

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دستور کار آزمایشگاه مدارهای منطقی

تهیه و تنظیم:

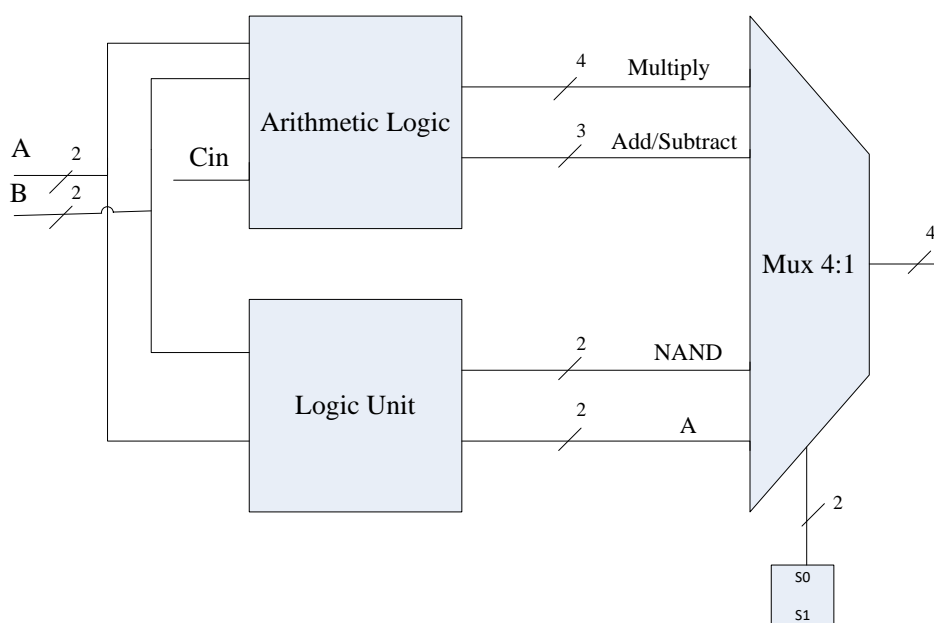
گروه مدرسين آزمایشگاه

سال تحصیلی ۹۶-۹۷

آزمایش ۸

هدف: پیاده‌سازی واحد محاسبه و منطق

بخش محاسبه و منطق یکی از مهم‌ترین قسمت‌های یک پردازنده است. ALU از سه قسمت محاسبه، منطق و کنترل تشکیل شده است. قسمت محاسباتی اعمالی نظیر جمع، تفریق، ضرب و تقسیم را انجام می‌دهد. در قسمت منطقی اعمالی نظیر NOT، XOR، OR، AND و NOR انجام می‌شود. در نهایت، قسمت کنترل نیز وظیفه تعیین واحد عملیاتی و عملیات مورد نظر را به عهده دارد. در این آزمایش هدف پیاده‌سازی یک ALU بسیار ساده است. شکل ۱



شکل ۱: بلوک دیاگرام یک واحد محاسبه و منطق

جدول ۱ کدهای عملیاتی را که توسط بخش محاسبه و منطق در این آزمایش مورد استفاده قرار می‌گیرد، نشان می‌دهد. در این آزمایش به دلیل محدودیت زمانی از پیاده‌سازی سایر عملگرها صرف نظر می‌شود. عملکرد مورد نظر و همچنین داده‌های ورودی را می‌توان با استفاده از سویچ‌های (DIP Switch) موجود بر روی برد تعیین نمود.

جدول ۱: کدهای تعیین کننده عملکرد ALU

	S1	S0
A	0	0
NAND	0	1
ADD/ Subtract _{A-B}	1	0
Multiply	1	1

واحد محاسبه و منطق متناسب با جدول ۱ را ابتدا در سطح ساختاری توصیف کنید. اکنون برنامه محکی بنویسید تا صحت عملکرد طرح را بررسی کند. بعد از اطمینان از صحت نتایج، برنامه را بر روی برد FPGA و توسط نمایشگر هفت قسمتی (7-Segment) تست نمایید (ورودی‌ها ۲-بیتی و خروجی ۴-بیتی در نظر گرفته شود. در حالت جمع و تفریق Cout بیت سوم خروجی است و بیت چهارم '0' است. در عملیات منطقی نیز دو بیت پرارزش خروجی مالتی‌پلکسر را '0' در نظر بگیرید).

توضیحات نحوه عملکرد نمایشگر هفت قسمتی و کدهای لازم جهت راه‌اندازی این نمایشگر روی برد FPGA موجود در آزمایشگاه، در ضمیمه ۴ آمده است.

پیش‌گزارش:

ساختار داخلی نمایشگرهای هفت قسمتی را مطالعه نموده و تفاوت دو نوع آند و کاتد مشترک را توضیح دهید.

آزمایش ۹

هدف: آشنایی با ساختارهای لچ و فلیپ فلاپ

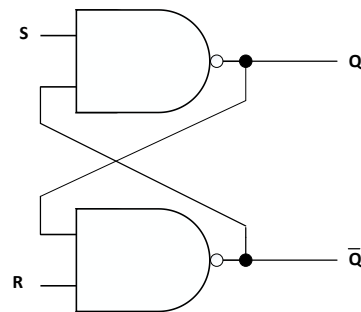
وسایل مورد نیاز:

منبع تغذیه، بردبرد، مالتی متر

تراشه‌های: 7400، 7404، 7408، 7432، 7486، 7476

۱- لچ SR: در این بخش با استفاده از دو گیت NAND که به صورت ضربدری بسته شده‌اند یک لچ SR طراحی نمایید. خروجی‌ها را برای حالت‌های مختلف آزمایش کنید.

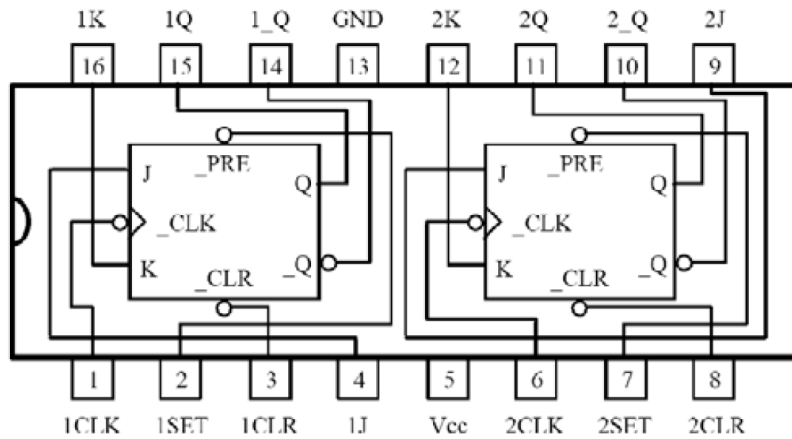
S	R	Q
0	0	
0	1	
1	0	
1	1	



شکل ۱: مدار لچ SR

۲- لچ D و فلیپ فلاپ D: با تغییر لچ SR آن را به لچ D تبدیل نمایید. اکنون با استفاده از این لچ، فلیپ فلاپ D حساس به لبه منفی تولید کنید.

۳- فلیپ فلاپ JK: تراشه 7476 شامل دو فلیپ فلاپ JK می‌باشد. نحوه قرار گرفتن پایه‌های این فلیپ فلاپ در شکل زیر نشان داده شده است. این فلیپ فلاپ را به ازای مقادیر مختلف J و K آزمایش کنید.



شکل ۲: راهنمای پایه‌های تراشه 7476

فلیپ‌فلاپ JK را به فلیپ‌فلاپ T تبدیل و صحت عملکرد را بررسی کنید. نهایتاً فلیپ‌فلاپ T حاصل را به فلیپ‌فلاپ D تبدیل کنید.

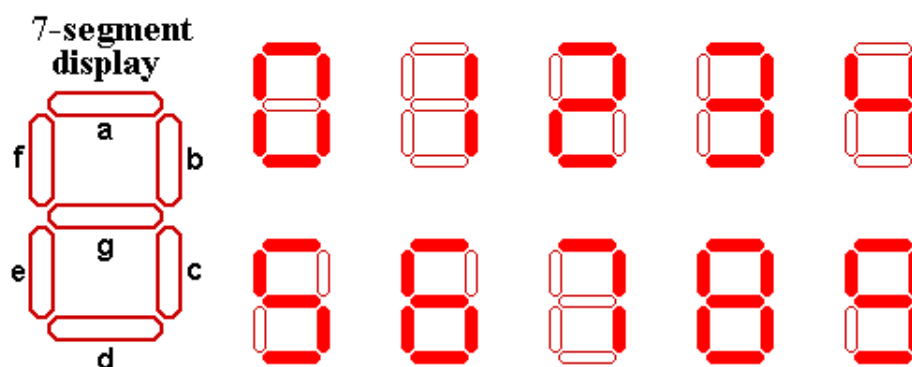
تکالیف پیش از آزمایش:

- ۱- تفاوت لچ و فلیپ‌فلاپ را بنویسید.
- ۲- روش‌های ساختن فلیپ‌فلاپ‌های حساس به لبه از روی لچ‌های حساس به سطح را بیان و یک مورد را توضیح دهید.
- ۳- چگونگی تبدیل فلیپ‌فلاپ T به فلیپ‌فلاپ D را توضیح دهید.

ضمیمه ۴

کار با نمایشگر هفت قسمتی

نمایشگر هفت قسمتی، از هفت قطعه فتوالکتریک، مانند کریستال مایع یا LED ساخته می‌شود. این قطعات به گونه‌ای که در شکل ۱ مشاهده می‌شود در کنار هم قرار داده شده‌اند. با روشن کردن برخی از این LEDها می‌توان اعداد و همچنین برخی از کاراکترها را نمایش داد.



شکل ۱: چگونگی نمایش اعداد با 7-seg [1]

Binary Inputs				Decoder Outputs							7-Segment Display Outputs
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

شکل ۲: نگاشت مقادیر BCD به 7-seg [2]

```

module bcd2seven_seg(a,b,cin,SEG_SEL,SEG_DATA);

input [1:0]a;
input [1:0]b;
input cin;
output [4:0]SEG_SEL;
output [7:0]SEG_DATA;
reg [7:0]SEG_DATA;

always @(mux_out)
begin
    case(mux_out)
        0:SEG_DATA = 8'b00111111;
        1:SEG_DATA = 8'b00000110;
        2:SEG_DATA = 8'b01011011;
        3:SEG_DATA = 8'b01001111;
        4:SEG_DATA = 8'b01100110;
        5:SEG_DATA = 8'b01101101;
        6:SEG_DATA = 8'b01111101;
        7:SEG_DATA = 8'b00000111;

    endcase
end

assign SEG_SEL=5'b00001;

//MODULE INSTANTIATIONS IN BASED ON TABLE 1

//DEFINITION OF CONTROL UNIT WITH MULTIPLEXER

```

```
#PlanAhead Generated physical constraints
```

```
NET "SEG_DATA[0]" LOC = P10;  
NET "SEG_DATA[1]" LOC = P7;  
NET "SEG_DATA[2]" LOC = P11;  
NET "SEG_DATA[3]" LOC = P5;  
NET "SEG_DATA[4]" LOC = P4;  
NET "SEG_DATA[5]" LOC = P12;  
NET "SEG_DATA[6]" LOC = P9;  
NET "SEG_DATA[7]" LOC = P3;
```

```
NET "SEG_SEL[0]" LOC = P15;  
NET "SEG_SEL[1]" LOC = P20;  
NET "SEG_SEL[2]" LOC = P19;  
NET "SEG_SEL[3]" LOC = P18;  
NET "SEG_SEL[4]" LOC = P16;
```

```
NET "a[0]" LOC = P161;  
NET "a[1]" LOC = P162;
```

```
NET "b[0]" LOC = P172;  
NET "b[1]" LOC = P175;
```

```
NET "cin" LOC = P183;
```

منابع:

- [1] "7-Segment Display." [Online]. Available: <http://www.physics.udel.edu/~watson/scen103/7seg.html>.
- [2] "BCD to 7-Segment Decoder." [Online]. Available: <http://macao.communications.museum/eng/exhibition/secondfloor/MoreInfo/Displays.html>.