

تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۱۷ 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 ۱۴۰۰/۰۲/۱۷ از طریق سامانه‌ی دروس

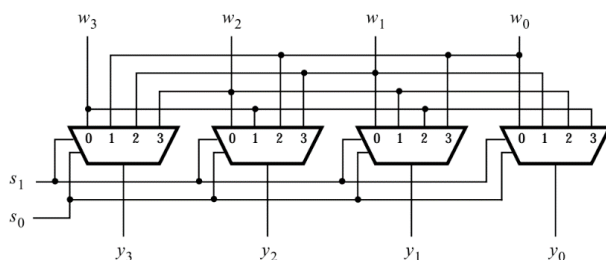
پرسش‌های مرور و تثبیت مفاهیم  
(نیازی به ارسال پاسخ این بخش نیست.)

الف) با استفاده از دستور Record برای اعداد مختلط یک داده‌ساختار طراحی نمایید.

$$\text{Complex Number} \begin{cases} \text{Real Part} \\ \text{Imaginary Part} \end{cases}$$

ب) در زبان VHDL تفاوت Buffer و Inout را در نظر بگیرید. آیا همواره می‌توان از Inout به جای Buffer استفاده کرد؟ چرا؟

پ) شکل ۲ یک Barrel-Shifter است که با استفاده از مالتی پلکسر طراحی شده است و عمل شیفت دورانی انجام می‌دهد. با ایجاد تغییرات در ساختار Barrel-Shifter مداری طراحی کنید که عملیات شیفت پرسش ۳ را انجام دهد. سپس یک توصیف ساختاری به زبان VHDL برای این مدار بنویسید.

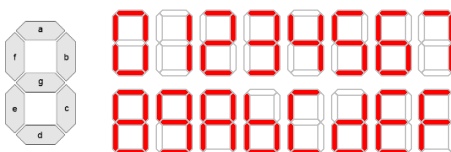


شکل ۲: Barrel-Shifter

ت) شکل ۳ ساختار یک نمایشگر هفت قسمتی<sup>۱</sup> را نشان می‌دهد. در هفت قسمتی آند مشترک، قسمت‌هایی روشن می‌شوند که ورودی متناظر آن برابر یک باشد. به عنوان مثال کد "0000111" برای نشان دادن عدد هفت به کار می‌رود. (قسمت‌ها در این رشته از راست به چپ به ترتیب متناظر با حروف a, b, c و ... قرار گرفته است).

۱) با استفاده از دستور with-select-when مدار مبدل عدد چهار بیتی بدون علامت به کد هفت قسمتی آند مشترک را توصیف نمایید.

۲) برای تبدیل هفت قسمتی آند مشترک به هفت قسمتی کاتد مشترک (قسمت مربوط به صفرها روشن می‌شود) چه راهکاری پیشنهاد می‌کنید؟ آیا می‌توان با ترکیب هفت قسمتی آند مشترک با یک واحد ترکیبی آن را به نوع کاتد مشترک تبدیل کرد؟ چگونه؟



شکل ۳: نمایشگر هفت قسمتی

<sup>۱</sup> Seven Segment Display

تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۱۷ از طریق سامانه‌ی دروس

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

پرسش ۱-۳

پاسخ هر یک از پرسش‌های زیر را به طور مختصر بیان کنید.

- الف) تفاوت اصلی دستور **when-else** با دستور **if-then-else** در چیست؟ آیا از نظر سخت‌افزاری یکسان هستند؟
- ب) تفاوت **signal** و **constant** در چیست؟ چگونه می‌توان از سیگنال برای تعریف ثابت استفاده کرد؟
- ج) تفاوت اصلی سیگنال (**signal**) با متغیر (**variable**) چیست؟
- د) در چه صورتی یک سیگنال خروجی می‌تواند بیش از یک درایور داشته باشد؟
- ه) تفاوت نوع داده‌ی **bit** و **std\_logic** را بیان کنید.

پرسش ۱-۳

در زبان توصیف سخت‌افزار VHDL، بردار دو بعدی (ماتریس دو بعدی) را می‌توان به دو روش تعریف کرد:

```
type row is array (3 downto 0) of std_logic;  
type col is array (0 to 7) of row;
```

(الف)

```
signal matrix : col;
```

```
type row_col is array (3 downto 0, 7 downto 0) of std_logic;
```

(ب)

```
signal matrix : row_col;
```

- ۱) شباهت‌ها و تفاوت‌های دو روش «الف» و «ب» را از مقداردهی اولیه، انتساب و دسترسی به یک درایه را بیان کنید.
- ۲) یک مثال مقداردهی اولیه، انتساب و دسترسی برای هر روش ارائه دهید.
- ۳) از نظر دسترسی به درایه کدام روش برای توصیف کننده (طراح سخت‌افزار) راحت‌تر است؟ برای دسترسی به سطر یا ستون (و یا بخشی از آن) چگونه؟ آیا در هر دو امکان‌پذیر است؟
- ۴) یک مثال برای انتخاب زیرماتریس ۲ در ۲ از ماتریس در هر دو روش ارائه دهید. آیا در هر دو این امکان وجود دارد؟

پرسش ۳-۳

شکل موج مربوط به هر یک از سیگنال‌ها را تا زمان ۵۰ نانو ثانیه رسم کنید. دلیل مقدار را در هر بخش مشخص کنید.

```
signal p,q,r : std_logic;  
signal s : std_logic := '1'  
-----  
p <= '1' after 5 ns, q after 10 ns;  
q <= '1' when p='0' else 'Z' when p='1' else s and p;  
r <= q when p='0' else not q;  
r <= 'Z';  
s <= not s after 5 ns;
```

تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۱۷ از طریق سامانه‌ی دروس

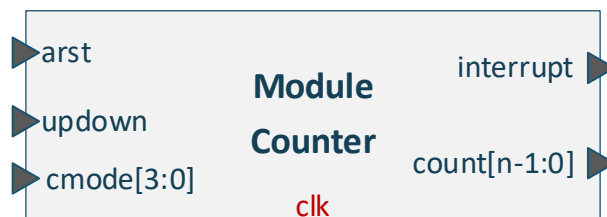
### پرسش‌های توصیف و پیاده‌سازی (ارسال پاسخ این بخش الزامی است.)

#### پرسش ۳-۴

شکل ۴ یک شمارنده‌ی پیمانه‌ای را نشان می‌دهد. در شمارنده‌ی پیمانه‌ای در هر چرخه‌ی ساعت<sup>۲</sup>، خروجی به تعداد مشخص (پیمانه) افزایش یا کاهش می‌یابد.

**الف)** یک شمارنده‌ی پیمانه‌ای دارای سیگنال reset ناهمگام در سطح انتقال ثبات (RTL) توصیف نمایید که:

- ورودی *cmode* پیمانه‌ی شمارنده است که یک عدد بدون علامت ۴ بیتی است.
- ورودی *updown* افزایشی یا کاهشی بودن شمارنده را نشان می‌دهد. اگر یک باشد شمارش افزایشی است.
- خروجی *interrupt* هنگامی یک می‌شود که سرزیر<sup>۳</sup> یا زیرریز<sup>۴</sup> رخ دهد. در شمارنده‌ی کاهشی هنگامی فعال می‌شود که مقدار خروجی کمتر از صفر شود و در شمارنده‌ی افزایشی زمانی که از خروجی بیشترین مقدار نمایش بزرگ‌تر شود، فعال می‌شود.
- خروجی یک عدد بدون علامت *n* بیتی است که مقدار شمارش در چرخه ساعت جاری را نشان می‌دهد. (*n* را به صورت generic با مقدار پیش‌فرض ۸ در نظر بگیرید.)



شکل ۴: شمارنده‌ی پیمانه‌ای

**ب)** یک فایل محیط آزمون طراحی نمایید که درستی مدار قسمت «الف» را در حالت‌های زیر بررسی کند.

- شمارنده‌ی افزایشی با پیمانه‌ی ۷ و خروجی ۸ بیتی (ریست شدن شمارنده پس از ۵ کلاک)
- شمارنده‌ی افزایشی با پیمانه‌ی ۱۳ و خروجی ۱۲ بیتی (ریست شدن شمارنده پس از ۸ کلاک)
- شمارنده‌ی نزولی با پیمانه‌ی ۹ و خروجی ۱۲ بیتی (ریست شدن شمارنده پس از ۱۰ کلاک)
- شمارنده با پیمانه‌ی ۳ و خروجی ۴ بیتی (۱۰ کلاک افزایشی و سپس کاهشی، ریست در شروع شبیه‌سازی)

#### پرسش ۳-۵

در شکل ۵، انواع دستورالعمل‌های پایه معماری RISC-V نشان داده شده است. هر دستورالعمل ۳۲ بیتی است.  
**الف)** با استفاده از دستورات *type*، *subtype* و ... انواع جدیدی برای پوشش بخش‌های مختلف دستورالعمل‌های این معماری تعریف کنید. (تعریف را در داخل پکیج *riscv\_types\_pkg.vhd* بنویسید.)

**ب)** با استفاده از انواع داده‌ی قسمت «الف» و دستور *record*، یک داده‌ساختار برای هر یک از دستورالعمل‌های معماری RISC-V طراحی کنید.

<sup>2</sup> Clock Cycle

<sup>3</sup> overflow

<sup>4</sup> underflow

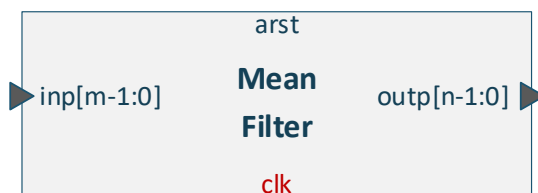
تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۱۷ از طریق سامانه‌ی دروس

31	25 24	20 19	15 14 12 11	7 6	0	
funct7	rs2	rs1	funct3	rd	opcode	R-type
imm [11:0]		rs1	funct3	rd	opcode	I-type
imm [11:5]		rs2	rs1	funct3	imm [4:0]	S-type
imm [12]	imm [10:5]	rs2	rs1	funct3	imm [4:1 11]	B-type
imm [31:12]				rd	opcode	U-type
imm [20 10:1 11 19:12]				rd	opcode	J-type

شکل ۵: مجموعه دستورالعمل‌های پایه معماری RISC-V

### پرسش ۳-۶

در یک فیلتر میانگیر<sup>۵</sup>، ورودی به صورت سریالی است و خروجی میانگین وزنی ورودی با یک یا چند داده‌ی ورودی قبلی است. هر فیلتر میانگیر با عمق فیلتر، وزن‌ها و عرض بیتی ورودی و خروجی مشخص می‌شود. شکل ۶ نمونه‌ای از آن را نشان می‌دهد.



شکل ۶: فیلتر میانگیر

**الف)** یک فیلتر میانگیر در سطح انتقال ثبات توصیف نمایید که ورودی ۱۶ بیتی، وزن‌ها ۳ بیتی و خروجی ۱۶ بیتی باشد. مقدار وزن‌ها در جدول ۱ آمده است. ضریب ورودی جاری برابر  $coef_0$  و ضریب ورودی در هفت کلاک قبل  $coef_7$  است. ضرایب بدون علامت هستند.

جدول ۱: ضرایب فیلتر میانگیر

$coef_7$	$coef_6$	$coef_5$	$coef_4$	$coef_3$	$coef_2$	$coef_1$	$coef_0$
۲	۳	۵	۶	۶	۵	۳	۲

**ب)** برای مدار قسمت «الف»، فایل محیط آزمون را نوشته و آن را در حالت‌های زیر شبیه‌سازی نمایید.

- ورودی در همه‌ی چرخه‌های ساعت برابر ۱۰ باشد. شکل موج خروجی را ارائه دهید.
  - ورودی به صورت (۱۰، ۵، ۱۵ و سپس ۲۰) به طور مرتب تکرار شود. شکل موج خروجی را ارائه دهید.
- ج)** مدار را در ابزار ویوادو<sup>۶</sup> سنتز و پیاده‌سازی نمایید. تعداد LUT و DSP استفاده شده را ارائه دهید.

<sup>۵</sup> Mean filter

<sup>۶</sup> Vivado

تحويل تا ساعت ۲۳:۵۵ روز جمعه مورخ ۱۴۰۰/۰۲/۱۷ از طریق سامانه‌ی دروس

پرسش ۳-۷ (اختیاری: نمره اضافی دارد.)

PicoBlaze یک پردازنده‌ی ۸ بیتی است که توسط شرکت زایلینکس ارائه شده است. جهت آشنایی بیشتر می‌توانید به لینک زیر مراجعه نمایید.

<https://en.wikipedia.org/wiki/PicoBlaze>

به لینک زیر مراجعه کرده و کد VHDL پردازنده را دانلود نمایید.

<https://github.com/binaryhardware/PicoBlazeMCU>

**الف)** ابتدا در محیط Vivado یک پروژه ایجاد کنید. سپس فایل توصیف‌شده را به آن اضافه کرده و طرح را سنتز و پیاده‌سازی<sup>۷</sup> نمایید. با توجه به گزارش‌های پس از سنتز و پیاده‌سازی، جدول ۲ را کامل کنید.

جدول ۲: تعداد منابع گزارش شده در ابزار

تعداد منبع تخمین‌زده شده		نوع منبع
پس از پیاده‌سازی	پس از سنتز	
		Lookup Table
		LUTRAM
		Flip-Flop
		BRAM
		DSP
		IO
		BUFF-Gates

**ب)** به نظر شما کد مربوط به پردازنده در چه سطح تجریدی توصیف شده است؟ دلیل این کار چیست؟ ذکر مثال توضیح دهید.

<sup>7</sup> Implementation