

Γλώσσες Περιγραφής Υλικού I

Κυκλώματα για προχωρημένους και στοιχεία λογικής σύνθεσης

Νικόλαος Καβαδιάς
nkavn@uop.gr

25 Μαΐου 2011

Σκιαγράφηση της διάλεξης

- Ειδικά κυκλώματα
 - Διαιρέτης ρολογιού
- Στοιχεία λογικής σύνθεσης
 - Εισαγωγή
 - Η ροή λογικής σύνθεσης
 - Παραδείγματα
 - Πλήρης αθροιστής
 - RTL αθροιστής προσημασμένων αριθμών των 4-bit
 - Καταχωρητής των 8-bit
 - Επεξεργαστής μέγιστου κοινού διαιρέτη (GCD processor)

Διαιρέτης ρολογιού (clock divider)

- Κύκλωμα παραγωγής σήματος ρολογιού, του οποίου η περίοδος είναι ρητό υποπολλαπλάσιο της περιόδου του αρχικού σήματος ρολογιού
- $f_{clkdiv} = (M/N) \times f_{clk}$
- Στα συστήματα με FPGA, η συχνότητα αναφοράς παράγεται από κάποιον πιεζοηλεκτρικό κρύσταλλο χαλαζία και στη συνέχεια τα χαρακτηριστικά του παλμού διαμορφώνονται από ειδικά on-chip κυκλώματα
- Κύκλος εργασίας (duty cycle): ο λόγος της διάρκειας της υψηλής (χαμηλής) λογικής στάθμης ως προς τη συνολική διάρκεια της περιόδου
- Χρησιμοποιούνται για το χρονοισμό διαύλων και γενικά πιο αργών διατάξεων σε σχέση με το σύστημα επεξεργασίας (CPU, FSMD) όπως είναι οι ελεγκτές περιφερειακών συσκευών (ελεγκτής διακοπών, οθόνης LCD/VGA)

Απλός παραμετρικός διαιρέτης ρολογιού (1)

- Διαιρέτης ρολογιού βασισμένος σε απαριθμητή
- Η παράμετρος LENGTH ρυθμίζει τη διάρκεια της περιόδου, η οποία είναι $\lfloor (LENGTH/2) \rfloor$
- Δεν διαθέτει γενικά κύκλο εργασίας ίσο με 50% (ισχύει μόνο για άρτια υποπολλαπλάσια)

```
// LENGTH should be in the range 2-255
`define LENGTH 3

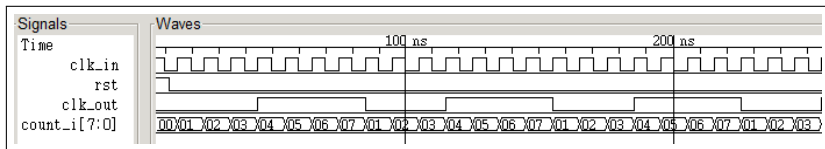
module clockdiv(clk_in, rst, clk_out);
  input  clk_in, rst;
  output clk_out;
  reg    clk_out;
  reg [7:0] count_i;

  always @(posedge clk_in)
  begin
    if (rst) begin
      count_i = 0;
      clk_out = 1'b0;
    end
  end
end
```

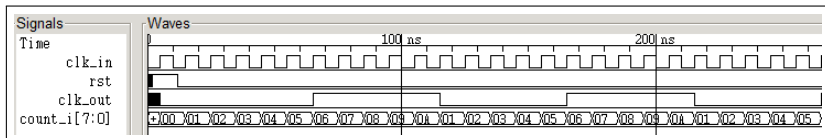
```
else begin
  if (count_i == (`LENGTH>>1))
    clk_out <= 1'b1;
  else if (count_i == `LENGTH)
  begin
    count_i = 0;
    clk_out <= 1'b0;
  end
  //
  count_i = count_i + 1;
end
end
endmodule
```

Απλός παραμετρικός διαιρέτης ρολογιού (2)

- Χρονικό διάγραμμα για $LENGTH = 7$



- Χρονικό διάγραμμα για $LENGTH = 10$

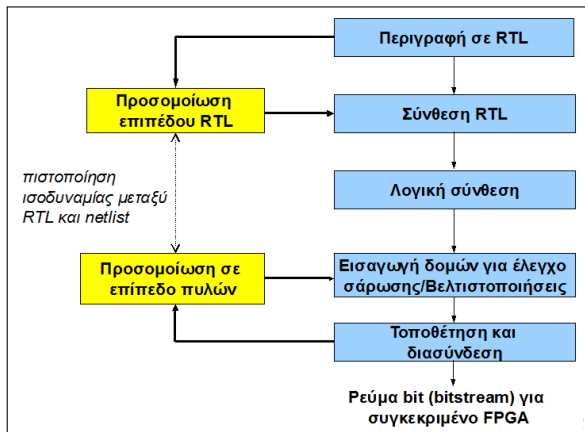


Εισαγωγή στη λογική σύνθεση

- Σκοπός της περιγραφής ψηφιακών κυκλωμάτων είναι η υλοποίησή τους σε ολοκληρωμένο
- Η λογική σύνθεση είναι είδος μεταγλώττισης από το υψηλό επίπεδο μιας HDL (δομική, RTL ή μικτή) περιγραφής στο χαμηλό επίπεδο της λίστας κόμβων (netlist) με τα στοιχειώδη κυκλωματικά στοιχεία της τεχνολογίας
- Δημοφιλείς τεχνολογίες: διεργασίες τυποποιημένου κελιού (standard cell VLSI), FPGA
- Υπάρχουν περιορισμοί στον τρόπο σχεδιασμού με μια HDL όπως είναι η Verilog ώστε η τελική περιγραφή να είναι συνθέσιμη
- Κατάλληλες τεχνικές στην ανάπτυξη του κώδικα οδηγούν στην επίτευξη καλύτερων επιδόσεων (ως προς ταχύτητα επεξεργασίας, επιφάνεια ολοκληρωμένου, κατανάλωση ισχύος/ενέργειας)
- Για τη σύνθεση χρησιμοποιούνται εμπορικά εργαλεία (Xilinx ISE Webpack, Altera Quartus, Cadence Design Suite, Synopsys DC) ή εργαλεία ανοικτού κώδικα (Icarus Verilog, VPR, Odin)

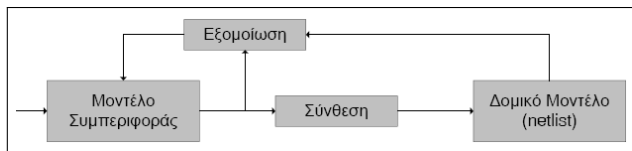
Η ροή της διαδικασίας λογικής σύνθεσης

- Η συγκεκριμένη ροή εστιάζει στις τεχνολογίες FPGA. Τα περισσότερα εργαλεία CAD χωρίζονται σε frontend και backend
- frontend** μετατροπή της περιγραφής του κυκλώματος σε netlist
- backend** λογική σύνθεση της netlist



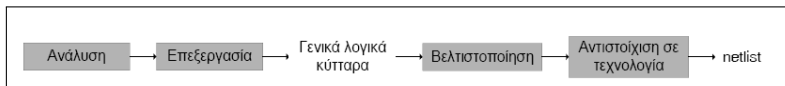
Λογική σύνθεση (1)

- Η σύνθεση λαμβάνει ως είσοδο μία περιγραφή και παράγει μία λίστα (netlist) από κύτταρα βιβλιοθήκης, με τις διασυνδέσεις τους
- Οι HDLs δημιουργήθηκαν αρχικά μόνο για λόγους τεκμηρίωσης και προσομοίωσης ενός κυκλώματος
- Αργότερα παρουσιάστηκε η ανάγκη για σύνθεση
- Όλες οι γλωσσικές δομές δεν μπορούν να είναι και συνθέσιμες. Κάθε περιγραφή είναι προσομοιώσιμη αλλά λίγες μόνο είναι συνθέσιμες



Λογική σύνθεση (2)

- Ένα από τα μεγάλα πλεονεκτήματα της σύνθεσης είναι η δυνατότητα βελτιστοποίησης του τελικού κυκλώματος



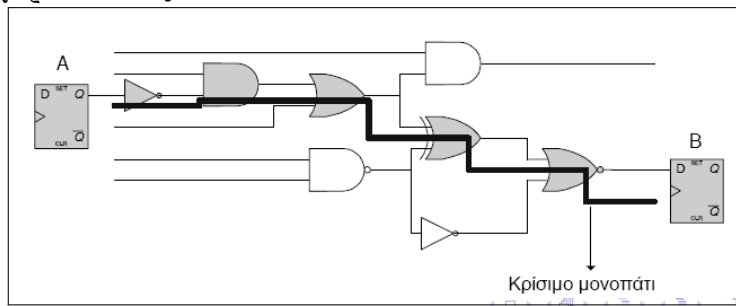
- Βελτιστοποίηση κυκλώματος
 - Ελαχιστοποίηση της επιφάνειας
 - Μεγιστοποίηση της ταχύτητας
 - Ικανοποίηση περιορισμών λειτουργίας
- Η αντιστοίχιση σε τεχνολογία (technology mapping, library binding) οδηγεί σε περαιτέρω βελτιστοποίηση του κυκλώματος

Προσομοίωση

- Η προσομοίωση είναι το βασικότερο εργαλείο επιβεβαίωσης της ορθότητας σχεδίασης ενός κυκλώματος
- Είδη προσομοίωσης
 - Προσομοίωση συμπεριφοράς (behavioral simulation)
 - Λειτουργική προσομοίωση (functional simulation-unit delay)
 - Στατική χρονική ανάλυση (static timing analysis), χωρίς διανύσματα εισόδου
 - Προσομοίωση σε επίπεδο πύλης (gate-level simulation)
 - Προσομοίωση διακοπών (switch-level simulation)
 - Προσομοίωση transistor (transistor-level simulation)
- Η πιο ακριβής προσομοίωση γίνεται μετά την οριστικοποίηση του layout (post-layout simulation)

Στατική χρονική ανάλυση

- Ο στατικός αναλυτής βρίσκει το/τα κρίσιμα μονοπάτια στο κύκλωμα χωρίς διανύσματα προσομοίωσης
- Η ανάλυση του κυκλώματος γίνεται στατικά
 - Αλγόριθμοι γράφων όπως all pairs longest paths
- Η στατική χρονική ανάλυση πρέπει να λαμβάνει υπόψη και τις καθυστερήσεις γραμμών που οφείλονται σε χωρητικότητες



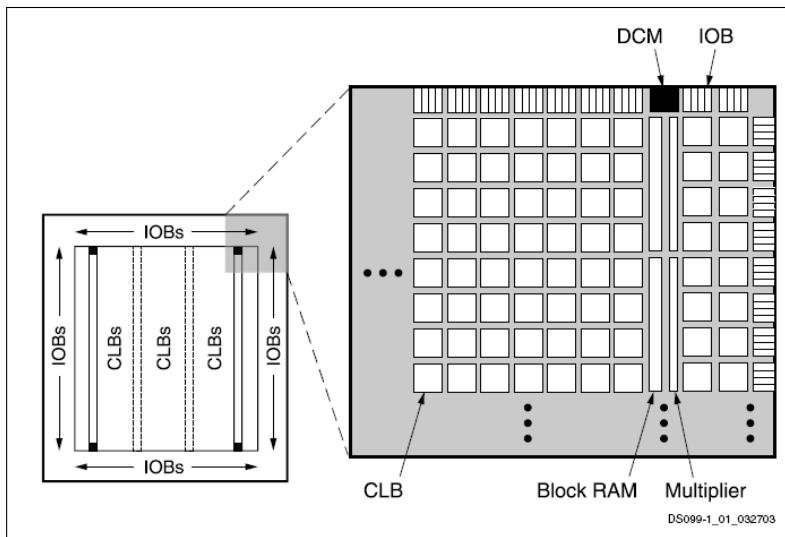
Διαμέριση, χωροθέτηση, τοποθέτηση και διασύνδεση

- Διαμέριση του δικτύου
 - Διαιρούμε τη σχεδίαση σε μικρότερα τμήματα
 - Απλοποίηση του προβλήματος με υποδιαίρεση σε υποπροβλήματα
 - Στόχοι: ελαχιστοποίηση των διασυνδέσεων, μικρό μέγεθος κάθε ομάδας διαμέρισης
- Χωροθέτηση
 - Τοποθετούμε τις ομάδες διαμέρισης στο χώρο του ολοκληρωμένου
 - Τοποθέτηση ομάδων με πολλές διασυνδέσεις μεταξύ τους, σε κοντινά σημεία
- Τοποθέτηση και διασύνδεση
 - Καθορίζει τη θέση των λογικών κυττάρων στο χώρο της διαμέρισης
 - Στόχος: ελαχιστοποίηση διασυνδέσεων που θα γίνουν σε επόμενο στάδιο
- Διασύνδεση
 - Πραγματοποιούνται οι διασυνδέσεις ανάμεσα στα κύτταρα
 - Καθορισμός καθολικών και τοπικών διασυνδέσεων, υπολογισμός της διαδρομής τους, ελαχιστοποίηση συνολικής επιφάνειας (NP-complete)

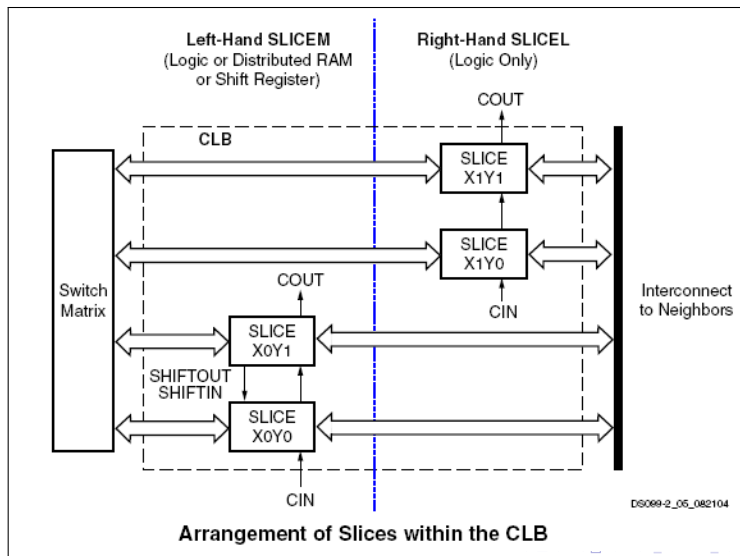
Αρχιτεκτονικά χαρακτηριστικά των συσκευών Xilinx Spartan-3

- Configurable Logic Blocks (CLBs) τα οποία περιέχουν RAM-based LUTs για την υλοποίηση λογικής και κατανεμημένης αποθήκευσης. Τα CLB διαθέτουν flip-flop/μανδαλωτές
- Μπλοκ εισόδου/εξόδου (Input/Output Blocks: IOBs) που ελέγχουν τη ροή των δεδομένων ανάμεσα στους ακροδέκτες της συσκευής και τα εσωτερικά στοιχεία λογικής
- Block RAM (BRAM) που παρέχουν τη δυνατότητα αποθήκευσης για 18-kbit (16-kbit data, 2-kbit parity bits)
- Ενσωματωμένοι πολλαπλασιαστές για εισόδους των 18-bit
- Ψηφιακός διαχειριστής ρολογιού (Digital Clock Manager: DCM) για την κατάλληλη ρύθμιση του ρολογιού προκειμένου την κατανομή του, για τον πολλαπλασιασμό/διαίρεση και την ολίσθηση ρολογιού

Η αρχιτεκτονική της οικογένειας Xilinx Spartan-3

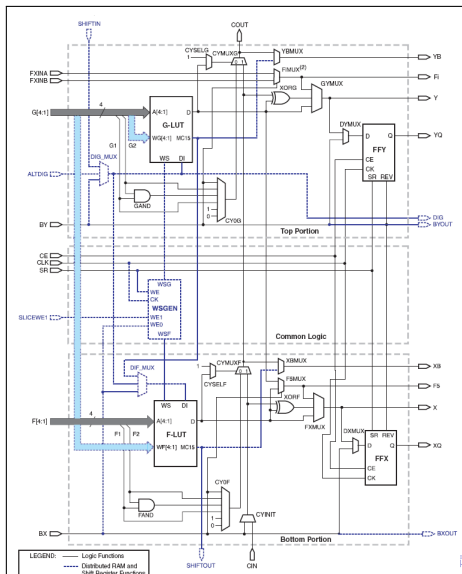


Spartan-3 CLB



Arrangement of Slices within the CLB

Η υπομονάδα SLICEM του Spartan-3 CLB



Παραδείγματα λογικής σύνθεσης με τη ροή Xilinx

- Στοχευόμενη συσκευή: Xilinx Spartan-3
- Αναπτυξιακή πλακέτα: Xilinx Spartan-3 starter kit
- Εργαλείο λογικής σύνθεσης: Xilinx ISE Webpack 9.2i
- Παραδείγματα
 - Πλήρης αθροιστής (1-bit)
 - Αθροιστής προσημασμένων (2's-complement) ακεραίων των 4-bit
 - Καταχωρητής των 8-bit
 - Μη προγραμματιζόμενος επεξεργαστής GCD (μέγιστου κοινού διαιρέτη)

Περιγραφή του πλήρους αθροιστή

```
`timescale 1 ns / 1 ns
module full_adder (a, b, cin, s, cout);
    input a;
    input b;
    input cin;
    output s;
    output cout;
    wire ab, bc, ac;

    xor (s, a, b, cin);
    or (cout, ab, bc, ac);
    and (ab, a, b);
    and (bc, b, cin);
    and (ac, a, cin);
endmodule
```

Αναφορά λογικής σύνθεσης (1)

```
...
Compiling verilog file "../fa.v" in library work
Module <full_adder> compiled
No errors in compilation
Analysis of file <"full_adder.prj"> succeeded.
Analyzing hierarchy for module <full_adder> in library <work>.
Analyzing top module <full_adder>.
Module <full_adder> is correct for synthesis.
Optimizing unit <full_adder> ...
Mapping all equations...
Building and optimizing final netlist ...
Found area constraint ratio of 100 (+ 5) on block full_adder, actual ratio is 0.
```

```
=====
*                               Final Report                               *
Design Statistics
# IOs                           : 5
Cell Usage :
# BELS                           : 2
#   LUT3                         : 2
# IO Buffers                     : 5
#   IBUF                         : 3
#   OBUF                         : 2
=====
```

Device utilization summary:

Selected Device : 3s200ft256-5

Number of Slices:	1	out of	1920	0%
Number of 4 input LUTs:	2	out of	3840	0%
Number of IOs:	5			
Number of bonded IOBs:	5	out of	173	2%

Αναφορά λογικής σύνθεσης (2)

TIMING REPORT

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.

=====
Timing constraint: Default path analysis

Total number of paths / destination ports: 6 / 2

Delay: 7.824ns (Levels of Logic = 3)

Source: b (PAD)

Destination: cout (PAD)

Data Path: b to cout

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
IBUF:I->O	2	0.715	1.040	b_IBUF (b_IBUF)
LUT3:I0->O	1	0.479	0.681	cout1 (cout_OBUF)
OBUF:I->O		4.909		cout_OBUF (cout)
Total		7.824ns	(6.103ns logic, 1.721ns route)	(78.0% logic, 22.0% route)

=====
CPU : 6.28 / 6.98 s | Elapsed : 6.00 / 7.00 s

-->

Total memory usage is 134872 kilobytes

Number of errors : 0 (0 filtered)

Number of warnings : 0 (0 filtered)

Number of infos : 0 (0 filtered)

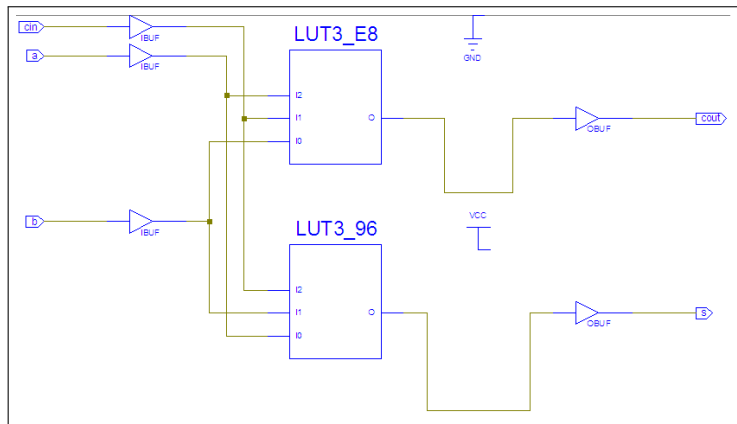
Λίστα κόμβων για τον πλήρη αθροιστή

```
`timescale 1 ns/1 ps
module full_adder (cout, a, b, s, cin);
  output cout;
  input a, b;
  output s;
  input cin;
  wire cout_OBUF_5;
  wire a_IBUF_6, b_IBUF_7;
  wire s_OBUF_8, cin_IBUF_9;
  wire N3, N4;
  defparam cout1.INIT = 8'hE8;
  X_LUT3 cout1 (
    .ADR0(b_IBUF_7),
    .ADR1(cin_IBUF_9),
    .ADR2(a_IBUF_6),
    .O(cout_OBUF_5)
  );
  defparam \Mxor_s_xo<1>1 .INIT = 8'h96;
  X_LUT3 \Mxor_s_xo<1>1 (
    .ADR0(a_IBUF_6),
    .ADR1(b_IBUF_7),
    .ADR2(cin_IBUF_9),
    .O(s_OBUF_8)
  );
);
```

```
X_BUF a_IBUF (.I(a), .O(a_IBUF_6));
X_BUF b_IBUF (.I(b), .O(b_IBUF_7));
X_BUF cin_IBUF (
  .I(cin),
  .O(cin_IBUF_9)
);
X_ZERO XST_GND (.O(N3));
X_ONE XST_VCC (.O(N4));
X_IPAD a_0 (.PAD(a));
X_IPAD b_1 (.PAD(b));
X_IPAD cin_2 (.PAD(cin));
X_OPAD cout_3 (.PAD(cout));
X_OPAD s_4 (.PAD(s));
X_OBUF cout_OBUF (
  .I(cout_OBUF_5),
  .O(cout)
);
X_OBUF s_OBUF (
  .I(s_OBUF_8),
  .O(s)
);
endmodule
```

Αυτόματα παραγόμενο σχηματικό του πλήρους αθροιστή

Σε επίπεδο λίστας κόμβων (για την τεχνολογία)



Περιγραφή του αθροιστή προσημασμένων αριθμών των 4-bit

```
`define DW      4
module add (a, b, sum);
  input[`DW-1:0] a;
  input[`DW-1:0] b;
  output[`DW-1:0] sum;
  wire[`DW:0] temp_sum;
  assign temp_sum = ({a[`DW-1], a}) + ({b[`DW-1], b});
  assign sum = temp_sum[`DW-1:0];
endmodule
```

Στατιστικά από τη σύνθεση

Device utilization summary:

Selected Device : 3s200ft256-5

Number of Slices:	3	out of	1920	0%
Number of 4 input LUTs:	6	out of	3840	0%
Number of bonded IOBs:	12	out of	173	6%

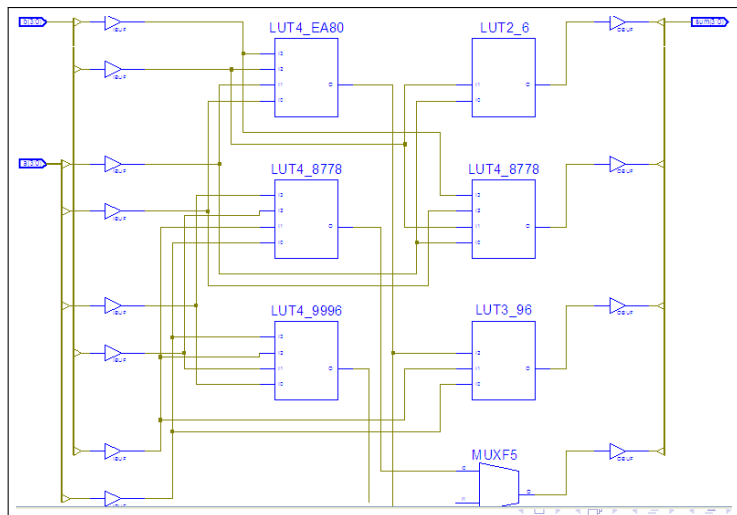
Timing Summary:

Delay: 9.110ns (Levels of Logic = 4)
Source: a<1> (PAD)
Destination: sum<3> (PAD)
Data Path: a<1> to sum<3>

Total 9.110ns (6.643ns logic, 2.467ns route)
(72.9% logic, 27.1% route)

Αυτόματα παραγόμενο σχηματικό για τον αθροιστή

Σε επίπεδο λίστας κόμβων (για την τεχνολογία)



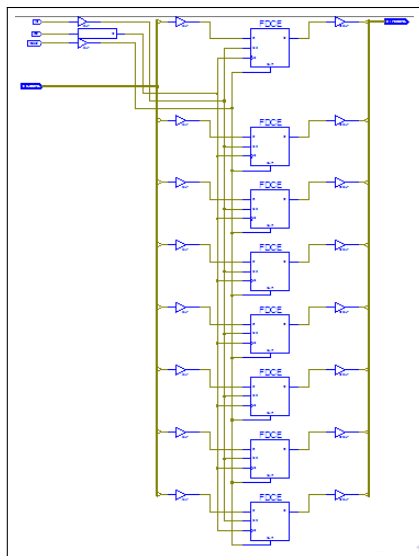
Περιγραφή του καταχωρητή των 8-bit

```
module register8 (clk, reset, ld, din, dout);
  input      clk, reset, ld;
  input [7:0] din;
  output [7:0] dout;
  reg [7:0]  dout;
  always @(posedge clk or posedge reset)
    if (reset == 1'b1)
      dout <= 0;
    else if (ld == 1'b1)
      dout <= din;
endmodule
```

Στατιστικά από τη σύνθεση

```
Selected Device : 3s200ft256-5
Number of Slices:                0 out of 1920    0%
Number of Slice Flip Flops:      8 out of 3840    0%
Number of bonded IOBs:           19 out of 173    10%
=====
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
Offset:                2.160ns (Levels of Logic = 1)
Source:                ld (PAD)
Destination:          dout_0 (FF)
=====
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
Offset:                6.216ns (Levels of Logic = 1)
Source:                dout_7 (FF)
Destination:          dout<7> (PAD)
```

Αυτόματα παραγόμενο σχηματικό για τον καταχωρητή



Περιγραφή της υλοποίησης FSMD του επεξεργαστή GCD (1)

```
module gcd (  
    clock, reset, start,  
    a, b,  
    outp, done);  
    parameter WIDTH = 8;  
    parameter s0 = 2'b00, s1 = 2'b01,  
              s2 = 2'b10, s3 = 2'b11;  
    input clock, reset, start;  
    input [WIDTH-1:0] a, b;  
    output [WIDTH-1:0] outp;  
    output done;  
    reg [WIDTH-1:0] outp;  
    reg done;  
    reg [1:0] state;  
    reg [WIDTH-1:0] x, y;  
  
    always @(posedge clock or posedge reset)  
    begin  
        done = 1'b0;  
        if (reset) begin  
            state = s0;  
            x = 0;  
            y = 0;  
            done = 1'b0;  
            outp = 0;  
        end  
end
```

```
    else begin  
        case (state)  
            s0: begin  
                if (start) begin  
                    x = a;  
                    y = b;  
                    state = s1;  
                end  
            end  
            else  
                state = s0;  
            end  
            s1: begin  
                if (x != 0 && y != 0)  
                    state = s2;  
                else begin  
                    outp = 0;  
                    state = s3;  
                end  
            end  
        end  
end
```

Περιγραφή της υλοποίησης FSMD του επεξεργαστή GCD (2)

```
s2: begin
  if (x > y) begin
    x = x - y;
    state = s2;
  end
  else if (x < y) begin
    y = y - x;
    state = s2;
  end
  else begin
    outp = x;
    state = s3;
  end
end
s3: begin
  done = 1'b1;
  state = s0;
end
endcase
end
endmodule
```

Αναφορά λογικής σύνθεσης για τον επεξεργαστή GCD

Found finite state machine <FSM_0> for signal <state>.

```
-----  
| States           | 4 |  
| Transitions     | 9 |  
| Inputs          | 5 |  
| Outputs         | 4 |  
| Clock           | clock (rising_edge) |  
| Reset           | reset (positive), asynchronous |  
| Reset State     | 00 |  
-----
```

Device utilization summary:

Selected Device : 3s200ft256-5

Number of Slices:	55	out of	1920	2%
Number of Slice Flip Flops:	28	out of	3840	0%
Number of 4 input LUTs:	106	out of	3840	2%
Number of IOs:	28			
Number of bonded IOBs:	28	out of	173	16%
Number of GCLKs:	1	out of	8	12%

Timing Summary:

Minimum period: 7.213ns (Maximum Frequency: 138.631MHz)
Minimum input arrival time before clock: 6.529ns
Maximum output required time after clock: 6.280ns
Maximum combinational path delay: No path found

Αυτόματα παραγόμενο σχηματικό για τον επεξεργαστή GCD

