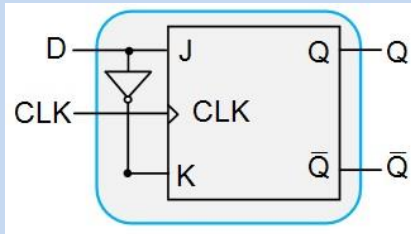
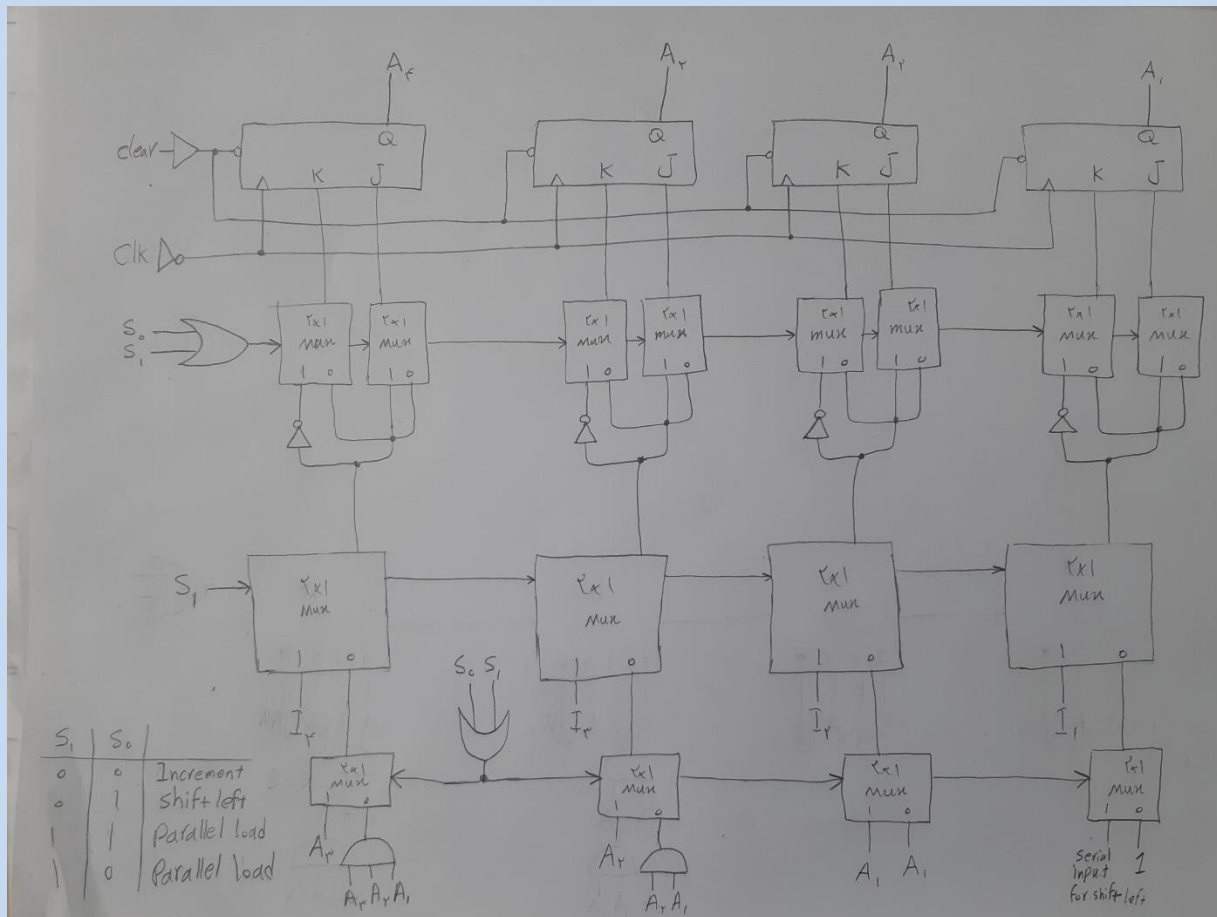


شماره تکلیف: ۷

ارجاع	پاسخ															
	<p>۲- یک شیفت رجیستر با ۳ عملیات بارگذاری، شیفت به چپ و افزایش (increment) طراحی کنید.</p> <div></div> <p>تبدیل فلیپ فلاپ JK به D</p> <div></div> <table data-bbox="260 1574 499 1753"><tr><th>S_1</th><th>S_0</th><th>Operation</th></tr><tr><td>0</td><td>0</td><td>Increment</td></tr><tr><td>0</td><td>1</td><td>Shift left</td></tr><tr><td>1</td><td>1</td><td>Parallel load</td></tr><tr><td>1</td><td>0</td><td>Parallel load</td></tr></table>	S_1	S_0	Operation	0	0	Increment	0	1	Shift left	1	1	Parallel load	1	0	Parallel load
S_1	S_0	Operation														
0	0	Increment														
0	1	Shift left														
1	1	Parallel load														
1	0	Parallel load														
۰	<p>۳- یک شیفت رجیستر ۴ بیتی با قابلیت بار شدن موازی را با استفاده از فلیپ فلاپ D و دو ورودی shift و load به گونه ای طراحی نمایید که شرایط زیر را داشته باشد:</p> <p>وقتی $shift = 1$ است محتوای ثابت یک مکان جابه جا شود.</p> <p>وقتی $load = 1$ و $shift = 1$ است داده جدید وارد رجیستر می گردد.</p> <p>اگر هر دو ورودی کنترل برابر صفر باشند، محتوای ثابت تغییر نمی کند.</p>															

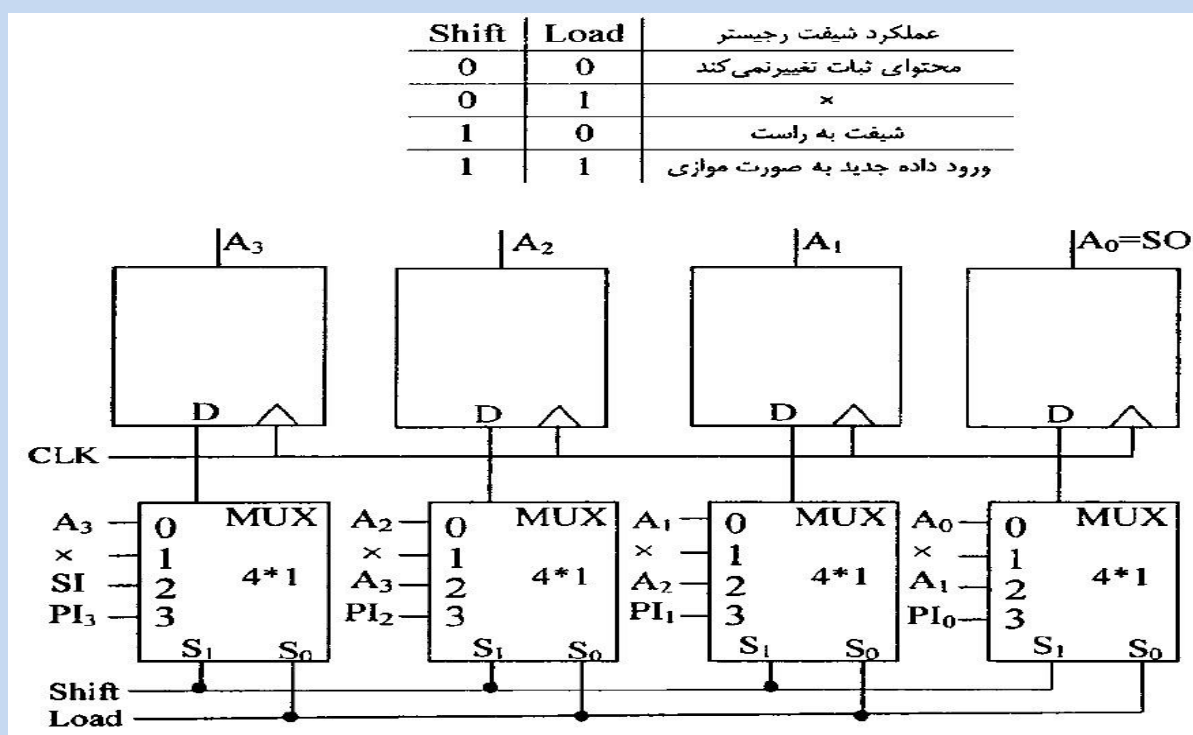


شماره تکلیف: ۷

پاسخ:

در ردیف دوم جدول درستی عملکرد مدار بی اهمیت (don't-care) است زیرا در صورت سوال عملکرد مدار را در این حالت تعیین نکرده و فرض بر این است که این حالت (shift = 0, load = 1) هرگز رخ نمی دهد. PI_0, PI_1, PI_2, PI_3 ورودی های موازی شیفت رجیستر هستند.

SI ورودی سریال (serial Input) و SO خروجی سریال (serial output) می باشند ضمناً مجموعه A_0, A_1, A_2, A_3 خروجی های موازی را تشکیل می دهند.



هنگامی که $shift = load = 0$ است در مولتی پلکسر سمت چپ، A_3 به عنوان خروجی انتخاب شده و به ورودی فلیپ فلاپ متناظرش منتقل می شود لذا در این حالت محتوای این فلیپ فلاپ بدون تغییر باقی می ماند. بقیه فلیپ فلاپ ها نیز به همین صورت محتوای خود را حفظ کرده و بنابر این در حالت $load = shift = 0$ محتوای ثابت بدون تغییر باقی می ماند. هنگامی که $shift = 1, load = 0$ بیت SI به فلیپ فلاپ سمت چپ وارد شده و بقیه فلیپ فلاپ ها، مقدار فلیپ فلاپ سمت چپ خود را به عنوان ورودی دریافت می کنند لذا در این حالت عمل شیفت به راست انجام می شود. هنگامی که $load = shift = 1$ باشد PI_0, PI_1, PI_2, PI_3 که ورودی های موازی هستند، همزمان با هم به فلیپ فلاپ های A_0, A_1, A_2, A_3 منتقل می شوند یعنی عمل بار شدن به صورت موازی (همزمان) انجام می گیرد.

۷- یک شمارنده صعودی باینری ۲ بیتی طراحی کنید.

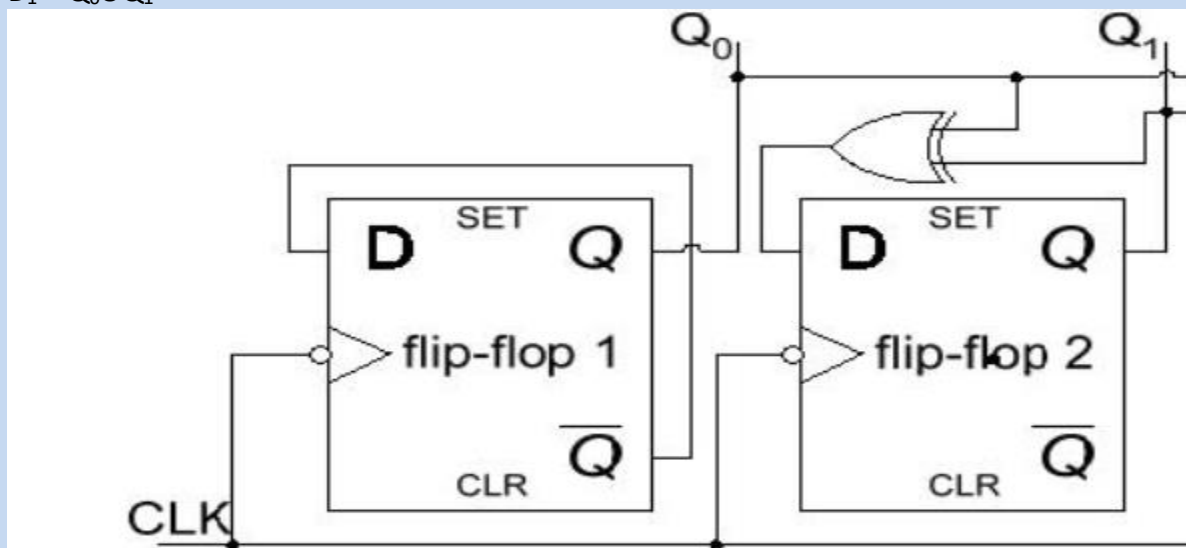
الف) با استفاده از فلیپ فلاپ D

Present State		Next State		D flip-flop inputs	
Q1	Q0	Q1	Q0	D1	D0
0	0	0	1	0	1

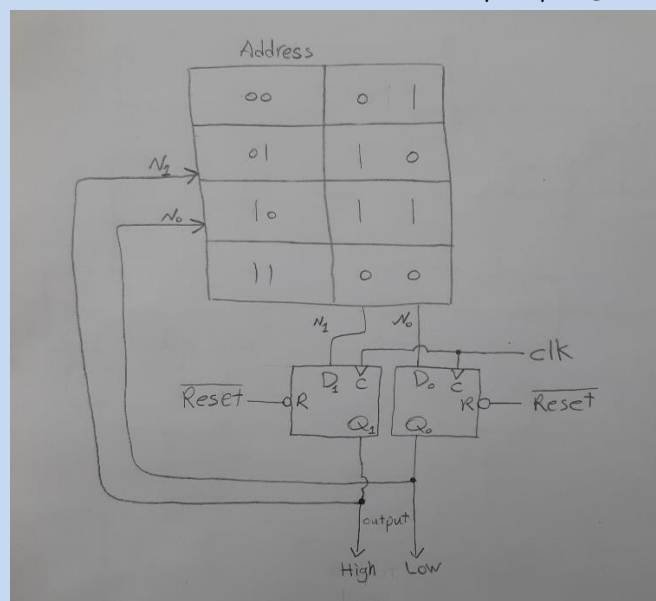
0	1	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0

$$D_0 = Q_0'$$

$$D_1 = Q_0 \oplus Q_1$$



(ب) یا استفاده از یک ROM و تعدادی فلیپ فلاپ D





شماره تکلیف: ۷

می توان باتوجه به جدول بدست آمده از قسمت قبل متوجه شد که برای مشخص کردن مقادیر درون حافظه نیازی به جدول حالت نیز نداریم و با توجه به این که نیاز به ورودی و خروجی نداریم (ورودی و خروجی نمودار حالت) پس به راحتی می توان تشخیص داد که اگر بخواهیم آدرس دهی ROM خود را طبق فرض از صفر برای فلیپ فلاپ در نظر بگیریم به راحتی با گذاشتن آدرس خانه ی بعدی ROM در خانه ی فعلی (و در خانه ی آخر، آدرس خانه اول) می توان یک شمارنده ساخت به طوری که ما فقط با فلیپ فلاپ D مقدار آن خانه حافظه را می خوانیم و نمایش می دهیم و در کلاک بعدی به خانه ی بعدی میرویم و این روند ادامه پیدا می کند. اما اگر نیاز بود (که با توجه به سوال در اینجا لزومی ندارد) که در خانه های میانی یک ROM بزرگتر این شمارنده را بسازیم (یعنی شروع از صفر نباشد)، آنگاه باز هم میتوانستیم با گذاشتن آدرس خانه بعدی در خانه فعلی این کار را انجام دهیم و فقط بیت های کم ارزش را به تعداد مورد نیاز بعنوان شمارنده بگیریم و اگر ترتیب نیز برای ما اهمیت داشت خروجی های مورد نظر را نیز به ترتیب در ROM ذخیره می کنیم. به طور مثال اگر آدرس حافظه در اختیار ما 1001 تا 1100 بود، که همان طور که پیداست یکان آن از 00 شروع نمی شود می توان در خروجی آدرس 1001 عدد 00 و در خروجی آدرس 1010 عدد 01 و در خروجی آدرس 1011 عدد 10 و در خروجی آدرس 1100 عدد 11 را گذاشت تا به جای استفاده از خروجی های کم ارزش فلیپ فلاپ ها که ترتیبشان از 01 به 00 است ترتیب درست یعنی 00 تا 11 را در خروجی داشت.