

۱- با فرض اینکه توابع f_1 و f_2 به فرم زیر تعریف شده و $f_3 = f_1 \oplus f_2$ باشد

$$f_1(a, b, c, d, e) = \sum m(0, 3, 4, 5, 11, 13, 14, 19, 21, 23, 27) + d(6, 10, 16, 22, 25)$$

$$f_2(a, b, c, d, e) = \sum m(2, 3, 5, 9, 12, 13, 18, 19, 20, 23, 29) + d(6, 10, 16, 22, 25)$$

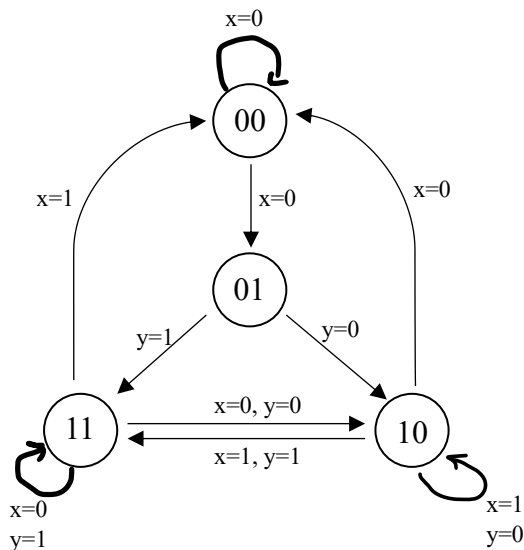
تابع f_3 را با استفاده از جدول کارنو ساده کرده و با مدارهای دوطبقه به روش NAND-AND و OR-NAND پیاده‌سازی نمایید.
 (راهنمایی: برای بدست آوردن تابع f_3 نیازی به رسم جدول درستی نیست) (۱۰ دقیقه - ۱۵ امتیاز)

۲- فقط با استفاده از واحدهای نیم جمع‌کننده (HA) و گیت‌های XOR مداری طراحی کنید که عدد علامت‌دار ۴ بیتی A (با نمایش سیستم متمم مینا) را در ورودی دریافت کرده و قدرمطلق عدد A را به عنوان خروجی ارائه دهد.

توجه: منظور از قدرمطلق عدد A عدد معادل بی‌علامت A می‌باشد.

مثال: قدرمطلق عدد 4- (عدد 1100 در مبنای ۲) برابر با عدد 4 (عدد 0100 در مبنای ۲) می‌باشد. (۱۵ دقیقه - ۱۵ امتیاز)

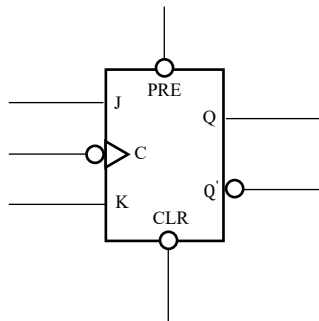
۳- فقط با استفاده از فلیپ‌فلاپ‌های D و یک ROM با اندازه مناسب، مداری طراحی کنید که از نمودار حالت زیر پیروی کند.
 (تعیین محتوای ROM الزامی است) (۱۵ دقیقه - ۳۰ امتیاز)



نکته: جایی که X یا y داده نشده مقدارش اهمیتی ندارد.

۴- الف) با استفاده از تعداد مناسبی از فلیپ‌فلاپ‌های JK نمایش داده شده در شکل زیر (دارای ورودی ناهمگام PRE (preset) و و CLR (clear) و تعدادی گیت مناسب یک شمارنده بالا / پایین شمار ناهمگام به گونه‌ای طراحی کنید که در هر دو وضعیت شمارش بالا یا پایین دقیقاً محدوده عدد 0 تا عدد 5 را بشمارد.

ب) فقط با استفاده از فلیپ‌فلاپ‌های نوع D که دارای ورودی RESET همگام (سنکرون) هستند و ماژول‌های نیم جمع‌کننده (HA) یک شمارنده بالا شمار (up-counter) به گونه‌ای طراحی کنید که قادر به شمارش اعداد از صفر تا ۸ باشد. (استفاده از هیچ عنصر دیگری به غیر از فلیپ‌فلاپ‌های نوع D و ماژول‌های نیم جمع‌کننده (HA) مجاز نیست). (۲۰ دقیقه - ۳۵ امتیاز)



۵- با توجه به سخت افزار شکل زیر و با استفاده از الگوریتم booth، مراحل ضرب دو عدد ۶ بیتی ۲۱- و ۲۱+ را با پر کردن جدول زیر در هر مرحله نشان دهید. (۱۵ دقیقه - ۲۰ امتیاز)

2	1	Iteration
	Initial values	Step
	010101	Multiplier
	101011	Multiplicand
	000000000000	product

