum_stage1, sum_stage2, sum_stage3;
    reg [width:0] carry_stage1, carry_stage2, carry_stage3;

    wire [width-1:0] sum_temp;
    wire [width:0] carry_temp;

    assign carry_temp[0] = 1'b0;

```

```

// Full adders

genvar i;

generate

  for (i = 0; i < width; i = i + 1) begin: full_adders

    fulladder fa (

      .a(a_i[i]),

      .b(b_i[i]),

      .cin(carry_temp[i]),

      .s(sum_temp[i]),

      .cout(carry_temp[i+1])

    );

  end

endgenerate

always @(posedge clk) begin

  // Pipeline stage 1

  sum_stage1 <= sum_temp;

  carry_stage1 <= carry_temp;

  // stage 2

  sum_stage2 <= sum_stage1;

  carry_stage2 <= carry_stage1;

  // stage 3

  sum_stage3 <= sum_stage2;

  carry_stage3 <= carry_stage2;

  sum_o <= {carry_stage3[width], sum_stage3};

end

endmodule

```

Ερώτημα 3.1

Η ελάχιστη καθυστέρηση είναι time=103 και η επιφάνεια των κυκλωμάτων area=450.

```
results_acc > timing_report.compile_ultra.100.rpt
1 Information: Updating design information... (UID-85)
2
3 *****
4 Report : timing
5 | | | | -path full
6 | | | | -delay max
7 | | | | -max_paths 1
8 Design : accumulator
9 Version: T-2022.03-SP5
10 Date : Sun May 19 17:49:32 2024
11 *****
12
13 Operating Conditions: PVT_0P7V_25C Library: asap7
14 Wire Load Model Mode: top
15
16 Startpoint: counter_reg[0]
17 | | | | (rising edge-triggered flip-flop)
18 Endpoint: ready (output port)
19 Path Group: (none)
20 Path Type: max
21
22 Point Incr Path
23 -----
24 counter_reg[0]/CLK (ASYNC_DFFHx1_ASAP7_75t_R) 0.00 0.00 r
25 counter_reg[0]/QN (ASYNC_DFFHx1_ASAP7_75t_R) 55.61 55.61 r
26 U135/Y (NAND3xp33_ASAP7_75t_R) 28.21 83.82 f
27 U129/Y (INVxp33_ASAP7_75t_R) 19.37 103.19 r
28 ready (out) 0.00 103.19 r
29 data arrival time
30 -----
31 (Path is unconstrained)
32
33
34 1
35
```

```
results_acc > area_report.compile_ultra.100.rpt
1
2 *****
3 Report : area
4 Design : accumulator
5 Version: T-2022.03-SP5
6 Date : Sun May 19 17:49:32 2024
7 *****
8
9 Library(s) Used:
10 |
11 | asap7 (File: /usr/local/eda/synLibs/asap7/7nm/db/asap7.db)
12
13 Number of ports: 46
14 Number of nets: 228
15 Number of cells: 193
16 Number of combinational cells: 147
17 Number of sequential cells: 46
18 Number of macros/black boxes: 0
19 Number of buf/inv: 34
20 Number of references: 17
21
22 Combinational area: 171.927360
23 Buf/Inv area: 24.261120
24 Noncombinational area: 279.002878
25 Macro/Black Box area: 0.000000
26 Net Interconnect area: undefined (No wire load specified)
27
28 Total cell area: 450.930239
29 Total area: undefined
30
31 Hierarchical area distribution
32 -----
33 | | | | | | | | | Global cell area Local cell area
34 |-----|-----|-----|-----|-----|-----|
35 | | | | | | | | | Absolute Percent Combi- Noncombi- Black- Design
36 | | | | | | | | | Total Total national national boxes
37 |-----|-----|-----|-----|-----|-----|-----|
38 | | | | | | | | |
39 accumulator 450.9302 100.0 171.9274 279.0029 0.0000 accumulator
40 |-----|-----|-----|-----|-----|-----|-----|
41 Total 171.9274 279.0029 0.0000
42
43 1
```

```

results_acc > E qor_report.compile_ultra.100.rpt
7 *****
10 Timing Path Group (none)
11 -----
12 Levels of Logic:          2.00
13 Critical Path Length:     103.19
14 Critical Path Slack:      uninit
15 Critical Path Clk Period:  n/a
16 Total Negative Slack:     0.00
17 No. of Violating Paths:   0.00
18 Worst Hold Violation:     0.00
19 Total Hold Violation:     0.00
20 No. of Hold Violations:   0.00
21 -----
22
23
24 Cell Count
25 -----
26 Hierarchical Cell Count:   0
27 Hierarchical Port Count:   0
28 Leaf Cell Count:          193
29 Buf/Inv Cell Count:        34
30 Buf Cell Count:           1
31 Inv Cell Count:            33
32 CT Buf/Inv Cell Count:     0
33 Combinational Cell Count:  147
34 Sequential Cell Count:     46
35 Macro Count:               0
36 -----
37
38
39 Area
40 -----
41 Combinational Area:        171.927360
42 Noncombinational Area:    279.002878
43 Buf/Inv Area:              24.261120
44 Total Buffer Area:         0.93
45 Total Inverter Area:       23.33
46 Macro/Black Box Area:     0.000000
47 Net Area:                  0.000000
48 -----
49 Cell Area:                  450.930239
50 Design Area:                450.930239
51

```

```

Design Rules
-----
Total Number of Nets:          228
Nets With Violations:          0
Max Trans Violations:          0
Max Cap Violations:            0
-----

Hostname: dagobah.ceid.upatras.gr

Compile CPU Statistics
-----
Resource Sharing:               0.04
Logic Optimization:             0.17
Mapping Optimization:           1.09
-----
Overall Compile Time:           7.98
Overall Compile Wall Clock Time: 8.35
-----

Design  WNS: 0.00  TNS: 0.00  Number of Violating Paths: 0

Design (Hold)  WNS: 0.00  TNS: 0.00  Number of Violating Paths: 0

```

```
module accumulator #(parameter k=4, parameter n=8, parameter m=$clog2(k))(input [k*n-1:0] data, input pl,  
input clk, input rstn, output ready, output [n+m-1:0] sum);
```

```
genvar gi;
```

```
reg [k*n-1:0] register;
```

```
wire add;
```

```
reg [n+m-1:0] result;
```

```
reg [m:0] counter;
```

```
reg r;
```

```
always @(posedge clk or negedge rstn)begin
```

```
    if(rstn==0)
```

```
        register<=0;
```

```
    else begin
```

```
        if(add)
```

```
            register<=register>>n;
```

```
        else begin
```

```
            if(pl)
```

```
                register<=data;
```

```
        end
```

```
    end
```

```
end
```

```
always @(posedge clk or negedge rstn)begin
```

```
    if(rstn==0)
```

```
        result<=0;
```

```
    else begin
```

```
        if(pl==1&&counter==0)
```

```
            result<=0;
```

```
        else
```

```
            if(add==1)
```

```
                result<=result+register[n-1:0];
```

```
        end
```

```
end
```

```
always @(posedge clk or negedge rstn)begin
```

```

        if(rstn==0)
            counter<=0;
        else begin
            if(counter!=0&&add==1)
                counter<=counter-1;
            else if(pl)
                counter<=k;
        end
    end

end

always @(posedge clk or negedge rstn)begin
    if(rstn==0)
        r<=0;

    else begin
        if(counter==0)
            r<=0;
        else
            r<=1;
    end
end

assign add=r;
assign ready=(counter==0)?1:0;
assign sum=result;

endmodule

```

Ερώτημα 3.2

Throughput: Ο αριθμός των εργασιών που υποβάλλονται σε επεξεργασία ανά μονάδα χρόνου.

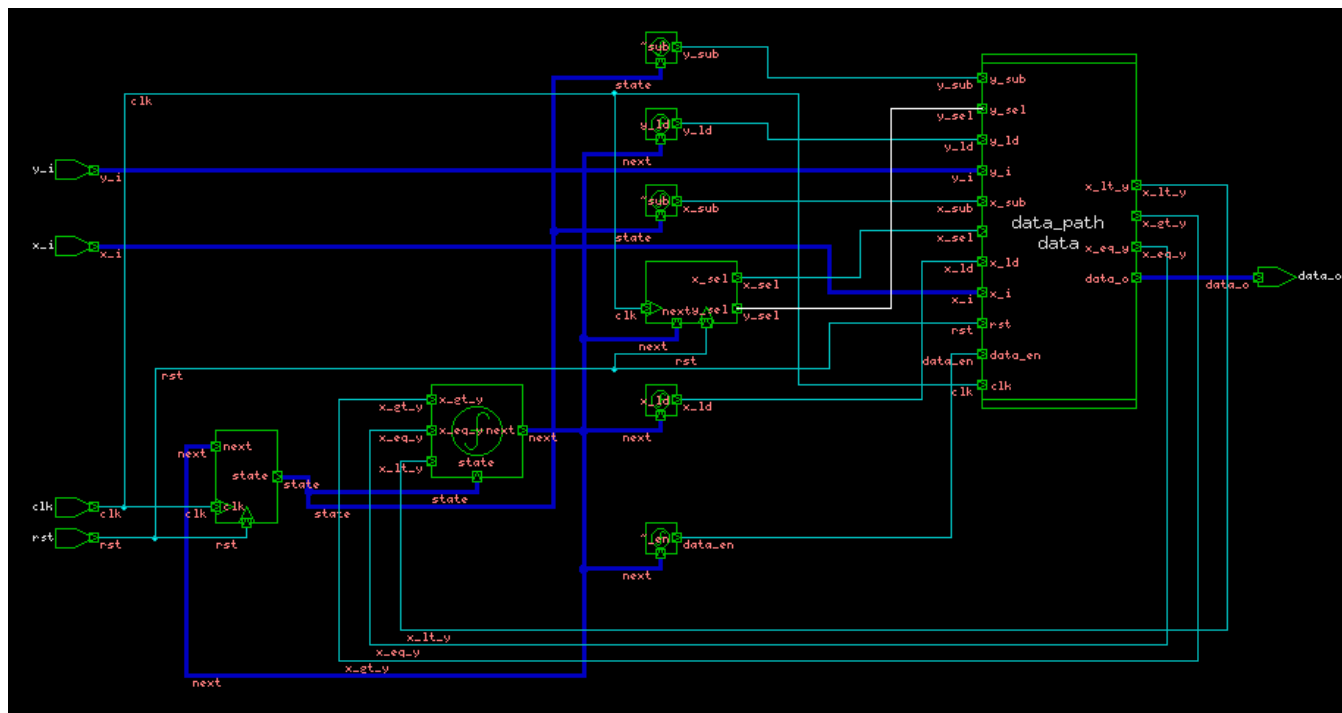
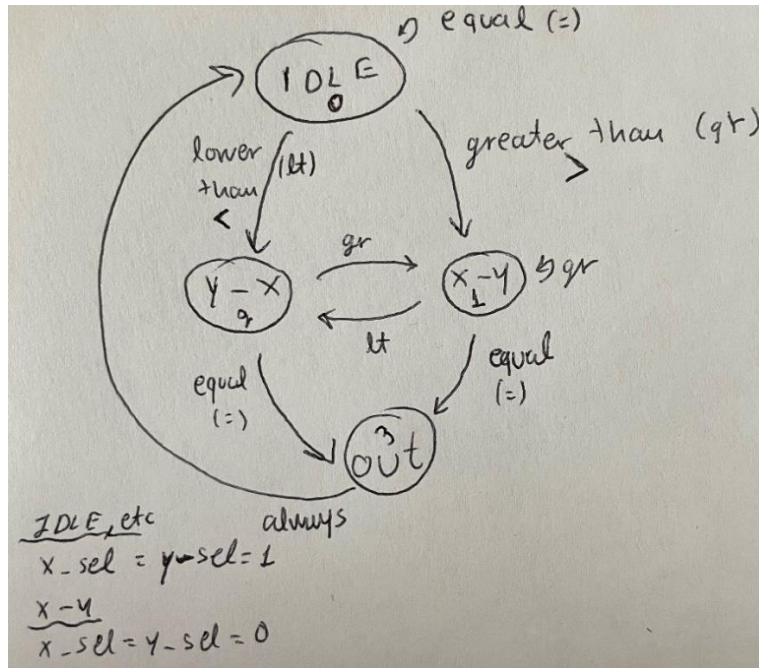
Latency: Ο χρόνος που χρειάζεται για να περάσει μια μεμονωμένη εργασία από το pipeline.

Ερώτημα 4

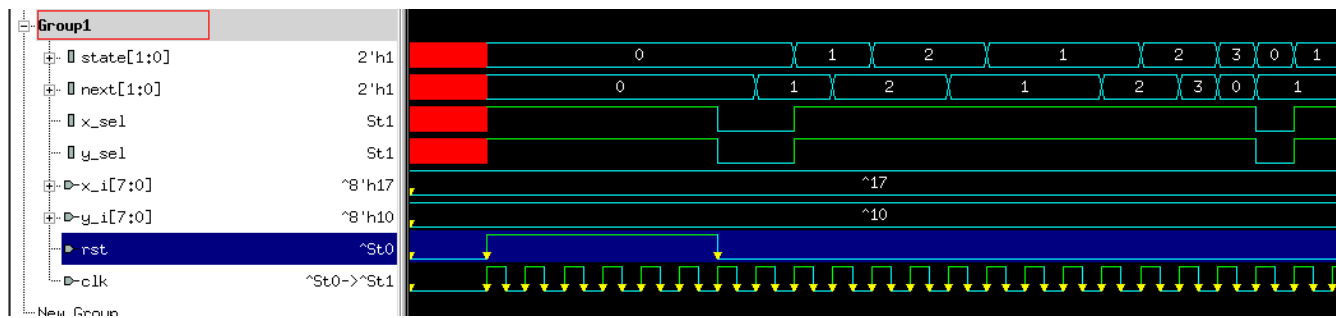
Παρακάτω σας παραθέτουμε το διάγραμμα καταστάσεων. Επιπλέον, υπάρχουν κάποιες αλλαγές στο FSM και για να εκτελέσουμε τους κώδικες φτιάξαμε αντίστοιχα σε ένα makefile:

gcd:

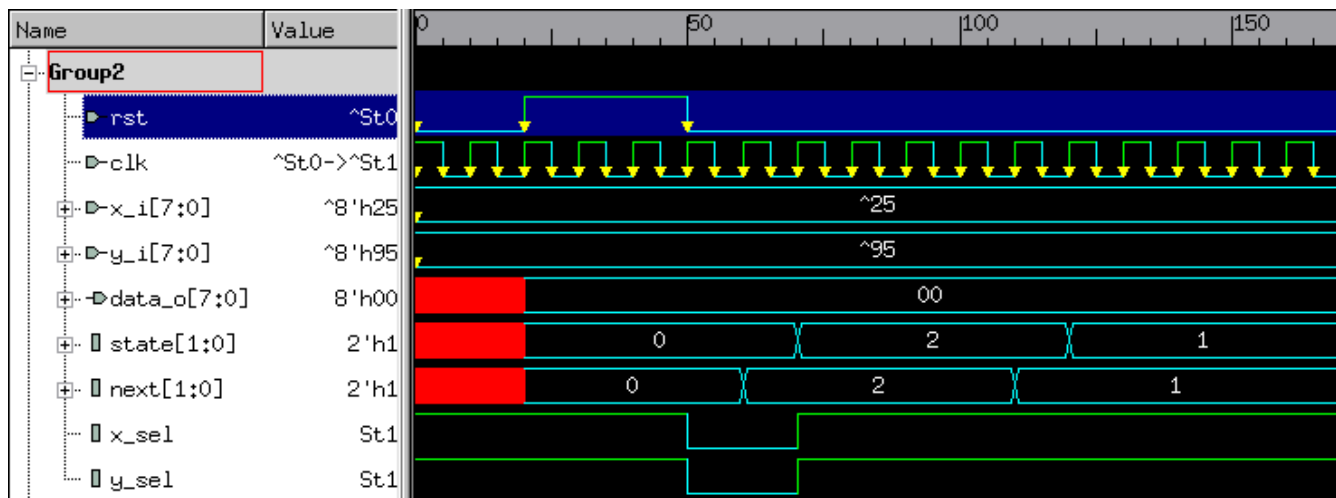
```
$(VCS) $(VCS_OPTS) $(SRC_DIR)/gcd_control.v $(SRC_DIR)/gcd_data.v -o gcd  
./gcd $(GUI_OPTS)
```



X>Y:



X<Y:



Κώδικας (gdc_control και gdc_data):

```
module control #(parameter n=8) (
    input rst,
    input clk,
    input [n-1:0] x_i,
    input [n-1:0] y_i,
    output [n-1:0] data_o
);

parameter [1:0]
idle=0,subtract_x=1,subtract_y=2,finish=3;

reg [1:0] state,next;

reg x_sel,y_sel;

wire x_gt_y,x_lt_y,x_eq_y,x_sub,y_sub,x_ld,y_ld,data_en;
```



```

data #(.n(n)) data_path
(.x_i(x_i),.y_i(y_i),.x_sel(x_sel),.x_ld(x_ld),.y_sel(y_sel),.y_ld(y_ld),.data_en(data_en),.x_sub(x_sub),.y_sub(y_sub),
.rst(rst),.clk(clk),.x_gt_y(x_gt_y),.x_eq_y(x_eq_y),.x_lt_y(x_lt_y),.data_o(data_o));

```

```

always @(posedge clk or posedge rst)

```

```

if(rst) state<=idle;

```

```

else state <=next;

```

```

always @(state or x_gt_y or x_eq_y or x_lt_y)begin

```

```

case(state)

```

```

    idle: if(x_gt_y) next=subtract_x;

```

```

        else if(x_lt_y) next=subtract_y;

```

```

        else next=idle;

```

```

    subtract_x:

```

```

        if(x_gt_y) next=subtract_x;

```

```

    else if(x_lt_y) next=subtract_y;

```

```

    else next=finish;

```

```

    subtract_y: if(x_gt_y) next=subtract_x;

```

```

    else if(x_lt_y) next=subtract_y;

```

```

    else next=finish;

```

```

    finish: next=idle;

```

```

endcase

```

```

end

```

```

assign x_sub=(state==1)?1:0;

```

```

assign y_sub=(state==2)?1:0;

```

```

assign x_ld=(next==2)?0:1;

```

```

assign y_ld=(next==1)?0:1;

```

```

assign data_en=(next==3)?1:0;

```

```

always @(posedge clk or posedge rst)begin

```

```

    if(rst==1)begin

```

```

        x_sel=1; y_sel=1;

```

```

    end

```

```

        else begin
            x_sel=1; y_sel=1;

            case(next)
            idle:begin
                x_sel<=0;
                y_sel<=0;

            end

            subtract_x;;

            subtract_y;;

            finish;;

        endcase
    end
end

endmodule

```

DATA FLOW

```

module data #(parameter n=8)
    (input [n-1:0] x_i,
    input [n-1:0] y_i,
    input x_sel,
    input x_ld,
    input y_sel,
    input y_ld,
    input data_en,
    input x_sub,
    input y_sub,
    input rst,
    input clk,
    output x_gt_y,
    output x_eq_y,
    output x_lt_y,
    output [n-1:0] data_o
    );

```

```

reg [n-1:0] x_reg;
reg [n-1:0] y_reg;
reg [n-1:0] result;

wire [n-1:0] x_sub_result;
wire [n-1:0] y_sub_result;
wire [n-1:0] x_mux;
wire [n-1:0] y_mux;

assign x_mux=(x_sel)?x_sub_result:x_i;
assign y_mux=(y_sel)?y_sub_result:y_i;

always @(posedge clk or posedge rst)begin
    if(rst==1)
        x_reg<=0;
    else
        if(x_ld==1) x_reg<=x_mux;
        else x_reg<=x_reg;
end

always @(posedge clk or posedge rst)begin
    if(rst==1)
        y_reg<=0;
    else
        if(y_ld==1) y_reg<=y_mux;
        else y_reg<=y_reg;
end

assign x_gt_y=(x_reg>y_reg)?1:0;
assign x_lt_y=(x_reg<y_reg)?1:0;
assign x_eq_y=(x_reg==y_reg)?1:0;

assign x_sub_result=(x_sub)?(x_reg-y_reg):x_reg;
assign y_sub_result=(y_sub)?(y_reg-x_reg):y_reg;

```

```
always @(posedge clk or posedge rst)begin
    if(rst==1)
        result<=0;
    else
        if(data_en==1) result<=x_reg;
        else result<=result;
end

assign data_o=result;

endmodule
```