

## HY430 – Εργαστήριο Ψηφιακών Κυκλωμάτων

Διδάσκων: Χ. Σωτηρίου, Βοηθός: (θα ανακοινωθεί)

<http://inf-server.inf.uth.gr/courses/CE430/>

1

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 2

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 3

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Τύποι Μοντελοποίησης Verilog

Δομική	<pre>module_name instance_name [instance_array_range] (signal, signal, ... );  module_name instance_name [instance_array_range] (.port_name(signal), (.port_name(signal), ...));</pre>	
(Structural)	<pre>counter counter_1( clk, enable, count_out); dff u2 (.clk(clock), .q(q[1]), .data(d[1]));</pre>	
Ροή Δεδομένων	<pre>gate_type #(delay) instance_name [instance_array_range] (terminal, terminal, ...);</pre>	
(Dataflow)	<pre>wire = (a &amp; b)   (c &amp; d);</pre>	
Συμπεριφοράς ή Διαδικαστική	<pre>initial   always @ (sensitivity list) begin--end</pre>	
RTL	<pre>always @(a or b or ci) begin     sum = a + b + ci; end</pre>	<pre>initial begin     bus = 16'h0000; #10 bus = 16'hC5A5;     #20 bus = 16'hFFAA; end</pre>

Συνθέσιμη;;

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιγραφή Συμπεριφοράς

- ▶ Χρήση Διαδικασιών
  - ▶ always ή initial
- ▶ Διαθέσιμες προτάσεις
  - ▶ for
  - ▶ while
  - ▶ Συναρτήσεις
  - ▶ Διεργασίες
  - ▶ fork – join
- ▶ Διαθέσιμοι Τύποι
  - ▶ reg και πίνακες reg
  - ▶ integer
  - ▶ real

```
initial begin
    // reset everything
end

always @(posedge clk) begin
    case (opcode)
        8'hAB: RegFile[dst] = #2 in;
        8'hEF: dst = #2 in0 + in1;
        8'h02: Memory[addr] = #2 data;
    endcase

    if (branch)
        dst = #2 br_addr;
    end
```

▶ 5

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιγραφή Συμπεριφοράς

- ▶ Απεικόνιση στο υλικό;;;
  - ▶ Το εργαλείο σύνθεσης κάνει αντιστοιχία
  - ▶ Μπορεί να ελαχιστοποιήσει την δυαδική λογική
  - ▶ Υπακούει όμως την δομική συμπεριφορά της περιγραφής



```
integer sum, i;
integer opcodes [31:0];
real average;

initial
    for (i=0; i<32; i=i+1)
        opcodes[i] = 0;

always @(posedge clk) begin
    sum = sum + 1;
    average = average + (c / sum);
    opcodes[d] = sum;
    $display("sum: %d, avg: %f",
        sum, average);
end
```

▶ 6

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

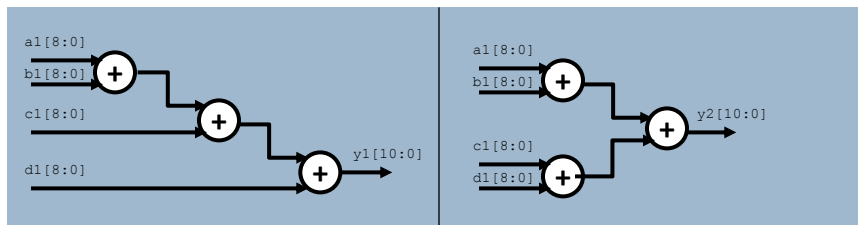
## Επιρροή Δομής

### Περιγραφή Verilog

```
module COMB_LOGIC_STRUCT(a1, b1, c1, d1, a2, b2, c2, d2, y1, y2);
  input [8:0] a1, b1, c1, d1, a2, b2, c2, d2;
  output [10:0] y1, y2;
  reg [10:0] y1, y2;

  always @(a1 or b1 or c1 or d1 or a2 or b2 or c2 or d2)
  begin
    y1 = a1 + b1 + c1 + d1;
    y2 = (a2 + b2) + (c2 + d2);
  end
endmodule
```

### Κυκλωματική Μορφή



► 7

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Παραδείγματα Περιγραφής Συμπεριφοράς

```
module test;

  task ShowValues;
    input [7:0] data;
    $display(..., data);
  endtask

  ...
  always @(posedge clk)
    ShowValues(counter);
  ...
endmodule
```

```
`define period 20

initial begin
  reset_ = 1'b0;
  reset_ = #(2*`period + 5) 1'b1;

  @(branch);
  reset_ = 1'b0;
  reset_ = #(2*`period + 5) 1'b1;
end
```

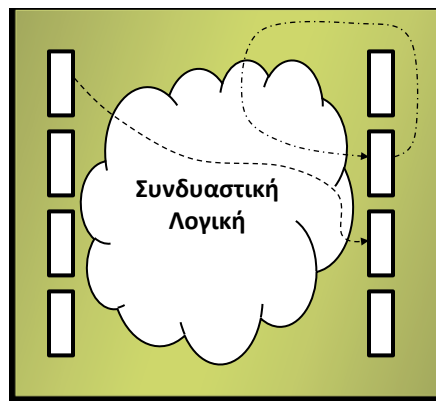
```
always @(negedge reset_) begin
  fork
    a = #2 8'h44;
    b = #(4*`period + 2) 1'b0;
    c = #(16*`period + 2) 8'h44;
  join
end
```

► 8

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιγραφή RTL (Register Transfer Level – Μεταβίβασης μεταξύ Καταχωρητών)

- ▶ Το κάθε τμήμα **always**
  - ▶ κάνει αναθέσεις
    - ▶ από καταχωρητές
    - ▶ σε καταχωρητές
- ▶ Σχεδιάζουμε
  - ▶ Κύκλο προς κύκλο...
  - ▶ Με οδηγό το ρολόι
  - ▶ 1<sup>ος</sup> κύκλος: ...
  - ▶ 2<sup>ος</sup> κύκλος: ...



▶ 9

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Δομική Περιγραφή

- ▶ Μόνο εμφανίσεις τμημάτων
- ▶ Μορφή Πλαισίου Δοκιμής
- ▶ Μορφή μετά από Σύνθεση

```

module top;
wire clk, reset;
wire [31:0] d_data, I_data;
wire [9:0] d_adr;
wire [5:0] i_adr;

  clock clk0(clk);
  processor pr0(clk, reset,
                d_adr, d_data,
                i_adr, i_data,
                ...);

  memory mem0(d_adr,
              d_data);

  memory mem1(i_adr, i_data) ;
  tester tst0(reset, ...);

endmodule

```

▶ 10

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Καλές Πρακτικές

### ▶ Ονόματα

- ▶ Συνοχή
- ▶ Νόημα
- ▶ Ευανάγνωστα

```
wire a,
      memory_data_write_enable;
wire mem_wr_en,
```

### ▶ Συνδυαστική Λογική

- ▶ Ευανάγνωστη

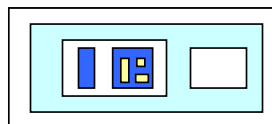
```
if (~req &&
    ((flag & prv_ack) |
     ~set) &&
    (count-2 == 0))
    ...
```

### ▶ Δομή

- ▶ Φυσική και όχι Λογική  
Ιεράρχηση του σχεδίου

### ▶ Συνδεσμολογία

- ▶ Κατά όνομα αντί Κατά θέση



- ▶ Χρήση κενών/tabs για ευθυγράμμιση του κώδικα

▶ 11

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Καλές Πρακτικές

### ▶ Σχόλια

- ▶ Κώδικας υλικού γενικά πιο δύσκολος στην κατανόηση
- ▶ Ακόμα και ο ίδιος ο σχεδιαστής σε 2 εβδομάδες δεν θυμάται απόλυτα την λειτουργία του κώδικα
- ▶ Αν δεν χρησιμοποιηθούν εν γένει δεν θα μπουν αργότερα

### ▶ Καλή πρακτική

- ▶ Σε κάθε μονάδα
- ▶ Σε κάθε τμήμα `always`

```
/******
 * Comments on module test:
 * Module test comprises of
 * the following components...
 *****/
module test;
// Line comment
```

▶ 12

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ **Καμπύλη Παρέτο**
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

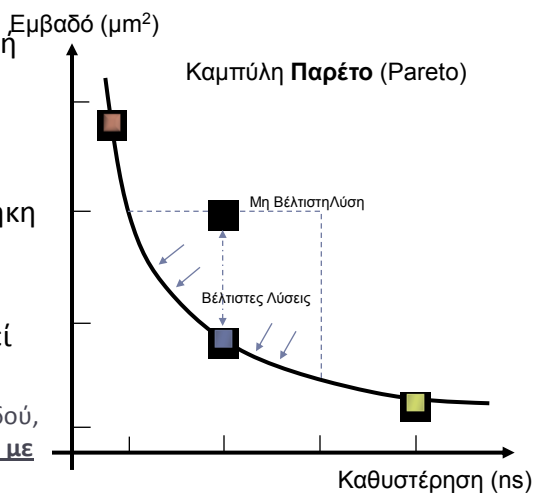
▶ 13

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Verilog και Λογική Σύνθεση

- ▶ Η διαδικασία Σύνθεσης μετατρέπει την περιγραφή Verilog σε:

1. Περιγραφή σε επίπεδο πυλών
2. Απεικόνιση σε βιβλιοθήκη πυλών
  - ▶ επιλογή μεγεθών
- ▶ Η διαδικασία ακολουθεί μια καμπύλη Παρέτο
  - ▶ Πολλαπλά σημεία (Εμβαδού, Καθυστερήσης) ανάλογα με τους περιορισμούς



▶ 14

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Συνθέσιμο Υποσύνολο της Verilog

Εκφράσεις	Λέξεις Κλειδιά	Σχόλια
Ορισμός Θυρών	input, output, inout	
Παράμετροι	parameter	
Ορισμός Μονάδων	module -- endmodule	
Σήματα και Μεταβλητές	wire, reg	
Εμφανίσεις	Εμφανίσεις Μονάδων, Θεμελιώδεις Πύλες	mymux(sel, a, b, out); nand nand4_i (out, a, b, c, d);
Προτάσεις Διαδικασιών	always, if, else, case	Όχι initial
Τμήματα Διαδικασιών	begin -- end	
Ροή Δεδομένων	assign	Αγνοούνται οι Καθυστερήσεις
Τελεστές	+, -, ~, &, ^, ==, != ...	Προσοχή: *, /, %
Συναρτήσεις, Διαδικασίες	function, task	Μόνο Συνδυαστική Λογική
Βρόχοι	for, while	Μόνο για αναθέσεις, εμφανίσεις

▶ 15

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ **D FF**
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 16

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση



## Καταχωρητές

### Περιγραφή Verilog

```

module Reg(Q, D, Clk);

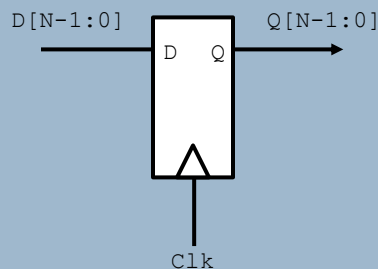
parameter N = 16;
input      Clk;
input  [N-1:0] D;
output [N-1:0] Q;
reg      [N-1:0] Q;

    always @(posedge Clk)
        Q <= D;

endmodule

```

### Κυκλωματική Μορφή



► 17

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Καταχωρητές

### Περιγραφή Verilog

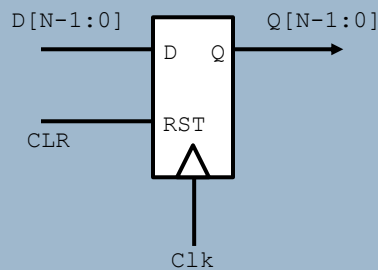
```

module RegRst(Q, D, CLR,
Clk);
parameter N = 16;
input  CLR, Clk;
input  [N-1:0] D;
output [N-1:0] Q;
reg    [N-1:0] Q;

    always @(posedge Clk or
posedge CLR)
    begin
        if (CLR)
            Q <= 0;
        else
            Q <= D;
    end
endmodule

```

### Κυκλωματική Μορφή



Ασύγχρονη αρχικοποίηση

► 18

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Καταχωρητές

### Περιγραφή Verilog

```

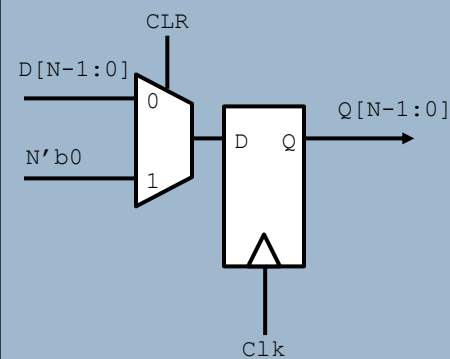
module Reg(Q, D, CLK, Clk);

parameter N = 16;
input      Clk;
input  [N-1:0] D;
output [N-1:0] Q;
reg      [N-1:0] Q;

    always @(posedge Clk)
    begin
        if (CLR)
            Q <= 0;
        else
            Q <= D;
    end
endmodule

```

### Κυκλωματική Μορφή



Σύγχρονη αρχικοποίηση

► 19

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Καταχωρητές

### Περιγραφή Verilog

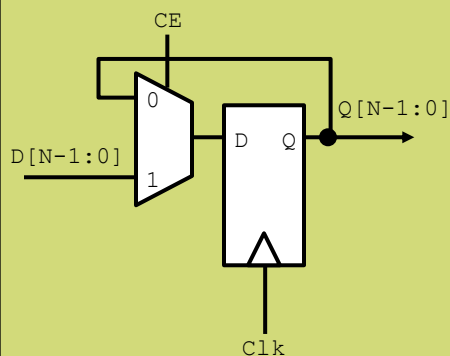
```

module RegId(Q, D, CE,
Clk);
parameter N = 16;
input  CE, Clk;
input  [N-1:0] D;
output [N-1:0] Q;
reg    [N-1:0] Q;

    always @(posedge Clk)
    begin
        if (CE)
            Q <= D;
    end
endmodule

```

### Κυκλωματική Μορφή



► 20

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Καταχωρητές

### Περιγραφή Verilog

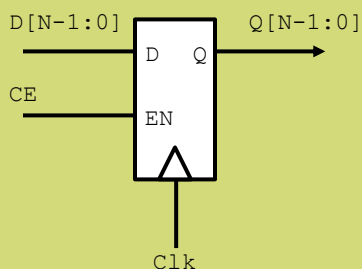
```

module RegLd(Q, D, CE,
  Clk);
  parameter N = 16;
  input  CE, Clk;
  input  [N-1:0] D;
  output [N-1:0] Q;
  reg    [N-1:0] Q;

  always @(posedge Clk)
  begin
    if (CE)
      Q <= D;
  end
endmodule

```

### Κυκλωματική Μορφή



Διατήρηση των Δεδομένων =>  
Χαμηλότερη Κατανάλωση

► 21

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Καταχωρητές

### Περιγραφή Verilog

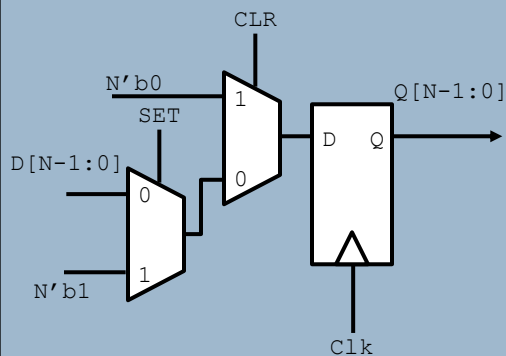
```

module Reg(Q, D, CLK,
  SET, Clk);
  parameter N = 16;
  input  Clk;
  input  [N-1:0] D;
  output [N-1:0] Q;
  reg    [N-1:0] Q;

  always @(posedge Clk)
  begin
    if (CLR)
      Q <= 0;
    else if (SET)
      Q <= 1;
    else
      Q <= D;
  end
endmodule

```

### Κυκλωματική Μορφή



► 22

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ **T FF**
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 23

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Καταχωρητές

### Περιγραφή Verilog

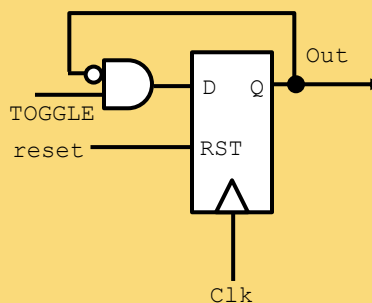
```

module Tff(Out, Toggle,
  Clk);

output Out;
input Toggle, Clk;
reg Out;
  always @(posedge Clk or
posedge reset)
    if (reset)
      Out <= 0;
    else
      if (Toggle)
        Out <= ~Out;
endmodule

```

### Κυκλωματική Μορφή



▶ 24

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ **Μετρητές**
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 25

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Μετρητής

### Περιγραφή Verilog

```

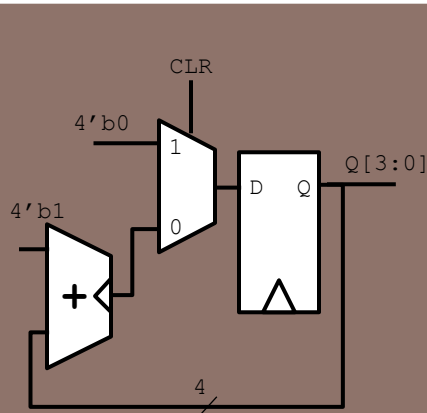
module counter (C, CLR, Q);
input C, CLR;
output [3:0] Q;
reg [3:0] tmp;

always @(posedge C)
begin
  if (CLR)
    tmp <= 4'b0000;
  else
    tmp <= tmp + 1'b1;
end

assign Q = tmp;
endmodule

```

### Κυκλωματική Μορφή



▶ 26

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Μετρητής

### Περιγραφή Verilog

```

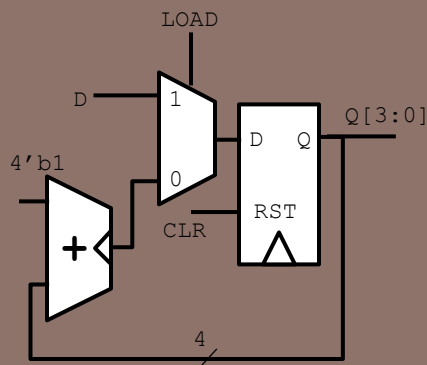
module counter(C, D, LOAD,
Q);
input C, D, LOAD;
output [3:0] Q;
reg [3:0] tmp;

always @(posedge C or
posedge CLR)
begin
    if (LOAD)
        tmp <= D;
    else
        tmp <= tmp + 1'b1;
    end

assign Q = tmp;
endmodule

```

### Κυκλωματική Μορφή



► 27

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Μετρητής

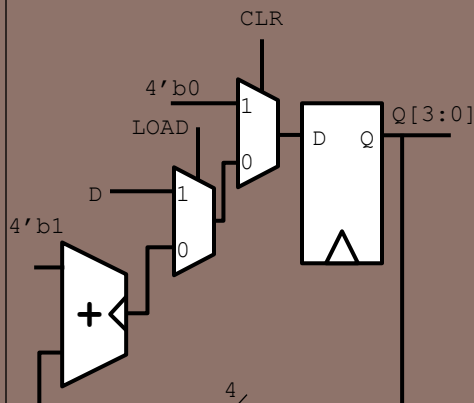
### Περιγραφή Verilog

```

module counter(C, D, LOAD,
CLR, Q);
input C, D, LOAD;
output [3:0] Q;
reg [3:0] tmp;
always @(posedge C)
begin
    if (CLR)
        tmp = 4'b0;
    else if (LOAD)
        tmp <= D;
    else
        tmp <= tmp + 1'b1;
    end
assign Q = tmp;
endmodule

```

### Κυκλωματική Μορφή



► 28

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Μετρητής

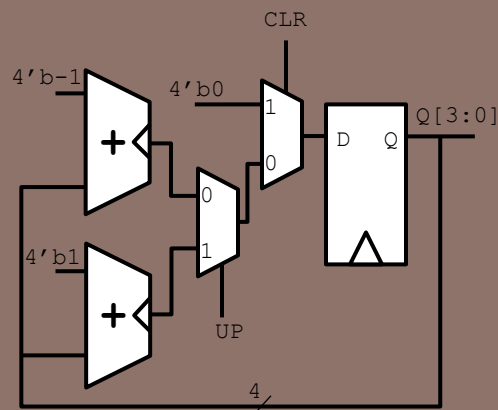
### Περιγραφή Verilog

```

module counter(C, D,
UP, CLR, Q);
input C, D, LOAD;
output [3:0] Q;
reg [3:0] tmp;
always @(posedge C)
begin
  if (CLR)
    tmp = 4'b0;
  else if (UP)
    tmp <= tmp + 1'b1;
  else
    tmp <= tmp - 1'b1;
end
assign Q = tmp;
Endmodule

```

### Κυκλωματική Μορφή



► 29

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Μετρητής

### Περιγραφή Verilog

```

module Cnt(Out, Zero, En, Clear, Clk);
parameter N = 32;
parameter MaxCnt = 100;
input En, Clear, Clk;
output Zero;
output [N-1:0] Out;
reg [N-1:0] Out;
reg Zero;
always @(posedge Clk) begin
  if(Clear) begin
    Out <= 0;
    Zero <= 0;
  end
  else if (En) begin
    if (Out == MaxCnt) begin
      Out <= 0;
      Zero <= 1;
    end
    else begin
      Out <= Out + 1;
      Zero <= 0;
    end
  end
end
endmodule

```

► Τι λειτουργία έχει αυτό το κύκλωμα;

► Τι κυκλωματική μορφή;

► 30

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ **Συσσωρευτής**
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 31

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Συσσωρευτής

### Περιγραφή Verilog

```

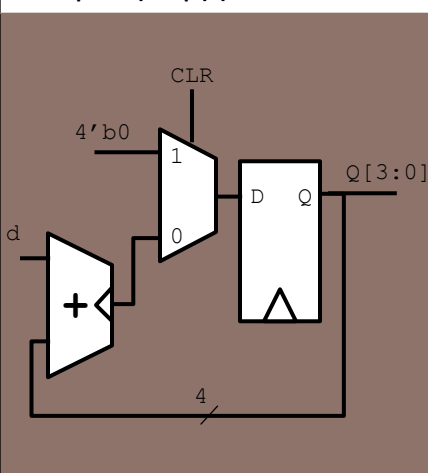
module counters (C, CLR, Q);
input C, CLR;
output [3:0] Q;
reg [3:0] tmp;

always @(posedge C)
begin
  if (CLR)
    tmp <= 4'b0000;
  else
    tmp <= tmp + d;
end

assign Q = tmp;
endmodule

```

### Κυκλωματική Μορφή



▶ 32

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση



## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ **Ολισθητές**
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 33

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Ολισθητής

### Περιγραφή Verilog

```

module shift(CLK, SI, SO);
input C, SI;
output SO;
reg [7:0] tmp;

always @(posedge CLK)
begin
    tmp = {tmp[6:0], SI};
end

assign SO = tmp[7];
endmodule

```

### Κυκλωματική Μορφή



▶ 34

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση



## Ολισθητής

### Περιγραφή με συνένωση

```
module SHIFT2(CLK, MULT,
LOAD, D, F);
input C, MULT, LOAD;
input [7:0] D;
output [7:0] F;
reg [7:0] tmp;
always @(posedge CLK)
begin
    if (LOAD)
        tmp = D;
    else if (MULT)
        tmp = {tmp[5:0], 2'b00};
    else
        tmp = {2'b00, tmp[7:2]};
end
assign F = tmp;
endmodule
```

### Περιγραφή με τελεστή ολισθησης

```
module SHIFT2(CLK, MULT,
LOAD, D, F);
input C, MULT, LOAD;
input [7:0] D;
output [7:0] F;
reg [7:0] tmp;
always @(posedge CLK)
begin
    if (LOAD)
        tmp = D;
    else if (MULT)
        tmp = tmp << 2;
    else
        tmp = tmp >> 2;
end
assign F = tmp;
endmodule
```

► 37

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

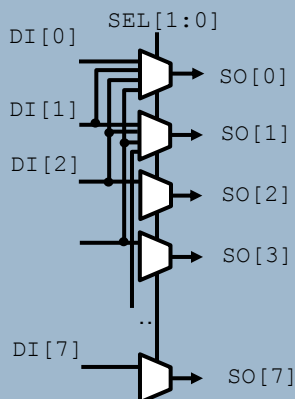
## Ολισθητής

### Περιγραφή Verilog

```
module SHIFT(DI, SEL, SO);
input [7:0] DI;
output [1:0] SEL;
reg [7:0] SO;

always @(DI or SEL)
begin
    case (SEL)
        2'b00 : SO = DI;
        2'b01 : SO = DI << 1;
        2'b10 : SO = DI << 2;
        default : SO = DI << 3;
    endcase
end
endmodule
```

### Κυκλωματική Μορφή



► 38

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιστροφικός Ολισθητής (Barrel Shift)

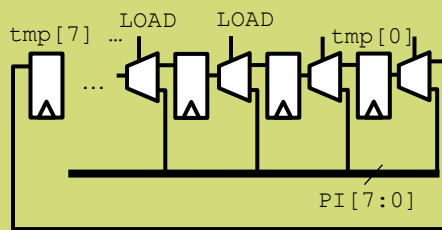
### Περιγραφή Verilog

```

module BARRELSHIFT(CLK, D,
LOAD, SI, SO);
input C, LOAD;
input [7:0] D;
output SO;
reg [7:0] tmp;
always @(posedge CLK or
posedge reset)
begin
if (reset)
    tmp = 8'b0;
else if (LOAD)
    tmp = D;
else
    tmp = {tmp[6:0], tmp[7]};
end
endmodule

```

### Κυκλωματική Μορφή



► 39

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Παραμετρικός Περιστροφικός Ολισθητής

```

module BarShiftReg(Out, In, Ld, Shift, Clk, Reset);
parameter N = 32;
input Ld, Shift, Clk, Reset;
input [N-1:0] In;
output [N-1:0] Out;
reg [N-1:0] Out;
always @(posedge Clk) begin
    if (~Reset) Out <= 0;
    else begin
        if (Ld) Out <= In;
        else if (Shift) begin
            Out <= {Out[N-2:0], Out[N-1]};
        end
    end
end
Endmodule

```

► 40

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 41

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Πολυπλέκτης

### Περιγραφή Verilog

```

module mux2(Out, In1, In0,
Sel);

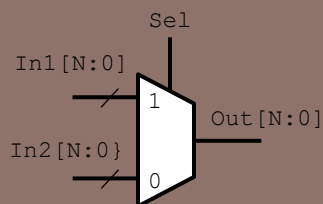
parameter N = 16;
output [N-1:0] Out;
input [N-1:0] In1, In0;
input Sel;

    wire [N-1:0] Out = Sel ?
In1 : In0;

endmodule

```

### Κυκλωματική Μορφή



▶ 42

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Πολυπλέκτης

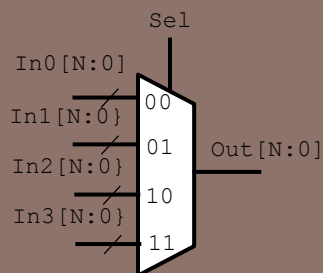
### Περιγραφή Verilog

```

module mux4(Out, In3, In2, In1,
In0, Sel);
parameter N = 32;
input  [ 1:0] Sel;
input  [N-1:0] In3, In2, In1,
In0;
output [N-1:0] Out;
reg    [N-1:0] Out;
always @(In0 or In1 or In2 or
In3 or Sel) begin
    case ( Sel )
        2'b00 : Out <= In0;
        2'b01 : Out <= In1;
        2'b10 : Out <= In2;
        2'b11 : Out <= In3;
    endcase
end
endmodule

```

### Κυκλωματική Μορφή



► 43

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Πολυπλέκτης

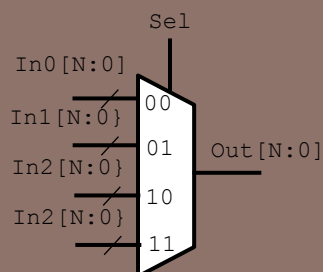
### Περιγραφή Verilog

```

module mux4(Out, In3, In2, In1,
In0, Sel);
parameter N = 32;
input  [ 1:0] Sel;
input  [N-1:0] In3, In2, In1,
In0;
output [N-1:0] Out;
reg    [N-1:0] Out;
always @(In0 or In1 or In2 or
In3 or Sel) begin
    case ( Sel )
        2'b00 : Out <= In0;
        2'b01 : Out <= In1;
        default : Out <= In2;
    endcase
end
endmodule

```

### Κυκλωματική Μορφή



► 44

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 45

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Κωδικοποιητής

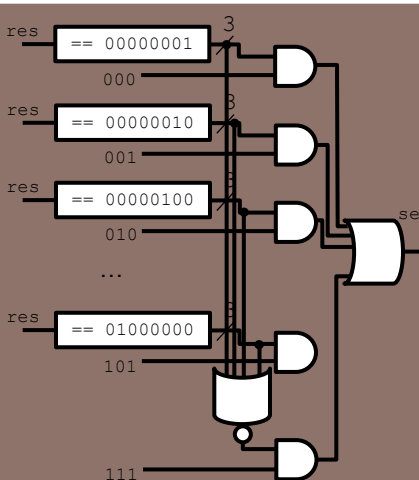
### Περιγραφή Verilog

```

module encoder (sel, res);
input  [7:0] res;
output [2:0] sel;
reg [7:0] sel;
always @(res)
begin
  case (res)
    8'b00000001 : sel = 3'b000;
    8'b00000010 : sel = 3'b001;
    8'b00000100 : sel = 3'b010;
    8'b00001000 : sel = 3'b011;
    8'b00010000 : sel = 3'b100;
    8'b00100000 : sel = 3'b101;
    8'b01000000 : sel = 3'b101;
    default      : sel = 3'b111;
  endcase
end
endmodule

```

### Κυκλωματική Μορφή



▶ 46

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Από-κωδικοποιητής

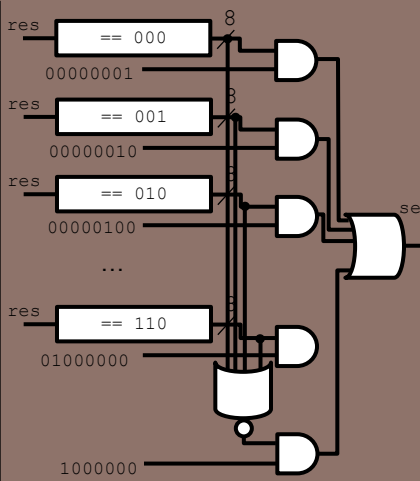
### Περιγραφή Verilog

```

module decoder (sel, res);
input  [2:0] sel;
output [7:0] res;
reg [7:0] res;
always @(sel)
begin
  case (sel)
    3'b000 : res = 8'b00000001;
    3'b001 : res = 8'b00000010;
    3'b010 : res = 8'b00000100;
    3'b011 : res = 8'b00001000;
    3'b100 : res = 8'b00010000;
    3'b101 : res = 8'b00100000;
    3'b110 : res = 8'b01000000;
    default : res = 8'b10000000;
  endcase
end
endmodule

```

### Κυκλωματική Μορφή



► 47

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Κωδικοποιητής

### Υλοποίηση με case

```

module encoder (sel, res);
input  [7:0] res;
output [2:0] sel;
reg [7:0] sel;
always @(res)
begin
  case (res)
    8'b00000001 : sel = 3'b000;
    8'b00000010 : sel = 3'b001;
    8'b00000100 : sel = 3'b010;
    8'b00001000 : sel = 3'b011;
    8'b00010000 : sel = 3'b100;
    8'b00100000 : sel = 3'b101;
    8'b01000000 : sel = 3'b101;
    default      : sel = 3'b111;
  endcase
end
endmodule

```

### Υλοποίηση με for

```

module encoder(In, Out);
input  [7:0] In;
output [2:0] Out;
reg [2:0] Out;
integer i;

always @(In) begin
  Out = 0;
  for (i=0; i<8; i=i+1)
    if (In[i])
      Out=i;
end
endmodule

```

► 48

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση



## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ **Κωδικοποίηση με Προτεραιότητες**
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 49

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Από-κωδικοποιητής

### Υλοποίηση με case

```

module decoder (sel, res);
input  [2:0] sel;
output [7:0] res;
reg [7:0] res;
always @(sel)
begin
  case (sel)
    3'b000 : res = 8'b00000001;
    3'b001 : res = 8'b00000010;
    3'b010 : res = 8'b00000100;
    3'b011 : res = 8'b00001000;
    3'b100 : res = 8'b00010000;
    3'b101 : res = 8'b00100000;
    3'b110 : res = 8'b01000000;
    default : res = 8'b10000000;
  endcase
end
endmodule

```

### Υλοποίηση με for

```

module decoder(In, Out);
input  [2:0] In;
output [7:0] Out;
reg [7:0] Out;
reg [7:0] tmp;
integer i;

  always @(In) begin
    tmp = 0;
    for (i=0; i<8; i=i+1)
      if (In == i)
        tmp[i]=1;
    Out = tmp;
  end
endmodule

```

▶ 50

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Κωδικοποιητής με if – Λογική με Προτεραιότητες

### Περιγραφή Verilog

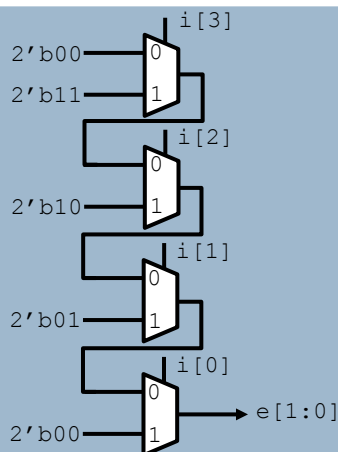
```

module encoder(i, e);
input [3:0] i;
output [1:0] e;
reg [1:0] e;

always @(i)
begin
if (i[0]) e = 2'b00;
else if (i[1]) e = 2'b01;
else if (i[2]) e = 2'b10;
else if (i[3]) e = 2'b11;
else e = 2'bxx;
end
endmodule

```

### Κυκλωματική Μορφή



► 51

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Κωδικοποιητής με if – Λογική με Προτεραιότητες

### Περιγραφή Verilog

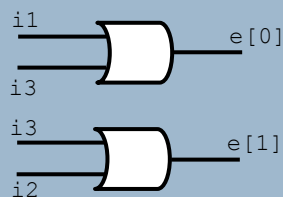
```

module encoder(i, e);
input [3:0] i;
output [1:0] e;
reg [1:0] e;

always @(i)
begin
if (i == 4'b0001) e = 2'b00;
else if (i == 4'b0010) e = 2'b01;
else if (i == 4'b0100) e = 2'b10;
else if (i == 4'b1000) e = 2'b11;
else e = 2'bxx;
end
endmodule

```

### Κυκλωματική Μορφή



► 52

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ **Αθροιστές**
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 53

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Αθροιστής

### Περιγραφή Verilog

```

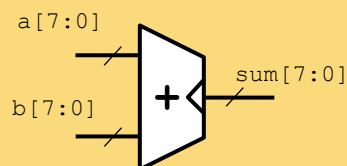
module adder (a, b, sum);
input    [7:0] a, b;
output   [7:0] sum;

assign sum = a + b;

endmodule

```

### Κυκλωματική Μορφή



▶ 54

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Αθροιστής με κρατούμενο

### Περιγραφή Verilog

```

module adder(a, b, sum);
input    [7:0] a, b;
output   [7:0] sum;
output    co;

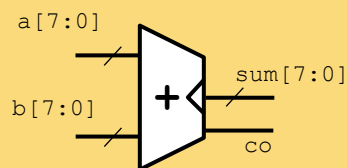
wire     [8:0] temp;

assign temp = a + b;
assign sum = temp[7:0];
assign co = temp[8];

endmodule

```

### Κυκλωματική Μορφή



► 55

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Αθροιστής με πρόσημο

### Περιγραφή Verilog

```

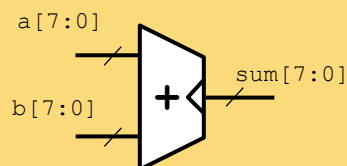
module adder(a, b, sum);
input signed [7:0] a, b;
output signed [7:0] sum;

assign sum = a + b;

endmodule

```

### Κυκλωματική Μορφή



► 56

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 57

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Συγκριτής

### Περιγραφή Verilog

```

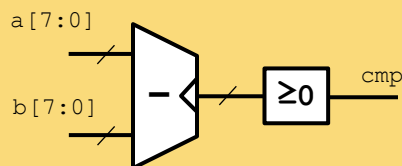
module comparator(a, b, cmp);
input    [7:0] a,b;
output    cmp;

assign cmp =
  (a>b) ? 1'b1 : 1'b0;

endmodule

```

### Κυκλωματική Μορφή



▶ 58

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ **Ανιχνευτής Ακμής**
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 59

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Ανιχνευτής Ακμής

### Περιγραφή Verilog

```

module posedgdet (Out, In,
  Clk);

  input  In, Clk;
  output Out;

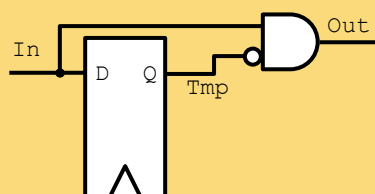
  reg Tmp;
  always @ (posedge Clk)
    Tmp <= In;

  wire Out = ~Tmp & In;

endmodule

```

### Κυκλωματική Μορφή



▶ 60

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ **D Latch**
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 61

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Μανταλωτής

### Περιγραφή Verilog

```

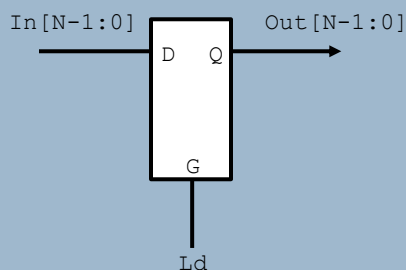
module Latch(In, Out, Ld);
parameter N = 16;
input    [N-1:0] In;
input    Ld;
output   [N-1:0] Out;

reg      [N-1:0] Out;

    always @(In or Ld)
        if (Ld)
            Out = In;
endmodule

```

### Κυκλωματική Μορφή



▶ 62

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Περιεχόμενα

- ▶ Περιγραφές και Συνθέσιμες Δομές
- ▶ Καμπύλη Παρέτο
- ▶ D FF
  - ▶ Σύγχρονη, ασύγχρονη αρχικοποίηση
  - ▶ Με ενεργοποίηση (enable)
- ▶ T FF
- ▶ Μετρητές
  - ▶ Με ανάθεση τιμής
  - ▶ Πάνω/Κάτω
- ▶ Συσσωρευτής
- ▶ Ολισθητές
  - ▶ Σειριακή είσοδος, Παράλληλη έξοδος
  - ▶ Παράλληλη είσοδος, Σειριακή έξοδος
  - ▶ Πολλαπλασιαστής/Διαφρέτης
  - ▶ Ολισθητής «Βαρέλι» (Barrel)
- ▶ Πολυπλέκτες
- ▶ Κωδικοποιητές/Από-κωδικοποιητές
- ▶ Κωδικοποίηση με Προτεραιότητες
- ▶ Αθροιστές
- ▶ Συγκριτές
- ▶ Ανιχνευτής Ακμής
- ▶ D Latch
- ▶ Σύγχρονη, Ασύγχρονη Μνήμη

▶ 63

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Σύγχρονη RAM

### Περιγραφή Verilog

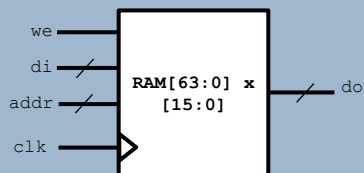
```

module ramsync (clk,we,addr,
di, do);
input clk, we, en;
input [5:0] addr;
input [15:0] di;
output [15:0] do;
reg [15:0] do;
reg [15:0] RAM [63:0];

always @(posedge clk)
begin
  if (we)
    RAM[addr]<=di;
  do <= RAM[addr];
end
endmodule

```

### Κυκλωματική Μορφή



▶ 64

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση



## Ασύγχρονη RAM ως προς την ανάγνωση

### Περιγραφή Verilog

```

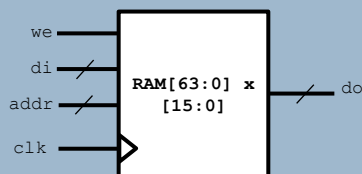
module ramsync (clk,we,addr,
di, do);
input clk, we, en;
input [5:0] addr;
input [15:0] di;
output [15:0] do;
wire [15:0] do;
reg [15:0] RAM [63:0];

always @(posedge clk)
begin
    if (we)
        RAM[addr]<=di;
end

assign do <= RAM[addr];
endmodule

```

### Κυκλωματική Μορφή



65

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση

## Synthesis Tool Manual

### Verilog While Statements Supported in XST

Table 7-19: Verilog While Statements Supported in XST

While Statement	Supported/Unsupported
All While statements	Supported

### Verilog For Statements Supported in XST

Table 7-20: Verilog For Statements Supported in XST

For Statement	Supported/Unsupported
All For statements	Supported (bounds must be static)

### Verilog Fork/Join Statements Supported in XST

Table 7-21: Verilog Fork/Join Statements Supported in XST

Fork/Join Statement	Supported/Unsupported
All Fork/Join statements	Unsupported

### Verilog Timing Control on Procedural Assignments Supported in XST

Table 7-22: Verilog Timing Control on Procedural Assignments Supported in XST

Timing Control on Procedural Assignment	Supported/Unsupported
delay (#)	Ignored
event (@)	Unsupported

Διαβάστε τι  
υποστηρίζει και  
τι όχι το  
synthesis tool



66

HY430 - Διάλεξη 7η - Verilog III - Σύνθεση