



مدارهای منطقی



دانشکده مهندسی کامپیوتر

اساتید: دکتر مهدی صدیقی، دکتر مرتضی صاحب‌الزمانی
تدریس‌یاران: رضا آدینه‌پور، مرتضی عادل‌خانی

دانشگاه صنعتی امیرکبیر

مهلت ارسال: ۲۷ آذر

طراحی و تحلیل مدارهای ترتیبی و FSM

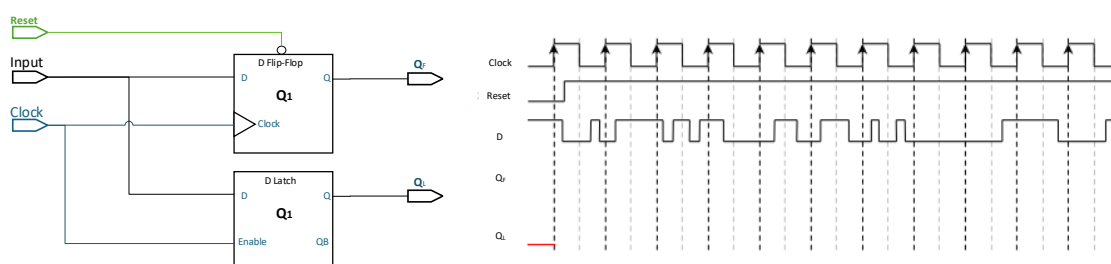
تمرین ششم

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص شده است. برای انجام تمرین زمان کافی اختصاص داده شده است.
- انجام آن را به هیچ‌وجه به روزهای پایانی موکول نکنید.
- سوالات خود را از طریق ایمیل‌های adinepour@aut.ac.ir و madelkhani@aut.ac.ir و یا در کلاس حل تمرین از تدریس‌یاران بپرسید.
- صرفاً تمرین آپلود شده در سامانه courses تصحیح می‌شوند.
- حتماً در نام گذاری فایل‌های آپلودی خود از قالب {HWx}_{STD_Number}_{Name} تبعیت کنید.
- پاسخ‌های ارسالی منحصراً باید حاصل تلاش‌های فردی شما باشد. در صورت استفاده از منابع خارجی یا هم‌فکری، حتماً این موارد را ذکر کنید.
- در صورت مشاهده هرگونه تقلب، نمره ۳ سری تمرین برای تمام افراد شرکت‌کننده، صفر لحاظ خواهد شد.

سوالات اختیاری

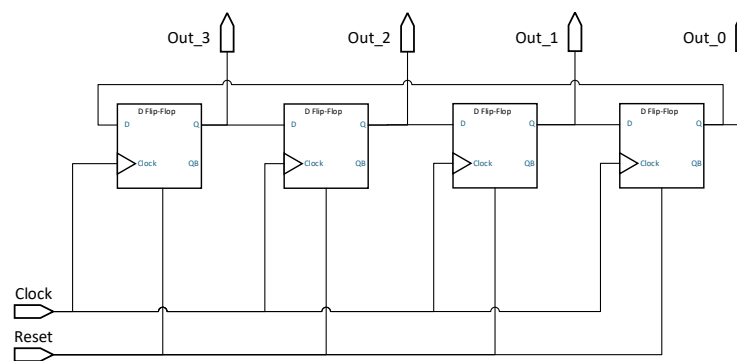
۱. با استفاده از یک فلیپ‌فلاپ D (بدون ورودی‌های enable یا reset) و همچنین گیت‌های استاندارد یک فلیپ‌فلاپ JK بسازید.

۲. نمودار زمانی زیر را با توجه به مدار داده شده تکمیل نمایید. سپس نمودار حالت آن را رسم کنید.



سوالات اصلی

۱. شمارنده‌های حلقوی، که به آن‌ها **Ring Counters** نیز گفته می‌شود، یک نوع شمارنده در مدارهای دیجیتال است که فلیپ‌فلاپ‌های مدار آن به‌صورت شیفت رجیستر حلقوی به هم وصل هستند. با این ساختار در هر لحظه فقط یکی از فلیپ‌فلاپ‌ها یک و بقیه صفر بوده و با هر کلاک این ۱ در شیفت رجیستر به صورت حلقوی حرکت می‌کند (به صورت شکل زیر). این نوع از شمارنده‌ها معمولاً در کاربردهایی مانند تولید سیگنال‌های زمانی یا کدگذاری توالی‌ها استفاده می‌شوند.



حال فرض کنید یک شمارنده حلقه‌ای که از ۰ تا ۵۷ می‌شمارد. این شمارنده دارای ۵۸ فلیپ‌فلاپ D خواهد بود، اما یک شمارنده باینری که از ۰ تا ۵۷ می‌شمارد دارای ۶ فلیپ‌فلاپ D خواهد بود. (ج) در یک فلیپ‌فلاپ نوع D، یک بازه کوتاه زمانی پس از لبه بالارونده کلاک وجود دارد که ورودی D باید ثابت باقی بماند، در غیر این صورت فلیپ‌فلاپ ممکن است به درستی عمل نکند. این بازه کوتاه زمانی Hold time نامیده می‌شود.

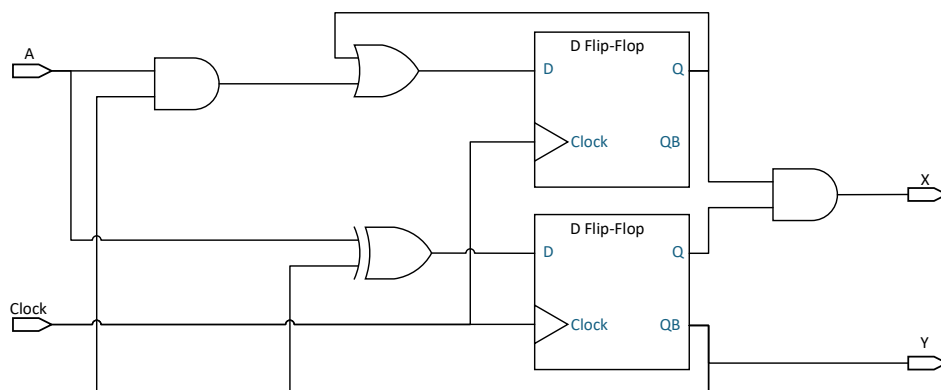
۲. یک فلیپ‌فلاپ جدید از نوع Set-Reset-Toggle (SRT) با نماد و جدول عملکرد زیر نشان داده شده است:

S	R	T	Function
X	X	1	Toggle (i.e., complement stored value)
1	0	0	Set
0	1	0	Reset
0	0	0	Hold
1	1	0	Hold

کدام یک از موارد زیر معادله ویژگی^۱ این فلیپ‌فلاپ است؟

- $Q^+ = TQ' + T'[Q(S \oplus R) + SR]$
- $Q^+ = (T \oplus Q) + S + R'Q$
- $Q^+ = T'Q + T[Q'(S + R') + SR]$
- $Q^+ = TQ' + T'[Q(S + R') + SR]$
- $Q^+ = T + [Q(S + R') + SR]$

۳. با توجه به مدار طراحی شده زیر نمودار حالت آن را رسم کنید. در این طراحی فرض کنید حالت اولیه زمانی است که خروجی هر دو فلیپ‌فلاپ برابر با "۰" است. همچنین فقط حالت‌هایی را که از حالت اولیه قابل دستیابی هستند، در طراحی خود بیاورید و نحوه طراحی خود را به صورت کامل شرح دهید.



Top is Dff1, bottom is Dff0

$$D_1 = Q_1 + A^*!Q_0$$

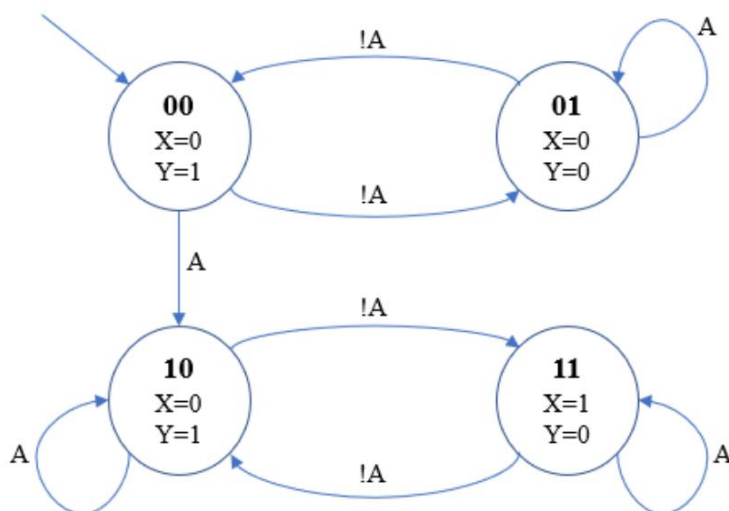
$$D_0 = A \oplus !Q_0$$

$$X = Q_1^* Q_0$$

$$Y = !Q_0$$

Next-State Table

Current State	Inputs/Next State		Outputs	
Q ₁ Q ₀	A = 0	A = 1	X	Y
00	01	10	0	1
01	00	01	0	0
10	11	10	0	1
11	10	11	1	0

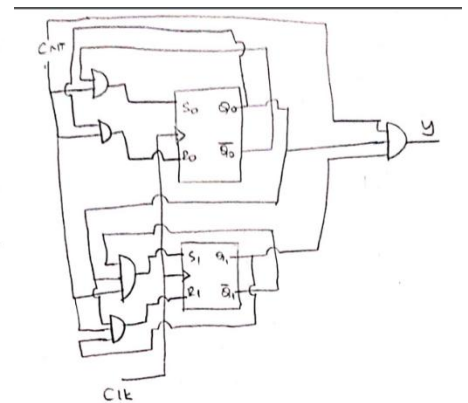
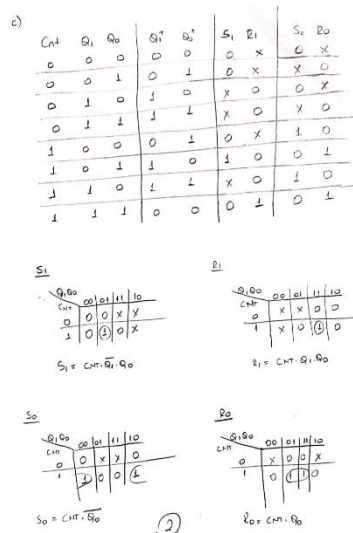
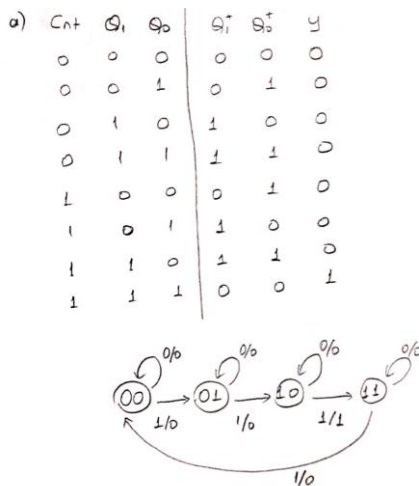
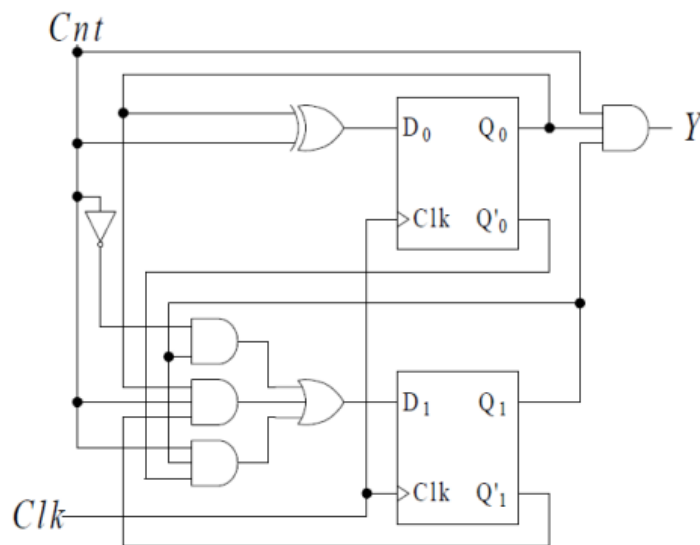


۴. مدار ترکیبی شکل زیر را در نظر بگیرید که دارای یک ورودی به نام Cnt و یک خروجی به نام Y است.

الف) نمودار حالت و جدول حالت مدار را به دست آورید.

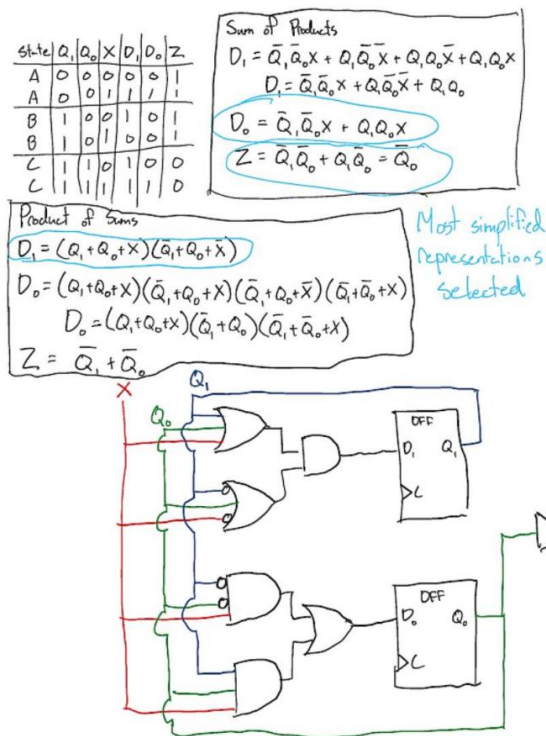
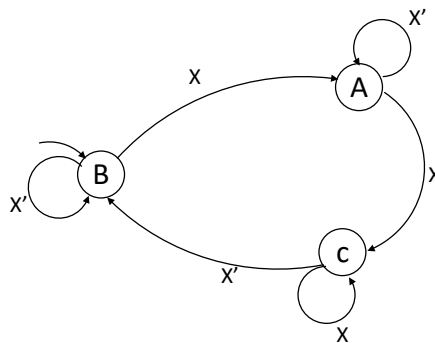
(ب) تعیین کنید که این مدار یک ماشین میلی (Mealy) یا مور (Moore) است. دلیل انتخاب خود را ذکر کنید.

ج) نمودار یا جدولی که در قسمت الف به دست آمده را با استفاده از فلیپ‌فلاپ‌های S-R حساس به لبه مثبت پیاده‌سازی کنید.

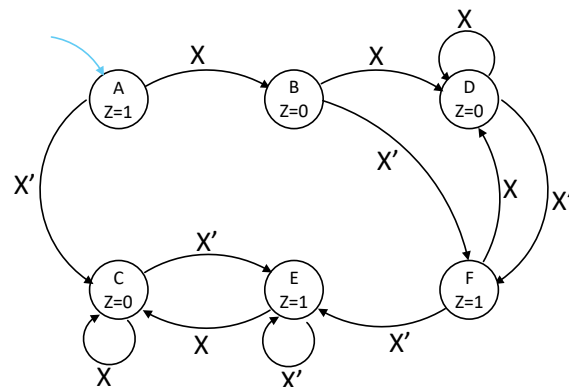


b) mealy-type

۵. با استفاده از تنها گیت‌های AND، OR و NOT (همچنین استفاده آزاد از حباب‌ها) و فلیپ‌فلاپ‌های نوع D، مدار مربوط به نمودار حالت مربوط به مداری با یک خروجی به نام Z و یک ورودی به نام X که به صورت شکل زیر نشان داده شده است را رسم کنید. می‌توانید فرض کنید حالت‌هایی که نشان داده نشده‌اند هرگز قابل دسترس نیستند.



۶. تعداد حالات را در نمودار انتقال حالت تا حد ممکن با استفاده از روش implication chart کاهش دهید. مراحل روش حل خود را نشان دهید و نمودار حالت کاهش یافته را رسم کنید.



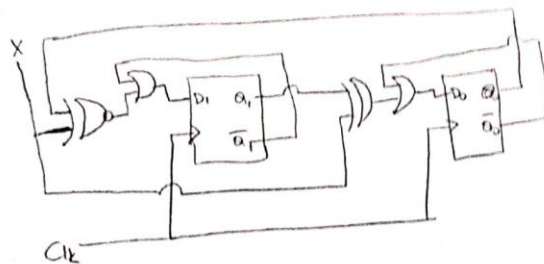
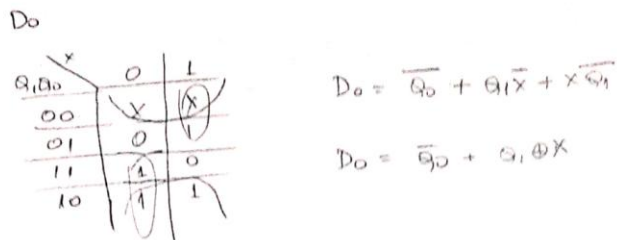
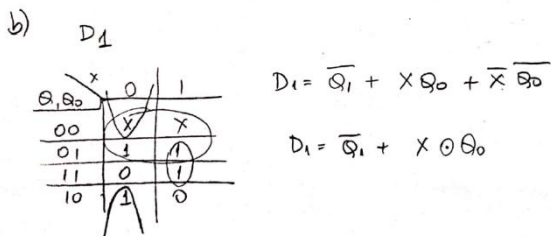
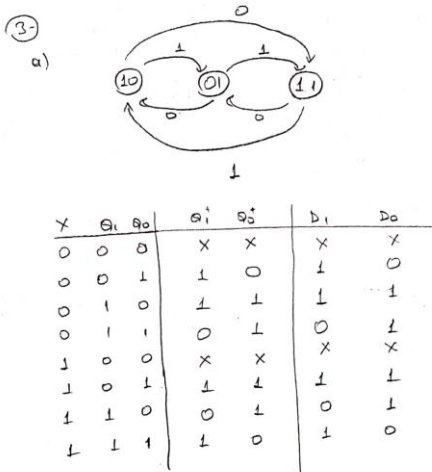
Final result has 3 states: A, BCD, EF.

A	EF				
B	EF	CD			
C	EF	CD	EC FD		
D	EF	CD	FD EC		
E	EC ED	CB	X	X	X
	F	A	B	C	D

۷. یک شمارنده با ورودی کنترلی طراحی کنید به طوری که در زمان یک بودن ورودی کنترلی شمارنده به ترتیب اعداد باینری ۱۰، ۰۱ و ۱۱ را به ترتیب پیمایش کند و این سیکل را تکرار کند. اما در صورتی که ورودی کنترلی صفر باشد شمارنده همان رشته اعداد را به ترتیب از آخر به اول پیمایش کند و این سیکل را تکرار کند. برای این طراحی موارد زیر را در نظر بگیرید:

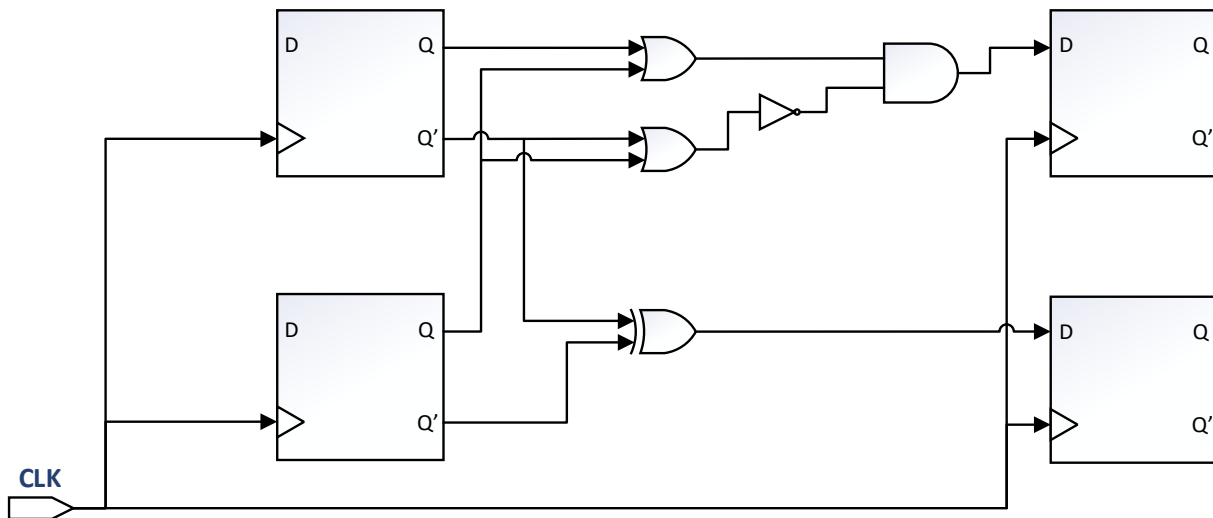
الف) نمودار حالت و جدول حالت را رسم کنید.

ب) شمارنده را با استفاده از فلیپ‌فلاپ‌های نوع D و گیت‌های منطقی پیاده‌سازی کنید.



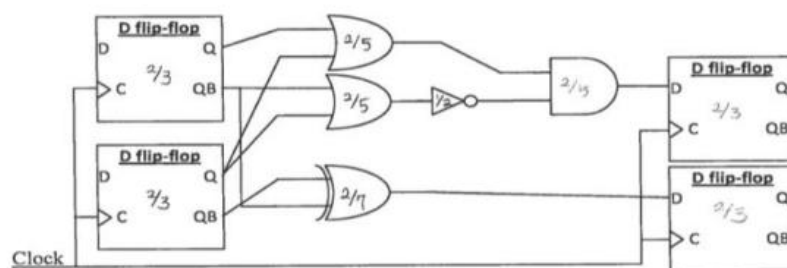
۸. فرض کنید مقادیر زیر برای فرآیندی که استفاده می‌کنید داده شده‌اند (توجه داشته باشید که زمان‌های نگهداری و راه‌اندازی مشخص نشده‌اند). فرض کنید قصد دارید این مدار را با فرکانس ۵۰ مگاهرتز کلاک کنید.
- (الف) با فرض اینکه شما یک مدار مطمئن می‌خواهید، بیشترین مقداری که زمان حفظ برای فلیپ‌فلاپ می‌تواند داشته باشد، چقدر است؟ کار خود را نشان دهید.
- (ب) با فرض اینکه شما یک مدار مطمئن می‌خواهید، بیشترین مقداری که زمان راه‌اندازی^۱ برای فلیپ‌فلاپ می‌تواند داشته باشد، چقدر است؟ کار خود را نشان دهید.
- (ج) همان‌طور که می‌دانید، هیچ دو رویدادی دقیقاً هم‌زمان اتفاق نمی‌افتند. به همین ترتیب، هیچ دو فلیپ‌فلاپی دقیقاً هم‌زمان لبه کلاک را نمی‌بینند. این پدیده به نام «عدم همزمانی کلاک» شناخته می‌شود. فرض کنیم که کلاک‌هایی که به هر دستگاه می‌روند می‌توانند حداکثر تا ۱ نانوثانیه از یکدیگر اختلاف داشته باشند. به عبارت دیگر، تنها چیزی که می‌دانیم این است که هر فلیپ‌فلاپ لبه صعودی کلاک را در حداکثر ۱ نانوثانیه از زمانی که هر فلیپ‌فلاپ دیگری لبه صعودی را می‌بیند، مشاهده خواهد کرد. فرض کنید هیچ اطلاعی از ترتیب مشاهده لبه صعودی کلاک توسط فلیپ‌فلاپ‌ها نداریم.
- (د) بخش «الف» را تحت این فرضیات مجدداً انجام دهید. توضیح مختصری از کار خود ارائه دهید.

ر) بخش «ب» را تحت این فرضیات مجدداً انجام دهید. توضیح مختصری از کار خود ارائه دهید.



Flipflop	Min	Max
Clock to Q	2ns	3ns
Set-up time		
Hold time		

Operator	Min	Max
OR / AND	2ns	5ns
NOT	1ns	2ns
XOR	2ns	7ns



4

- a. Assuming you want a reliable circuit, what is the highest value the flip-flop could have for a hold time? Show your work. [4 points]

$$\text{Clk to } Q + \text{CL Delay} \geq \text{hold time}$$

$$2 + 2 \geq \text{hold time}$$

$$4 \geq \text{hold time}$$

$$4 \text{ ns}$$

- b. Assuming you want a reliable circuit, what is the highest value the flip-flop could have for a setup time? Show your work. [4 points]

$$\text{Clk to } Q + \text{Clk Delay} + \text{Setup time} = 20$$

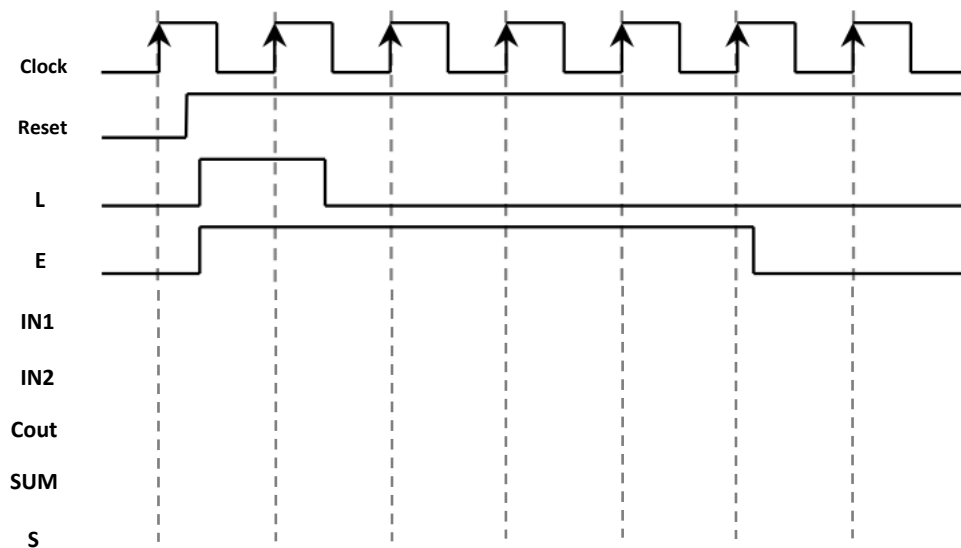
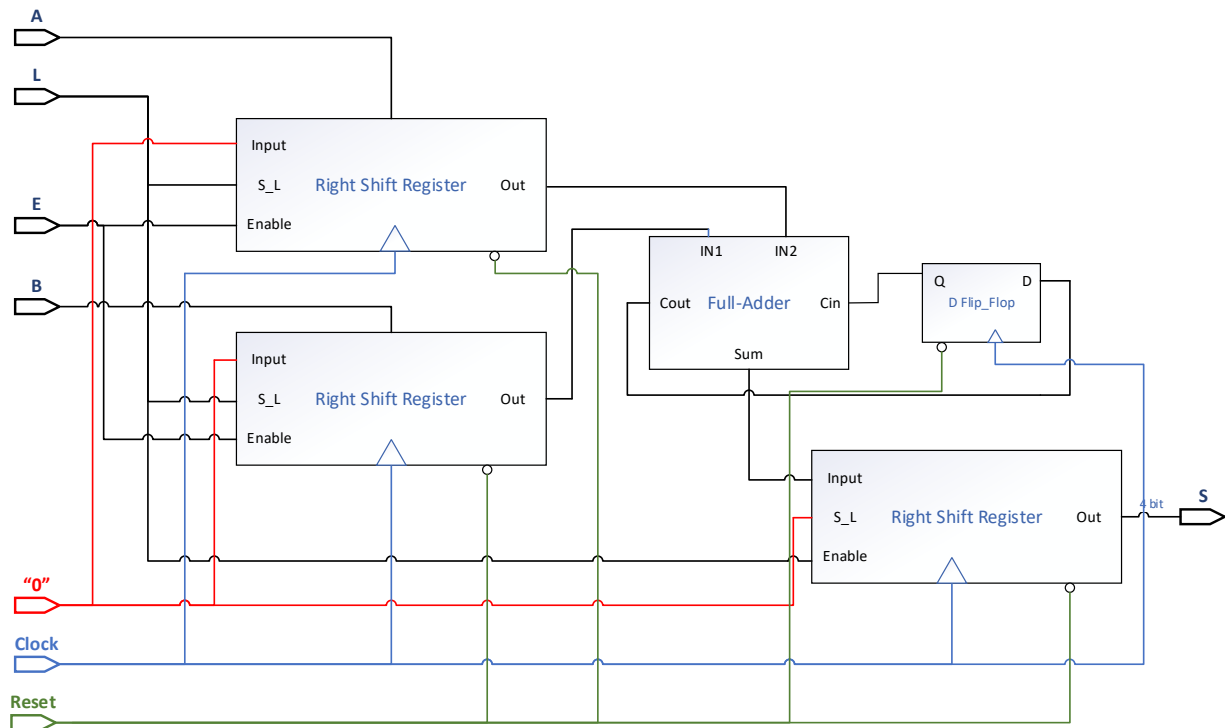
$$3 + 12 + \text{setup time} = 20$$

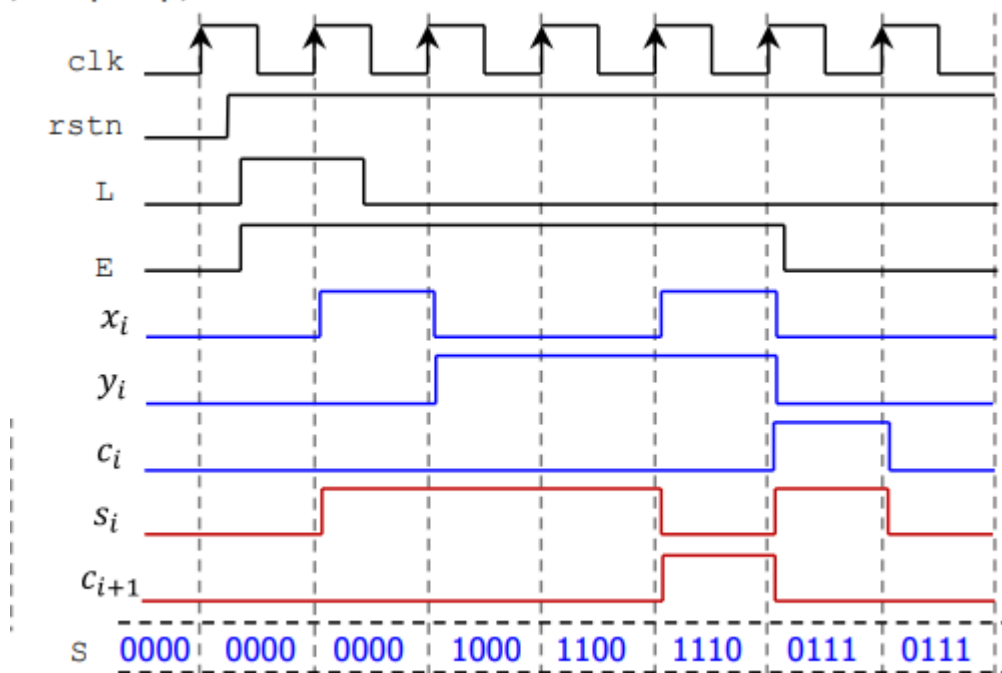
$$15 + \text{setup time} = 20$$

$$5 \text{ ns}$$

(Problem continued on the next page)

۹. شکل زیر نشان دهنده یک جمع کننده سریال بیتی برای ۲ ورودی A و B به صورت زیر است که از سه شیفت رجیستر ۴ بیتی با دسترسی موازی، یک فلیپ فلاپ و یک تمام جمع کننده تشکیل شده است. با در نظر گرفتن $A=1001$ و $B=1110$ نمودار زمانی^۱ این مدار را تکمیل کنید.





سوالات امتیازی

۱. شمارنده ۸ بیتی اشباع شونده (Saturating 8-bit Adder) یک نوع شمارنده است که در آن پس از رسیدن به یک مقدار خاص (بیشترین مقدار ممکن برای آن بیت‌ها) دیگر شمارش نمی‌کند و مقدار آن باقی می‌ماند. به عبارت دیگر، در شمارنده‌های معمولی، وقتی شمارنده به حداکثر مقدار خود برسد و یک واحد به آن اضافه شود، باید از ابتدا شروع کند (یعنی به ۰ برگردد). اما در یک شمارنده اشباع شونده، وقتی شمارنده به بیشترین مقدار ممکن (مثلاً ۲۵۵ در یک شمارنده ۸ بیتی) برسد، دیگر افزایش نمی‌یابد و همان مقدار را حفظ می‌کند.

برای شمارنده ۸ بیتی اشباع شونده، پس از رسیدن به مقدار ۲۵۵ (بیشترین مقدار ممکن برای ۸ بیت)، اگر به آن یک واحد دیگر اضافه شود، همچنان مقدار ۲۵۵ را نگه می‌دارد.

در این قسمت شما باید یک ماژول Verilog بنویسید که یک شمارنده رو به بالا (از صفر شروع می‌کند و هر بار یک واحد به آن اضافه می‌شود) ۸ بیتی اشباع شونده با ورودی‌های enable و reset پیاده‌سازی کند که ورودی و خروجی‌های آن به صورت زیر است:

ورودی‌ها: کلاک (clock)، ریست (reset) و فعالساز (enable)

خروجی: مقدار شمارنده، (Q[7:0])

توجه داشته باشید که در صورتی که در لبه بالا رونده کلاک ورودی ریست فعال شد شمارنده باید مقدار خود را به ۰ تنظیم کند و اگر ریست برابر با ۰ و فعالساز برابر با ۱ بود شمارنده باید مقدار خود را در لبه بالارونده کلاک یک واحد افزایش دهد. در غیر این صورت، رجیستر باید مقدار خود را نگه دارد.

```
module Counter(clock, reset, enable, Q);  
    input clock; input reset;  
    input enable;  
    output reg [7:0] Q;  
    always @(posedge clock) begin  
  
        if (reset) begin  
            Q <= 8'h0;  
        end  
        else if (enable && (Q!=8'hff)) begin  
            Q <= Q + 1;  
        end  
    end  
endmodule
```

موفق باشید
