



مهلت تحویل تا ساعت ۲۳:۵۵ روز دوشنبه مورخ ۹۹/۰۳/۰۵ 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. برای پرسش‌هایی که ماهیت کد نویسی دارند، فایل قالب تمرین در سامانه قرار داده شده است. آن را تکمیل نموده و در پوشه SRC قرار دهید.
  - زمان تحویل 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.2020 رفع نمایید.

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

محمدپور

موفق و پیروز باشید!

مهلت تحویل تا ساعت ۲۳:۵۵ روز دوشنبه مورخ ۹۹/۰۳/۰۵ از طریق سامانه‌ی دروس

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

- الف) روش‌های کدگذاری حالات مختلف یک ماشین حالت را نام برده و مزایا و معایب هریک را بیان کنید.
- ب) توضیح دهید در چه صورت سیگنال‌ها و متغیرهای موجود در یک فرایند<sup>۱</sup> در حین سنتز به حافظه تبدیل می‌شوند.
- ج) فرض کنید جهت طراحی یک شمارنده که ورودی فعال‌ساز همگام و ریست غیرهمگام دارد، از قطعه کد ۴-۱ استفاده شود.

قطعه کد ۴-۱

```
-- type of signal 'count' is std_logic_vector(7 downto 0)

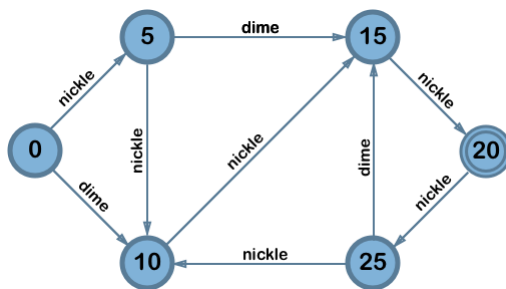
active_clock <= clock and enable;

process (active_clock)
begin
    if reset = '1' then
        count <= (others => '0')
    else
        if rising_edge(active_clock) then
            count <= count + '1';
        end if;
    end if;
end process
```

- آیا ممکن است کد توصیف‌شده مشکلی ایجاد کند؟ توضیح دهید.
- در صورتی که کد مشکلی دارد آن را تغییر دهید تا یک شمارنده دارای فعال‌کننده را پیاده‌سازی کند. همچنین فضای کمتری اشغال کند و Latch ناخواسته نداشته باشد.

د) نقش و دلایل شبیه‌سازی قبل و بعد از سنتز را بیان کنید.

ه) مدار مربوط به ماشین حالت شکل ۴-۱ را توصیف نمایید. سپس در ابزار Vivado آن را به صورت One-Hot و Gray سنتز نمایید. میزان منابع مصرفی دو روش را با یکدیگر مقایسه کنید.



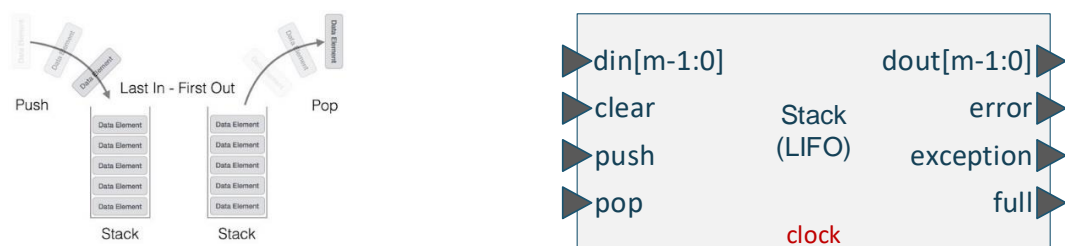
شکل ۴-۱

<sup>۱</sup> Process

## پرسش‌های مرور و تثبیت مفاهیم (ارسال پاسخ این بخش الزامی است.)

### پرسش ۱-۴

یک حافظه‌ی پشته با ابعاد  $2^n \times m$  طراحی نمایید.  $2^n$  تعداد خانه‌های حافظه و  $m$  طول هر یک از داده‌ها است. سیاست حافظه‌ی پشته *LIFO (Last In First Out)* است. جهت پیاده‌سازی آرایه‌ای از داده‌های  $m$  بیتی تعریف نمایید. سپس یک متغیر یا سیگنال جهت اشاره به محل داده‌ی خروجی یا ورودی تعریف نمایید و در هنگام نوشتن در پشته یک واحد به آن اضافه کنید و در هنگام خواندن از آن یک واحد از آن کم کنید.



شکل ۲-۴ پشته

- خروجی *empty* زمانی فعال می‌شود که پشته خالی باشد.
- اگر ورودی *clear* برابر یک شود، پشته در کلاک بعدی باید خالی شود. یعنی همه‌ی محتوا نامعتبر شوند.
- اولویت *clear* از *pop* و *push* بالاتر است.
- خروجی *full* زمانی فعال می‌شود که پشته پر باشد.
- خروجی *expection* زمانی فعال می‌شود که سرریز<sup>۲</sup> یا زیرریز<sup>۳</sup> رخ دهد.

### پرسش ۲-۴

فیلتر دیجیتالی را در نظر بگیرید که در آن داده‌های ورودی  $x[t]$  به صورت سریالی در هر کلاک وارد سیستم می‌شوند و خروجی لحظه‌ای  $y[t]$  برابر با میانگین وزنی چهار داده‌ی قبلی است. در این فیلتر رابطه‌ی زیر برقرار است:

$$y[t] = \frac{(c_0 \cdot x[t] + c_1 \cdot x[t-1] + c_2 \cdot x[t-2] + c_3 \cdot x[t-3])}{4}$$

داده‌ی ورودی و داده‌ی خروجی را ۸ بیتی در نظر بگیرید. ضرایب  $(c_i)$  ۴ بیتی هستند.

(الف) مدار مربوط به فیلتر را توصیف نمایید.

(ب) یک فایل محیط آزمون برای فیلتر طراحی شده بنویسید که مقدار همه‌ی ضرایب در آن برابر یک باشد و داده‌های ورودی از فایلی با نام *filter.data* خوانده شود. شکل موج ۱۰ خروجی اول را ذخیره کرده و ضمیمه کنید.

(ج) گزارش مربوط به توان مصرفی و میزان منابع مصرفی پس از پیاده‌سازی<sup>۴</sup> را ارائه دهید.

<sup>2</sup> Overflow

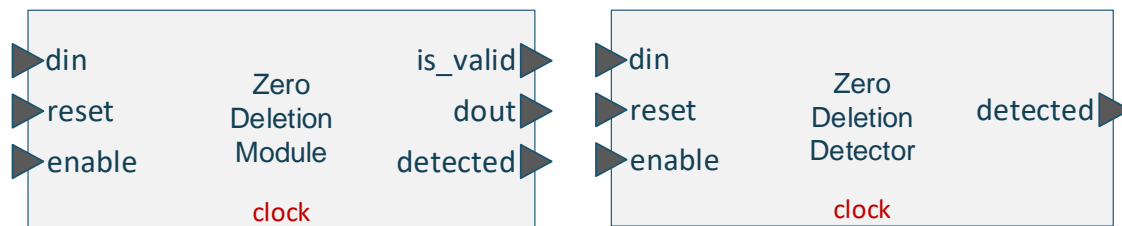
<sup>3</sup> Underflow

<sup>4</sup> Implementation

### پرسش ۳-۴

در انتقال داده‌های دیجیتال به صورت سریالی جهت کاهش نویز و تصحیح خطا از روش‌های متفاوتی استفاده می‌شود. یکی از این روش‌ها Zero Deletion and Insertion نام دارد. در سمت فرستنده‌ی داده، مدار Zero Insertion قرار می‌گیرد که به ازای هر ۵ داده‌ی ۱ پشت سر هم، یک صفر اضافه می‌کند. به عنوان مثال داده‌ی 11011111101 به صورت 1101111110101 ارسال می‌شود. در سمت گیرنده‌ی داده، مدار Zero Deletion قرار می‌گیرد که صفرهای اضافه شده در سمت فرستنده را کشف کرده و آن را حذف می‌کند. بدیهی است که تشخیص دهنده‌ی Zero Deletion و Zero Insertion نوعی ماشین حالت است.

**الف)** مطابق شکل ۳-۴ مدار مربوط به تشخیص دهنده‌ی Zero Deletion Detector را به صورت ماشین حالت Mealy توصیف نمایید. در این مدار اگر ورودی enable فعال باشد، در صورت تشخیص صفر اضافی خروجی detected برابر یک می‌شود و اگر enable غیرفعال باشد، خروجی برابر صفر است.



شکل ۴-۴ مدار Zero Deletion

شکل ۳-۴ مدار Zero Deletion Detector

**ب)** مطابق شکل ۴-۴ مدار مربوط به Zero Deletion را توصیف نمایید.

- اگر ورودی reset فعال شود، ماشین حالت مربوط به تشخیص دهنده به حالت اولیه برمی‌گردد و همه خروجی‌ها صفر می‌شوند.
- مدار در صورتی بیت صفر اضافی را حذف می‌کند که ورودی enable فعال باشد.
- خروجی is\_valid هر زمان که داده‌ی خروجی معتبر باشد برابر یک می‌شود.
- خروجی detected همان خروجی ماشین حالت تشخیص دهنده است.

**ج)** برای مدار بخش (ب) یک فایل محیط آزمون طراحی نمایید که:

اولاً محتوای ورودی din را از فایل ضمیمه شده با نام datain.txt در هر کلاک بخواند و خروجی dout را در فایل با نام dataout.txt ذخیره نماید.

ثانیاً در صورتی که خروجی detected برابر با یک شود با استفاده از دستور assert در محیط کنسول پیغامی با فرمت زیر نمایش دهد:

```
zero deletion detected at [now-time]
total number of deletions: [number of deletions]
```

منظور از [now-time] زمان جاری شبیه‌سازی است. دستور now در محیط شبیه‌سازی زمان جاری را برمی‌گرداند. منظور از [number of deletions] تعداد کل صفرهای تشخیص داده شده است. بنابراین تعداد صفرها باید شمرده شود.



مهلت تحویل تا ساعت ۲۳:۵۵ روز دوشنبه مورخ ۹۹/۰۳/۰۵ از طریق سامانه‌ی دروس

## پرسش ۴-۴ (نمره‌ی اضافی دارد و حل آن اختیاری است.)

مدار تشخیص دهنده‌ی دنباله‌ی ۱۰۰۱۰۰۱ و ۱۰۱۱۰۰۱۱ را به صورت همپوشان<sup>۵</sup> توصیف نمایید. این مدار دارای یک ورودی است که با هر کلاک، یک بیت را دریافت می‌کند و وقتی هر کدام از دنباله‌های بالا را دریافت کرد، خروجی خود را یک کند. همپوشان بودن به معنی این است که مثلاً اگر دنباله ۱۰۰۱۰۰۱۰۰۱ دریافت شد، خروجی در کلاک هفتم و کلاک سیزدهم یک می‌شود.

**(الف)** با قرار دادن attribute ها در کد VHDL، مدار را طوری در ابزار Vivado سنتز نمایید که ماشین حالت به صورت One-Hot سنتز شود. خروجی log سنتز و شماتیک پس از سنتز را ارائه دهید.

**(ب)** همین کار را برای کد حالت جانسون تکرار کنید.

**(ج)** قطعه کد ۲-۴ (xmodule.vhd) را سنتز نموده و لیست اتصالات پس از سنتز را نشان دهید. شماتیک پس از سنتز را ذخیره کرده و ارائه دهید.

**(د)** با استفاده از attribute ها مدار را طوری سنتز نمایید که سیگنال‌های wire\_a و wire\_b هنگام بهینه‌سازی در مراحل سنتز حذف نشوند. (از keep استفاده کنید).

لیست اتصالات پس از سنتز و پیاده‌سازی را ارائه دهید. شماتیک پس از سنتز و پیاده‌سازی را ضمیمه کنید.

**(ه)** با استفاده از attribute ها مدار را طوری سنتز نمایید که سیگنال‌های wire\_a و wire\_b هنگام بهینه‌سازی در مراحل سنتز حذف نشوند. (از dont\_touch استفاده کنید).

لیست اتصالات پس از سنتز و پیاده‌سازی را ارائه دهید. شماتیک پس از سنتز و پیاده‌سازی را ضمیمه کنید.

**(و)** تفاوت keep و dont\_touch را یافته و با ذکر مثال توضیح دهید.

**راهنمایی:** جهت مشاهده و ذخیره لیست اتصالات در ابزار ویوآدو از سربرگ File/از منوی Export گزینه‌ی Export Netlist را انتخاب کنید و یا در محیط TCL Terminal دستور export\_edif را وارد نمایید.

توضیحات و لیست attribute های ابزار Vivado در فایل Vivado Design Suite User Guide: Synthesis (UG901) موجود است.

<sup>5</sup> Overlapped