



دانشکده مهندسی کامپیوتر

طراحی کامپیوتری سیستم‌های دیجیتال

تمرین سری دوم

استاد

دکتر بیت الهی

طراحان

امیرمحمد درپوش

سارا ایرانپور

نازنین محمودی

بهار ۱۴۰۴

آداب نامه‌ی تمرین

- دقت کنید که هر سوال رو در قسمت خودش در کوئرا آپلود کنید
- برای هر سوال عملی نیاز به پیاده‌سازی Testbench هست که تمامی حالتها را پوشش دهد.
- برای تمامی سوالات (حتی سوالات پیاده‌سازی) نیاز هست که در یک فایل PDF جداگانه توضیحات لازم رو ارائه بدید و در صورتی که سوال عملی باشد، نتایج حاصل از Testbench را در آن قرار دهید
- **توجه داشته باشید که ارسال با تاخیر برای تکالیف وجود ندارد.**

Q1)

در این سوال شما باید یک تایمر دیجیتال ۸ بیتی طراحی کنید.

در این تایمر، ورودی‌ها شامل یک سیگنال ساعت clk با فرکانس ۱ MHz، یک سیگنال ریست reset، و یک سیگنال هشدار alarm بیتی است. خروجی‌ها شامل زمان گذرانده شده seconds که به صورت ۸ بیتی نمایش داده می‌شود و یک سیگنال هشدار alarming یک بیتی است که در صورتی که زمان گذرانده شده به مقدار alarm برسد فعال می‌شود. این تایمر باید با گذشت هر ثانیه مقدار seconds را افزایش دهد. همچنین در صورت رسیدن تایمر به مقدار داده شده در سیگنال alarm باید تایمر متوقف شده و بیت alarming شروع به چشمک زدن کند. (بازه تناوب هر چشمک را می‌توانید دلخواه تنظیم کنید.) در صورت ۱ بودن بیت reset باید مقدار seconds به ۰ بازگردانده شده و تایمر از ابتدا شروع به کار کند. این تایمر باید شامل پایه‌های زیر باشد.

ورودی

- (1 bit) Clk
- (1bit) Reset
- (8 bit) Alarm

خروجی

- (8 bit) Seconds
- (1 bit) Alarming

Q2)

درون زبان VHDL سیگنال‌ها و متغیرها هر دو برای ذخیره و مدیریت اطلاعات استفاده می‌شوند.

الف) تفاوت این دو ساختار را به همراه کاربرد هر کدام شرح دهید.

ب) مقدارهای ابتدایی و نهایی var_a var_b signal_a signal_b را در ۳ کلاک سایکل ابتدایی این کد به دست بیاورید. (باید روند تغییر این ۴ متغیر مشخص باشد.)

architecture Behavioral of SignalVsVariableExample is

```
signal signal_a : std_logic := '0';
```

```
signal signal_b : std_logic := '0';
```

```
begin
```

```
process(clk, reset)
```

```
variable var_a : std_logic := '0';
```

```
variable var_b : std_logic := '0';
```

```
begin
```

```
if rising_edge(clk) then
```

```
    signal_a <= var_a;
```

```
    signal_b <= var_b;
```

```
    var_a := not var_a;
```

```
    signal_b := not signal_b;
```

```
end if;
```

```
end process;
```

```
end Behavioral;
```

Q3)

طراحی دیکودر 2 به 4 فعال-بالا با اولویت و تأخیر قابل تنظیم یک مدار دیکودر ۲ به ۴ طراحی کنید که در آن ورودی‌ها شامل دو بیت A و B، یک سیگنال enable، و یک پارامتر generic با نام DELAY از نوع time هستند. در صورتی که enable='1' باشد، یکی از خروجی‌های Y0 تا Y3 باید مطابق مقدار AB یعنی A & B فعال شود. از دستور with-select-when به همراه تأخیر پارامتری (generic) برای خروجی‌ها استفاده کنید.

نکات:

مقدار DELAY باید از طریق generic قابل تغییر باشد.

از ساختار concurrent استفاده شود. (نه process)

در صورتی که enable='0' باشد، همه خروجی‌ها باید صفر باقی بمانند. (استفاده از شرط اضافی روی enable مجاز است).

Q4)

طراحی رجیستر 8×4 با نوع آرایه‌ی دوبعدی و مقداردهی اولیه یک رجیستر 8×4 طراحی کنید که از نوع داده‌ی آرایه دوبعدی std_logic_vector استفاده می‌کند. رجیستر باید دارای سیگنال‌های زیر باشد:

- Clk : کلاک
- wr_en : فعال‌سازی نوشتن
- wr_addr : آدرس سطر (۲ بیت)
- data_in : داده ورودی ۸ بیتی
- Mem : آرایه 8×4 خروجی برای مشاهده مقدار درونی حافظه

در هر لبه‌ی بالا رونده کلاک، اگر wr_en='1' باشد، مقدار data_in در آدرس مشخص شده با wr_addr در آرایه ذخیره شود.

نکات:

نوع آرایه باید توسط خودتان تعریف شود. (نه استفاده از std_logic_vector به تنهایی)
مقدار اولیه رجیسترها را صفر قرار دهید.

از ساختارهای TYPE و SIGNAL برای تعریف آرایه استفاده کنید.

Q5)

طراحی ماژول پردازشگر بیت با استفاده از عملگرهای شیفت، مقایسه‌ای و خصوصیات سیگنال‌ها (Attributes) یک ماژول طراحی کنید که دارای سیگنال‌های زیر است:

- input_vec : ورودی ۸ بیتی std_logic_vector

- output_shifted : خروجی که برابر است با input_vec بعد از دو بار شیفت چپ منطقی (sl 2)
- output_rotated : چرخش راست دایره‌ای (ror 3) روی input_vec
- is_zero : خروجی Boolean که اگر تمام بیت‌های input_vec صفر باشد، برابر true شود (با استفاده از عملگر مقایسه‌ای)
- last_event_time : سیگنال از نوع time که زمان آخرین تغییر input_vec را با استفاده از attribute 'LAST_EVENT در خود نگه دارد.

نکات:

استفاده از attribute 'LAST_EVENT اجباری است.

از عملگرهای شیفت و مقایسه به طور هم‌زمان در این طراحی استفاده کنید.

پیاده‌سازی به صورت ترکیبی (بدون کلاک) نیز پذیرفته است، اما استفاده از کلاک برای سنجش زمان نیز قابل قبول است.

Q6)

یک تایمر دیجیتال طراحی کنید که زمان را به صورت دقیقه و ثانیه نمایش دهد و دارای ورودی‌های clk کلاک ورودی، rst (ریست)، و run (اجرای تایمر) باشد. کلاک ورودی clk با فرکانس frequency مشخص شده و از آن برای تولید یک پالس 1 Hz استفاده می‌شود که شمارش زمان را کنترل می‌کند. شمارنده زمان از 00:00 شروع شده و تا 09:59 ادامه می‌یابد و در صورت فعال بودن run مقدار ثانیه‌ها افزایش می‌یابد. در صورت فعال بودن rst مقدار تایمر در هر لحظه صفر شده. مقدار minutes و seconds باید برای نمایش در سون سگمنت ارسال شوند.

ورودی‌ها:

- Fclk : فرکانس کلاک، که به عنوان مقدار GENERIC تعریف شده.
- Clk : سیگنال کلاک ورودی.
- Rst : سیگنال ریست که مقدار تایمر را صفر می‌کند.
- Run : سیگنال کنترلی که تعیین می‌کند تایمر باید شمارش

خروجی‌ها:

- digit1, digit2, digit3 : سه نمایشگر سون سگمنت برای نمایش مقدار دقیقه و ثانیه.

Q7)

در طراحی یک سیستم دیجیتال برای پردازش سیگنال، نیاز است که سیگنال ورودی با استفاده از یک مجموعه از ضرایب مشخص، پردازش شده و نویزهای ناخواسته حذف شوند. در این سیستم، برای ذخیره‌سازی مقادیر ورودی و ضرایب فیلتر، از شیفت رجیسترها استفاده می‌شود که به صورت پیوسته مقادیر جدید را دریافت کرده و مقادیر قدیمی را جابجا می‌کنند. سپس هر کدام از این ورودی‌ها با ضرایب متناظر خودشان چند برابر شده و در نهایت نتایج در یک جمع‌کننده جمع می‌شوند تا مقدار نهایی به دست آید. این طراحی باید به گونه‌ای باشد که برای تعداد دلخواه n و دقت m بیت به درستی عمل کند و همچنین مکانیزم مناسبی برای مدیریت overflow در نظر گرفته شود. سیستم باید قابلیت ریست شدن (Reset) داشته باشد تا در شروع مجدد، مقادیر داخلی آن به حالت اولیه (مانند صفر) برگردند.

ورودی‌ها :

- یک GENERIC تعریف شده تا تعداد ضربه‌های فیلتر (n) و تعداد بیت‌های هر مقدار ورودی (m) را مشخص کنیم.
- ورودی‌های m بیتی (x coefficient)
- load: برای بارگذاری ضرایب فیلتر استفاده می‌شود در صورتی که برابر با 1 باشد
- run: برای گرفتن ورودی، اجرای فیلتر و تولید خروجی نهایی در صورتی که برابر با 1 باشد

خروجی‌ها:

- y : خروجی نهایی
- overflow : برای بررسی وجود یا عدم وجود سرریز