



Lab 5: Character LCD Control

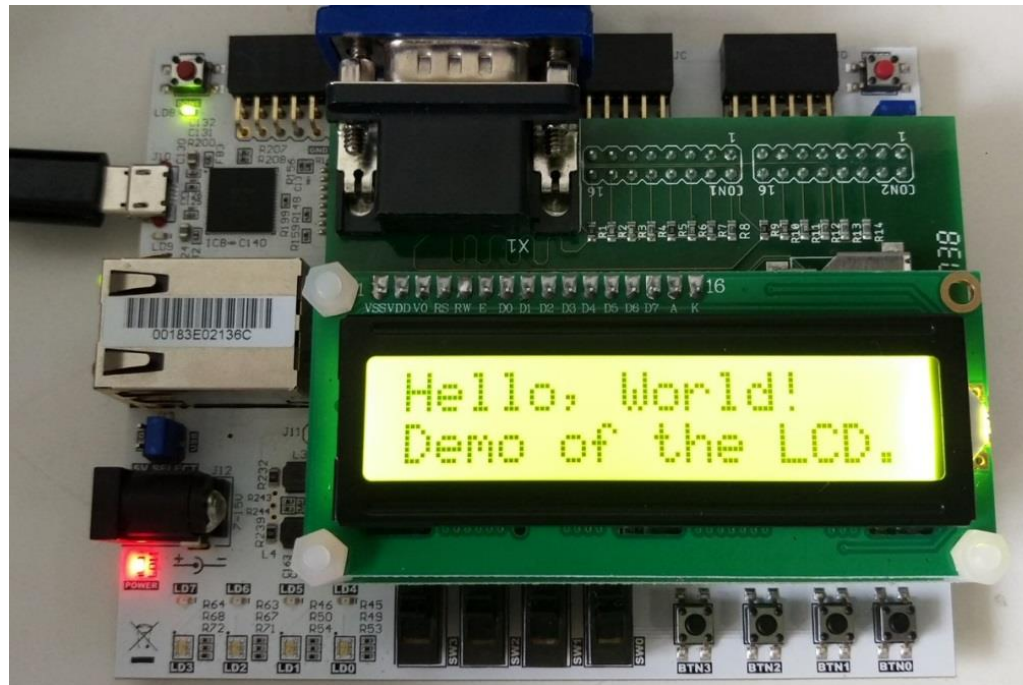
Chun-Jen Tsai and Lan-Da Van
Department of Computer Science
National Yang Ming Chiao Tung University
Taiwan, R.O.C.
Fall, 2022



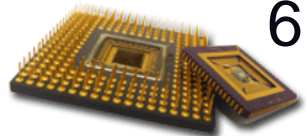
Lab 5: Character LCD Control

Lab 5

- ◆ In this lab, you will compute the first 25 Fibonacci numbers, and use the standard 1602 character LCD to display the numbers.



- ◆ The lab file submission deadline is on 10/31 by 6:00pm.





1602 Character LCD Display

Lab 5

- ◆ The Arty board has only simple I/O devices such as the LEDs, switches, buttons, and UART.
- ◆ We have designed an expansion board, Arty_IO, that adds three more peripherals to Arty:
 - a 1602 character LCD device (supports only 4-bit mode)
 - a SD card socket (supports only the SPIF mode)
 - a 12-bit color VGA interface





Memory Map of the LCD

Lab 5

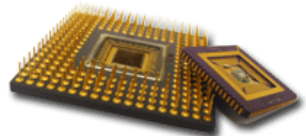
- ◆ The LCD device can be treated as a 32-byte memory.
 - Each memory cell corresponds to a character on the display.
 - Writing an ASCII code to a cell will display the character on the corresponding location on the LCD:



Note: the LCD device is slow, you should not update the screen faster than 2 Hz.

- ◆ Display data memory (DD RAM) addresses:

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





Character LCD Interface (1/2)

Lab 5

- ◆ The LCD interface has 8 data wires (DB0 ~ DB7) and 3 control wires (LCD_E, LCD_RS, LCD_RW):
 - LCD_E enables/disables the inputs to the LCD module.
 - The rest of the wires are defined depending on the functions:

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

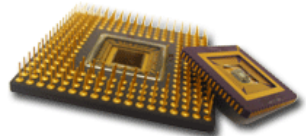
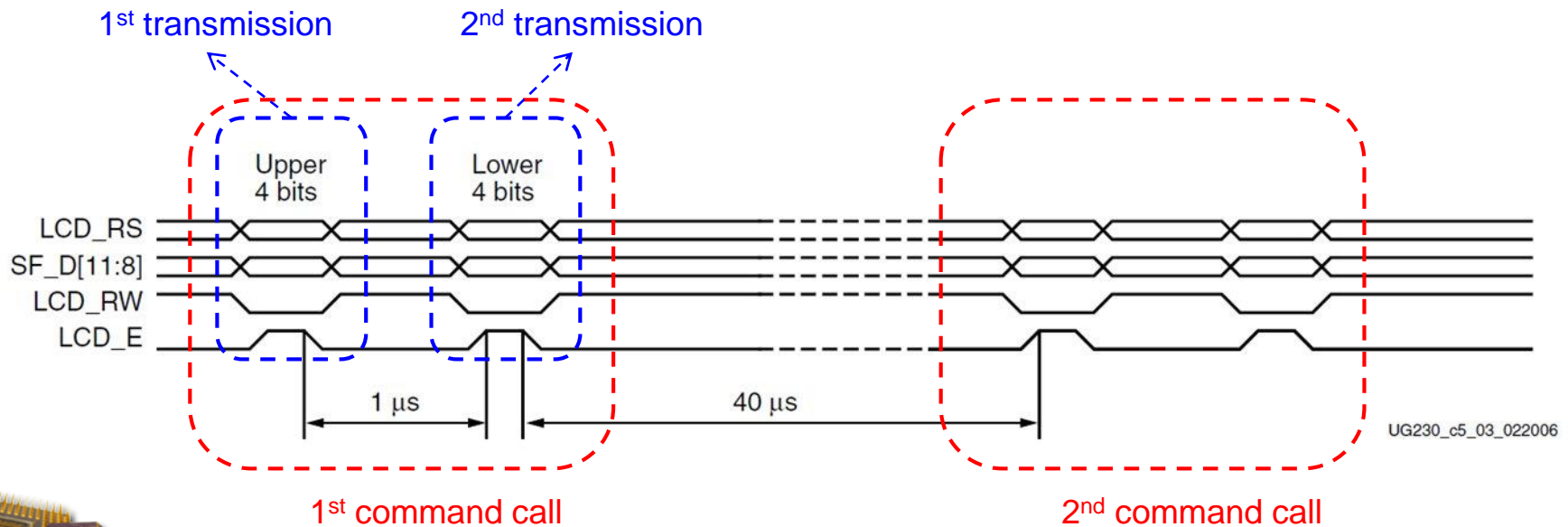




Character LCD Interface (2/2)

Lab 5

- ◆ However, the Arty_IO board uses the 4-bit operating mode of the LCD device, that is, only DB4~DB7 are connected to the FPGA.
 - Execution of a function will need two transmissions, using only LCD_E, LCD_RS, LCD_RW, and DB4~DB7:

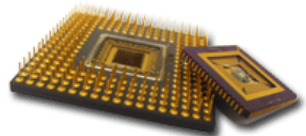
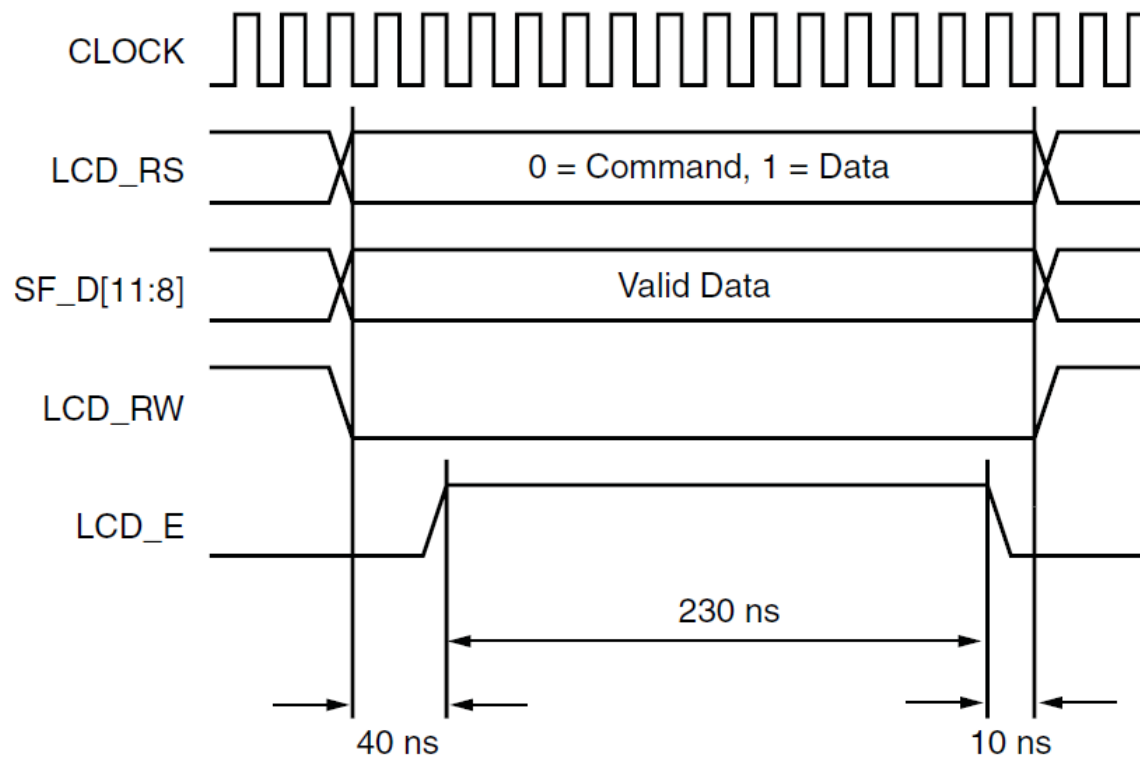




Timing Diagrams for Transmission

Lab 5

- ◆ The timing diagram for one transmission in four-bit mode is as follows:
 - Note that execution of a function requires two transmissions.



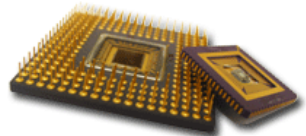


Sample Circuit of Lab 5

Lab 5

- ◆ Two Verilog program files will be provided to you:
 - LCD_Module.v – An LCD controller module
 - Lab5.v – a sample top-level module that prints a “Hello, World!” message using the LCD controller module

```
module LCD_module(  
    input clk,  
    input reset,  
    input [127:0] row_A, // top-row of text  
    input [127:0] row_B, // bottom-row of text  
    output reg LCD_E,  
    output reg LCD_RS,    //register select  
    output reg LCD_RW,    //read / write  
    output reg [3:0] LCD_D //data  
);
```





Fibonacci Number C-Model

Lab 5

- ◆ The first two Fibonacci numbers are 0 and 1. Each remaining number is the addition of the previous two.
- ◆ A short C-model that computes the first 25 Fibonacci numbers is as follows:

```
int fibo[25], idx;  
  
fibo[0] = 0, fibo[1] = 1;  
for (idx = 0; idx < 25; idx++)  
{  
    if (idx >= 2)  
    {  
        fibo[idx] = fibo[idx-1] + fibo[idx-2];  
    }  
    printf("Fibo #%02x is %04x.\n", idx+1, fibo[idx]);  
}
```





What to Do in Lab 5

Lab 5

- ◆ In Lab 5, it is **mandatory** to do the following things:
 - Design a circuit to compute and store the first 25 Fibonacci numbers in a register array
 - Once they are stored, the LCD will start displaying numbers:
 - ◆ Roughly every 0.7 sec, the LCD scrolls up one number cyclically.
 - ◆ If BTN3 is pressed, the scrolling direction will be reversed (scroll-up becomes scroll-down, and vice versa).
 - Example display: cyclic scroll-up (**numbers are hexadecimal**)

