

lab8 Report

Lab8_1 (Implement Key Board)

Design Specification

IO:

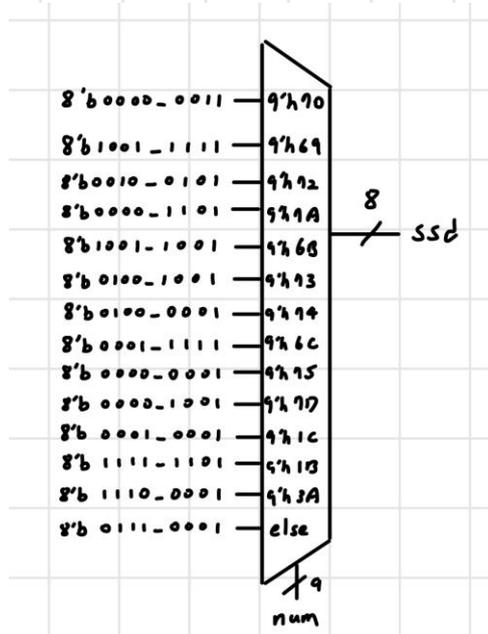
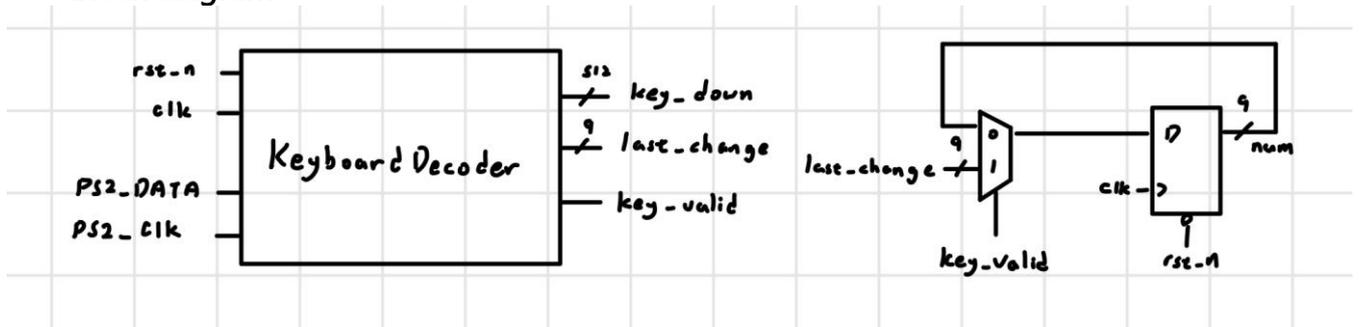
Input: clk, rst_n.

Inout: PS2_DATA, PS2_CLK.

Output: [7:0] ssd, [3:0] yes

Design Implementation

Block diagram



KeyboardDecoder :

此 module 將鍵盤訊號轉換成可以得知哪些按鍵目前被按著的訊號(key_down)、最近一個被操作的按鍵訊號(last_change)以及按下或放開任一按鍵時的通知訊號(key_valid)。

接下來利用 key_valid 決定是否要更新 num 的值，並進一步利用 case 查表輸出對應的 ssd

ssd[7]	ssd[6]	ssd[5]	ssd[4]	ssd[3]	ssd[2]	ssd[1]	ssd[0]
W7	W6	U8	V8	U5	V5	U7	V7

yes[3]	yes [2]	yes [1]	yes [0]	clk	rst_n	PS2_CLK	PS2_DATA
W4	V4	U4	U2	W5	V17	C17	B17

Discussion

這個題目是感覺是要讓我們熟悉 KeyboardDecoder 這個 module 的使用，相對來說較輕鬆，利用簡單的查表就可以達到題目的要求。

Lab8_2

Design Specification

IO:

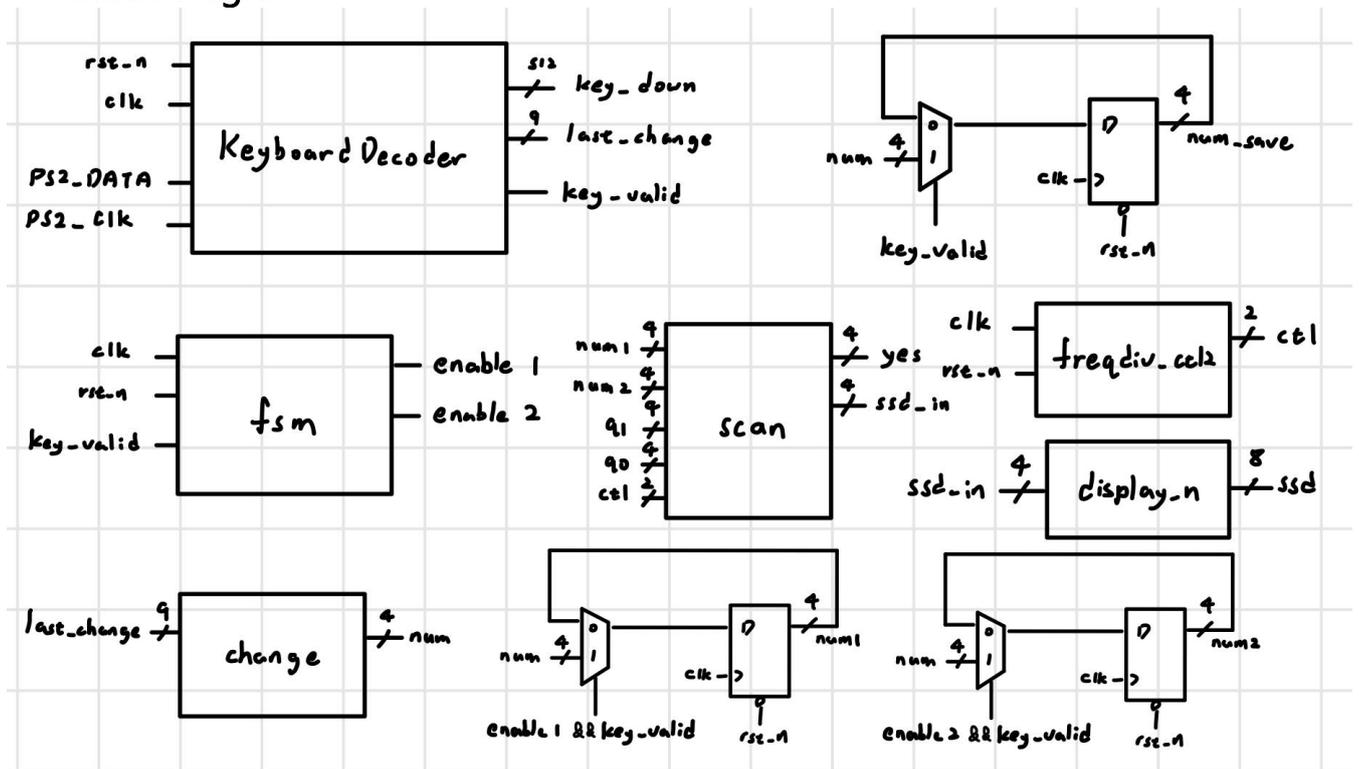
Input: clk, rst_n.

Inout: PS2_DATA, PS2_CLK.

Output: [7:0] ssd, [3:0] yes

Design Implementation

Block diagram



接下來利用 key_vaid 決定是否要更新 num_save 的值，更進一步利用 fsm 輸出的 enable1,enable2 決定是否更新 num1,num2 的值

最後設定一個變數 sum，使得 $sum = num1 + num2$; $q1 = sum / 10$; $q0 = sum \% 10$;

ssd[7]	ssd[6]	ssd[5]	ssd[4]	ssd[3]	ssd[2]	ssd[1]	ssd[0]
W7	W6	U8	V8	U5	V5	U7	V7

yes[3]	yes [2]	yes [1]	yes [0]	clk	rst_n	PS2_CLK	PS2_DATA
W4	V4	U4	U2	W5	V17	C17	B17

Discussion

本實驗要做出個位數的加法，一樣直接利用查表得知按下的數字為何，利用 fsm 去決定要儲存哪一位，接下來就是利用數學式子算出總和，並將算出的二位數字變成分開的兩個數字，再丟上七段顯示器就大功告成。

Lab8_3

Design Specification

IO:

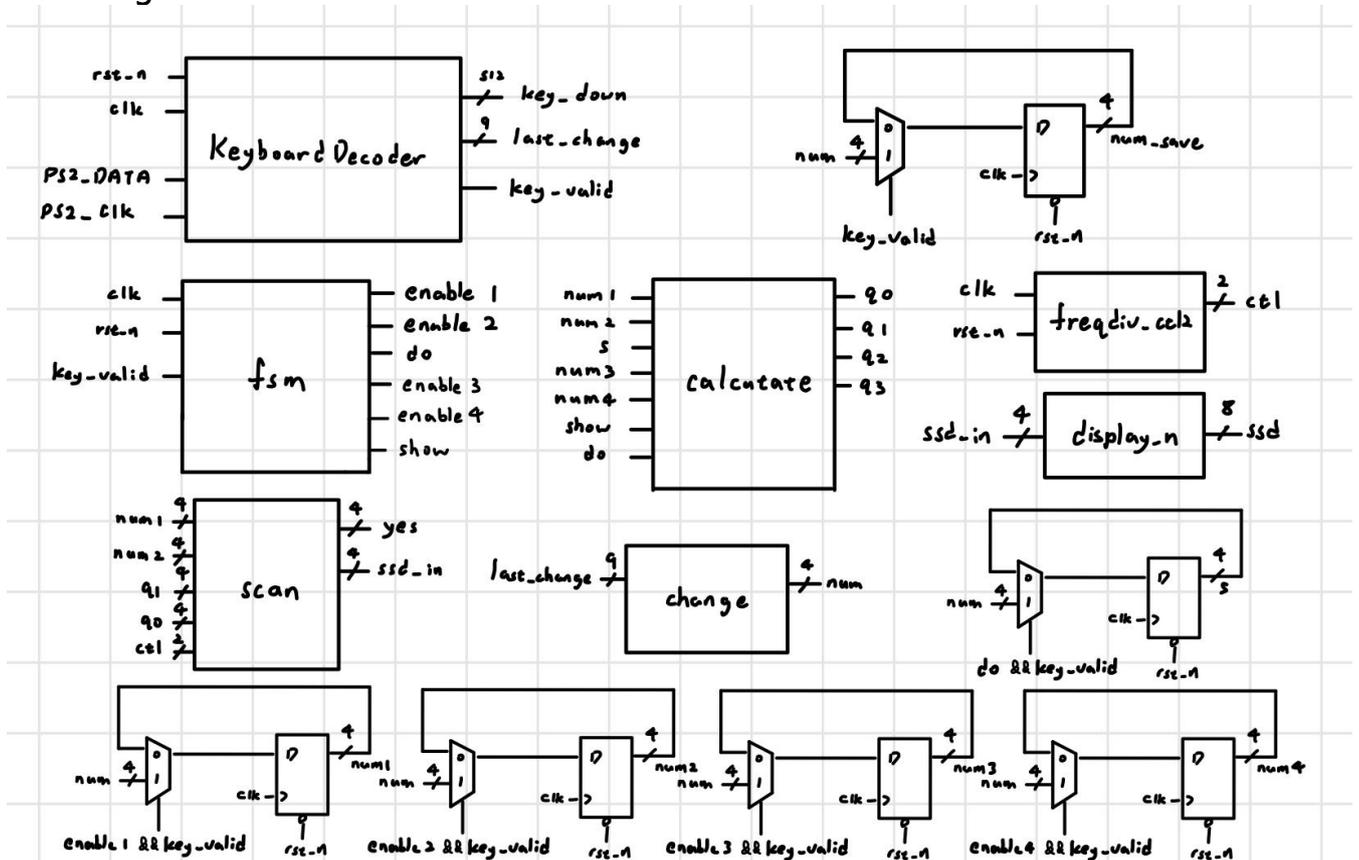
Input: clk, rst_n.

Inout: PS2_DATA, PS2_CLK.

Output: [7:0] ssd, [3:0] yes, [15:0] light(for check)

Design Implementation

Block diagram



Calculate: 是利用儲存的值去做判斷跟計算，當 show 等於 1 時為按下 enter 之後，所以當 show=1 在開始計算就好，題目要求輸入時需看見按下的按鍵，所以 do 等於 1 時顯示按下的符號，且判斷如果是減的時候且結果是負的時候，第一位顯示負號，剩下的就是一些數學上的計算。

ssd[7]	ssd[6]	ssd[5]	ssd[4]	ssd[3]	ssd[2]	ssd[1]	ssd[0]
W7	W6	U8	V8	U5	V5	U7	V7

yes[3]	yes [2]	yes [1]	yes [0]	clk	rst_n	PS2_CLK	PS2_DATA
W4	V4	U4	U2	W5	V17	C17	B17

light [15]	light [14]	light [13]	light [12]	light [11]	light [10]	light [9]	light [8]
L1	P1	N3	P3	U3	W3	V3	V13
light [7]	light [6]	light [5]	light [4]	light [3]	light [2]	light [1]	light [0]
V14	U14	U15	W18	V19	U19	E19	U16

Discussion

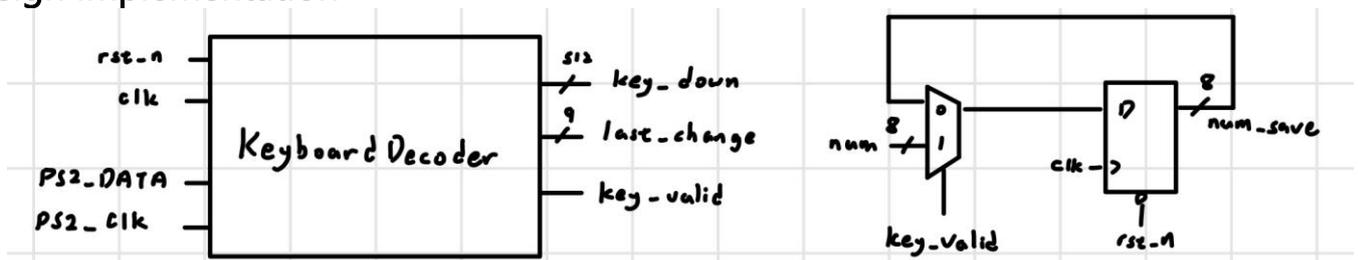
這題比上一題稍微複雜，也算是上一題的延伸，就是要同時做出加減乘，較困難的點應該在於七段顯示器的設計，要同時讓按下的按鍵顯現在七段顯示器上，所以我的作法是輸入完前兩位時接下來的顯示器將顯現運算符號並記錄下來，輸入第三位時數字將覆蓋剛剛的運算符號，在按下 enter 之後將做數學運算跟顯示答案為了方便如有負號，就顯示在第一位。

Lab8_4

Design Specification

- IO:
- Input: clk, rst_n.
- Inout: PS2_DATA, PS2_CLK.
- Output: [7:0] ssd, [3:0] yes, [15:0] light(for check)

Design Implementation



KeyboardDecoder :

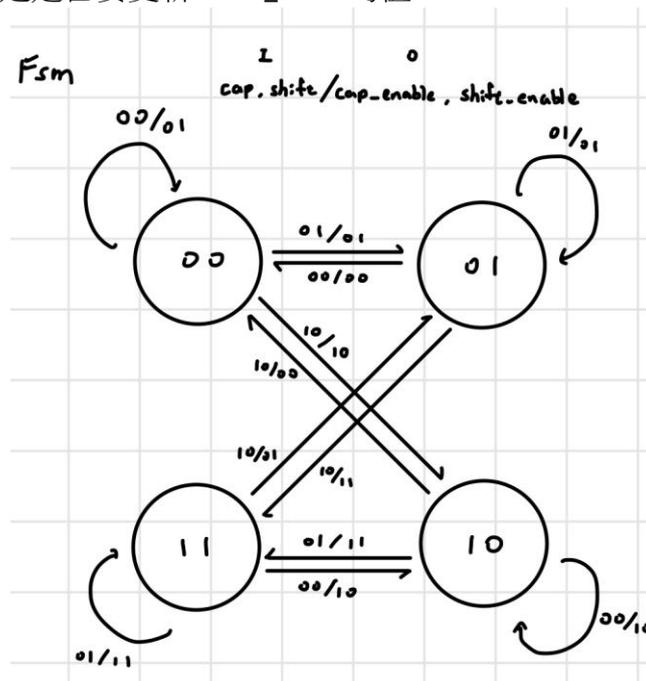
此 module 將鍵盤訊號轉換成可以得知哪些按鍵目前被按著的訊號(key_down)、最近一個被操作的按鍵訊號(last_change)以及按下或放開任一按鍵時的通知訊號(key_valid)。

Dec	Hx	Oct	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051)	{	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

利用 case 查表找出對應的數字

接下來利用 key_vaild 決定是否要更新 num_save 的值



不會同時輸入 11 故無標

最後利用 fsm 的 output 判斷是否需要轉換大小寫

light [15]	light [14]	light [13]	light [12]	light [11]	light [10]	light [9]	light [8]
L1	P1	N3	P3	U3	W3	V3	V13
light [7]	light [6]	light [5]	light [4]	light [3]	light [2]	light [1]	light [0]
V14	U14	U15	W18	V19	U19	E19	U16
clk	rst_n	PS2_CLK	PS2_DATA				
W5	V17	C17	B17				

Discussion

最後一題相對複雜的第三題輕鬆許多，只需要做好是否切換大小寫就結束了，剩餘的利用查表及前面幾題的應用就可完美呈現。

Conclusion

原本看到鍵盤的時候超害怕很難，還好有之前學長留給我們的.v 檔，只需要搞清楚 last_change,key_down 的運作就輕鬆許多，一開始還卡在第一題，因為忘記 reset 要使用 high active 所以都沒跑出想要的結果，還好有發現，剩下的照著一題一題寫就會發現是漸進下去的，所以第二題處理好，最複雜的第三題也就迎刃而解，希望這也能幫到最後的 final project

References

老師的ppt