CS150 Project

Checkpoint 1

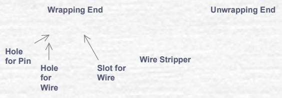
Keypad and LCD Interface

Keypad and LCD Interface

1) Physical Interface
2) Keypad
3) LCD (Optrex DMC-24227)
4) Design Structure and FSM
5) Design implementation
6) Hints and cautions

1) Establishing the physical interface: wire wrapping

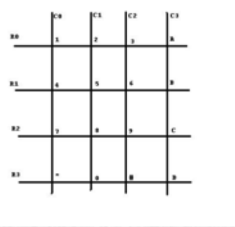
- Materials needed: wire wrap, wire wrap tool, and a little practice.



- Wire wraps available from the IEEE office at 204a Cory. (Next door)
- Pins are given in chkpt1 specification.
- Wire wrap according to diagram on the right.

2) Keypad

- 4x4 keypad, each pin is connected to a specific row or column.
- Remember to connect a pull-down resistor to each output line of the keypad into your design.
- You will need to cycle through each row and check the columns for any button presses.



3) LCD (Optrex DMC-24227)

- Based off Hitachi HD44780 controller chip.
- We will be using 4-bit mode, 5 by 7 dot matrix

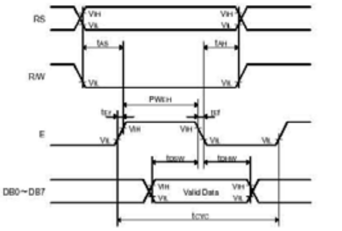
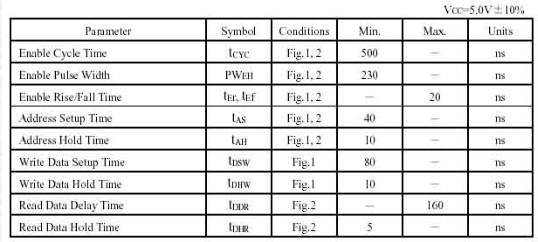


Fig.1 Write Operation Timing



Vcc=5.0V ± 10%

Parameter	Symbol	Conditions	Min.	Max.	Units
Enable Cycle Time	t_{CYC}	Fig.1, 2	500	—	ns
Enable Pulse Width	PW_{EH}	Fig.1, 2	230	—	ns
Enable Rise/Fall Time	t_{Er}, t_{Ef}	Fig.1, 2	—	20	ns
Address Setup Time	t_{AS}	Fig.1, 2	40	—	ns
Address Hold Time	t_{AH}	Fig.1, 2	10	—	ns
Write Data Setup Time	t_{DSW}	Fig.1	80	—	ns
Write Data Hold Time	t_{DHW}	Fig.1	10	—	ns
Read Data Delay Time	t_{DDR}	Fig.2	—	160	ns
Read Data Hold Time	t_{DHR}	Fig.2	5	—	ns

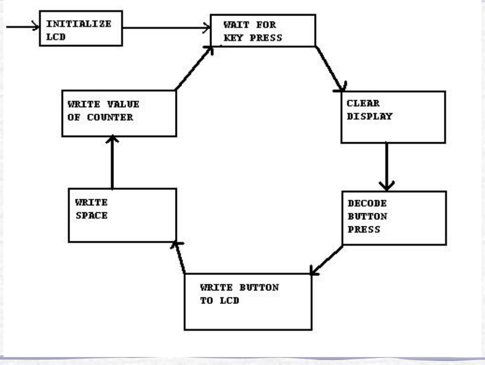
- The enable pulse can be created using a 490 Hz clock.
- In the 4-bit mode, the control bytes are sent on consecutive enable cycles. The most significant 4 bits (DB[7:4]) are sent first, followed by the remaining 4 bits of the byte (DB[3:0]).

- When sending instructions to the LCD, you will need to remember to send the RS, R/W and enable signals along with the databytes.

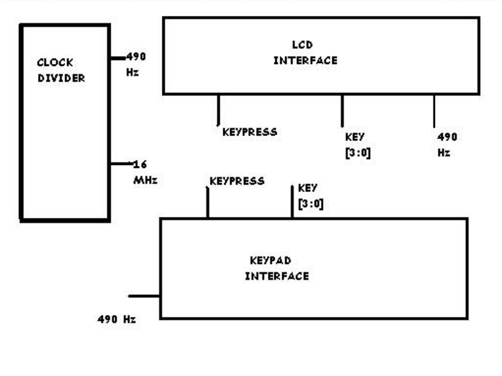
eg.	RS	R/W	DB7	DB6	DB5	DB4
	0	0	0	0	0	0
	0	0	1	0	0	0

this “Display Off” instruction would be implemented on two enable cycles.

[in 8 bit: you would send 00 0000 1000 with enable]

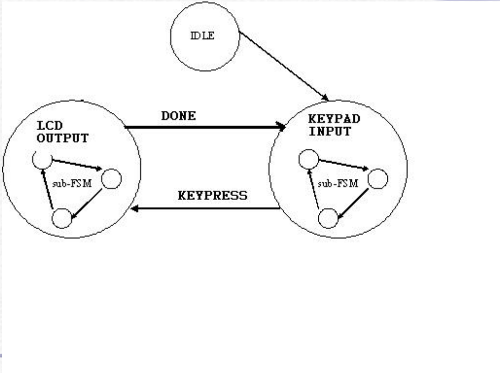


4) Design Structure



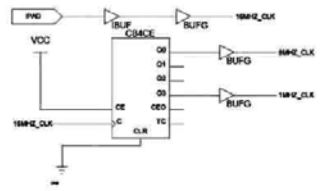
- At start up, (after downloading to the Xilinx chip) the LCD must be initialized before displaying anything.
- Remember to clear line whenever you give it a new input.
- You can ground R/W since we are not reading from the LCD in this checkpoint.

FSM



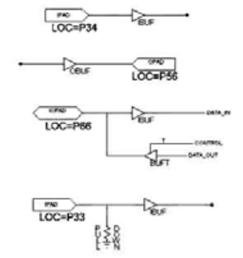
5) Design Implementation

- **Clock Divider:** a counter can divide clocks by multiples of two.
- Make sure you use BUFG before sending the clock out.



IPADS & OPADS:

- Make sure there is always a BUF corresponding to the type of PAD connected.
- Avoid the use of copy and paste in your schematics, ESPECIALLY IPADs and OPADs.



6) Hints and Cautions

- Really sit down and plan your FSM before doing anything in Xilinx. Write it down, draw state diagrams… etc - understand what’s suppose to happen.
- Minimize as much as possible.
- Look at error messages.
- The oscilloscope is your friend!
- Use a potentiometer for the LCD power (if there’s nothing on the screen, give the potentiometer a tweak).