

تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۳۱ از طریق سامانه‌ی دروس

نکاتی در رابطه با نوع تمرین

هر سری از تمرین‌ها، از چهار بخش تشکیل شده است:

- مرور و تثبیت مفاهیم: پرسش‌های این بخش جهت مرور و یادآوری مفاهیم درسی آورده شده است و با مطالعه مفاهیم درسی گفته‌شده در کلاس درس و اسلایدهای درس خواهید توانست به آن‌ها پاسخ دهید. پاسخ آن‌ها مورد ارزیابی قرار نخواهد گرفت، لذا نیازی به ارسال پاسخ آن‌ها نیست.
- تحلیل و طراحی مدار: پرسش‌های این بخش جهت درک عمیق مفاهیم درسی و افزایش قدرت تحلیل و طراحی سیستم‌های دیجیتال آورده شده است. پاسخ به آن‌ها الزامی بوده و مورد ارزیابی قرار خواهد گرفت.
- توصیف و پیاده‌سازی: پرسش‌های این بخش جهت افزایش مهارت شما در پیاده‌سازی مدارهای دیجیتال، بررسی درستی عملکرد آن و استفاده از ابزارهای طراحی آورده شده است. پاسخ به آن‌ها الزامی بوده و مورد ارزیابی قرار خواهد گرفت.
- طراحی و پیاده‌سازی سامانه پیشرفته: پرسش‌های این بخش ممکن است کمی پیچیده‌تر و دشوارتر از سایر بخش‌ها باشد. الزامی یا اختیاری بودن آن‌ها در صورت پرسش ذکر شده است.

نکاتی در رابطه با نحوه‌ی ارسال تمرین

- ارسال تمرینات به‌صورت الکترونیکی و از طریق [سامانه دروس](#) خواهد بود. فایل ارسالی شما فایل zip با نام sid_hs.zip است که sid شماره دانشجویی و hs شماره سری تمرین است. یک قالب آماده در سامانه دروس قرار داده شده است تا پاسخ تمرین را در قالب تعیین شده بنویسید. پرسش‌هایی که پاسخ آن‌ها ماهیت تشریحی و تحلیلی دارد را به صورت تایپ‌شده یا دستی نوشته و به صورت یک فایل PDF ذخیره نمایید.
- زمان تحويل هر سری از تمرینات مشخص بوده و پاسخ تمرین پس از موعد مقرر شده در سامانه دروس قرار داده خواهد شد لذا امکان تغییر آن وجود ندارد. در حل تمرینات، می‌توانید به‌صورت دوتایی یا چندتایی با یکدیگر همفکری و بحث نمایند ولی هر شخص می‌بایست در نهایت پاسخ و استدلال خود را به‌صورت انفرادی بنویسد و در صورت شباهت پاسخ، تمامی افراد نمره تمرین را از دست خواهند داد.
- بازای هر روز ارسال زودهنگام تمرین‌ها ۵ درصد نمره اضافی تا سقف ۲۰ درصد (در صورتی که نمره خام تمرین بالاتر از ۷۵ درصد شود) در نظر گرفته خواهد شد. بازای هر روز دیرکرد در ارسال تمرین ۱۵ درصد جریمه اعمال خواهد شد. ارسال دیرهنگام فقط تا ۲ روز پس از زمان مقرر شده امکان‌پذیر است و پس از آن نمره تمرین صفر در نظر گرفته خواهد شد.

چنانچه ابهامی در زمینه تمرینات دارید، می‌توانید اشکالات خود را از طریق پست الکترونیکی زیر با موضوع PDS.2021 رفع نمایید.

[ali\[mohammadpour\[at\]aut\[dot\]ac\[dot\]ir](mailto:ali[mohammadpour[at]aut[dot]ac[dot]ir)

محمدپور

موفق و پیروز باشید!

تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۳۱ از طریق سامانه‌ی دروس

پرسش‌های مرور و تثبیت مفاهیم
(نیازی به ارسال پاسخ این بخش نیست.)

الف) به پرسش‌های زیر پاسخ دهید.

- مزیت تعریف و انتساب ثابت‌ها^۱ در بدنه‌ی پکیج^۲ چیست؟
 - تفاوت اصلی روال^۳ و تابع^۴ در زبان توصیف سخت‌افزار VHDL چیست؟
 - چرا در طراحی سیستم‌های دیجیتال از اصطلاح **چرخه‌ی طراحی** استفاده می‌شود؟
 - دو ساختار رایج بسته‌بندی تراشه یعنی BGA و QFP را مقایسه کرده و مزایا و معایب هر یک را بیان کنید.
 - با توجه به اینکه فقط بخشی از زبان توصیف سخت‌افزار سنتزپذیر است، آیا در طراحی سیستم‌های دیجیتال به ساختارهایی که قابلیت سنتز ندارند نیاز است؟
 - در جریان طراحی سیستم‌های دیجیتال، انتساب پایه‌های تراشه قبل از چه مرحله‌ای باید تعیین شود؟ چرا؟ چه دلایلی باعث می‌شود تا انتساب پایه‌ها پس از انتساب اولیه تغییر کند؟
- ب) یک روال در زبان VHDL بنویسید که یک بردار از نوع STD_LOGIC_VECTOR با طول نامشخص را بگیرد و طول بردار (تعداد بیت‌ها) را به صورت یک عدد از نوع Integer برگرداند.

پ) قطعه کدهای زیر چه مداری را توصیف می‌کند؟ آن را رسم کنید.

```
Process (a, b, p, q)                                process (a, b, c, x, y)
begin                                                begin
    y <= x when p='1' else (not x);                if (a='1') then
    x <= (a and b) when p='1' else                  z <= x;
        (a or b);
    z <= y when q='1' else 'z';
end process;                                         elsif (b='1') then
                                                    z <= x or y;
                                                    elsif (c='1') then
                                                    z <= x and y;
                                                    else
                                                    z <= not y;
                                                    end if;
end process;
```

ت) نوع داده‌ی زیر را در نظر بگیرید. این نوع داده برای مدل‌سازی زمان مورد استفاده قرار می‌گیرد. عملگر + و - را برای این نوع داده، گرانبار کنید. به عنوان مثال اگر زمان ۱:۲۳:۴۵ و ۵:۴۳:۲۱ باهم جمع شوند حاصل ۰۷:۰۷:۰۶ خواهد بود. در نظر داشته باشید که زمان منفی وجود ندارد، لذا در صورتی که حاصل عددی منفی شود، مقدار حاصل از این عملگر صفر خواهد بود.

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Hour					Minute					Second						

¹ Constant

² Package Body

³ Procedure

⁴ Function

تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۳۱ از طریق سامانه‌ی دروس

پرسش‌های تحلیل و طراحی مدار، توصیف و پیاده‌سازی (ارسال پاسخ این بخش الزامی است.)

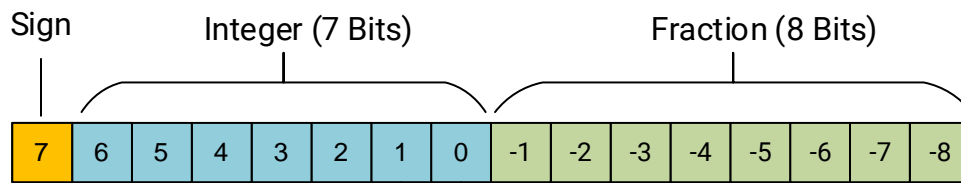
پرسش ۱-۴

هدف از این تمرین گرانبار کردن^۵ عملیات جمع و تفریق و ضرب برای اعداد ممیز ثابت است که قابلیت سنتز داشته باشد. از نوع داده‌ی real برای بررسی درستی عملیات استفاده می‌کنیم.

- حتماً از قطعه کدهای قرار داده شده در صفحه‌ی گیت‌هاب درس استفاده نمایید.
- یک پکیج با نام xpoint_pkg.vhd تعریف شده است. نوع داده‌ی xfixed یک بردار با طول متغیر است.

```
type xfixed is array (integer range <>) of std_logic;
```

- شکل ۱ ساختار اعداد ممیز ثابت ۱۶ بیتی با ۸ بیت در بخش اعشار و ۷ بیت در بخش صحیح و یک بیت علامت را نشان می‌دهد.



شکل ۱: ممیز ثابت ۱۶ بیتی

- کد زیر نحوه‌ی تعریف نوع داده‌ی ممیز ثابت شکل ۱ را نشان می‌دهد.

```
signal sig_xfixed: xfixed (7 downto -8);
```

- رابطه‌ی زیر نحوه محاسبه مقدار در این نوع نمایش اعداد را برای $Q1.n.m$ نشان می‌دهد که دارای یک بیت علامت، n بیت صحیح و m بیت اعشار است.

```
signal sig_xfixed: xfixed (n downto -m);
```

$$-S \times 2^n + \sum_{i=0}^{n-1} 2^i + \sum_{j=0}^m 2^{-j}$$

- هنگام جمع و تفریق دو عدد ممیز ثابت ابتدا باید محل ممیز را یکسان کرد. این کار با شیفت به چپ یا راست و با افزودن تعدادی صفر به بخش کم‌ارزش قابل انجام است. هنگام ضرب نیز باید توجه کرد که محل ممیز هنگام ضرب باید به درستی انتخاب شود. راهنمایی: لینک زیر نحوه‌ی جمع، تفریق و ضرب این نوع اعداد را توضیح داده است.

[https://en.wikipedia.org/wiki/Q_\(number_format\)](https://en.wikipedia.org/wiki/Q_(number_format))

- یک فایل با نام xdsp.vhd قرار داده شده است که در واقع یک واحد محاسبات ممیز ثابت ۱۶ بیتی است.
- یک فایل محیط آزمون با نام tb_xdsp.vhd برای بررسی درستی پکیج و واحد محاسبات ممیز ثابت استفاده می‌شود.

⁵ Overloading

تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۳۱ از طریق سامانه‌ی دروس

- (الف) مطابق با موارد توضیح داده شده، عملگر + (جمع) را برای نوع داده‌ی **xfixed** گرانبار کنید. جهت سادگی مسأله عرض بیتی نوع خروجی را مطابق با عملوند سمت چپ علامت + در نظر بگیرید.
- (ب) مطابق با موارد توضیح داده شده، عملگر - (تفریق) را برای نوع داده‌ی **xfixed** گرانبار کنید. جهت سادگی مسأله عرض بیتی نوع خروجی را مطابق با عملوند سمت چپ علامت - در نظر بگیرید.
- (ج) مطابق با موارد توضیح داده شده، عملگر * (ضرب) را برای نوع داده‌ی **xfixed** گرانبار کنید. خروجی دارای یک بیت علامت است و تعداد بیت‌های بخش صحیح آن برابر با مجموع تعداد بیت‌های بخش صحیح در دو عملوند و تعداد بیت‌های بخش اعشار آن برابر با بیشینه‌ی تعداد بیت‌های بخش اعشار دو عملوند است.
- (د) روالی طراحی نمایید که یک عدد ممیز ثابت را بگیرد و تعداد بیت‌های اعشار و تعداد بیت‌های صحیح را به شکل **integer** برگرداند.
- (ه) تابعی طراحی نمایید که یک عدد ممیز ثابت را بگیرد و آن را به نوع داده‌ی **real** تبدیل کند.
- (و) واحد محاسبات ممیز ثابت (**xdsp.vhd**) دارای دو ورودی ۱۶ بیتی ممیز ثابت (به شکل Q1.7.8) و یک ورودی سه بیتی **opr** است. خروجی آن یک عدد ممیز ثابت ۲۴ بیتی (به شکل Q1.15.8) است. با توجه به جدول ۱ آن را توصیف نمایید.

جدول ۱: واحد محاسبه ممیز ثابت ۱۶ بیتی

حالت	عملیات
$opr = 000$	$outp \leftarrow inpX$
$opr = 001$	$outp \leftarrow inpX + inpY$
$opr = 010$	$outp \leftarrow inpX - inpY$
$opr = 011$	$outp \leftarrow inpX \times inpY$
$opr = 100$	$outp \leftarrow (-1) \times inpX$
$opr = 101$	$outp \leftarrow 16 \times inpX$
$opr = 110$	$outp \leftarrow inpX / 16$
$opr = 111$	$outp \leftarrow inpY + 8.125$

- (ز) یک فایل محیط آزمون طراحی نمایید و درستی همه‌ی حالت‌های ممکن **opr** در واحد محاسبات ممیز ثابت را برای دو بردار آزمون زیر بررسی کنید.

جدول ۲: بردار آزمون

شماره بردار	$inpX$	$inpY$
۱	۲۴/۰	-۳۲/۰
۲	۱۶/۵۶۲۵	-۱۶/۵۶۲۵

- (ح) یک فایل محیط آزمون طراحی نمایید که مطابق قالب زیر، نتیجه‌ی هر یک از ۱۶ حالت را در فایلی با نام **test.txt** بنویسد.

`(inpX, inpY, opr, outp) :: (inpX_real, inpY_real, outp_real)`

برای تبدیل اعداد به نوع داده‌ی **real** از تابعی که در بخش (ه) طراحی کرده‌اید استفاده نمایید.

تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۳۱ از طریق سامانه‌ی دروس

پرسش ۲-۴

تابع غیرخطی سیگموئید در شبکه‌های عصبی مورد قرار می‌گیرد.

$$\tanh(x) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}} \right)$$

- ورودی مدار (x) یک بردار ممیز ثابت ۶ بیتی به‌صورت (Q1.2.3) است.
- خروجی مدار یک بردار ۱۶ بیتی در قالب ممیز شناور است.
- شکل ۲ قالب نوع داده‌ی ممیز شناور را نشان می‌دهد.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S	Exponent					Fraction									

شکل ۲: ممیز ثابت ۱۶ بیتی

$$result = (-1)^S \times (1.Fraction) \times 2^{Exponent}$$

الف) ماژول مربوط به تابع \tanh را با استفاده از حافظه طراحی نمایید. معماری را `arch_lut` نام‌گذاری کنید.

ب) ماژول مربوط به تابع \tanh را می‌توان با استفاده از رابطه‌ی زیر پیاده‌سازی کرد. معماری را `arch_hard` نام‌گذاری کنید.

$$\text{sigmoid}(x) = \begin{cases} 1 & x < -1 \\ x & -1 \leq x \leq 1 \\ -1 & x > 1 \end{cases}$$

ج) یک فایل محیط آزمون طراحی نمایید که بازای همه‌ی حالات ورودی خروجی ماژول \tanh را با ساختار زیر در فایل `nonlinear.txt` بنویسد.

```
Arch_lutx: [Input as Vector], [Input as real], [Output as Vector], [Output as Real]
Arch_hard: [Input as Vector], [Input as real], [Output as Vector], [Output as Real]
```

د) هر دو طرح را سنتز و پیاده‌سازی و جدول زیر را کامل کنید.

جدول ۳: گزارش منابع مصرفی

گزارش منابع در روش HARD		گزارش منابع در روش LUT		نوع منبع
پس از پیاده‌سازی	پس از سنتز	پس از پیاده‌سازی	پس از سنتز	
				Lookup Table
				Flip-Flop
				BRAM
				DSP
				IO
				BUFF-Gates

ه) دو روش پیاده‌سازی را تحلیل و مزایا و معایب هر یک را بیان کنید.