LCD Calculator

ENEE 245: Digital Circuits and Systems Laboratory
Lab 12

Objectives

The objectives of this laboratory are the following:

- Learn how to build an LCD driver and interact with the LCD output device.
- Build a calculator FSM.
- Interface with an external keyboard

This lab involves the design and implementation of a calculator using the LCD display on the FPGA board. A user will input values and operation codes via a keyboard interfaced with the FPGA, and the calculator will perform addition, subtraction, ANDing, and ORing of the input values, depending on the operation code.

Keyboard Controller

On the course website is a tarball containing a keyboard controller. There is one known bug to this keyboard controller—the bug and its solution are further down in this email. There are 5 files in the compressed directory:

1. clkdiv.v	Provides clk dividing to a 25MHz clock
2. keyboard_ctrl.v	Keyboard Driver to gather inputs from PS2
3. keyboard_top.v	Converts Keyboard inputs to ASCII and displays on LCD
4. lcd_driver.v	LCD Driver to display characters to a streaming output
5. Nexys2_500General.ucf	UCF File with pin ports

This code should work out of the box. To test, add all files to a new project and generate a programming file. When you first boot up the code, the LCD screen should be blank. Pressing any NUMPAD key should display the corresponding character on the LCD screen. Only the top row is active, but an easy change can activate any characters on the bottom row (This also allows students to display the answer on the bottom line if they want to).

Note: there is one known bug that occurs with Xilinx!!!

If you receive the error below:

> Pl_Uap_Flow1FitterRuleFastFeedbacks: bad index to sec_nodes array

Do the following to fix it:

> Project -> Design Goals & Strategies
>
> Change “Timing Goal” to “Timing Performance”

Calculator

Use the code given to you to set up a connection between the numeric keypad and the LCD driver. Without making any changes to the attached code, you should be able to see whatever you type on the keypad printed out on the LCD screen.

Implement a simple calculator that accepts two input variables and performs one of four possible operations on them:

+	add
-	subtract
/	AND
	anything else: OR

Print the results to the LCD screen (either on the top row or below …).

Note: this will require some trial and error on the actual FPGA boards, as not all of this will work in simulation.

Pre-Lab Preparation

1. Look through the LCD and keyboard code given to you and familiarize yourself with it, to minimize the time it will take for you to get it working in the lab.
2. Develop Verilog codes for the calculator.

In-Lab Procedure

Ask the TA questions regarding any procedures about which you are uncertain.

1. Show your code and circuit to your TA.
2. Demonstrate the operation of your calculator.

Post-Lab Report

Write up your code, schematics, and lab procedures.

1. Suggest additional features that can make your calculator more user-friendly.
2. What digital logic elements are generally used to represent the “states” in the finite state machine? Suggest an alternative way of implementing state information.

Appendix: ASCII Table

To save you a few minutes of Googling, here is the ASCII table: the 8-bit values that represent alphanumeric characters as well as a number of non-printing characters like end-of-line, and return.

Dec	Hx	Oct	Html	Char	Dec	Hx	Oct	Html	Char	Dec	Hx	Oct	Html	Char
0	0	0		NUL	43	2B	53	&#43;	+	86	56	126	&#86;	V
1	1	1		SOH	44	2C	54	&#44;	,	87	57	127	&#87;	W
2	2	2		STX	45	2D	55	&#45;	-	88	58	130	&#88;	X
3	3	3		ETX	46	2E	56	&#46;	.	89	59	131	&#89;	Y
4	4	4		EOT	47	2F	57	&#47;	/	90	5A	132	&#90;	Z
5	5	5		ENQ	48	30	60	&#48;	0	91	5B	133	&#91;	[
6	6	6		ACK	49	31	61	&#49;	1	92	5C	134	&#92;	\
7	7	7		BEL	50	32	62	&#50;	2	93	5D	135	&#93;	]
8	8	10		BS	51	33	63	&#51;	3	94	5E	136	&#94;	^
9	9	11		TAB	52	34	64	&#52;	4	95	5F	137	&#95;	_
10	A	12		LF	53	35	65	&#53;	5	96	60	140	&#96;	`
11	B	13		VT	54	36	66	&#54;	6	97	61	141	&#97;	a
12	C	14		FF	55	37	67	&#55;	7	98	62	142	&#98;	b
13	D	15		CR	56	38	70	&#56;	8	99	63	143	&#99;	c
14	E	16		SO	57	39	71	&#57;	9	100	64	144	&#100;	d
15	F	17		SI	58	3A	72	&#58;	:	101	65	145	&#101;	e
16	10	20		DLE	59	3B	73	&#59;	;	102	66	146	&#102;	f
17	11	21		DC1	60	3C	74	&#60;	<	103	67	147	&#103;	g
18	12	22		DC2	61	3D	75	&#61;	=	104	68	150	&#104;	h
19	13	23		DC3	62	3E	76	&#62;	>	105	69	151	&#105;	i
20	14	24		DC4	63	3F	77	&#63;	?	106	6A	152	&#106;	j
21	15	25		NAK	64	40	100	&#64;	@	107	6B	153	&#107;	k
22	16	26		SYN	65	41	101	&#65;	A	108	6C	154	&#108;	l
23	17	27		ETB	66	42	102	&#66;	B	109	6D	155	&#109;	m
24	18	30		CAN	67	43	103	&#67;	C	110	6E	156	&#110;	n
25	19	31		EM	68	44	104	&#68;	D	111	6F	157	&#111;	o
26	1A	32		SUB	69	45	105	&#69;	E	112	70	160	&#112;	p
27	1B	33		ESC	70	46	106	&#70;	F	113	71	161	&#113;	q
28	1C	34		FS	71	47	107	&#71;	G	114	72	162	&#114;	r
29	1D	35		GS	72	48	110	&#72;	H	115	73	163	&#115;	s
30	1E	36		RS	73	49	111	&#73;	I	116	74	164	&#116;	t
31	1F	37		US	74	4A	112	&#74;	J	117	75	165	&#117;	u
32	20	40	&#32;	Space	75	4B	113	&#75;	K	118	76	166	&#118;	v
33	21	41	&#33;	!	76	4C	114	&#76;	L	119	77	167	&#119;	w
34	22	42	&#34;	"	77	4D	115	&#77;	M	120	78	170	&#120;	x
35	23	43	&#35;	#	78	4E	116	&#78;	N	121	79	171	&#121;	y
36	24	44	&#36;	$	79	4F	117	&#79;	O	122	7A	172	&#122;	z
37	25	45	&#37;	%	80	50	120	&#80;	P	123	7B	173	&#123;	{
38	26	46	&#38;	&	81	51	121	&#81;	Q	124	7C	174	&#124;	|
39	27	47	&#39;	'	82	52	122	&#82;	R	125	7D	175	&#125;	}
40	28	50	&#40;	(	83	53	123	&#83;	S	126	7E	176	&#126;	~
41	29	51	&#41;	)	84	54	124	&#84;	T	127	7F	177	&#127;	DEL
42	2A	52	&#42;	*	85	55	125	&#85;	U					