



PRELAB 5: BÀI TỔNG HỢP THỰC HIỆN MẠCH SỐ TRÊN BREADBOARD – GIAO TIẾP VỚI FPGA

Họ và tên: Lâm Thành Phát	Lớp TN: L21
MSSV: 2111974	Ngày: 26/11/2022

A) PRELAB

Phần I: Tạo Clock 1 Hz cho Kit DE 2

Kit De 2 cung cấp chân CLOCK_50 được nối tới một bộ tạo clock 50 MHz. Nếu nối module tới chân này, ta sẽ nhận được CLOCK 50 MHz (Giao động 50 triệu lần 1 giây). Trong các bài toán thực tế, ta thường thực hiện các thay đổi ở mức 1 giây (Clock 1 Hz).

Để tạo Clock 1 Hz trên Kit DE 2, hoạt động thực tế, ta có thể dùng một bộ nạp thanh ghi giá trị 50M, sau đó trừ đi 1 cho mỗi clock. Sau khi hết 50 M Clock, một bộ so sánh giá trị thanh ghi với 0 sẽ cho ra kết quả 1 và cứ thế lặp lại.

Kết quả của bộ so sánh đó chính là một tín hiệu flag cho kết quả tương tự Clock 1 Hz. Nếu ta đặt các module có enable hoạt động với tín hiệu flag này, chúng sẽ hoạt động (nháy số,...) mỗi 1 giây.

Yêu cầu: Sinh viên viết mã phần cứng SystemVerilog cho module Clock 1 Hz thực hiện nháy nháy đèn LEDR0 trên Kit De2 mỗi 1 giây.

Đoạn mã System Verilog tham khảo cho module này:

```
// Code your design here
///////////////////////////////
// Ho Chi Minh University of Technology
// Faculty of Electrical & Electronics Engineering
// Department of Electronics
// EE1016 - EE1010 : Ky Thuat So
//
// Lab 5 : Thiet ke he thong den giao thong
// Thuc hien giao tiep DE2 - GPIO - Breadboard - ICs/LEDs/LCDs
// Module change 50 MHz to 1 sec
```



```
//  
//////////  
  
module secgen (  
    input clk,  
    input rst,  
  
    output sec  
);  
  
//////////  
  
parameter CLKCONST = 50000000; //clock 50 MHz  
//////////  
  
reg [25:0] temp; // 26-bit 67M  
  
wire tmpzr;  
  
assign tmpzr = temp == 0;  
  
always_ff @(posedge clk) begin  
    if (rst)  
        temp <= CLKCONST; //50M  
    else if (tmpzr)  
        temp <= CLKCONST;  
    else  
        temp <= temp - 1;  
end  
  
assign sec = tmpzr;
```



```
endmodule
```

Đoạn mã System Verilog tạo clock 1 Hz

```
// Code your design here
module clock1hz(input clkin, output clkout);
reg [24:0] counter;
output reg clkout;
input clkin
initial begin
    counter = 0;
    clkout = 0;
end
always @(posedge clkin) begin
    if (counter == 0) begin
        counter <= 24999999;
        clkout <= ~clkout;
    end else begin
        counter <= counter -1;
    end
end
endmodule
```



Phần II: Viết máy trạng thái đèn giao thông điều khiển

Hệ thống đèn giao thông điều khiển được thiết kế theo yêu cầu của Thí nghiệm 1 Lab 5 yêu cầu hoạt động theo một máy trạng thái.

THÍ NGHIỆM 1

Mục tiêu: Thiết kế hệ thống đèn giao thông điều khiển thủ công

Yêu cầu: Thiết kế một hệ thống đèn giao thông cho một ngã tư có các thông số sau:

Đầu vào (input):

- Các nút nhấn bao gồm:
 - 1 nút **RESET**. (Sử dụng nút Key0 trên Kit De 2)
 - 1 Switch thay đổi hướng xe được di chuyển (Sử dụng nút SW1 trên Kit De2)

Ngõ ra (output):

- Sinh viên sử dụng ngõ ra Expansion Header trên Kit De 2 (dùng GPIO_0) để kết nối tới các linh kiện sau:
- Hai bộ đèn (Hai bộ còn lại đổi diện ở mỗi hướng là tùy chọn) bao gồm:
 - 1 Led 7 đoạn đếm giây
 - 1 Led Đỏ
 - 1 Led Xanh lá
 - 1 Led Vàng



Mô tả hoạt động:

- Sau khi RESET, hệ thống đèn giao thông tự động hoạt động theo chế độ
- Ở chế độ điều khiển:
 - Hướng xe được cho phép di chuyển sẽ sáng đèn xanh lá
 - Hướng còn lại sáng đèn đỏ
 - Tất cả đèn đếm hiển thị 9 giây và không thay đổi giá trị
- Khi hướng xe thay đổi (ở chế độ điều khiển)
 - Hướng xe được di chuyển trước đó chuyển từ đèn xanh lá sang đèn vàng, đèn đếm hiển thị 3 giây và đếm ngược xuống 0 rồi chuyển sang đèn đỏ
 - Hướng còn lại giữ nguyên đèn đỏ cho đến khi hướng trước kia chuyển sang đèn đỏ, thời gian đếm ngược từ 3 giây đếm xuống 0.
 - Lúc này, hướng được chuyển sang đèn xanh. Tất cả đèn đếm hiển thị 9 giây và không thay đổi giá trị

Yêu cầu: Sinh viên hãy tham khảo đề bài và vẽ máy trạng thái tương ứng cho bộ điều khiển chính của hệ thống này:

Đặt các ký hiệu:

Hai hướng đối diện là A,B. Mỗi hướng sẽ có 3 trạng thái R(red), Y(yellow), G(green).

VD: AG-BR: Bên A có đèn xanh, bên B có đèn đỏ ...

SW : Tác động SW đổi hướng, \overline{SW} : Không tác động SW.

$Timer$: Bộ đếm ngược đã đếm từ 3-0, \overline{Timer} bộ đếm ngược chưa đếm hết từ 3-0.

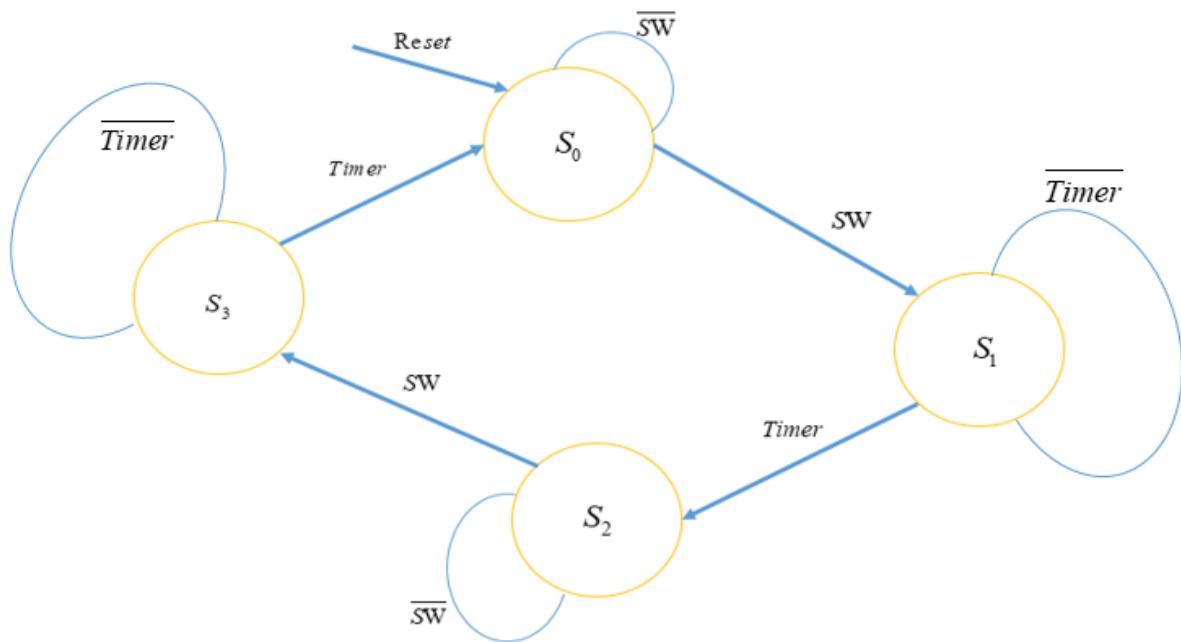
Giả sử trạng thái ban đầu là AG-BR.

Các trạng thái:

- S_0 : AG-BR

- S_1 : AY-BR
- S_2 : AR-BG
- S_3 : AR-BY

Máy trạng thái cho bộ điều khiển của hệ thống:



Phần III: Viết máy trạng thái đèn giao thông tự động

Hệ thống đèn giao thông tự động theo yêu cầu của Thí nghiệm 2 Lab 5 yêu cầu hoạt động theo một máy trạng thái.

Mục tiêu: Thiết kế hệ thống đèn giao thông tự động, hai chế độ

Yêu cầu:

Thiết kế một hệ thống đèn giao thông cho một ngã tư có các thông số sau:

Đầu vào (input):

- Các nút nhấn bao gồm:
 - o 1 nút RESET. (Sử dụng nút Key0 trên Kit De 2)



- 1 Switch chuyển giữa chế độ tự động và điều khiển (Sử dụng SW0 trên Kit De2)
- 1 Switch thay đổi hướng xe được di chuyển (Sử dụng nút SW1 trên Kit De2)

Ngõ ra (output):

- Thay thế kết nối giữa GPIO_0 Kit De 2 đến Led 7 đoạn thành Kết nối từ GPIO_0 Kit De 2 đến IC 74LS47 đến Led 7 đoạn.

Mô tả hoạt động:

- Sau khi RESET, hệ thống đèn giao thông tự động hoạt động theo chế độ cài ở SW0.
- Chế độ điều khiển như thí nghiệm 1.
- Ở chế độ tự động:
 - Mỗi hướng xe thay nhau di chuyển.
 - Hướng được di chuyển có đèn xanh trong thời hạn 5 giây đếm xuống 0, chuyển sang đèn vàng và từ 2 giây đếm xuống 0. Lúc này hướng chuyển sang đèn đỏ, thời gian đếm ngược 9 đếm xuống 0.
 - Hướng còn lại hiển thị đèn đỏ và thời gian đếm ngược 9 đếm xuống 0 (Trùng thời điểm hướng còn lại vừa đèn đỏ được 1 giây) và chuyển sang đèn xanh trong thời hạn 5 giây đếm xuống 0, chuyển sang đèn vàng và từ 2 giây đếm xuống 0. Và cứ thế tiếp tục.

Yêu cầu: Sinh viên hãy tham khảo đề bài và vẽ máy trạng thái tương ứng cho bộ điều khiển chính của hệ thống này:

Gọi trạng thái ban đầu sau khi nhấn RESET là S_{begin} .

Đối với SW_0 , ứng với $SW_0 = 1$ đặt là chế độ tự động, $SW_0 = 0$ là chế độ điều khiển.

Ở chế độ điều khiển, SW_1 : Tác động SW1 đổi hướng, $\overline{SW_1}$: Không tác động SW1.



Giả sử ở cả hai chế độ, trạng thái ban đầu là AG-BR.

$Timer_i$: Bộ đếm ngược đã đếm từ i đến 0 giây, $\overline{Timer_i}$ bộ đếm ngược chưa đếm hết từ i đến 0.

Xét chế độ điều khiển, các trạng thái được trình bày như ở phần II.

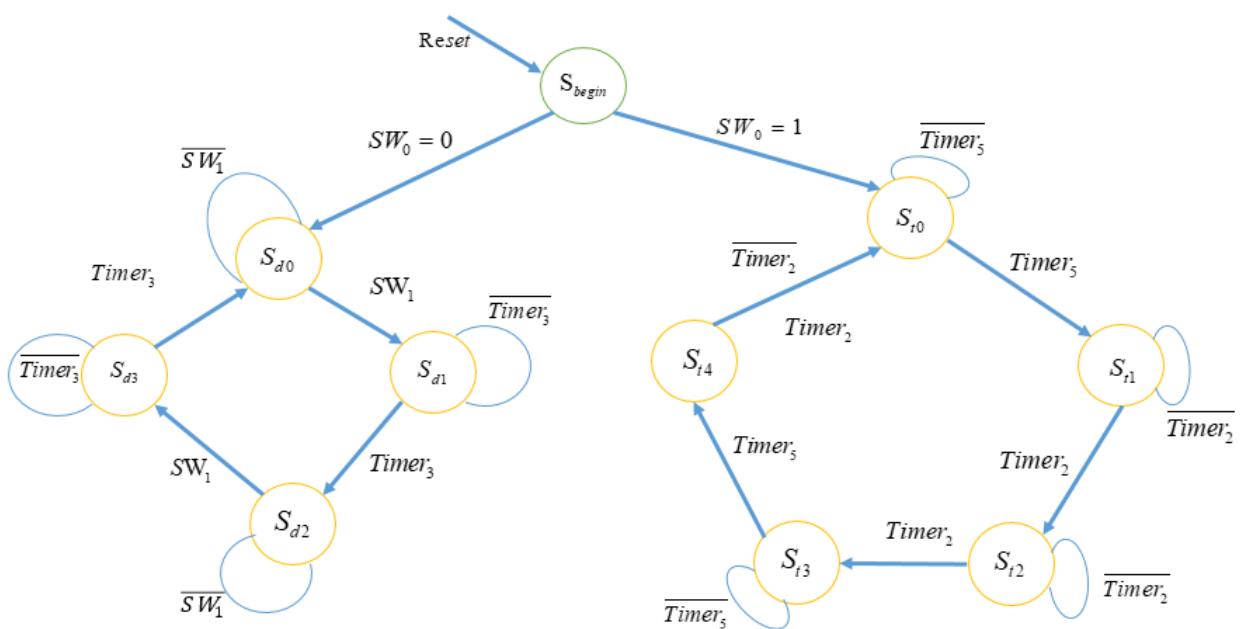
Xét chế độ tự động, ta có sơ đồ hoạt động của từng hướng như sau:

$$\begin{cases} AG : G(Timer_5) \rightarrow Y(Timer_2) \rightarrow R(Timer_9) \\ BR : R(Timer_9) \rightarrow G(Timer_5) \rightarrow Y(Timer_2) \end{cases}$$

Các trạng thái trong chế độ tự động:

- S_{t0} : AG-BR, duy trì trong khoảng thời gian $Timer_5$.
- S_{t1} : AY-BR, duy trì trong khoảng thời gian $Timer_2$.
- S_{t2} : AR-BR, duy trì trong khoảng thời gian $Timer_2$.
- S_{t3} : AR-BG, duy trì trong khoảng thời gian $Timer_5$.
- S_{t4} : AR-BY, duy trì trong khoảng thời gian $Timer_2$.

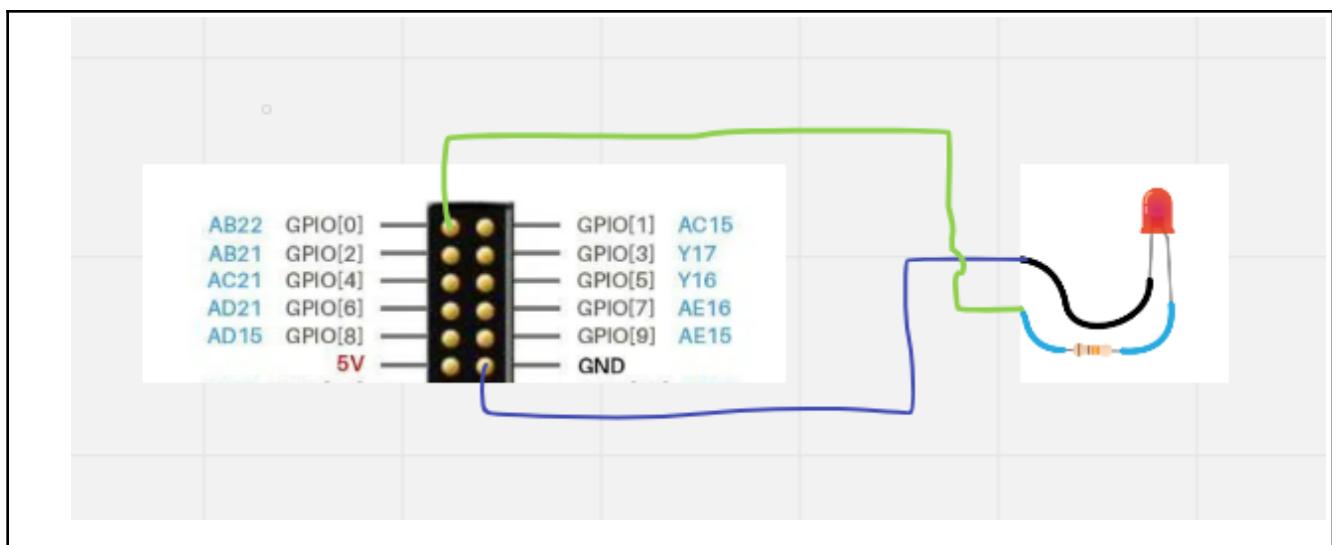
Máy trạng thái cho bộ điều khiển của hệ thống:



Phần IV: Thành phần GPIO của Kit DE2

Sinh viên tham khảo datasheet kit DE2 Terrasic và tìm hiểu thứ tự của GPIO_0 và GPIO_1, cách thức lắp đèn LED tới GPIO_0 của kit DE2.

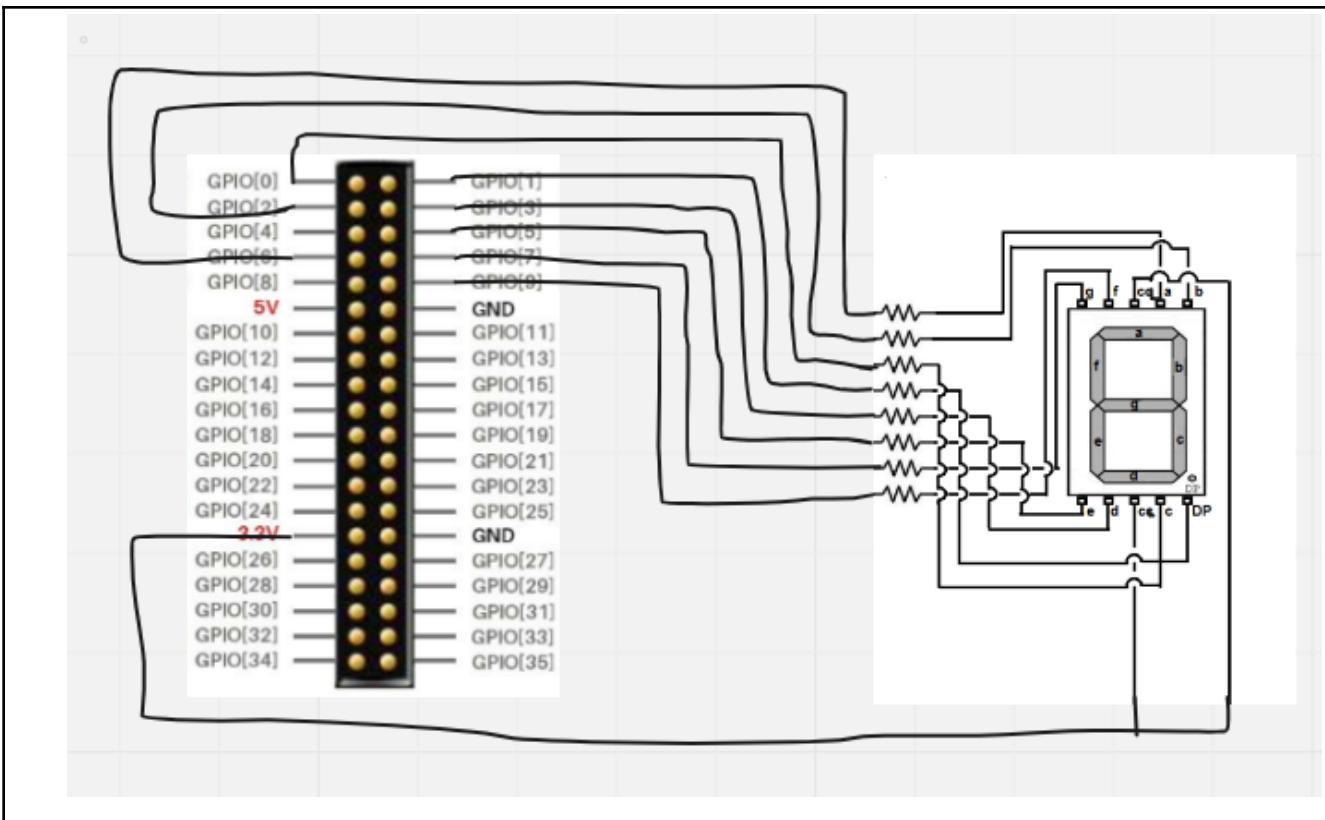
Yêu cầu: Sinh viên vẽ cách thức lắp mạch nối với GPIO_0 của Kit DE2 tới LED (lưu ý tính thẩm mỹ)



Phân V: LED 7 đoạn 5161BS

Sinh viên tham khảo datasheet IC led 7 đoạn 5161BS (Cho LED Common Anode)

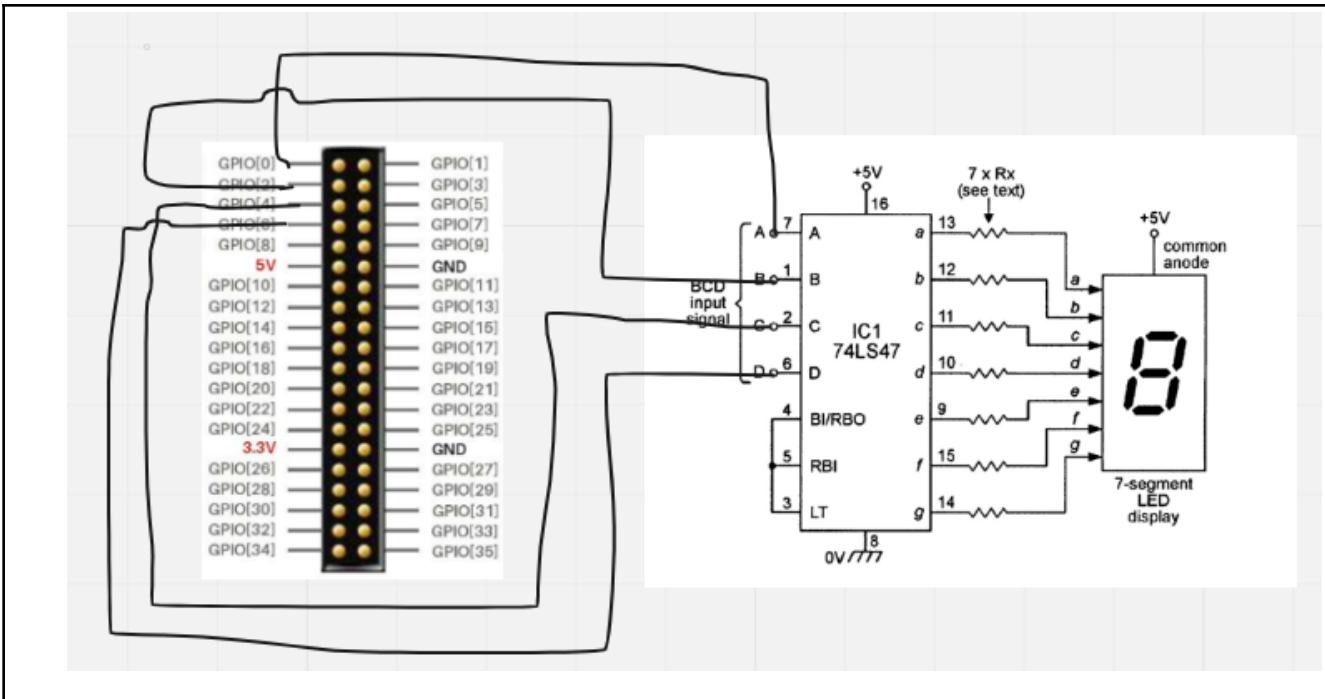
Yêu cầu: Sinh viên vẽ cách thức lắp mạch nối với GPIO_0 của Kit DE2 tới LED 7 đoạn (lưu ý tính thẩm mỹ)



Phân VI: IC giải mã led 7 đoạn 74LS47

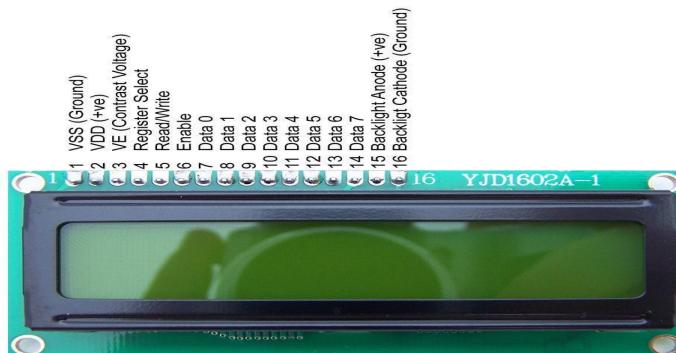
Sinh viên tham khảo datasheet IC giải mã led 7 đoạn 74LS47 (Cho LED Common Anode)

Yêu cầu: Sinh viên vẽ cách thức lắp mạch nối với GPIO_0 của Kit DE2 tới IC 74LS47 tới LED 7 đoạn (lưu ý tính thẩm mỹ)



Phần VII: Giao tiếp LCD 16x2

Sinh viên cần tham khảo datasheet LCD 16x2, đa phần sử dụng chip điều khiển HD44780 với sơ đồ chân cắm như sau:



Hình 5.1 LCD 16 x 2

LCD yêu cầu sinh viên gửi các mã Hex code dưới dạng các command, trong đó bao gồm mã HEX code khởi động và mã HEX Code để thực hiện ghi / xóa nội dung

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

xxxx0000	CG RAM (1)	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000		Q	q	P	P					-	¶	¤	¤	¤	¤	¤	¤
xxxx0001	(2)	!	1	A	Q	a	q			¤	¤	¤	¤	¤	¤	¤	¤
xxxx0010	(3)	"	2	B	R	b	r			¤	¤	¤	¤	¤	¤	¤	¤
xxxx0011	(4)	#	3	C	S	c	s			¤	¤	¤	¤	¤	¤	¤	¤
xxxx0100	(5)	\$	4	D	T	d	t			¤	¤	¤	¤	¤	¤	¤	¤
xxxx0101	(6)	%	5	E	U	e	u			¤	¤	¤	¤	¤	¤	¤	¤
xxxx0110	(7)	&	6	F	V	f	v			¤	¤	¤	¤	¤	¤	¤	¤
xxxx0111	(8)	'	7	G	W	g	w			¤	¤	¤	¤	¤	¤	¤	¤
xxxx1000	(1)	(8	H	X	h	x			¤	¤	¤	¤	¤	¤	¤	¤
xxxx1001	(2))	9	I	Y	i	y			¤	¤	¤	¤	¤	¤	¤	¤
xxxx1010	(3)	*	:	J	Z	j	z			¤	¤	¤	¤	¤	¤	¤	¤
xxxx1011	(4)	+	;	K	L	k	l			¤	¤	¤	¤	¤	¤	¤	¤
xxxx1100	(5)	,	<	L	¥	l	l			¤	¤	¤	¤	¤	¤	¤	¤
xxxx1101	(6)	-	=	M	J	m	j			¤	¤	¤	¤	¤	¤	¤	¤
xxxx1110	(7)	.	>	N	^	n	^			¤	¤	¤	¤	¤	¤	¤	¤
xxxx1111	(8)	/	?	O	_	o	_			¤	¤	¤	¤	¤	¤	¤	¤

Note: The user can specify any pattern for character-generator RAM.

Hình 5.2: Mã ký tự và dạng hiển thị tương ứng

Các lệnh cho mô-đun LCD 16x2

Mô-đun LCD 16×2 có một bộ hướng dẫn lệnh đặt trước. Mỗi lệnh sẽ làm cho mô-đun thực hiện một nhiệm vụ cụ thể. Các lệnh thường được sử dụng và chức năng của chúng được đưa ra trong bảng dưới đây

Lệnh	Chức năng
0F	LCD bật, con trỏ bật, con trỏ nhập nháy bật
01	Xoá toàn màn hình
02	Quay về màn hình chính



Lab 5: Bài tổng hợp thực hiện mạch số trên breadboard – giao tiếp với FPGA.

04	Giảm con trỏ
06	Tăng con trỏ
0E	Màn hình bật, con trỏ nháy tắt
80	Bắt con trỏ trở về vị trí đầu tiên của hàng 1
C0	Bắt con trỏ trở về vị trí đầu tiên của hàng 2
38	Sử dụng 2 hàng và ma trận 5x7
83	Con trỏ hàng 1 vị trí 3
3C	Kích hoạt dòng 2
08	Tắt màn hình hiển thị và con trỏ
C1	Nhảy đến dòng 2 vị trí 1
OC	Bật màn hình hiển thị, tắt con trỏ
C2	Nhảy đến hàng 2, vị trí 2

Khởi tạo LCD

Các bước phải được thực hiện để khởi tạo màn hình LCD được đưa ra dưới đây và các bước này là phổ biến cho hầu hết các ứng dụng.

B1: Gửi 38H đến dòng dữ liệu 8 bit để khởi tạo

B2: Gửi 0FH để bật LCD, con trỏ BẬT và con trỏ nháy ON.

B3: Gửi 06H để tăng vị trí con trỏ.

B4: Gửi 01H để xóa màn hình và trả về con trỏ.

Đưa dữ liệu vào LCD

Các bước để gửi dữ liệu đến mô-đun LCD được đưa ra dưới đây. Mô-đun LCD có các chân RS, R / W và E. Chính trạng thái logic của các chân này làm cho mô-đun xác định xem đầu vào dữ liệu đã cho là lệnh hay dữ liệu được hiển thị.

Đặt R / W mức thấp.

Đặt RS = 0 nếu byte dữ liệu là lệnh và tạo RS = 1 nếu byte dữ liệu là dữ liệu sẽ được hiển thị.

Đặt byte dữ liệu trên thanh ghi dữ liệu.

Xung E từ cao xuống thấp.

Lặp lại các bước trên để gửi dữ liệu khác.

Dữ liệu chữ cái và chữ số, ký hiệu sinh viên có thể tham khảo bảng mã ASCII

Dec	Hx	Char	Dec	Hx	HTML	Char	Dec	Hx	HTML	Char	Dec	Hx	HTML	Char
0	0	NUL (null)	32	20	 	Space	64	40	@	�	96	60	`	`
1	1	SOH (Start of heading)	33	21	!	!	65	41	A	A	97	61	a	a
2	2	STX (Start of text)	34	22	"	"	66	42	B	B	98	62	b	b
3	3	ETX (End of text)	35	23	#	#	67	43	C	C	99	63	c	c
4	4	EOT (End of transmission)	36	24	$	\$	68	44	D	D	100	64	d	d
5	5	ENQ (Enquiry)	37	25	%	%	69	45	E	E	101	65	e	e
6	6	ACK (Acknowledge)	38	26	&	&	70	46	F	F	102	66	f	f
7	7	BEL (Bell)	39	27	'	'	71	47	G	G	103	67	g	g
8	8	BS (Backspace)	40	28	({	72	48	H	H	104	68	h	h
9	9	TAB (Horizontal tab)	41	29))	73	49	I	I	105	69	i	i
10	A	LF (NL line fd, new line)	42	2A	*	*	74	4A	J	J	106	6A	j	j
11	B	VT (Vertical tab)	43	2B	+	+	75	4B	K	K	107	6B	k	k
12	C	FF (NP form fd, new page)	44	2C	,	,	76	4C	L	L	108	6C	l	l
13	D	CR (Carriage return)	45	2D	-	-	77	4D	M	M	109	6D	m	m
14	E	SO (Shift out)	46	2E	.	.	78	4E	N	N	110	6E	n	n
15	F	SI (Shift in)	47	2F	/	/	79	4F	O	O	111	6F	o	o
16	10	DLE (Data link escape)	48	30	0	0	80	50	P	P	112	70	p	p
17	11	DC1 (Device control 1)	49	31	1	1	81	51	Q	Q	113	71	q	q
18	12	DC2 (Device control 2)	50	32	2	2	82	52	R	R	114	72	r	r
19	13	DC3 (Device control 3)	51	33	3	3	83	53	S	S	115	73	s	s
20	14	DC4 (Device control 4)	52	34	4	4	84	54	T	T	116	74	t	t
21	15	NAK (Negative acknowledge)	53	35	5	5	85	55	U	U	117	75	u	u
22	16	SYN (Synchronous idle)	54	36	6	6	86	56	V	V	118	76	v	v
23	17	ETB (End of trans. block)	55	37	7	7	87	57	W	W	119	77	w	w
24	18	CAN (Cancel)	56	38	8	8	88	58	X	X	120	78	x	x
25	19	EM (End of medium)	57	39	9	9	89	59	Y	Y	121	79	y	y
26	1A	SUB (Substitute)	58	3A	:	:	90	5A	Z	Z	122	7A	z	z
27	1B	ESC (Escape)	59	3B	;	,	91	5B	[[123	7B	{	{
28	1C	FS (File separator)	60	3C	<	<	92	5C	\	\	124	7C	|	
29	1D	GS (Group separator)	61	3D	=	=	93	5D]	^	125	7D	}	}
30	1E	RS (Record separator)	62	3E	>	>	94	5E	^	_	126	7E	~	-
31	1F	US (Unit separator)	63	3F	?	?	95	5F	_	DEL	127	7F		

Hình 5.3: Bảng mã ASCII

Yêu cầu: Sinh viên tìm hiểu ý nghĩa chân cảm của LCD. Sinh viên viết mã phần cứng SystemVerilog có chức năng nạp vào LCD dòng chữ “Chuan bi TN5”

Tham khảo module LCD:



Module LCD Sinh viên viết:

```
module lcddisp_ex
    input clk,
    output reg lcd_e, lcd_rs,
    output reg [7:0] data
);
//preset lcd write
integer j = 1;

reg [7:0] Datas [1:17];

always @(posedge clk) begin
    Datas[1] = 8'h38; //-- control instruction : configure - 2 lines, 5x7 matrix --
    Datas[2] = 8'h0C; //-- control instruction : Display on, cursor off --
    Datas[3] = 8'h06; //-- control instruction : Increment cursor : shift cursor to right --
    Datas[4] = 8'h01; //-- control instruction : clear display screen --
    Datas[5] = 8'h80; //-- control instruction : force cursor to begin at first line --
    Datas[6] = 8'h43; //-- C --
    Datas[7] = 8'h68; //-- h --
    Datas[8] = 8'h75; //-- u --
    Datas[9] = 8'h61; // -- a --
    Datas[10] = 8'h6E; //-- n --
    Datas[11] = 8'h20; //-- --
    Datas[12] = 8'h62; //-- b --
    Datas[13] = 8'h69; //-- i --
    Datas[14] = 8'h20; //-- --
```



```
Datas[15] = 8'h54; //-- T --
Datas[16] = 8'h4E; //-- N --
Datas[17] = 8'h35 //-- 5 --
end
///////////////////////////////
integer i = 0; //

always @(posedge clk) begin

//-- Delay for writing data

if (i <= 1000000) begin
i = i + 1; lcd_e = 1;
data = Datas[j];
end

else if (i > 1000000 & i < 2000000) begin
i = i + 1; lcd_e = 0;
end

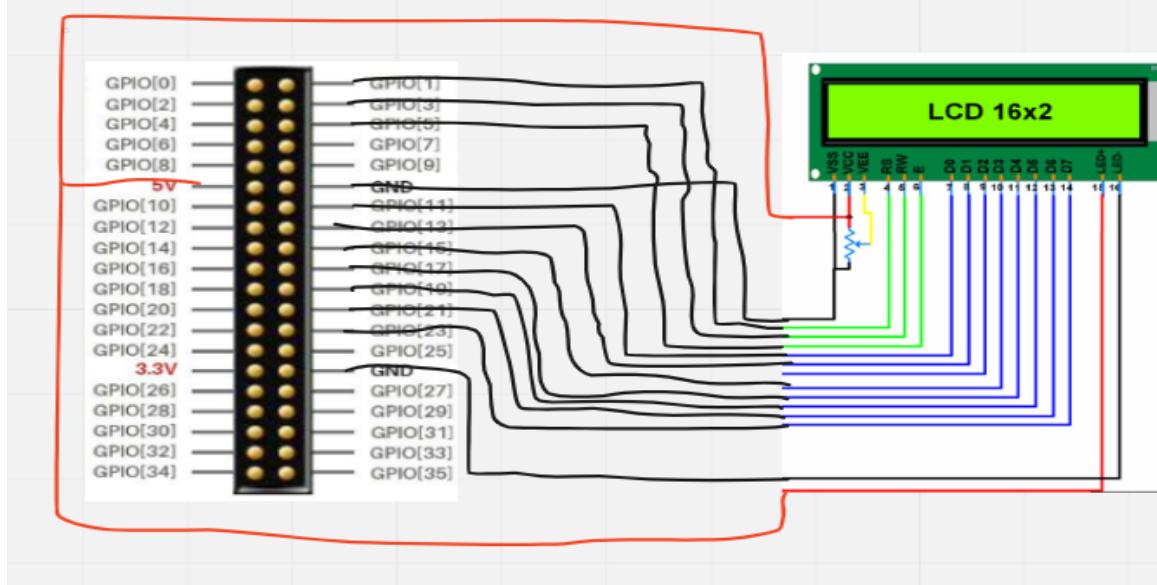
else if (i == 2000000) begin
j = j + 1; i = 0;
end
else i = 0;

//-- LCD_RS signal should be set to 0 for writing commands and to 1 for writing data

if (j <= 5 )
```

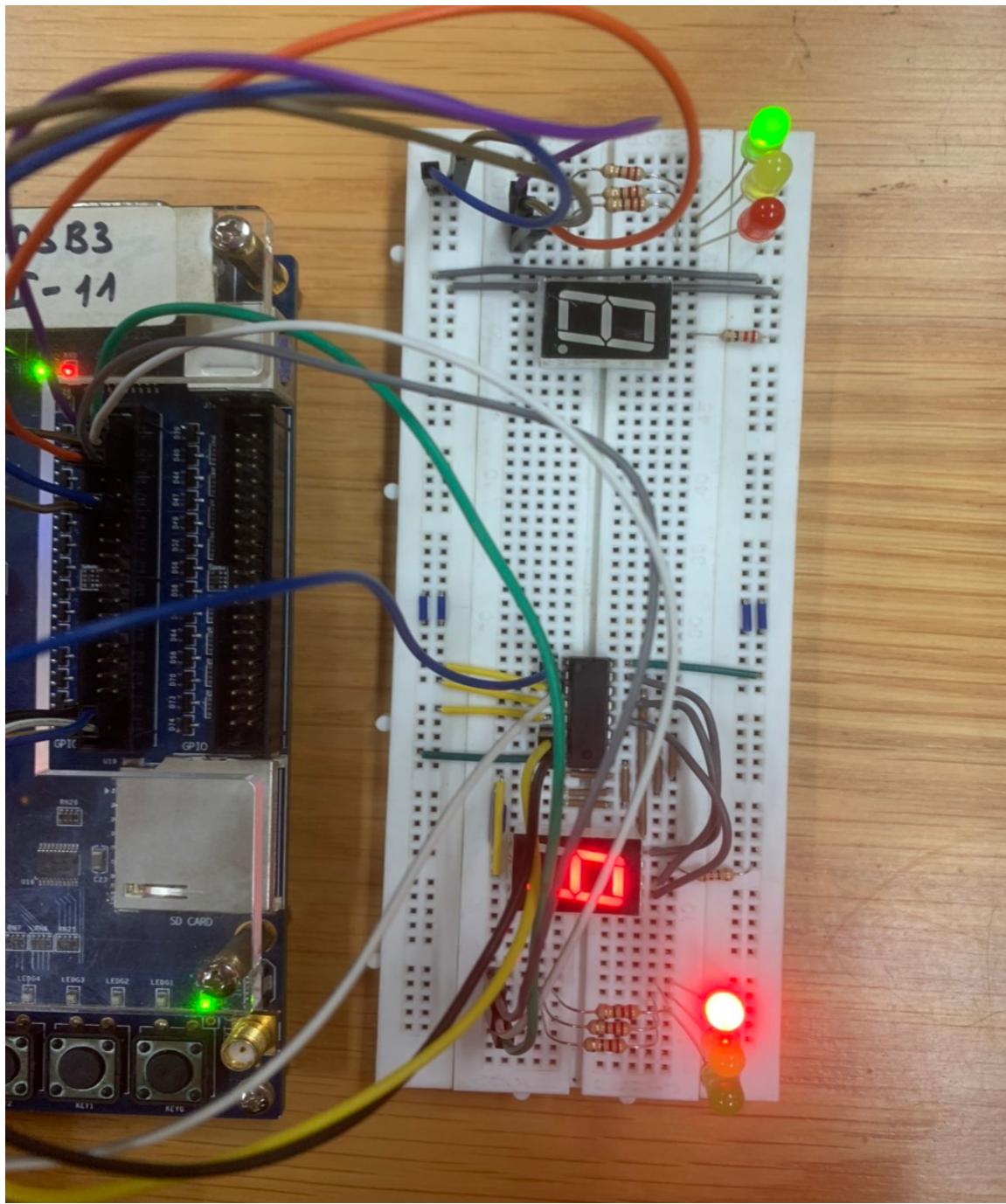
```
lcd_rs = 0;  
else if (j > 5 & j < 18)  
    lcd_rs = 1;  
else if (j == 18)  
    lcd_rs = 0;  
else if (j > 27) begin  
    lcd_rs = 1;  
    j = 5;  
end  
else  
    lcd_rs = 1;  
end  
endmodule
```

Ngoài ra, với chân Contrast Voltage, sinh viên cần lắp biến trở để điều chỉnh. Sinh viên vẽ lại sơ đồ mạch kết nối GPIO với LCD

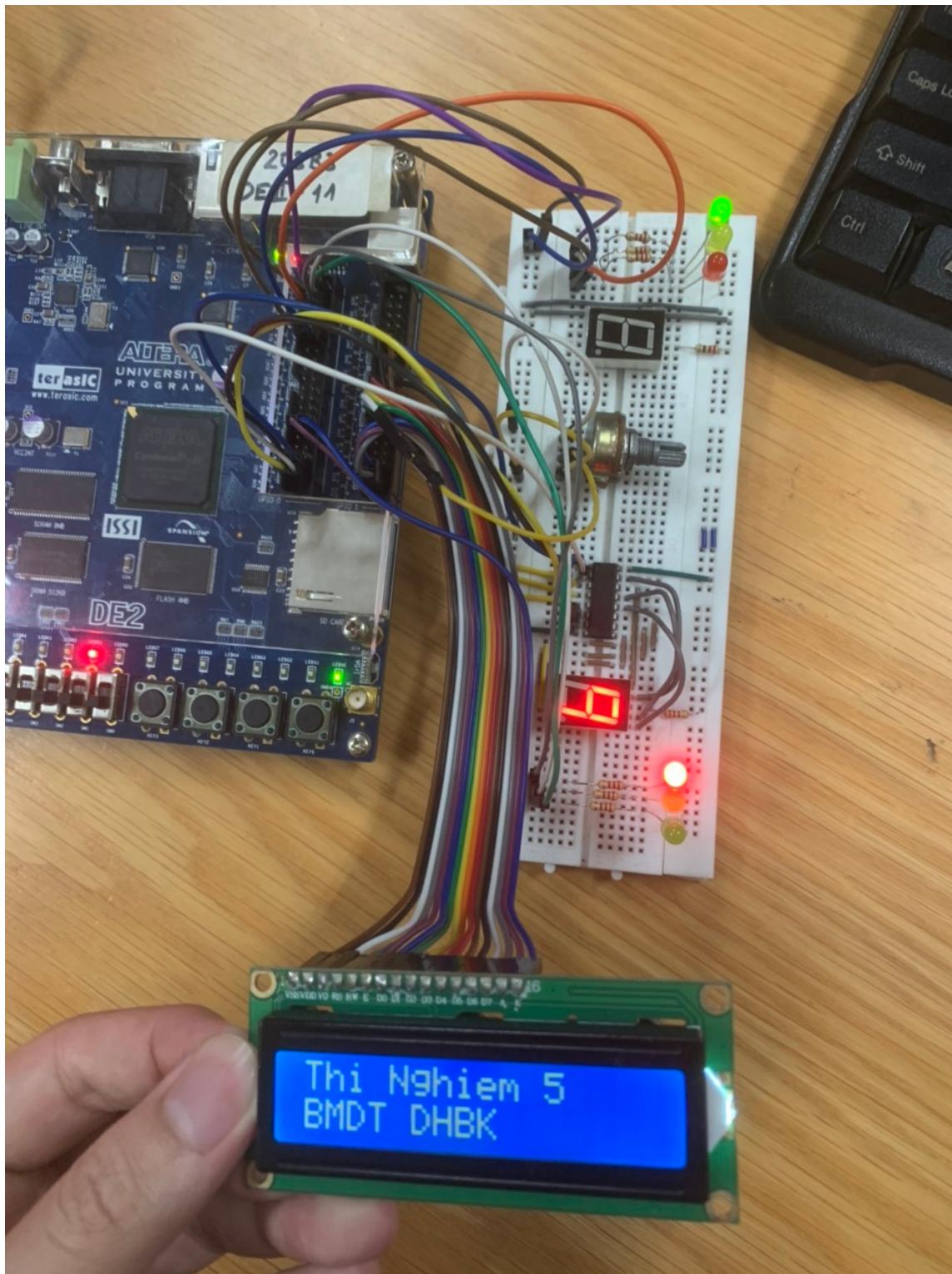


Phần VIII: Mạch lắp tham khảo

Sinh viên tham khảo hình mạch lắp mẫu dưới đây để có thể làm sơ đồ lắp mạch tương tự cho thí nghiệm 1, 2 và 3



Hình 5.4 Sơ đồ lắp mạch ví dụ - 1



Hình 5.5 Sơ đồ lắp mạch ví dụ - 2