

Laborator 3

1. Scopul laboratorului

Scopul acestui laborator este interfatarea FPGA-ului cu un circuit extern. Aceasta presupune înțelegerea specificațiilor circuitului extern pentru a putea scrie modulul de comunicare de pe FPGA. În acest laborator circuitul extern s-a ales să fie LCD-ul pus la dispoziție de placa Spartan 3E Starter Kit. După scrierea modului de comunicare cu LCD-ul, se va scrie un modul de specificare a unui ceas care, prin intermediul primului modul, va fi afișat pe LCD.

2. Descrierea functionarii LCD-ului

Pe placa Spartan 3E Starter Kit se afla un LCD ce poate afisa 2 linii a cate 16 caractere fiecare. Legarea LCD-ului la FPGA este facuta ca in Figura 2.1 .

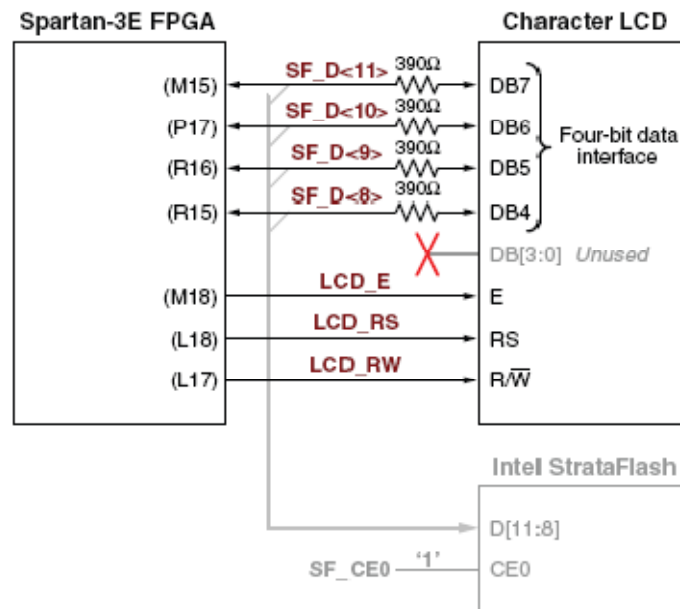


Figura 2.1

Semnificatia pinilor este data in Tabelul 2.1.

Signal Name	FPGA Pin	Function	
SF_D<11>	M15	Data bit DB7	Shared with StrataFlash pins SF_D<11:8>
SF_D<10>	P17	Data bit DB6	
SF_D<9>	R16	Data bit DB5	
SF_D<8>	R15	Data bit DB4	
LCD_E	M18	Read/Write Enable Pulse 0: Disabled 1: Read/Write operation enabled	
LCD_RS	L18	Register Select 0: Instruction register during write operations. Busy Flash during read operations 1: Data for read or write operations	
LCD_RW	L17	Read/Write Control 0: WRITE, LCD accepts data 1: READ, LCD presents data	

Tabelul 2.1

LCD-ul contine un controler grafic Sitronix ST7066U. Acesta utilizeaza 3 memorii interne:

- Display Data RAM – stocheaza codurile caracterelor ce sunt afisate pe ecranul de 2 linii a cate 16 caractere (Figura 2.2 – adresele sunt date in hexazecimal):

Character Display Addresses																Undisplayed Addresses			
1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	...	27
2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	...	67
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...	40

Figura 2.2

- Character Generator ROM – stocheaza o harta a caracterelor (font bitmap) predefinite, ce pot fi afisate de LCD (Figura 2.3)

		Upper Data Nibble															
		DB7	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
		DB6	0	0	0	1	1	1	1	0	0	1	1	1	1	1	
		DB5	0	1	1	0	0	1	1	1	1	0	0	1	1	1	
		DB4	0	0	1	0	1	0	1	0	1	0	1	0	1	1	
		↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
		xxxx0000															
		xxxx0001															
		xxxx0010															
		xxxx0011															
		xxxx0100															
		xxxx0101															
		xxxx0110															
		xxxx0111															
		xxxx1000															
		xxxx1001															
		xxxx1010															
		xxxx1011															
		xxxx1100															
		xxxx1101															
		xxxx1110															
		xxxx1111															
		DB3															
		DB2															
		DB1															
		DB0															

UG230_c6_02_030906

Figura 2.3

- Character Generator RAM – memorie ce permite crearea a inca 8 caractere, pe langa cele aflate in CG ROM.

Pe cei 4 biti SF_D, LCD-ul poate accepta comenzi si date/adese (e.g. adresele caracterelor din font bitmap). Codurile comenzilor acceptate de LCD se gasesc in Tabelul 2.2 (mai multe detalii legate de functia fiecarei comenzi, precum si durata de executie, se gasesc in „ug230.pdf”).

Function	LCD_RS	LCD_RW	Upper Nibble				Lower Nibble			
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear Display	0	0	0	0	0	0	0	0	0	1
Return Cursor Home	0	0	0	0	0	0	0	0	1	-
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S
Display On/Off	0	0	0	0	0	0	1	D	C	B
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	-	-
Function Set	0	0	0	0	1	0	1	0	-	-
Set CG RAM Address	0	0	0	1	A5	A4	A3	A2	A1	A0
Set DD RAM Address	0	0	1	A6	A5	A4	A3	A2	A1	A0
Read Busy Flag and Address	0	1	BF	A6	A5	A4	A3	A2	A1	A0
Write Data to CG RAM or DD RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0
Read Data from CG RAM or DD RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0

Tabelul 2.2

Modul in care este efectuata o operatie de scriere la LCD (comenzi si date/adrese) este ilustrat in Figura 2.4. (Observatie: Deoarece pini SF_D[11:8] sunt comuni cu memoria Intel StrataFlash, pentru a dezactiva aceasta memorie si a folosi LCD-ul, pinul SF_CE0 va trebui sa aiba valoarea 1). Deoarece LCD-ul este interfatat la FPGA doar prin 4 biti de date (Figura 2.1), transferul a 8 biti va fi facut in 2 transe. Initial vor fi transmisi bitii cei mai semnificativi ai octetului, apoi se asteapta cel putin 1us, dupa care se vor transmite si ceilalti 4 biti. Pentru a transmite 4 biti catre LCD, se seteaza semnalul LCD_RS (0, daca vrem sa transmitem o comanda, sau 1, daca dorim sa transmitem date/adrese), se seteaza semnalul LCD_RW (0, pentru o operatie de scriere), se pun cei patru biti pe bitii SF_D[11:8], apoi se asteapta cel putin 40us pentru stabilizarea acestor semnale, dupa care se pune valoarea 1 pe pinul LCD_E. Acest pin va trebui sa mentina valoarea 1 timp de cel putin 230 ns. Intre doua scrieri succesive trebuie asteptat un timp de cel putin 40us (sau 1.64 ms in cazul operatiei de **Clear Display**).

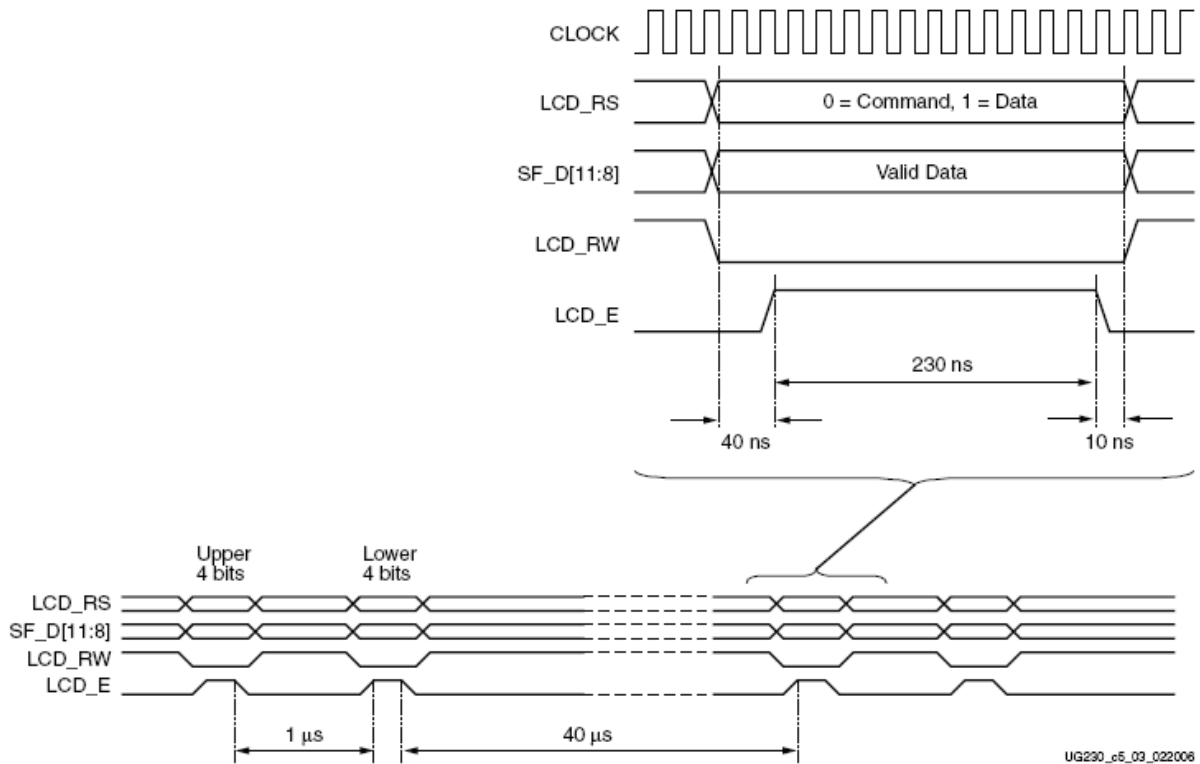


Figura 2.4

Înainte de a putea efectua vreo operație asupra LCD-ului, acesta trebuie inițializat la pornire și apoi configurat. Pentru aceasta, se procedează ca în Figura 2.5 și Figura 2.6

Power-On Initialization

The initialization sequence first establishes that the FPGA application wishes to use the four-bit data interface to the LCD as follows:

- Wait 15 ms or longer, although the display is generally ready when the FPGA finishes configuration. The 15 ms interval is 750,000 clock cycles at 50 MHz.
- Write SF_D<11:8> = 0x3, pulse LCD_E High for 12 clock cycles.
- Wait 4.1 ms or longer, which is 205,000 clock cycles at 50 MHz.
- Write SF_D<11:8> = 0x3, pulse LCD_E High for 12 clock cycles.
- Wait 100 µs or longer, which is 5,000 clock cycles at 50 MHz.
- Write SF_D<11:8> = 0x3, pulse LCD_E High for 12 clock cycles.
- Wait 40 µs or longer, which is 2,000 clock cycles at 50 MHz.
- Write SF_D<11:8> = 0x2, pulse LCD_E High for 12 clock cycles.
- Wait 40 µs or longer, which is 2,000 clock cycles at 50 MHz.

Figura 2.5

Display Configuration

After the power-on initialization is completed, the four-bit interface is now established. The next part of the sequence configures the display:

- Issue a **Function Set** command, 0x28, to configure the display for operation on the Spartan-3E Starter Kit board.
- Issue an **Entry Mode Set** command, 0x06, to set the display to automatically increment the address pointer.
- Issue a **Display On/Off** command, 0x0C, to turn the display on and disables the cursor and blinking.
- Finally, issue a **Clear Display** command. Allow at least 1.64 ms (82,000 clock cycles) after issuing this command.

Figura 2.6

3. Aplicatie de laborator

Sa se realizeze un proiect Xilinx ce descrie functionalitatea unui ceas folosind limbajul Verilog. Ceasul va fi afisat pe LCD-ul de pe placa.

I) Prima parte a aplicatiei are ca scop scrierea unui driver pentru LCD, pornind de la specificatiile din sectiunea a doua a acestui laborator. Veti primi un proiect Xilinx cu majoritatea codului gata facut, pe care va trebui sa-l completati. In urma completarii acestui cod, pe ecranul LCD-ului va trebui sa apara secventa de caractere: 00:00:00. Proiectul trebuie completat cu fisierul ucf iar codul proiectului va trebui completat in sectiunile in care apare un „//TODO”, conform indicatiilor din comentarii.

Detalii de implementare:

Driverul pentru LCD este implementat ca un automat cu stari finite. Astfel:

- stările 0-4 realizeaza Power on init (Figura 2.5)
- stările 5-8 realizeaza Display Configuration (Figura 2.6)
- starea 9 este utilizata pentru aducerea cursorului la inceputul afisajului. Acest lucru se intampla dupa fiecare afisare completa a secventei hh:mm:ss. Deci automatul este adus in aceasta stare la sfarsitul afisarii.
- stările 10-17 sunt folosite pentru a afisa cele 8 caractere pe ecran (Observatie – se presupune ca LCD-ul este comandat cu autoincrementare a adresei cursorului – vezi sursa proiectului).
- starea 18 este folosita pentru a transmite 8 biti de date, respectand toate specificatiile de timp descrise in Figura 2.4

II) A doua parte a aplicatiei are ca scop completarea modulului de descriere a ceasului.

Detalii de implementare:

Ceasul va avea ca intrari 3 butoane + un semnal de ceas si va afisa orele, minutele si secunde pe display-ul de pe placa in formatul **hh:mm:ss**. Ceasul trebuie sa functioneze in timp real si de asemenea, sa poata fi setat cu ajutorul a 3: reset, plus si minus:

reset: reseteaza (sterge) valorile iesirilor, dupa apasarea acestuia afisarea trebuie sa fie 00:00:00;

plus: incrementeaza valoarea minutelor (daca inainte de apasare minutele aveau valoarea 59 efectul apasarii butonului va fi acela de a incrementa ora iar valoarea minutelor va fi 00);

minus: decrementeaza valoarea minutelor (daca inainte de apasare minutele aveau valoarea 00 efectul apasarii butonului va fi acela de a decrementa ora iar valoarea minutelor va fi 59).

Pentru interpretarea corecta a semnalelor primite de la butoane, la fiecare “citire” se va lua in considerare starea anterioara a butoanelor (o apasare oricat de lunga a unui buton va avea efect o singura data). De asemenea tot pentru o interpretare corecta a semnalelor de intrare provenite de la butoane, se va folosi debouncer-ul.

O reprezentare schematica a modulului de ceas se prezinta in Figura 3.1:

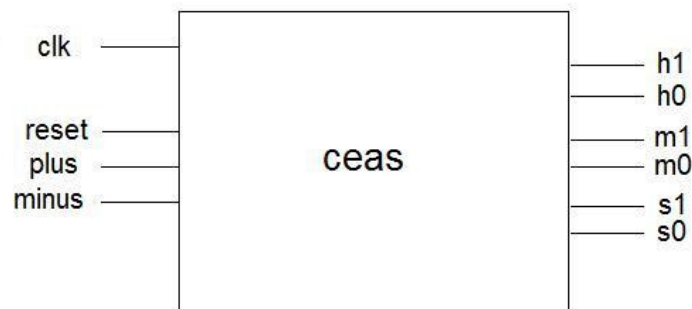


Figura 3.1

Iesirile din modulul ceas reprezinta cate o cifra din valorile dorite la afisare.

Atentie! Limbajul de descriere a circuitelor nu permite impartirea la 10 (deoarece nu este o putere de-a lui 2, caz in care s-ar fi sintetizat printr-o shiftare de biti) asa ca trebuie gasita o metoda de a obtine aceste valori cifra cu cifra.

4. Tema de casa

Sa se realizeze un proiect Xilinx care sa descrie functionalitatea unui cronometru folosind limbajul Verilog.

Asemănător cu cerința precedentă (pentru implementarea ceasului), modulul pentru descrierea cronometrului are ca intrări 3 butoane + semnal de ceas, iar la ieșire va prezenta cifra cu cifra valorile pentru a afișa pe display minutele, secunde și sutimile de secundă în formatul **mm:ss:sss**.

Cronometrul funcționează în timp real și poate fi setat cu ajutorul butoanelor de setare: reset, start și stop:

reset: resetează (sterge) valorile ieșirilor doar dacă cronometrul este oprit (nu se poate reseta în timp ce cronometrează);

start: porneste cronometrul doar dacă acesta este oprit sau resetat;

stop: opreste cronometrarea timpului.

Schema bloc a modulului ce descrie cronometrul este prezentată în Figura 4.1:

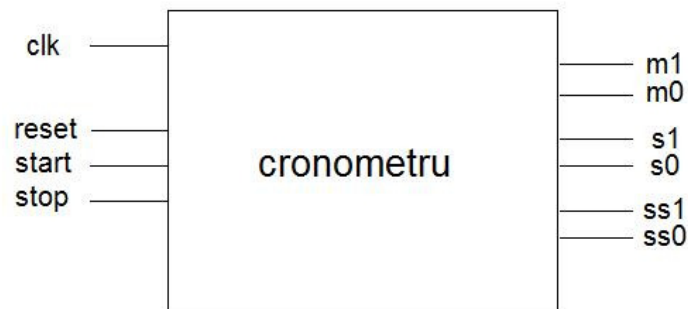


Figura 4.1

Observație: Tema va fi uploadată pe moodle sub formă unei arhive .zip (cu formatul : Tema2_NumePrenume_grupa.zip) ce va conține :

- folderul ce conține proiectul Xilinx
- un fișier README.txt în care se va descrie soluția temei

Deadline: 21 Noiembrie 2009, ora 23:55