

## **APLICAȚIA 8**

### **AUTOMATE CU STĂRI FINITE**

#### **1. Rezumat**

Acest laborator își propune implementarea unui automat cu stări finite folosind limbajul Verilog HDL. Exemplu prezentat în cadrul acestui laborator este cel al unei unități de control pentru un automat de cafea. Circuitul descris este un circuit secvențial ce prezintă 4 semnale de intrare, 3 semnale de ieșire precum și un număr de 8 stări.

#### **Obiectivele lucrării**

Obiectivul acestui laborator este acela de realizare a unui circuit secvențial ce implementează un automat cu stări finite simplu folosind Verilog HDL. De asemenea se cere descrierea Verilog a modulului și verificare funcționării corecte folosind placa Nexys-2

#### **Obiective tehnice**

1. Modelarea automatelor cu stări finite în Verilog HDL.
2. Implementarea unui automat cu stări finite de tip Moore ce modelează controlul unui automat de cafea.
3. Sinteza și implementare design-ului pe placa FPGA Nexys-2.

#### **Timp necesar**

2-3 ore

#### **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

## AUTOMATE CU STĂRI FINITE

Necesar	Cantitate
Software ISE® WebPACK™ 14.4 de pe pagina de WEB Xilinx, <a href="http://www.xilinx.com">www.xilinx.com</a>	1
Plugin Digilent ( <a href="http://www.digilent.com">www.digilent.com</a> )	1
Placă Digilent Nexys 2	1
Cablu PMOD	1
Placă de expansiune – PMODSw	1

## 2. Automate cu stări finite

### 2.1 Definirea problemei

Se dorește realizarea unei unități de control pentru un automat de cafea care să funcționeze în felul următor:

1. Automatul funcționează cu monede de 0.5 lei, respectiv bancnote de 1 leu.
2. Există trei variante de băuturi: espresso, espresso lung și cappuccino
3. Fiecare băutură costă 2 lei.
4. Selecția uneia dintre cele 3 băuturi se poate face după ce s-a introdus suma de 2 lei.
5. Automatul nu dă rest; în cazul introducerii unei sume mai mari de 2 lei, automatul va permite selecția unei singure băuturi.

Pentru modelarea unității de control pentru problema descrisă mai sus, va fi folosit un circuit digital ce va avea 4 semnale de intrare și 3 semnale de ieșire. Semnalele de intrare au următoarea semnificație:

1. Două semnale de intrare (*Credit05* și *Credit10*) pentru cele două modalități de introducere a banilor; valoarea 1 aferentă acestor două semnale indică faptul că a fost introdusă una dintre cele două.
2. Două semnale de intrare aferente celor 3 băuturi (*coffee[1:0]*): valoarea 00 reprezintă niciun fel de selecție a băuturii, valoarea 01 este folosită pentru espresso, 10 pentru espresso lung, iar 11 pentru cappuccino

Fiecărui tip de cafea i se asociază un semnal de ieșire.

## 2.2 Diagrama de stări

Modelarea unui astfel de automat cu stări finite necesită ca prim pas realizarea diagramei de stări aferente. Pentru unitatea de control descrisă anterior se va folosi un automat de tip Moore. Diagrama de stări asociată acestui automat este prezentată mai jos.

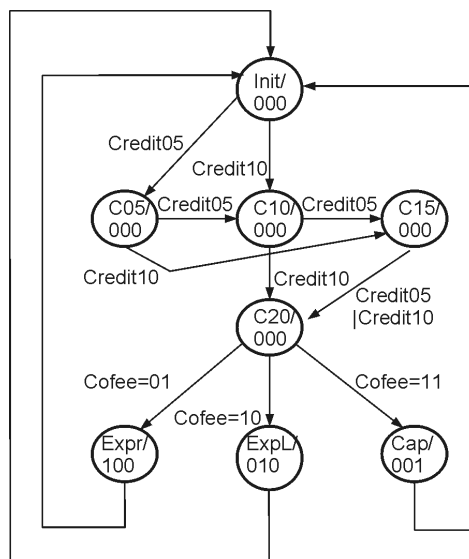


Fig. 8.1 – Diagramă automat cafea

Automatul cu stări finite din Fig. 1 prezintă un număr de 8 stări. Tabelele 1, 2 și 3 prezintă codificarea stărilor, tabelul de tranziții asociat acestui automat, precum și tabelul aferent ieșirilor. Fiind un automat de tip Moore, ieșirile sunt dependente doar de starea automatului, nu și de intrările acestuia.

## AUTOMATE CU STĂRI FINITE

Tabelul 8.1 – Codificarea stărilor

Stare	Codificare		
	Q2	Q1	Q0
Init	0	0	0
C05	0	0	1
C10	0	1	0
C15	0	1	1
C20	1	0	0
Expr	1	0	1
ExpL	1	1	0
Cap	1	1	1

Tabelul 8.2 – Tabelul de tranziții

Stare curentă			Intrări				Stare viitoare		
Q2	Q1	Q0	credit05	credit10	coffee[1]	coffee[0]	D2	D1	D0
0	0	0	0	0	x	x	0	0	0
0	0	0	0	1	x	x	0	1	0
0	0	0	1	0	x	x	0	0	1
0	0	1	0	0	x	x	0	0	1
0	0	1	0	1	x	x	0	1	1
0	0	1	1	0	x	x	0	1	0
0	1	0	0	0	x	x	0	1	0
0	1	0	0	1	x	x	1	0	0
0	1	0	1	0	x	x	0	1	1
...	...	...	...	...	...	...	...	..	..

Tabelul 8.3 – Tabelul ieșirilor

Stare curentă			Ieșiri		
Q3	Q2	Q1	expr	expr_l	capp
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	1	0	0	1

Automatul cu stări finite va conține un registru de stare de 3 biți. Pe baza tabelului 8.2 se pot deriva ecuațiile logice pentru fiecare din cele 3 bistabile din cadrul registrului de stare, iar pe baza tabelului 8.3 se pot determina ecuațiile logice pentru cele 3 ieșiri.

### 3. Implementarea automatului cu stări finite

Se cere implementarea unui automatului descris la 2.1. Modelarea acestui automat se va face folosind o metodologie bazată pe 2 segmente (vezi lucrarea anterioară).

#### Pas 1 – Crearea unui proiect Xilinx ISE și descrierea unei circuit barrel shifter

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: *coffee\_fsm*
- 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*.

Proiectul va avea o singură sursă:

- *coffee\_fsm* – reprezintă modulul asociat automatului; interfața și parte din descrierea modului se găsește mai jos

```
`define INIT 3'b000
```

```
`define C05 3'b001
```

```
`define C10 3'b010
```

```
// DE COMPLETAT
```

```
module coffee_fsm(  
    input clk, rst,  
    input credit05, credit10,  
    input [1:0] cofee,  
    output [2:0] current_state  
    output reg expr, expr_l, capp);
```

## AUTOMATE CU STĂRI FINITE

---

```
reg [2:0] state_nxt, state_reg; //starea următoare a registrului de stare,
respectiv starea curentă

//segmentul secvential– memorează starea curentă
always @( posedge clk or posedge rst)
begin
    //dacă resetul este activ (0 logic)
    if (rst)
        state_reg <= 0; //q_reg denotă starea curentă
    else
        state_reg <= state_nxt;

end

//segmentul combinational – pregătește starea următoare, care va fi încărcată
//în registru de stare
always @( state_reg, coffee, credit05, credit10) //lista de senzitivități conține
toate semnalele a căror modificare
begin //produce re-evaluarea codului dintre begin ... end
    state_nxt = state_reg;
    expr = 0;
    expr_l = 0;
    capp = 0;d

    //descrierea automatului
    case (state_reg)
        `INIT : begin
            state_nxt = `INIT;
            expr = 0;
            expr_l = 0;
            capp = 0;
            if(credit05)
                state_nxt = `C05;
            if(credit10)
                state_nxt = `C10;

        end

end

// DE COMPLETAT
```

După cum se poate observa, circuitul va avea două tipuri de ieșiri: ieșirile principale ale circuitului, respectiv starea curentă.

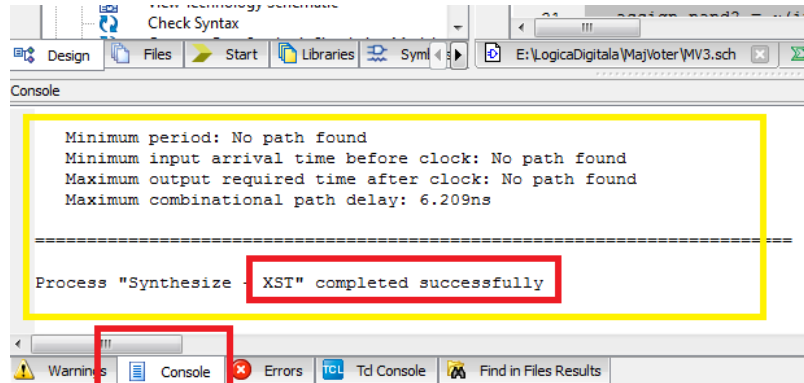
### Pas 3 – Sinteza circuitului

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) – *coffee\_fsm*.

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

Remarcați la output-ul din tab-ul *Console*, 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 6 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 <sup>1</sup>
JA2: K12	JA8: L16	JB2: R18	JB8: R16	JC2: J16	JC8: F14	JD2: M18	JD8: K15 <sup>2</sup>
JA3: L17	JA9: M14	JB3: R15	JB9: T18	JC3: G13	JC9: G16	JD3: N18	JD9: J15 <sup>3</sup>
JA4: M15	JA10: M16	JB4: T17	JB10: U18	JC4: H16	JC10: J12	JD4: P18	JD10: J14 <sup>4</sup>

Notes: <sup>1</sup> shared with LD3 <sup>2</sup> shared with LD3 <sup>3</sup> shared with LD3 <sup>4</sup> shared with LD3

Ieșirile circuitului vor fi legate la 6 LED-uri de pe placa Nexys2.

Va fi creat fișierul *coffee\_fsm.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:

```
djtgcfg prog -d Nexys2 -i 0 -f coffee_fsm.bit
```

Verificați funcționarea corectă a design-ului pe placa FPGA!

## Bibliografie:

- [1] Xilinx - Xilinx UG695 ISE In Depth Tutorial - [http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx14\\_1/ise\\_tutorial\\_ug695.pdf](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
- [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](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](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.