

# Lucrarea 6

## Decodificatoare. Multiplexoare. Sumatoare.

### 6.1 Obiective

Lucrarea are următoarele obiective:

- Fixarea cunoștințelor despre circuitele combinaționale, decodificatoare, multiplexoare, sumatoare;
- Exersarea utilizării pachetului de programe XILINX ISE Webpack pentru implementarea circuitelor combinaționale pe FPGA, pornind de la descrierea schematică;
- Testarea circuitelor combinaționale pe macheta FPGA, cu utilizarea osciloscopului și a analizorului logic.

### 6.2 Decodificator

Decodificatorul este un circuit combinațional cu  $n$  intrări și  $2^n$  ieșiri, care identifică un cod de intrare prin activarea unei singure linii de ieșire, corespunzătoare acestui cod.

Figura 6.1 prezintă simbolul unui decodificator de 2 biți iar tabelul 6.1 prezintă tabelul de adevăr care descrie funcția acestuia.

*Tabelul 6.1*

**Tabelul de adevăr al decodificatorului de 2 biți.**

E	A1	A0	D3	D2	D1	D0
1	x	x	0	0	0	0
0	0	0	0	0	0	1
0	0	1	0	0	1	0
0	1	0	0	1	0	0
0	1	1	1	0	0	0

Realizați un proiect schematic în XILINX ISE pentru studiul comportamentului unui decodificator

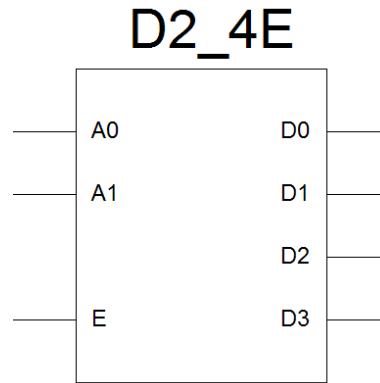


Figura 6.1 Simbolul decodicatorului de 2 biți.

de 2 biți. Utilizați simbolul **D2\_4E** și bus-uri pentru intrările și ieșirile decodicatorului, așa cum este prezentat în figura 6.2.

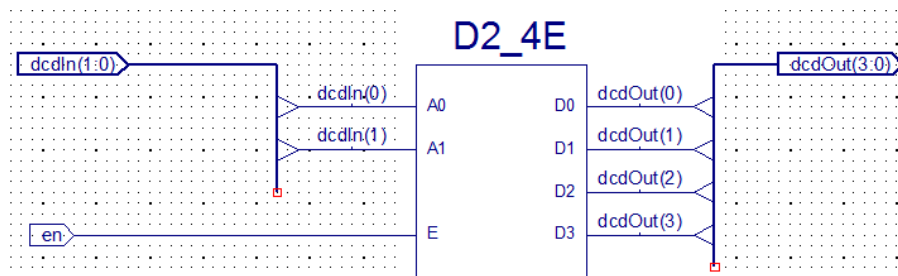


Figura 6.2 Schema de conectare a simbolului decodicatorului de 2 biți.

Ca generator de semnal, folosiți modulul **genSignal8** ale cărui schemă (**genSignal8.sch**) și simbol (**genSignal.sym**) se găsesc pe situl de web al laboratorului. Preluțați acele fișiere și adăugați-le la proiectul ISE.

Desenați o schemă care să includă simbolul decodicatorului studiat (**dcd2.sym**) și cel al generatorului de semnal (**genSignal8.sym**) conectate așa ca în figura 6.3. De remarcat corespondența prin nume a bus-urilor **dcdIn**.

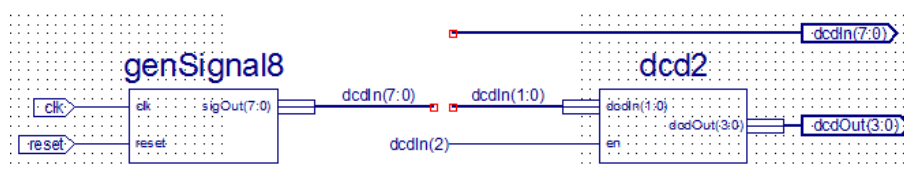


Figura 6.3 Schema de testare a decodicatorului de 2 biți.

Preluțați de pe situl de web al laboratorului fișierele de contrângeri XILINX ISE (**dcd2test.ucf**) și fișierul de proiect al analizorului logic (**dcd2test.lpf**).

Studiați comportamentul decodicatorului vizualizând semnalele de intrare și de ieșire atât cu osciloscopul cât și cu analizorul logic. În cazul osciloscopului, vizualizați semnalul **en** pe canalul 1 și setați-l tot pe acesta ca semnal de trigger-are. Mutați sondele canalelor 1, 2 și 3 pe intrările și ieșirile decodicatorului. Puteți vizualiza astfel corelația lor în timp, luând ca referință semnalul **en**.

Tabelul 6.2 prezintă corespondența dintre porturi, pini FPGA și canalele analizorului logic, pentru testarea decodicatorului de 2 biți.

Tabelul 6.2

**Asocierea porturilor, pinilor FPGA și sondele analizorului logic pentru studiul decodificatorului de 2 biți.**

Port	Pin FPGA	Conector placă	Canal, Conector analizor logic
en	D7	J4-I09	0, negru/alb
dcdin(1)	C7	J4-I010	1, maro/alb
dcdin(0)	F8	J4-I011	2, roșu/alb
dcdOut(3)	B4	J1-IO1	4, galben/alb
dcdOut(2)	A4	J1-IO2	5, verde/alb
dcdOut(1)	D5	J1-IO3	6, albastru/alb
dcdOut(0)	C5	J1-IO4	7, mov/alb

Desenați formele de undă vizualizate pe osciloscop și pe analizorul logic (țineți cont de corelațiile temporale).

Verificați funcționarea butonului de reset (**BTN South (RESET)(K17)**) utilizând analizorul logic. Cum procedați?

Preluați de pe situl de web al laboratorului fișierele Verilog care descriu comportamentul acelorași module (**dcd2.v**, **genSignal8.v**, **dcd2test.v**) și modelul unui mediu de simulare pentru decodificator (**dcd2tb.v**). Realizați simularea circuitului decodificator.

Refaceți implementarea cu FPGA pornind de la fișierele sursă Verilog.

Comparați rezultatele simulării cu cele obținute în urma investigației cu analizorul logic.

## 6.3 Multiplexor

Circuitul multiplexor selectează una din cele  $2^n$  linii de intrare de date pe baza unui cod de selecție de  $n$  biți.

Circuitul multiplexor poate prezenta o intrare de validare a ieșirii.

Figura 6.4 prezintă simbolul bloc al multiplexorului 4:1 din biblioteca grafică a XILINX ISE, iar tabelul 6.3 prezintă tabelul de adevăr al acestui circuit.

Tabelul 6.3

**Tabelul de adevăr al multiplexorului cu validare.**

E	S1	S0	O
1	-	-	0
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3

Realizați un proiect schematic în XILINX ISE pentru studiul comportamentului unui multiplexor 4:1. Utilizați simbolul **M4\_1E** și bus-uri pentru intrările și ieșirile multiplexorului, așa cum este prezentat în figura 6.5.

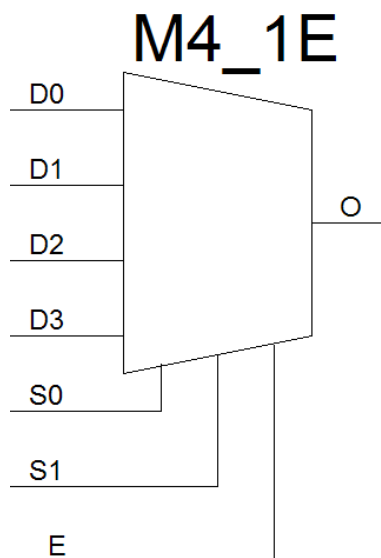


Figura 6.4 Schema bloc a multiplexorului MUX4:1 cu intrare de validare (XILINX ISE).

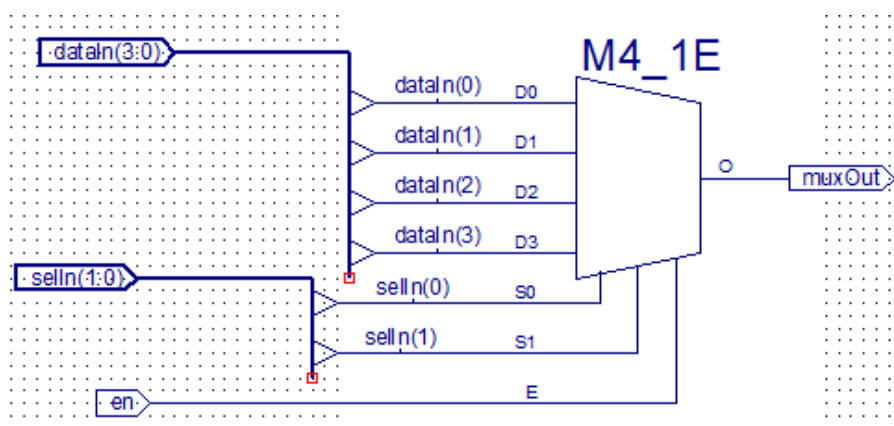


Figura 6.5 Schema de conectare a simbolului multiplexorului 4:1.

Ca generator de semnal, folosiți modulul **genSignal8** ale cărui schemă (**genSignal8.sch**) și simbol (**genSignal.sym**) se găsesc pe situl de web al laboratorului. Preluțați acele fișiere și adăugați-le la proiectul ISE.

Desenați o schemă care să includă simbolul multiplexorului studiat (**mux4x1.sym**) și cel al generatorului de semnal (**genSignal8.sym**) conectate așa ca în figura 6.6.

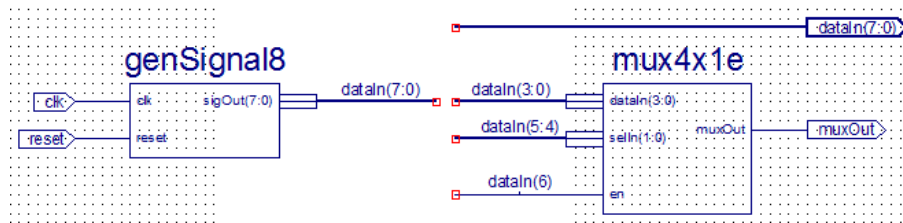


Figura 6.6 Schema de testare a multiplexorului 4:1.

Preluțați de pe situl de web al laboratorului fișierele de contrângerii XILINX ISE (**mux4x1e-Test.ucf**) și fișierul de proiect al analizorului logic (**mux4x1eTest.lpf**).

Studiați comportamentul multiplexorului vizualizând semnalele de intrare și de ieșire atât cu osciloscopul cât și cu analizorul logic. În cazul osciloscopului, vizualizați semnalul **en** pe canalul 1 și setați-l tot pe acesta ca semnal de trigger-are. Mutați sondele canalelor 1, 2 și 3 pe intrările și ieșirile multiplexorului. Puteți vizualiza astfel corelația lor în timp, luând ca referință semnalul **en**.

Tabelul 6.4 prezintă corespondența dintre porturi, pini FPGA și canalele analizorului logic, pentru testarea multiplexorului 4:1.

Tabelul 6.4

**Asocierea porturilor, pinilor FPGA și sondele analizorului logic pentru studiul multiplexorului 4:1.**

Port	Pin FPGA	Conector placă	Canal, Conector analizor logic
en	D7	J4-I09	0, negru/alb
selIn(1)	C7	J4-I010	1, maro/alb
selIn(0)	F8	J4-I011	2, roșu/alb
dataIn(3)	E8	J4-I012	3, portocaliu/alb
dataIn(2)	B4	J1-IO1	4, galben/alb
dcdOut(1)	A4	J1-IO2	5, verde/alb
dcdOut(0)	D5	J1-IO3	6, albastru/alb
muxOut	C5	J1-IO4	7, mov/alb

Desenați formele de undă vizualizate pe osciloscop și pe analizorul logic (țineți cont de corelațiile temporale).

## 6.4 Sumatorul/scăzătorul pe 2 biți

Schema bloc a unui circuit sumator/scăzător este prezentată în figura 6.7. Corepondența porturilor pe placă este prezentată în tabelul 6.5.

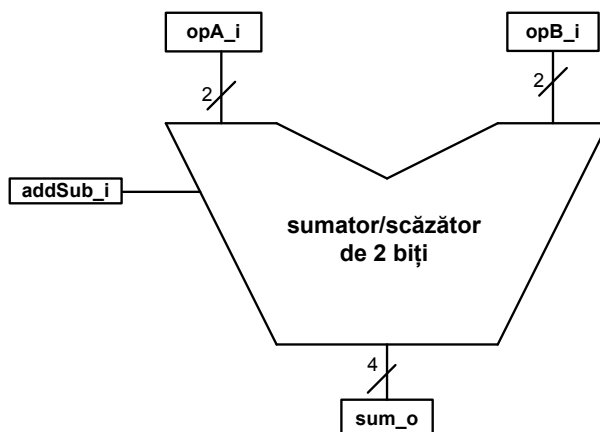


Figura 6.7 Schema bloc a circuitului sumator/scăzător pe 2 biți.

Tabelul 6.5

Corespondența porturilor sumatorului/scăzătorului pe 2 biți.

Port	Direcție	Număr biți	Locație pe placă	Pin FPGA	Semnificație
opA_i	in	2	SW1 SW0	L14, L13	Primul operand
opB_i	in	2	SW3 SW2	N17, H18	Al doilea operand
sum_o	out	4	LED3-0	F11, E11, E12, F12	Rezultat
addSub_i	in	1	BTN South	K17	Comanda: 0 = adunare, 1 = scădere

Modulul sumator-scăzător poate fi descris în Verilog prin următorul cod (prezent pe situl laboratorului în fișierul **addSub.v**):

```

module addSub(
    opA_i,
    opB_i,
    sum_o,
    addSub_i
);

input [1:0]  opA_i;
input [1:0]  opB_i;
output [3:0] sum_o;
input       addSub_i;

reg [3:0]    sum_o;

always @(opA_i or opB_i or addSub_i)
if (addSub_i) sum_o <= opA_i - opB_i; else
    sum_o <= opA_i + opB_i;

endmodule

```

Asocierile pinilor se vor edita în fișierul de constrângeri ISE (prezent pe situl laboratorului în fișierul **addSub.ucf**).

Preluțați fișierele sursă (**addSub.v** și **addSub.ucf**) și realizați un proiect ISE specificând ca metodologie de descriere "HDL". Ca o alternativă, programați macheta FPGA cu datele din fișierul **addSub.bit** existent pe situl laboratorului. Programarea se poate face prin executarea utilitarului **impact**. Apelarea acestui executabil cu parametrii necesari este încapsulată în fișierul **prog.bat**.

Programarea se poate face lansând comanda:

```
prog addSub
```

Încărcați pe placă fișierul bitstream **addSub.bit** (fie realizând proiectul pornind de la fișierele sursă, fie preluând fișierul **.bit**).

Utilizați circuitul proiectat pentru a completa tabelul 6.6. Verificați funcționalitatea prin conversia numerelor din binar în baza 10.

Tabelul 6.6

Verificarea sumatorului/scăzătorului.

Operația zecimal	a binar	b binar	s binar	s zecimal
$1 + 1 = 2$				
$1 + 2 = 3$				
$2 + 1 = 3$				
$3 + 3 = 6$				
$1 - 1 = 0$				
$1 - 2 = -1$				
$2 - 1 = 1$				
$1 - 3 = -2$				

Ce observați în legătură cu operațiile care produc rezultat negativ?