

APLICAȚIA 5

IMPLEMENTAREA DECODIFICATORULUI PENTRU AFIȘAJUL CU 7 SEGMENTE

1. Rezumat

Acest laborator își propune folosirea tehnicilor de minimizare bazate pe diagrame Karnaugh, sau metoda Quine McCluskey pentru a minimiza ecuațiile corespunzătoare unui decodificator pentru afișajul cu 7 segmente. Pentru sinteză se pot folosi oricare din metodele anterior amintite. Se cere minimizarea funcțiilor logice care realizează decodificarea pentru o cifră hexazecimală pentru afișajul cu 7 segmente. Circuitul descris este un circuit combinațional care are 4 biți de intrare pentru operanzi (reprezentând o cifră hexazecimală) și 7 ieșiri corespunzătoare funcțiilor logice.

Obiectivele lucrării

Obiectivul acestui laborator este acela de aplicare a tehnicilor de minimizare pentru un circuit combinațional clasic. De asemenea se cere descrierea Verilog HDL a modulului și verificare funcționării corecte folosind placa Nexys-2.

Obiective tehnice

1. Minimizarea funcțiilor logice folosind diagrame Karnaugh sau metoda Quine-McCluskey.
2. Realizarea decodificatorului pentru afișajul cu 7 segmente.
3. Sinteza și implementare design pe placa FPGA Nexys-2.

Timp necesar

2-3 ore

DECODIFICATOR AFISAJ 7 SEGMENTE

Pregătirea pentru laborator

- Citiți documentul înainte de a începe realizarea practică.
- Salvați output-urile pentru fiecare cerință sau anunțați cadrul didactic în vederea prezentării rezultatelor.

Echipamente și Materiale

Acces la software-ul Xilinx

Necesar	Cantitate
Software ISE® WebPACK™ 14.4 de pe pagina de WEB Xilinx, www.xilinx.com	1
Plugin Digilent (www.digilent.com)	1
Placă Digilent Nexys 2	1
Cablu PMOD	1
Placă de expansiune - PMODSw	1
Placa de expansiune – PMODSSD – Seven-segment display	1

2. Minimizarea funcțiilor logice pentru decodificatorul pentru afișaje cu 7 segmente

2.1 Decodificatorul pentru afișajul cu 7 segmente

Decodificatoarele cu segmente sunt de două tipuri:

- Cu anod comun
- Cu catod comun

Un decodificator cu anod comun are toate legăturile la anodul comun legate împreună. Celelalte legături (a, b, c, \dots, g) sunt conectate la pinii FPGA-ului. Dacă dorim să aprindem segmentul a , trebuie să avem la pinul respectiv 0 logic. Dacă dorim ca acesta să fie stins trebuie să avem 1 logic. Pentru catod comun modul de comandă este invers: pinul a trebuie comandat cu 1 logic dacă dorim segmentul a să fie aprins, sau 0 logic când dorim segmentul să fie stins.

DECODIFICATOR AFISAJ 7 SEGMENTE

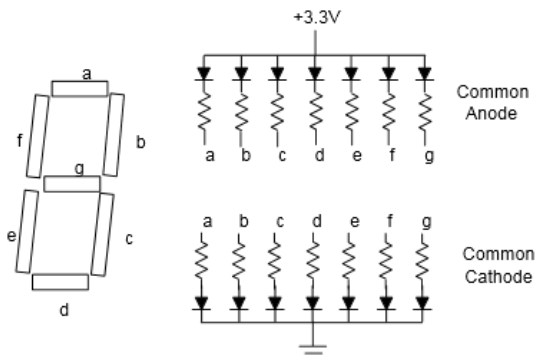


Fig. 5.1 Afișajul cu segmente mod de conectare: anod sau catod comun
(sursa 1.[7])

Afișajele de pe placa Digilent Nexys-2 are afișajele conectate în manieră anod comun. Modulul de expansiune PMODSSD are afișajele cu segmente conectate în maniera catod comun. În ambele cazuri, ieșirile *a*, *b*, ..., *g* sunt multiplexate (adică comune pentru toate afișajele de pe placă sau modulul de expansiune).

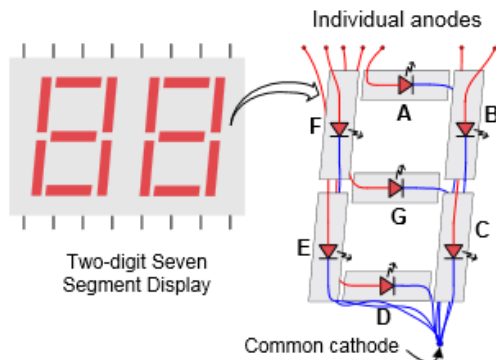


Fig. 5.2 PMODSSD – schema bloc (sursa 1.[9])

Pentru varianta cu catod comun dacă dorim afișarea cifrei 0, atunci trebuie să avem aprinse segmentele: *a,b,c,d,e,f*. Asta presupune ca pinii respectivi să aibe 1 logic. Afișajul *g* se dorește stins, așadar pinul lui primește 0 logic.

DECODIFICATOR AFISAJ 7 SEGMENTE

Cifră	c3	c2	c1	c0	A	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
...											

Intrările pentru circuitul decodificator așa cum se poate vedea mai sus sunt cei patru biți ($c3-c0$) care denotă cifra hexazecimală, iar ieșirile (funcțiile de ieșire) sunt 7 biți aferenți comenzii afișajului cu segmente.

3. Implementarea decodificatorului pentru afișajul cu 7 segmente

Se cere implementarea decodificatorului pentru afișajul cu 7 segmente folosind atât modelarea bazată pe flux de date, precum și modelarea pur comportamentală. Circuitul va avea 4 intrări, ce corespund unei cifre hexazecimale, și va avea 7 ieșiri, ce controlează fiecare led aferent afișajului.

Pas 1 – Tabelul de adevăr și minimizarea funcțiilor logice

Fiecare ieșire a circuitului îi corespunde câte o funcție logică. Se va completa tabelul de adevăr aferent celor 7 funcții logice. Tabelul de adevăr va avea 4 variabile de intrare. Se vor minimiza cele 7 funcții logice și se va scrie forma lor minimizată.

Pas 2 – Crearea unui proiect Xilinx ISE și descrierea unei circuit decodificator pentru afișajul cu 7 segmente

Succint vor fi punctate etapele realizării unui proiect nou:

- Pentru pornire ISE: deschideți un terminal și tastați *ise*
- Creați un proiect nou în directorul workspace: *hex_7display*
- În continuare realizați utilizând limbajul de descriere hardware Verilog componenta din figura de mai jos. La *Hierarchy* în tab-ul de *Design* selectați *Project* → *New source* deschide fereastra *New Source Wizard*. Pentru implementarea folosind descrierea Verilog HDL alegeți la *Select Source Type* – *Verilog Module*.

DECODIFICATOR AFISAJ 7 SEGMENTE

Proiectul va avea o singură sursă:

- *hex_7display* – este modulul care va implementa circuitul dorit; descrierea acestuia va fi fie una de tip flux de date, bazat pe funcțiile logice minimizate, fie una de tip comportamentală; interfața modulului este dată mai jos

```
`timescale 1 ns/1 ps
```

```
module hex_7display  
  (input [3:0] c,  
    output [6:0] hex);
```

```
//DE COMPLETAT LISTA DE SEMNALE INTERNE
```

```
//DE COMPLETAT DESCRIEREA MODULULUI
```

```
endmodule
```

În ceea ce privește descrierea comportamentală, aceasta se va baza pe blocul de tip *always*, respectiv pe instrucțiunea *case*. Modulul aferent este prezentat mai jos.

```
module hex_7display  
  (input [3:0] c,  
    output reg [6:0] hex);  
  
  always  
    @c  
  begin  
    hex = 7'b1111111;  
    case( c)  
      4'b0000: hex = 7'b0111111;
```

```
//DE COMPLETAT DESCRIEREA MODULULUI
```

```
endmodule
```

DECODIFICATOR AFISAJ 7 SEGMENTE

- Adăugați la proiect un fișier de tip testbench. *Project*→*New source* deschide fereastra *New Source Wizard*, alegeți la *Select Source Type* – *Verilog Test Fixture*

Fișierul testbench este următorul.

```
module hex_7display_tb;

    // Inputs
    reg [3:0] c; //hex digit

    //Outputs
    wire [6:0] hex;

    // Instantiate the Unit Under Test (UUT)
    // realizati instanta pentru circuitul testat

    initial begin
        // Initialize Inputs
        c = 0;
    end

    always //toggle inputs for two bit comparator
    begin
        //adaugați combinațiile de numere
    end

endmodule
```

Simulați circuitul folosind simulatorul ISIM.

Pas 3 – Sinteza circuitului

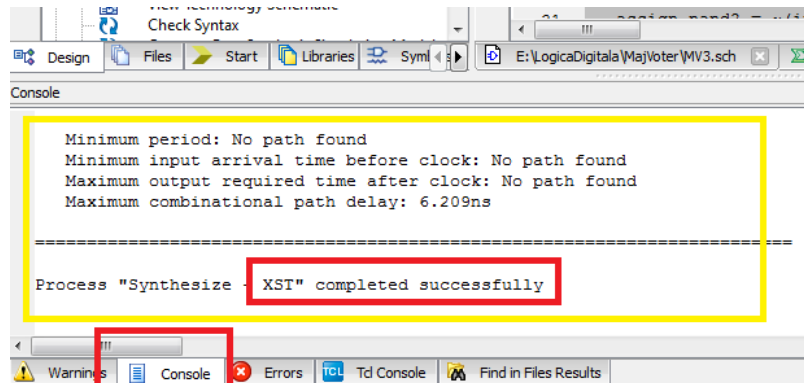
La *Hierarchy* în tab-ul de *View* selectați *Implementation*. Se poate observa că fișierul testbench a dispărut.

În continuare selectați modulul care doriți să-l setați ca și top-level (cel al cărui design va fi programat pe FPGA) – *hex_7display*.

În tabul de *Design* dați click pe *Synthesize*→*Run*. Alternativa este să dați dublu click pe *Synthesize*.

DECODIFICATOR AFISAJ 7 SEGMENTE

La output-ul din tab-ul *Console*, trebuie să vă apară finalizarea cu succes a operației de sinteză.



Pas 3 – Implementarea circuitului

Înainte de a trece la configurarea design-ului pe placă mai avem nevoie de crearea fișierului .UCF. Placa folosită este Nexys-2 cu FPGA-ul Spartan3-E 500 FG320. Toate aceste informații se găsesc specificate în manualul plăcii (Nexys-2 Board Reference Manual).

Circuitul pe care dorim să-l verificăm folosește 4 comutatoare pentru intrări și 1 afișaj cu 7 segmente.

Va fi folosită componenta PmodSWT care este conectată la interfața PMOD2, atunci trebuie consultat manualul aferent acestuia și trebuie identificați pinii pentru conectorul PMOD2 al plăcii Digilent Nexys-2.

Pentru placa Nexys-2, din manual studiați specificația pentru PMOD2 și extrageți informațiile referitoare la pini. Vor fi folosiți pinii indicați mai jos:

Table 3: Nexys2 Pmod Connector Pin Assignments							
Pmod JA		Pmod JB		Pmod JC		Pmod JD	
JA1: L15	JA7: K13	JB1: M13	JB7: P17	JC1: G15	JC7: H15	JD1: J13	JD7: K14 ¹
JA2: K12	JA8: L16	JB2: R18	JB8: R16	JC2: J16	JC8: F14	JD2: M18	JD8: K15 ²
JA3: L17	JA9: M14	JB3: R15	JB9: T18	JC3: G13	JC9: G16	JD3: N18	JD9: J15 ³
JA4: M15	JA10: M16	JB4: T17	JB10: U18	JC4: H16	JC10: J12	JD4: P18	JD10: J14 ⁴

Notes: ¹ shared with LD3 ² shared with LD3 ³ shared with LD3 ⁴ shared with LD3

DECODIFICATOR AFISAJ 7 SEGMENTE

În ceea ce privește afișajul cu 7 segmente, se va folosi modulul de expansiune PMODSSD. Acesta va fi conectat la portul de expansiune JC. Astfel, segmentul *a* va fi conectat la pinul G15, segmentul *b* la pinul J16 etc..

Table 3: Nexys2 Pmod Connector Pin Assignments							
Pmod JA		Pmod JB		Pmod JC		Pmod JD	
JA1: L15	JA7: K13	JB1: M13	JB7: P17	JC1: G15	JC7: H15	JD1: J13	JD7: K14 ¹
JA2: K12	JA8: L16	JB2: R18	JB8: R16	JC2: J16	JC8: F14	JD2: M18	JD8: K15 ²
JA3: L17	JA9: M14	JB3: R15	JB9: T18	JC3: G13	JC9: G16	JD3: N18	JD9: J15 ³
JA4: M15	JA10: M16	JB4: T17	JB10: U18	JC4: H16	JC10: J12	JD4: P18	JD10: J14 ⁴

Notes: ¹ shared with LD3 ² shared with LD3 ³ shared with LD3 ⁴ shared with LD3

Va fi creat fișierul *hex_7display.ucf*.

Se va continua prin implementarea și crearea fișierului de configurare .bit.

Pas 4 – Configurare placă FPGA

Ultimul pas constă în descărcarea design-ului pe placă.

Din Terminal tastați:

```
djtcfg prog -d Nexys2 -i 0 -f hex_7display.bit
```

Verificați funcționarea corectă a design-ului!

Bibliografie:

- [1] Xilinx - Xilinx UG695 ISE In Depth Tutorial - http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/ise_tutorial_ug695.pdf , 2012
- [2] C. Kief, A. Vera, A. Haddad, Q. Cao. COSMIAC FPGA Tutorials <http://cosmiac.org/thrust-areas/education-and-workforce-development/fpga/ate-developed-material/>.
- [3] J. F. Wakerly – Digital Design: Principles and Practices, 3rd Edition, Prentice Hall, 2000
- [4] J. Bhasker - A Verilog HDL Primer, Third Edition - Star Galaxy Publishing, 2005

DECODIFICATOR AFISAJ 7 SEGMENTE

- [5] P. Chu - RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability, Wiley – IEEE Press, 2006
- [6] S. Brown, Z. Vrsanec - Fundamentals of Digital Logic with Verilog Design - McGraw-Hill, 2007
- [7] R. Haskell, D. Hanna - Introduction to Digital Design Using Digilent FPGA Boards – Block Diagram/Verilog Examples – LBE Books, 2009
- [8] Digilent Nexys 2 Reference Manual -
https://www.digilentinc.com/Data/Products/NEXYS2/Nexys2_rm.pdf
- [9] Digilent PMODSSD Reference Manual -
<http://www.digilentinc.com/Products/Detail.cfm?Prod=PMOD-SSD>
- [10] Digilent PMODSWT Reference Manual -
https://www.digilentinc.com/Data/Products/PMOD-SWITCH/Pmod%20SWT_rm.pdf
- [11] O. Boncalo, A. Amăricăi. “Proiectarea circuitelor digitale folosind Verilog HDL – Analiza si Sinteza”. Editura Politehnica, 2011.