حل مسائل فصل ۵

کلاس حل تمرین معماری کامپیوتر دکتر زینالی، رامتین کوثری، ۶ آذر ۱۴۰۳ کلاس ۲۰۳

حل مسائل با چند دستور در هر خانه حافظه :

در ادامه مطالب جلسه قبل، نکته ای که باید توجه شود این است که اگر خانه های حافظه برای مثال ۳۲ بیتی بودند ولی رجیستری که به خروجی حافظه متصل بود، ۱۶ بیتی بود و هر خانه حافظه ۲ دستور داشت یعنی هر دستور ۱۶ بیت، در مرحله Fetch باید یکبار بخش اول خانه حافظه را در رجیستر بریزیم و بار دیگر بخش دوم. اگر این حالت برقرار باشد، هرکجا که از حافظه چیزی را می خواندیم یا در آن چیزی را می ریختیم باید از فلگ کمکی m استفاده کنیم. در غیر این صورت اگر رجیستر متصل به خروجی حافظه، همان ۳۲ بیت بود فلگ کمکی m را در هر حالت باید بررسی کنیم و بنویسیم. در نتیجه حواستان به این نکته باشد.

جلسه قبل ما فرمت دستور با ۲ بخش را حل کردیم، این جلسه مثال امتحان با ۳ بخش را حل میکنیم.

در این نوع از فرمت دستور ما باید این مسئله را با استفاده از یک فلگ کمکی حل کنیم، حال پیش از آنکه این فلگ را طراحی کنیم باید ابتدا خود فرمت دستور را بررسی کنیم. فرمت دستور ما مجموعا ۴۲ بیت است که بخش سمت چپی دستور اول و بخش میانی دستور دوم و بخش سمت راست بخش سوم میباشد. اگر ما دستور بخش اول را انجام دهیم باید به بخش دوم رفته و آن را انجام دهیم ولی حواستان باشد که از مراحل چرخه دستورات مرحله Fetch را برای بخش های دوم و سوم انجام نمی دهیم چون Fetch می آید و از حافظه یک بلوک را میخواند، حال که بلوک حافظه ما از ۳ دستور تشکیل شده است هنگامی که دستور اول را خواندیم در همان بلوک Fetch شده باید به سراغ دستور دوم در بخش دوم برویم سپس به دستور سوم در بخش سوم و نباید Fetch کنیم. حال اگر در بخش سوم یا آخر باشیم باید برویم بخش اول بلوک بعدی حافظه پس اینجاست که دوباره باید Fetch را انجام دهیم زیرا بلوک بعدی را میخواهیم. تا اینجا در نظر داشته باشید که ما برای تعیین بودن در بخش اول و بخش دوم نیاز داریم که یک فلگ کمکی یا جانبی داشته باشیم و SC را فقط هنگامی که در بخش سوم باید ریست (صفر) کنیم. این نکته را هم توجه داشته باشید که مرحله بعد از Fetch مرحله Decode میباشد و در این مرحله اجزای دستور را تفکیک می کردیم، حال اگر از بخش اول گذر کنیم باید برای بخش دوم نیز این کار را انجام دهیم و سپس بخش سوم و همچنین مراحل بعدی مانند E.A و Execute. پس برای اینکه مرحله Fetch را انجام ندهیم ولی مرحله Decode را انجام دهیم باید Sequencer را به روی زمانی ببریم که مرحله Decode شروع میشود.

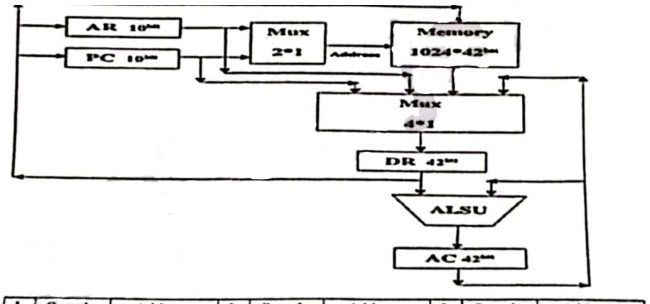
به حل مثال زیر از امتحان دقت کنید که میخواهیم مراحل چرخه اجرای دستورات را برای این حالت بنویسیم :

در این مثال ابتدا باید آن فلگ کمکی را تعین کنیم، ما ۳ دستور در یک بلوک بیشتر نداریم پس با ۲ بیت میتوانیم آن را کد کنیم که مثلا با m

نمایش میدهیم حال اگر $m = 00$ باشد قرارداد می کنیم بخش اول

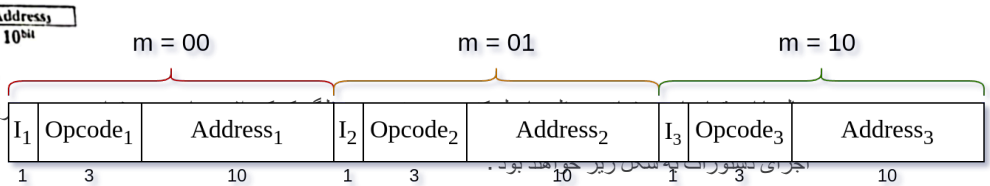
و اگر $m = 01$ باشد قرارداد می کنیم بخش دوم و اگر $m = 10$ باشد

قرارداد میکنیم بخش سوم اجرا شود :



I_1 1bit	$Opcode_1$ 3bit	$Address_1$ 10bit	I_2 1bit	$Opcode_2$ 3bit	$Address_2$ 10bit	I_3 1bit	$Opcode_3$ 3bit	$Address_3$ 10bit

Symbol	Opcode	Function
ADDM	111	M[AR] ← M[AR] + AC
SUB	011	AC ← AC - M[AR]



Fetch : $IR \leftarrow M[PC]\ ,\ PC \leftarrow PC + 1$

این مرحله چون ۱ بلوک حافظه را در مقدار PC میخواند پس نیازی نداریم که فلگ جانبی را لحاظ کنیم و از آنجایی که شکل رجیستر IR ندارد باید از رجیستر DR استفاده کنیم، همچنین PC مستقیم به آدرس مموری متصل است پس خواهیم داشت :

$$T_0 : \quad DR \leftarrow M[PC] \ , \ PC \leftarrow PC + 1$$

Decode :

این مرحله باید حواسمان باشد که باید برای هر ۳ حالت فلگ کمکی نوشته شود یعنی :

$$m_1'm_0'T_1 : \ I \leftarrow DR_{(13)} \ , \ D_0D_1D_2 \dots D_{31} \leftarrow DR_{(12-10)} \ , \ AR \leftarrow DR_{(9-0)}$$
$$m_1'm_0\,T_1 : \ I \leftarrow DR_{(27)} \ , \ D_0D_1D_2 \dots D_{31} \leftarrow DR_{(26-24)} \ , \ AR \leftarrow DR_{(23-14)}$$
$$m_1\,m_0'T_1 : \ I \leftarrow DR_{(41)} \ , \ D_0D_1D_2 \dots D_{31} \leftarrow DR_{(40-38)} \ , \ AR \leftarrow DR_{(37-28)}$$

Effective Address : $if\ (I = 1)\ then\ AR \leftarrow M[AR]$

با توجه به نکات جلسه قبل داریم :

$$I\,T_2 : \quad NOP$$
$$I'T_2 : \quad DR \leftarrow M[AR]$$
$$I'T_3 : \quad AR \leftarrow DR$$

Execute : $D_0D_1D_2 \dots D_{31}$

برای **مثال** مرحله اجرا دستور ADDM را می نویسیم :

$$D_7(ADDM) : \quad AC \leftarrow (AC \oplus M[AR])'$$

$$D_7T_4 \quad : \quad DR \leftarrow M[AR]$$
$$D_7T_5 \quad : \quad AC \leftarrow AC + DR$$
$$D_7T_6 \quad : \quad DR \leftarrow AC$$
$$D_7T_7 \quad : \quad M[AR] \leftarrow DR$$

حال در اینجا دستور تمام شده و باید سراغ دستور بعدی برویم ولی حواستان باشد که دستور بعد در خانه بعدی حافظه نیست پس نباید SC را صفر کنیم تا مرحله Fetch اتفاق بیفتد بلکه فقط باید به بخش دوم $(m \leftarrow 01)$ برویم و همچنین SC را ۱ کنیم تا چرخه از مرحله Decode در دستور بخش دوم شروع شود و وقتی دستور بخش دوم تمام شد SC نباید تغییر کند (باید همان ۱ بماند) و همچنین فلگ جانبی را نیز باید روی بخش سوم تنظیم کنیم $(m \leftarrow 10)$ که به دستور بعد برود و حال که برای دستور سوم فلگ جانبی باید دوباره صفر شود $(m \leftarrow 00)$ و همچنین SC باید صفر شود که چرخه اجرای دستور بعدی از بخش اول در خانه بعدی حافظه شروع شود پس کافی است به RTL های بالا، RTL های **جدیدی** اضافه کنیم که این فرایند را انجام دهد :

$$D_7T_4 \quad : \quad DR \leftarrow M[AR]$$
$$D_7T_5 \quad : \quad AC \leftarrow AC + DR$$
$$D_7T_6 \quad : \quad DR \leftarrow AC$$
$$m_1'm_0'D_7T_7 \quad : \quad M[AR] \leftarrow DR \ , \ SC \leftarrow 1 \ , \ m_0 \leftarrow 1$$
$$m_1'm_0\,D_7T_7 \quad : \quad M[AR] \leftarrow DR \ , \ SC \leftarrow 1 \ , \ m_1 \leftarrow 1 \ , \ m_0 \leftarrow 0$$
$$m_1\,m_0'D_7T_7 \quad : \quad M[AR] \leftarrow DR \ , \ SC \leftarrow 0 \ , \ m_1 \leftarrow 0$$

پس میتوانیم به کمک فلگ جانبی این فرایند ها را انجام دهیم ولی باید توجه داشته باشیم که به ازای هر فلگی که اضافه میکنیم باید یک $J.K \ - \ f.f$ (جی کا فلیپ فلاپ) برای آن طراحی کنیم که در اینجا ۲ فلگ اضافی داریم.