دانیال مینی - ۴۰۱۳۰۰۲۳

۱- ۰: مقدار خونهٔ [illegible] X مموری داخل رجیستر R1 ریخته می‌شود.

۱: بیت E (Carry) و رجیستر R2 هردو برابر صفر می‌شوند.

۲: رجیستر R1 به راست شیفت می‌خورد و بیت کم ارزش R1 درون E ذخیره می‌شود و پر ارزش‌ترین بیت R1 درون E ریخته می‌شود.

۳: مقدار E با رجیستر R2 جمع می‌شود و در درون R2 ریخته می‌شود.

۴: مقدار خانهٔ Y مموری را یک واحد افزایش می‌دهیم و چک می‌کنیم اگر مقدار خانهٔ Y مموری + یک برابر با صفر است ← مقدار PC را یک واحد اضافه می‌کنیم و در PC ذخیره می‌کنیم ← اگر مقدار خانهٔ Y مموری برابر با صفر باشد خط بعدی دستورات اجرا نمی‌شود و خط بعد از آن اجرا می‌شود.

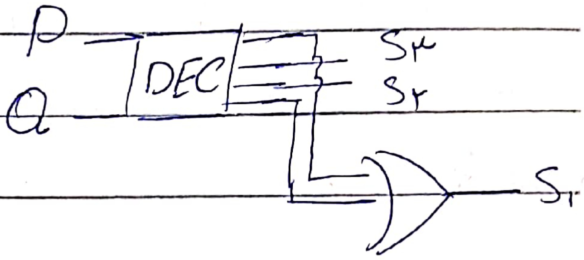
۵: مقدار ۲ در PC ریخته می‌شود که باعث می‌شود برنامه به خط ۲ باز گردد

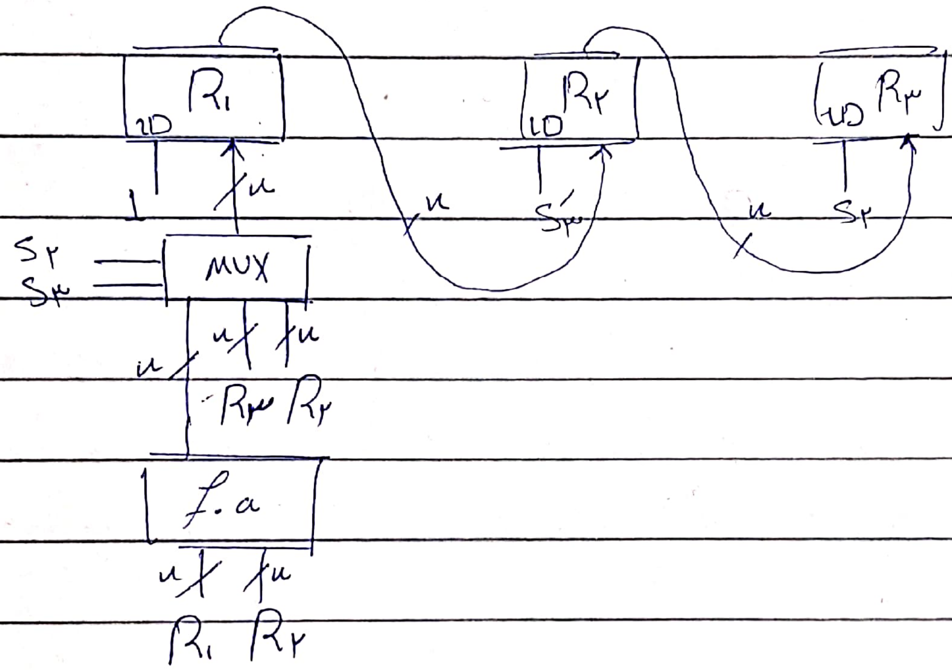
۶: چک می‌کند که اگر رجیستر R2 برابر صفر است بعد مقدار [illegible] در خانهٔ X مموری ذخیره کند.

این دستورات برای چک کردن این است که X بر 2^6 بخش‌پذیر می‌باشد یا نه که اگر بخش‌پذیر باشد [illegible] مقدار حافظه در خانه X تغییر نمی‌کند و اگر نباشد آنگاه مقدار حافظه در خانه X برابر با صفر خواهد شد.

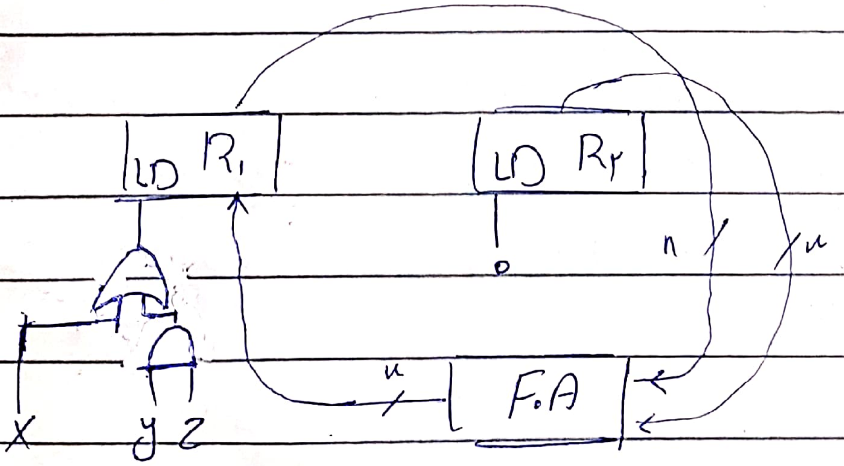
۳- الف :

P	Q	P⊕Q	Q'P	QP'
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0





ب :



دانیال حسینی ٤٠١٣٠٠٢٣

۳- از طریق بررسی کردن نحوه ساخت سیگنال های ورودی انکدر مدارمان را تحلیل می کنیم.

زمانی که حاصل $NOT(R_1)$ و $NOT(R_2)$ برابر ۱ شوند (یعنی هر دو برابر صفر شوند)

در این صورت کد خروجی انکدر باید ۰۰ باشد یعنی سلکت انتخاب می شود و

رجیسترهای R_1 و R_2 برابر صفر می شوند: $R_1 \leftarrow 0$, $R_2 \leftarrow 0$: R_1, R_2

در صورت صفر بودن R_1 مقدار R_2 درون R_1 ریخته می شود:

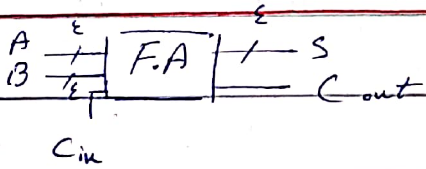
$T_1 : R_1 \leftarrow R_2$

باید حالت load رجیسترهای R_1 و R_2 هر دو با T_1 و R_1 اند شده اند فقط R_1 وجود

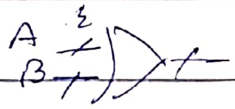
ولی R_1 , R_2 با not رجیستر R_1 اند شده است به عنوان ورودی نمی توانند باید load آنها فعال باشند.

٤-

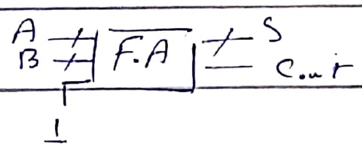
$T_1)\ A+B$:



$T_2)\ A \vee B$:



$T_3)\ A-B+1$:



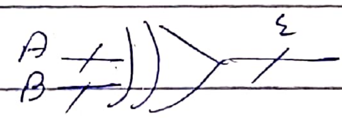
$T_4)\ A \wedge B$:

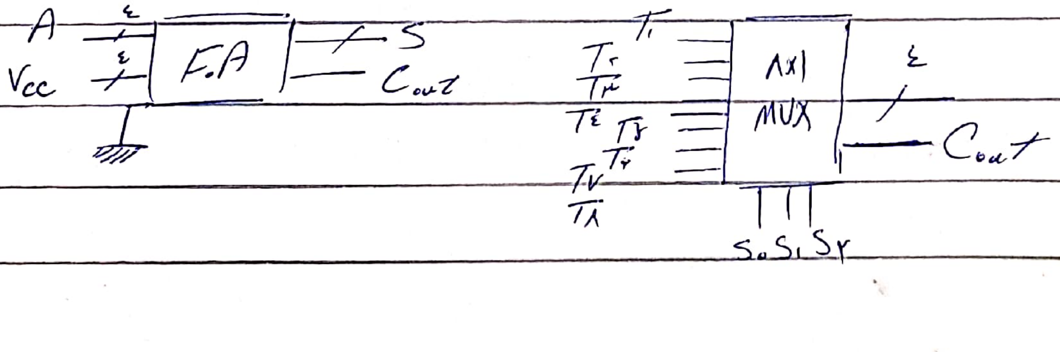
$T_5)\ A$:

$T_6)\ A'$:

$T_7)\ A-1 : A - 0001 = A + 1111$

$T_8)\ A \oplus B$:





دانیال حسینی ۴۰۱۳۰۰۲۳

۵- $xy: R_1 \leftarrow R_2 + R_1 , R_1 \leftarrow 0$

در این دستور ما می‌خواهیم به صورت همزمان و در یک سیکل ساعت مقدار R_1 را ۲ بار تغییر دهیم یکبار مقدار آن را صفر کنیم و بار دیگر R_2+R_1 که در خروجی حاصل R_1 ابهام و اشکال به وجود می‌آید.

$yz: R_1 \leftarrow R_2 , R_1 \leftarrow R_1+1$

مانند دستور بالا ما در یک سیکل ساعت اقدام به مقداردهی با R_1 ۲ مقدار متفاوت کرده‌ایم که اینکار درست نیست و در مقدار R_1 اشکال ایجاد می‌کند.

۶- یک سیکل برای ریختن R_1 به A و یک سیکل برای ریختن R_2 به B و یک سیکل هم برای جمع A و B که در کل ۳ سیکل لازم است.