

دانشکده کامپیوتر

دانشگاه علم و صنعت

طراحی سیستم های دیجیتال

تکلیف شماره ۱

۱- در یک ماژول یک full-adder ۱۶ بیتی طراحی کنید.

۲- در یک ماژول دیگر یک subtractor ۱۶ بیتی طراحی کنید  
(borrow الزامی نیست).

۳- در یک ماژول دیگر یک ضرب کننده ۱۶ بیت در ۱۶ بیت طراحی کنید.

۴- در یک ماژول دیگر یک مدار شیفت به چپ منطقی که تعداد شیفت را از ورودی میگیرد و یک مدار XOR طراحی کنید.

\*\* نکته در طراحی های بالا حق استفاده از اوپرند های خود زبان مانند  
(..., <, +, -, /, \*) را ندارید و تمامی ماژول ها با استفاده از دستورات  
منطقی ساخته شوند (&, |, !, ...).

۵ - تمامی ماژول های طراحی شده بالا را در یک ماژول به هم متصل کنید و آپ کد های زیر را بهشان متصل نمایید.

Example	Instuction	Upcode
$S=a+b$	Add	000
$S=a-b$	Subtract	001
$Hi,Lo=a*b$	Multiply	010
$Out = a \text{ xor } b$	Xor	011
$Out = a \ll 1$	Shift left logical	100

۶- تمامی دستورات بالا خروجی هایی برای **flag** ها دارند که در ماژول طراحی شده در شماره ۵ این فلگ ها را مقدار دهی کنید.

فلگ ها:

Z	زمانی ۱ میشود که حاصل صفر باشد
N	زمانی ۱ میشود که حاصل منفی باشد
C	زمانی ۱ میشود که حاصل <b>carry</b> داشته باشد
O	زمانی ۱ میشود که حاصل <b>over-flow</b> داشته باشد

- تاپ ماژول (ماژول قسمت ۵):

تاپ ماژول یک در این تمرین عملکردی شبیه به **alu** دارد. شما باید در ورودی این ماژول آپکد ها و دو ورودی ۱۶ بیتی بگیرید و در خروجی دو مقدار ۱۶ بیتی قرار دهید (لازم به ذکر است فلگ ها هم میتوانند از نوع **reg** بوده و در خروجی ظاهر نشوند هم میتوانند خروجی باشند)

- نحوه اتصال در تاپ ماژول:

<نام انتخابی در تاپ ماژول> <نام فایل ماژول که همانام با خوده ماژول است>  
(  
, (چیزی که به آن متصل میکنیم) <نام متغیر>.  
(چیزی که به آن متصل میکنیم) <نام متغیر>.  
);

سینتکس اتصال وریلاگ برای adder ای که در کلاس نوشته شد بدین گونه است:

```
Reg first_in, second_in, CIN, COUT, result;
```

```
adder my_adder (  
  .a(first_in),  
  .b(second_in),  
  .cin(CIN),  
  .cout(COUT),  
  .sum(result)  
);
```

مشاهده میکنید که در بالا رجیترهای تعریف شده به ورودی های adder وصل شده اند.

مهلت تحویل دو شنبه ۸ آبان ماه ساعت ۱۲