

## Laboratorul 3

### Modelarea circuitelor combinaționale

#### Introducere

Se consideră un circuit combinațional cu  $n$  intrări și  $m$  ieșiri. În Fig. 1 este dată o reprezentare prin schemă bloc a circuitului. Pentru acest circuit putem scrie setul de funcții:

$$y_1 = f_1(x_1, x_2, \dots, x_n)$$

$$y_2 = f_2(x_1, x_2, \dots, x_n)$$

.....

$$y_m = f_m(x_1, x_2, \dots, x_n)$$

unde  $f_1, f_2, \dots, f_m$  sunt funcții booleene cu argumentele  $x_1, x_2, \dots, x_n$

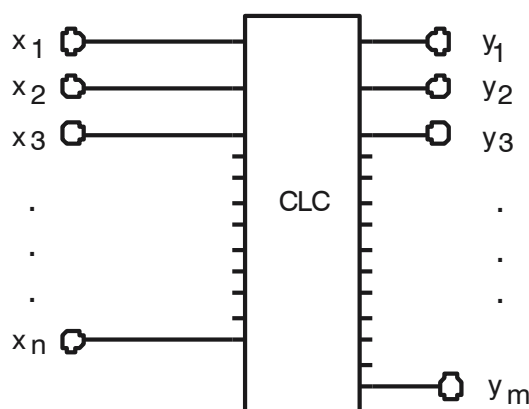


Fig. 1

În general, în cazul proiectării unui sistem numeric care conține circuite combinaționale, se parcurg etapele de sinteză și analiză a acestor componente.

Problema de sinteză a circuitelor combinaționale se definește în modul următor: cunoscând modul de funcționare a circuitului combinațional exprimat prin valorile semnalelor de ieșire corespunzătoare diferitelor combinații ale variabilelor de intrare, se cere să se stabilească structura circuitului.

Ținând cont de această formulare, rezultă că rezolvarea problemei de sinteză presupune parcurgerea a două etape:

1. *sinteza abstractă*, care constă în stabilirea expresiilor funcțiilor booleene care corespund condițiilor impuse între semnalele de ieșire și de intrare. Pentru ca circuitul realizat să fie cât mai simplu și deci mai economic, se caută expresiile minime ale funcțiilor. Din punct de vedere matematic, problemele de sinteză presupun minimizarea funcțiilor booleene în sistemul de funcții elementare alese din considerente practice;
2. *sinteza structurală*, care constă în determinarea structurii fizice a circuitului sintetizat. Sinteza structurală se face în funcție de tipul circuitelor logice elementare (module) și de numărul de intrări ale acestora, de semnalele disponibile în diferite puncte ale sistemului etc.

Algoritmul sintezei circuitelor combinaționale va fi:

1. Din condițiile problemei se stabilesc corespondențele între combinațiile semnalelor de intrare și ieșire folosind tabelul de adevăr, diagrama V-K etc.
2. Se realizează minimizarea funcțiilor booleene care rezultă din etapa precedentă
3. Se implementează dacă este cazul cu funcțiile elementare impuse de realizarea practică

4. Se stabilește logigrama plecând de la forma minimă obținută pentru funcțiile de ieșire în pasul al treilea
5. Se analizează circuitul obținut pentru a vedea dacă corespunde condițiilor impuse inițial (etapă facultativă).

În cazul problemelor de analiză se cunoaște structura circuitului și se cere să se stabilească valorile posibile la ieșiri pentru toate combinațiile posibile ale valorilor semnalelor de la intrări.

În cadrul proiectării utilizând pachete de programe bazate pe utilizarea limbajelor de descriere hardware majoritatea etapelor proiectării sunt automatizate. Ca atare, la descrierea componentelor combinaționale se va căuta ca descrierea să fie cât mai simplă, ușor de înțeles și ușor de modificat pentru situația în care trebuiesc făcute corecții sau modificări ale funcționării acestora.

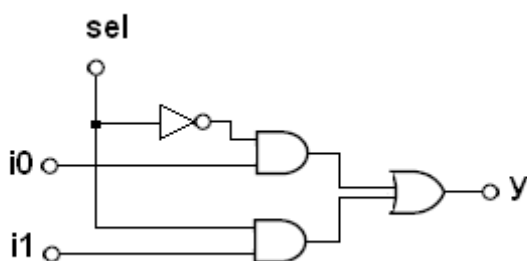
Descrierea circuitelor combinaționale în limbajul Verilog poate fi realizată folosind cele două variante disponibile: descriere structurală și descriere comportamentală.

### Descrierea structurală a circuitelor combinaționale

Această descriere poate fi:

- explicită, când se bazează pe utilizarea primitivelor limbajului Verilog sau a unor componente descrise anterior sau preluate din biblioteci disponibile în cadrul sistemului de proiectare

Exemplu: Circuit multiplexor 2 la 1.



```

module mux121(y, i0, i1, sel);
  input i0,i1;
  input sel;
  output y;
  wire w1, w2, w3;

  not (w1,sel);
  and (w2,w1,i0);
  and (w3,sel,i1);
  or (y,w2,w3);

endmodule
    
```

- implicită, când se utilizează operatorii limbajului Verilog, programul de sinteză fiind cel care inferează structura circuitului din expresiile respective.

Exemplu: Descrierea unui multiplexor 2 la 1 pe 8 biți folosind operatorul „? ... :...”

```

module mux821(i0,i1,sel,y);
  input [7:0] i0,i1;
  input sel;
  output [7:0] y;
    
```

```
assign y = sel ? i1 : i0;  
endmodule
```

**Exemplul 1:** Descrierea unui sumator și al unui circuit de scădere pe 8 biți folosind operatorii de adunare și scădere

```
module add8(i0,i1,sum);  
input [7:0] i0,i1;  
output [7:0] sum;  
  
assign sum = i0 + i1;  
endmodule
```

```
module dif8(i0,i1,dif);  
input [7:0] i0,i1;  
output [7:0] dif;  
  
assign dif = i0 - i1;  
endmodule
```

### Descrierea comportamentală

Pentru descrierea comportamentală a circuitelor combinaționale se utilizează instrucțiunile procedurale utilizând blocuri de tip *always* sensibile la evenimente.

În cazul circuitelor combinaționale, în lista de sensibilitate a blocului *always* trebuie să apară toate intrările circuitului combinațional. Lipsa unei intrări nu afectează sinteza circuitului dar poate să ducă la simularea greșită a circuitului.

**Exemplul 2:** Descrierea unui multiplexor 2 la 1 pe 8 biți folosind instrucțiunea *case*.

```
module mux821(i0,i1,sel,y);  
input [7:0] i0,i1;  
input sel;  
output [7:0] y;  
reg [7:0] y;  
  
always @ (i0 or i1 or sel)  
begin  
case (sel)  
1'b0: y = i0;  
1'b1: y = i1;  
endcase  
end  
endmodule
```

**Exemplul 3:** Descrierea unui multiplexor 3 la 1 pe 8 biți folosind instrucțiunea *case*.

```
module mux831(i0,i1,i2,sel,y);  
input [7:0] i0,i1,i2;
```

```

input [1:0] sel;
output [7:0] y;
reg [7:0] y;

always @ (i0 or i1 or i2 or sel)
begin
  case (sel)
    2'b00: y = i0;
    2'b01: y = i1;
    2'b10: y = i2;
    default: y = 8'bx;
  endcase
end
endmodule

```

În cazul în care este cunoscută tabela de adevăr a circuitului combinațional, este bine ca aceasta să fie implementată direct în cod fără a se mai face minimizarea ecuațiilor booleene ale circuitului.

**Exemplul 4:** Circuitul majoritate.

i2	i1	i0	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

```

module maj3(i0,i1,i2,y);
  input i0,i1,i2;
  output y;
  reg y;
  always @ (i0 or i1 or i2)
  begin
    case ({i2, i1, i0})
      3'b000: y = 1'b0;
      3'b001: y = 1'b0;
      3'b010: y = 1'b0;
      3'b011: y = 1'b1;
      3'b100: y = 1'b0;
      3'b101: y = 1'b1;
      3'b110: y = 1'b1;
      3'b111: y = 1'b1;
    endcase
  end
endmodule

```

În cazul în care se dorește modificarea descrierii, astfel încât aceasta să corespundă unui circuit detector 1 din 3 (o intrare activă din 3), modificarea se face foarte simplu prin schimbarea valorilor funcției corespunzătoare din tabelul de adevăr.

**Exemplul 5.** Circuit detector 1 din 3.

i2	i1	i0	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

```

module detect13(i0,i1,i2,y);
  input i0,i1,i2;
  output y;
  reg y;
  always @ (i0 or i1 or i2)
  begin
    case ({i2, i1, i0})
      3'b000: y = 1'b0;
      3'b001: y = 1'b1;
      3'b010: y = 1'b1;
      3'b011: y = 1'b0;
      3'b100: y = 1'b0;
      3'b101: y = 1'b0;
      3'b110: y = 1'b0;
      3'b111: y = 1'b0;
    endcase
  end
endmodule

```

Dacă s-ar fi dorit descrierea folosind ecuațiile booleene minime, acestea ar fi trebuit obținute folosind metodele de minimizare cunoscute.

Pentru circuitul majoritate descrierea ar fi fost:

```

module maj3(i0,i1,i2,y);
  input i0,i1,i2;
  output y;
  assign y= (i0&i1) | (i0&i2) | (i1&i2);
endmodule

```

În timp ce pentru funcția de detector 1 din 3 descrierea ar fi:

```

module detect13(i0,i1,i2,y);
  input i0,i1,i2;

```

```
output y;  
assign y= (i0&i1&i2) | (!i0&i1&i2) | (!i0&!i1&i2);  
endmodule
```

### Procedeu experimental

A) Să se scrie descrierea structurală și comportamentală a unui circuit ce realizează funcția majoritate de 4 variabile. Pentru descrierea structurală se determină funcția logică minimizată a ieșirii.

Se creează câte un stand de test și se verifică funcționarea fiecăreia din cele două descrieri.

Se creează un stand de test care să verifice echivalența celor două descrieri.

B) Să se scrie descrierea structurală și comportamentală a unui codificator binar cu prioritate 4 la 2. Codificatorul are o a treia ieșire care este activă atunci când cel puțin o intrare este activă. Pentru descrierea structurală se determină funcțiile logice minimizate ale celor 3 ieșiri.

Se creează câte un stand de test și se verifică funcționarea fiecăreia din cele două descrieri.

Se creează un stand de test care să verifice echivalența celor două descrieri.

C) Să se scrie descrierea structurală și comportamentală a unui decodor BCD-7 segmente care să comande afișarea cifrelor de la 0 la 9. Pentru combinațiile neutilizate, segmentele afișorului vor fi stinse.

Se creează câte un stand de test și se verifică funcționarea fiecăreia din cele două descrieri.

Se creează un stand de test care să verifice echivalența celor două descrieri.

D) Să se scrie descrierea structurală și comportamentală a unui sumator complet pe 8 biți. Pentru descrierea structurală se va utiliza descrierea structurală a unui sumator de 1 bit.

Se creează câte un stand de test și se verifică funcționarea fiecăreia din cele două descrieri.

Se creează un stand de test care să verifice echivalența celor două descrieri.

### Conținutul referatului

Referatul va conține codul Verilog pentru descrierea comportamentală și structurală a circuitelor de la procedeul experimental, codul Verilog pentru standurile de test, precum și formele de undă obținute la simularea circuitelor.