



تمرین سری دوم

- پرسش‌های خود را در سامانه CW و تالار مربوط به تمرین مطرح نمایید.
- پاسخ سوالات را تایپ نمایید.
- پاسخ تمرین را به صورت یک فایل زیپ با فرمت HW2_401234567.zip آپلود کنید. فایل زیپ باید به صورتی باشد که پس از باز کردن آن بدون هیچ پوشه‌ای فایل‌های زیر با ساختار زیر قرار گرفته باشند:

```
1 .
2 |-- HW2T_401234567.pdf
3 |-- practical
4 |   |-- HW2P_401234567_401234568.pdf
5 |   |--Q1
6 |       |-- schematic.circ
7 |       |-- verilog
8 |           |-- circuit
9 |               |-- main.v
10 |                   |-- ...
11 |       |-- gates
12 |           |-- ...
13 |       |-- toplevel
14 |           |-- ...
15 |--Q2
16 |   |-- schematic.circ
17 |   |-- verilog
18 |       |-- circuit
19 |           |-- main.v
20 |               |-- ...
21 |       |-- gates
22 |           |-- ...
23 |       |-- toplevel
24 |           |-- ...
```

- در صورت عدم تطابق فایل آپلود شده با فرمت بالا، تمرین شما تصحیح نخواهد شد.
- پاسخ سوالات تئوری و گزارش تمرین‌های عملی باید به فرمت pdf باشد.
- هر دانشجو می‌تواند حداکثر سه تمرین را با دو روز تأخیر بدون کاهش نمره ارسال نماید.
- تمرینات عملی به صورت گروه‌های دو نفر تحویل داده شود.
- هر دو عضو گروه موظف هستند تمرینات خود را بارگذاری کنند.
- عواقب عدم تطابق بین پاسخ دو عضو گروه برعهده خودشان است.
- تحویل تمرین به صورت انگلیسی مجاز نیست. در صورت تحویل تمرین به صورت انگلیسی (حتی بخشی از تمرین) نمره تمرین موردنظر صفر در نظر گرفته می‌شود.
- در صورت مشاهده تقلب برای بار اول نمره هر دو طرف صفر می‌شود. در صورت تکرار نمره کل تمرینات صفر خواهد شد.

- استفاده از ابزارهایی مانند ChatGPT به منظور ابزار کمک آموزشی مجاز است به شرط آن که به خروجی آن اکتفا نشود.
- توجه شود که پروژه نهایی درس در گروه‌های چهار نفر تحویل گرفته می‌شود.
- سوالات با عنوان اختیاری نمره‌ای ندارند اما جواب دادن به آن‌ها کمک به‌سزایی در یادگیری درس می‌کند.

تمارین تئوری

۱. فرض کنید یک بانک ثبات شامل ۳۲ ثبات ۳۲ بیتی داریم. می‌خواهیم یک RTL پیاده‌سازی کنیم به طوری که بتوان میان هر دو ثبات عملیات MOV را انجام داد. (فرض کنید که تمام قطعات ما تک بیتی هستند)

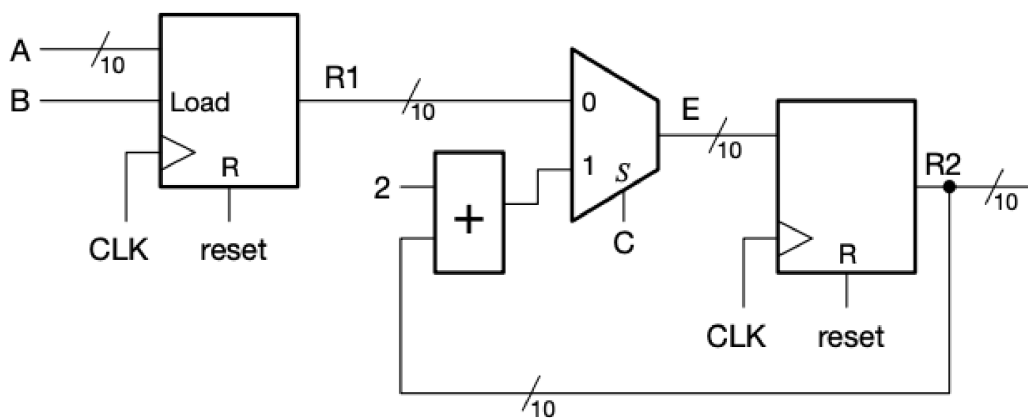
(آ) در هر یک از حالات زیر چه قطعاتی لازم است و در هر روش به چند بیت سیگنال کنترلی نیاز داریم؟

- اتصال نقطه به نقطه^۱
- ارتباط common bus

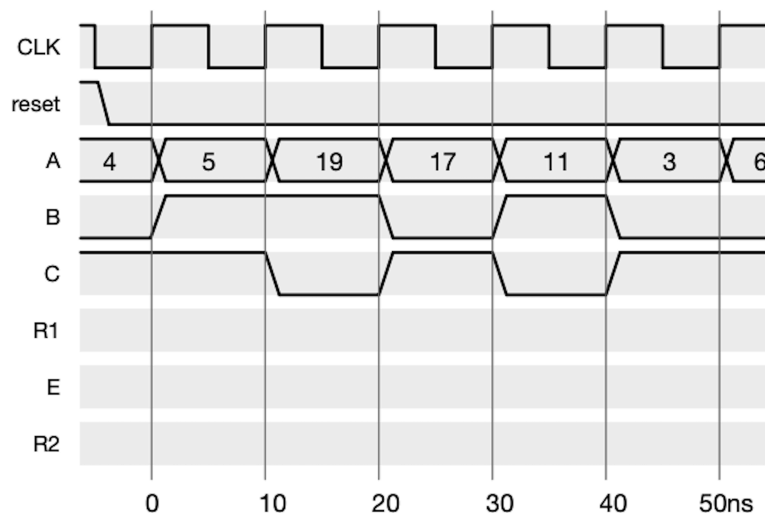
(ب) حال فرض کنید که تعداد ثبات‌های خود را به ۶۴ افزایش داده‌ایم. چه تغییراتی در بخش الف ایجاد می‌شود؟

(ج) اگر تعداد بیت‌های هر ثبات را به ۶۴ افزایش دهیم چه تغییراتی در بخش الف ایجاد می‌شود؟

۲. (آ) با توجه به شکل زیر، مقدار R1، E و R2 را به صورت RTL بیان کنید.



(ب) با توجه به شکل موج^۲ زیر، مقدار R1، E و R2 را در هر واحد زمانی مشخص کنید. (فرض کنید که reset، active-high است و مقدار ثبات‌ها را برابر با صفر می‌گذارد.)

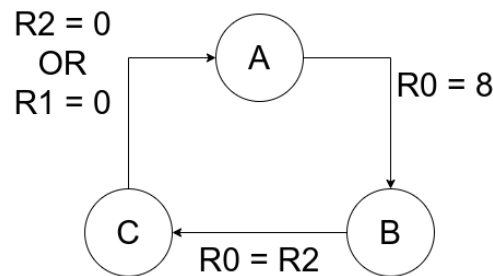


۳. ماشین حالت^۳ زیر را در نظر بگیرید.

^۱Point-to-Point

^۲Waveform

^۳State Machine



این ماشین مربوط به سیستمی دارای ۳ ثبات ۸ بیتی با نام‌های $R0$, $R1$, $R2$ است. بسته به حالتی^۴ که در حال حاضر در آن هستیم، اتفاقات خاصی در هر لبه ساعت^۵ می‌افتد. اتفاقات مربوطه به صورت زیر هستند:

- حالت A: در این حالت در هر لبه ساعت ثبات $R0$ با عدد ۱ جمع می‌شود.
- حالت B: در این حالت در هر لبه ساعت مقدار $R2$ برابر حاصل عملیات xor بین $R0$ و $R2$ قرار داده می‌شود. همچنین مقدار ثبات $R0$ با یک جمع می‌شود (ولی در عملیات xor مذکور مقدار قدیمی آن استفاده می‌شود).
- حالت C: در این حالت ابتدا مقدار $R1$ یکی کم می‌شود و سپس مقدار $R2$ برابر حاصل جمع مقدار $R2$ بعلاوه مقدار (جدید) $R1$ قرار داده می‌شود.

دقت کنید در تمام عملیات‌های جمع مذکور، بیت نقلی در صورت ایجاد شدن دور ریخته می‌شود. همچنین دقت کنید در لبه ساعتی که در حال خروج از یک حالت هستیم هم عملیات‌های مربوط به آن حالت انجام خواهند شد.

(آ) با استفاده از زبان RTL این سیستم را شبیه‌سازی کنید.

(ب) شمای مدار منطقی مناسب برای پیاده‌سازی این سیستم را بکشید. اجازه استفاده از بلوک‌های جمع، مقایسه و دیگر بلوک‌های رایج را دارید.

۴. مجموعه دستورات زیر را در نظر بگیرید. در هر سیکل زمانی، بر اساس مقدار فعلی ثبات‌ها، یکی از دستورات زیر اجرا می‌شود:

- $(R1 \neq 0)(R0 = 0) : R0 \leftarrow 1, \quad R2 \leftarrow R2 + R1$
- $(R1 \neq 0)(R0 = 1) : R0 \leftarrow 2, \quad R1 \leftarrow R1 - 1$
- $(R1 \neq 0)(R0 = 2) : R0 \leftarrow 3, \quad R2 \leftarrow R2 \times 2$
- $(R1 \neq 0)(R0 = 3) : R0 \leftarrow 4, \quad R2 \leftarrow R2 \% R1$
- $(R1 \neq 0)(R0 = 4) : R0 \leftarrow 0, \quad R1 \leftarrow R1 - (R2 \& 1)$
- $(R1 = 0) : \text{Terminate}$

فرض کنید ابتدا $R0 = 0$, $R1 = 5$ و $R2 = 1$ هستند.

۱. مقادیر نهایی ثبات‌های $R0$, $R1$ و $R2$ را پس از اجرای این دستورات گزارش کنید.
۲. چند سیکل زمانی (چند مرحله) از اجرای دستورات گذشته است؟
۳. در هر سیکل، کدام دستور (متناسب با مقدار $R0$) اجرا شده است؟
۴. برای رسیدن به این مقادیر نهایی، هر کدام از حالت‌های $R0$ چند بار تکرار شده‌اند؟

⁴State

⁵Clock Edge

۵. یک طراحی به زبان RTL ارائه دهید که مقدار ب.م.م (بزرگ‌ترین مقسوم‌علیه مشترک) دو مقدار r_1 و r_2 را محاسبه کرده و نتیجه را در r_3 ذخیره کند. در واقع طراحی ارائه شده توسط شما باید به صورت ترتیبی، در چند چرخه ساعت مقدار ب.م.م را محاسبه کرده و در نهایت در یک وضعیت ثابت بماند که در آن مقدار داخل r_3 مقدار ب.م.م است. مقدار نهایی ثبات‌های r_1 و r_2 فاقد اهمیت است. (در حل این سوال مجاز به استفاده از تقسیم نمی‌باشید!)

۶. در مجموعه دستورات RTL زیر، R_0, R_1, R_2, R_3 ثبات‌های n بیتی هستند و فلیپ‌فلاپ‌های E, S, F_0, F_1, F_2, F_3 در سیستم وجود دارند. مجموعه دستورات نشان داده شده چه عملی را انجام می‌دهند؟ عملکرد آن را به طور کلی توضیح داده و با یک مثال ساده، مرحله به مرحله نشان دهید. همچنین مدار مناسبی برای توصیف آن رسم کنید.

$$S : R_3 \leftarrow 0, S \leftarrow 0, E \leftarrow 0, F_0 \leftarrow 1, R_1 \leftarrow R_0$$

$$F_0 : R_1 \leftarrow \overline{R_1}, F_0 \leftarrow 0, F_1 \leftarrow 1$$

$$F_1 : R_1 \leftarrow R_1 + 1, F_1 \leftarrow 0, F_2 \leftarrow 1$$

$$F_2 : R_2 \leftarrow R_2 + R_1, F_2 \leftarrow 0, F_3 \leftarrow 1$$

$$F_3 : F_3 \leftarrow 0, (R_2 < 0) : [R_2 \leftarrow R_2 + R_0, E \leftarrow 1], (R_2 \geq 0) : [R_3 \leftarrow R_3 + 1, F_2 \leftarrow 1]$$

۷. با توجه به دستورات زیر:

```

1 p: X <= A, B <= A
2 q: M <= X
3 r: Y <= M, R <= M
4 s: A <= R XOR Y
5 t: B <= A + Y

```

با استفاده از گذرگاه مشترک و *tri-state buffer*، مداری طراحی کنید که دستورات را در یک کلاک انجام شود. همچنین A و B و X و M و R و Y ، فلیپ‌فلاپ‌های تک‌بیتی هستند. (p و q و r و s و t سیگنال‌های کنترلی هستند).

۸. سوال اختیاری - با توجه به ساختار RTL و Micro-Ops به بخش‌های زیر پاسخ دهید.

(آ) فرض کنید که مجموعه دستورات ما ۲ آدرس هستند و در مجموع ۴ ثبات R_1, R_2, R_3, R_4 داریم، حال مقدار نهایی O را بر اساس دستورات RTL بنویسید. فرض کنید که هر کدام از حروف در اصل آدرسی از حافظه هستند و باید در ثبات بارگذاری^۶ شود.

$$O = \frac{(a + b + c) \times (d \times e - f)}{g + h \times i}$$

(ب) حال فرض کنید که دستورات ما تک آدرس هستند و باید از انباشتگر^۷ استفاده کنید و با این فرض جدید کد مربوط به بخش (آ) را پیاده سازی کنید.

(ج) فرض کنید همان ثبات‌های بخش (آ) را داریم و هر کدام ۸ بیتی هستند. در ثبات اول مقدار $(F2)_H$ و ثبات دوم مقدار $(FF)_H$ و ثبات سوم مقدار $(B9)_H$ و در ثبات آخر مقدار $(EA)_H$ به ترتیب از چپ به راست بارگذاری می‌شوند. محتوای هر ثبات را بعد از اجرای کامل کد زیر با راه حل کامل مشخص کنید.

$$(i) R_1 \leftarrow R_1 + R_2, R_3 \leftarrow R_3 + \text{shl}(R_4)$$

$$(ii) R_3 \leftarrow R_3 \wedge R_4, R_2 \leftarrow R_2 + 1$$

$$(iii) R_1 \leftarrow R_1 - R_3$$

$$(iv) R_1 \leftarrow \text{sar}(R_3) \oplus \text{cir}(R_4)$$

تمارین عملی

برای آزمون تمرین به آخرین نسخه سامانه داوری نیازمندید. بنابراین کافیسیت مخزن سامانه داوری را git pull کنید.

۱. عملیات‌های انجام‌شده روی ثابت‌های R1 و R2 را در RTL زیر در نظر بگیرید.

$R2 \leftarrow -R1$
 $R2 \leftarrow R1 \& R2$
 $R2 \leftarrow -R2$
 $R2 \leftarrow R1 + R2$

(آ) عملکرد این مدار را توضیح دهید.

(ب) در نرم‌افزار logisim-evolution مداری مطابق RTL توصیف‌شده طراحی کنید.

Inputs:

- in1 (32 bits)
- load
- clk

Outputs:

- out1 (32 bits)

نحوه طراحی این مدار باید به صورتی باشد که در ابتدا سیگنال Load فعال شود و مقدار ورودی در R1 قرار بگیرد. سپس پس از چهار کلاک در R2 حاصل قرار بگیرد و در خروجی نمایش داده شود. نحوه داوری این سوال به صورت زیر خواهد بود:

```
1 ./synth_valid.sh schematic.circ HW2/tb1.v
```

۲. Flowchart موجود در شکل ۱ را در نظر بگیرید. مدار تقسیم‌کننده‌ای با توجه به عملکرد توصیف‌شده طراحی کنید.

Inputs:

- divisor (32 bits)
- dividend (32 bits)
- start
- clk

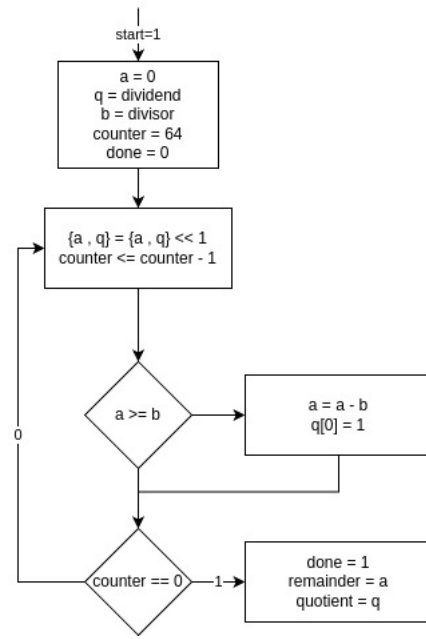
Outputs:

- quotient (32 bits)
- remainder (32 bits)
- done

طراحی شما باید به گونه‌ای باشد که با فعال شدن سیگنال start فرآیند تقسیم شروع شود و پس از پایان آن، سیگنال done فعال شود و مقادیر خارج‌قسمت و باقی‌مانده در خروجی قرار بگیرند.

نحوه داوری این سوال به صورت زیر خواهد بود:

```
1 ./synth_valid.sh schematic.circ HW2/tb2.v
```



شکل ۱: Flowchart تقسیم‌کننده

همچنین برای اطمینان از توصیف می‌توانید به پیاده سازی وریلاگ قرار گرفته در مخزن توجه کنید.