خلاصه درس

معماری کامپیوتر-فصل ۷

تهیه کننده و گردآورنده:

BORNA66

Power By WWW.Borna66.IR

WWW.PNU-CLUB.COM

باشگاه علمی و آموزشی دانشجویان پیام نور

ارایه رایگان جدیدترین اخبار و نمونه سوالات امتحانی و جزوات درسی و منابع ارشد کلیه رشته ها ی پیام نور

==========

WWW. COMPUTER-ENG.IR <-> وب سایت مهندسی کامپیوتر <-> WWW.BORNA66.IR

كنترل ريزبرنامه نويسي شده

✓ حافظه كنترل :

وظيفه يك واحد كنترل به كارگيري دنباله هايي از ريزدستورالعمل هاست . پيچيدگي سيستم ديجيتال به تعداد ريزعمل هاي متوالي كه اجرا مي كند بستگي دارد . ريزبرنامه نويسي روشي شاخص و سيستماتيك در كنترل رشته ريز عمل ها است .

تابع كنترلي كه مشخص كننده يك ريزعمل است يك متغير دودويي مي باشد . وتي كه اين متغير در يكي از حالات دودويي باشد ، ريزعمل متناظر آن اجرا مي شود . اگر متغير كنترلي در وضعيت دودويي مخالف باشد وضعيت ثبات هاي سيستم را تغيير نمي دهد . وضعيت فعال يك متغير كنترل ممكن است بسته به كاربرد هر يك از دو حالت **1** يا **0** باشد . در سيستم مبتني بر گذرگاه **(Bus Organized System)** سيگنال هاي كنترلي كه ريزعمل را مشخص مي نمايند به صورت گروه هايي از بيت ها هستند كه مسيرها را در مولتي پلكسر ها ، ديكدرها و واحدهاي حساب و منطق انتخاب مي نمايند .

متغيرهاي كنترلي در هر لحظه از زمان مي توانند با رشته اي از **1** ها و**0** ها كه كلمه كنترل ناميده مي شود نمايش داده شوند .

واحد كنترلي كه متغيرهاي دودويي كنترل آن در حافظه ذخيره شده باشد واحد كنترل ريزبرنامه نويسي شده نام دارد . هر كلمه در واحد كنترل حاوي يك ريز دستورالعمل است . ريزدستورالعمل مشخص كننده يك يا چند ريزعمل براي سيستم است . رشته اي از ريزدستورالعمل ها ، ريزبرنامه را تشكيل مي دهند .

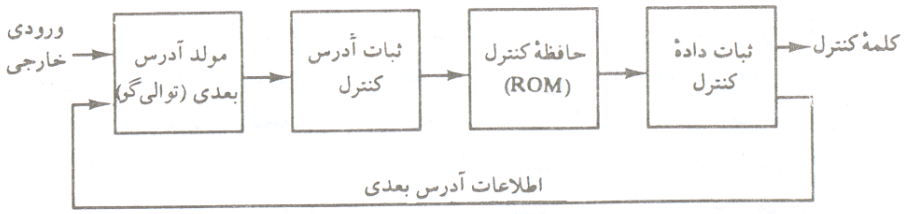
حافظه اي كه بخشي از واحد كنترل باشد حافظه كنترل ناميده مي شود .

كامپيوتري كه از واحد كنترل ريزبرنامه نويسي شده استفاده مي كند دو حافظه مجزا خواهد داشت : يك حافظه اصلي و يك حافظه كنترل . حافظه اصلي در اختيار كاربر است تا برنامه هاي خود را در آن ذخيره كند . محتواي حافظه اصلي به هنگام دستكاري داده ها و يا هر بار كه برنامه عوض شود تغيير مي نمايد . حافظه كنترل يك ريزبرنامه ثابتي را نگهداري مي كند و به وسيله هر كاربري قابل تغيير نيست .

ريز برنامه از ريزدستورالعمل هايي تشكيل شده كه به نوبه خود انواع سيگنال هاي كنترل را براي اجراي ريزاعمال روي ثبات ها معين مي كنند. هر دستورالعمل ماشين موجب اجراي رشته اي از ريزدستورالعمل هاي موجود در حافظه كنترل مي شود . وظيفه اين ريزدستورالعمل ها برداشت دستورالعمل ماشين از حافظه اصلي ، محاسبه آدرس موثر ، اجراي عملي كه توسط دستورالعمل ماشين مشخص شده ، و بازگرداندن كنترل به فاز برداشت براي تكرار سيكل مربوط به دستورالعمل بعدي است .

آرايش كلي واحد كنترل ريز برنامه نويسي شده در دياگرام شكل **1 – 7** نشان داده شده است.

WWW.PNU-CLUB.COM



شکل **1-7** : سازمان کنترل ریزبرنامه نویسی شده

مولد آدرس بعدي گاهي اوقات توالي گر **(Sequencer)** ریزبرنامه نامیده مي شود ، زیرا آدرس هاي متوالي کلماتي را که از حافظه کنترل خوانده مي شوند معین مي سازد . آدرس ریزدستورالعمل بعدي بسته به ورودي توالي گر به چند طریق معین مي شود . عملیات توالي گر ریزبرنامه عمدتا" عبارت است از افزایش ثبات آدرس کنترل به میزان یك واحد ، بارکردن آدرس حافظه کنترل در ثبات آدرس کنترل ، انتقال آدرس خارجي ، یا بار کردن یك آدرس اولیه براي آغاز اعمال کنترل .

مزیت عمده کنترل ریزبرنامه نویسي شده این است که وقتي پیکربندي سخت افزاري تکمیل شد ، سخت افزار اضافي و یا تغییر سیم بندي نخواهیم داشت . اگر بخواهیم کنترل متفاوتي را براي سیستم ایجاد کنیم ، فقط لازم است مجموعه اي از دستورالعمل هاي متفاوتي را براي حافظه کنترل اختیار کنیم . پیکربندي سخت افزار براي عملیات متفاوت تغییر نمي کند و تنها تغییري که باید صورت گیرد ریزبرنامه اي است که در حافظه کنترل جاي داده مي شود .

✓ توالي آدرس :

ریز دستورالعمل ها به صورت گروهي در حافظه ذخیره مي شوند و هر گروه نیز یک روال را تشکیل مي دهند . هر دستورالعمل کامپیوتر داراي روال ریزبرنامه متعلق به خود در حافظه کنترل است که ریزاعمال مربوط به اجراي دستورالعمل را تولید مي نماید. سخت افزاري که توالي آدرس حافظه کنترل را کنترل مي نماید باید قادر به دنبال کردن ریزدستورالعمل هاي یک روال بوده و قابلیت انشعاب از یک روال به روال دیگر را هم داشته باشد .

هنگامي که اجراي دستورالعملي پایان یابد ، کنترل باید به روال برداشت (واکشي) بازگردد . این عمل با اجراي یک ریزدستورالعمل انشعاب غیر شرطي به ابتداي روال برداشت تحقق مي یابد . به طور خلاصه قابلیت هاي لازم براي توالي آدرس در حافظه کنترل عبارتند از :

افزایش ثبات آدرس کنترل

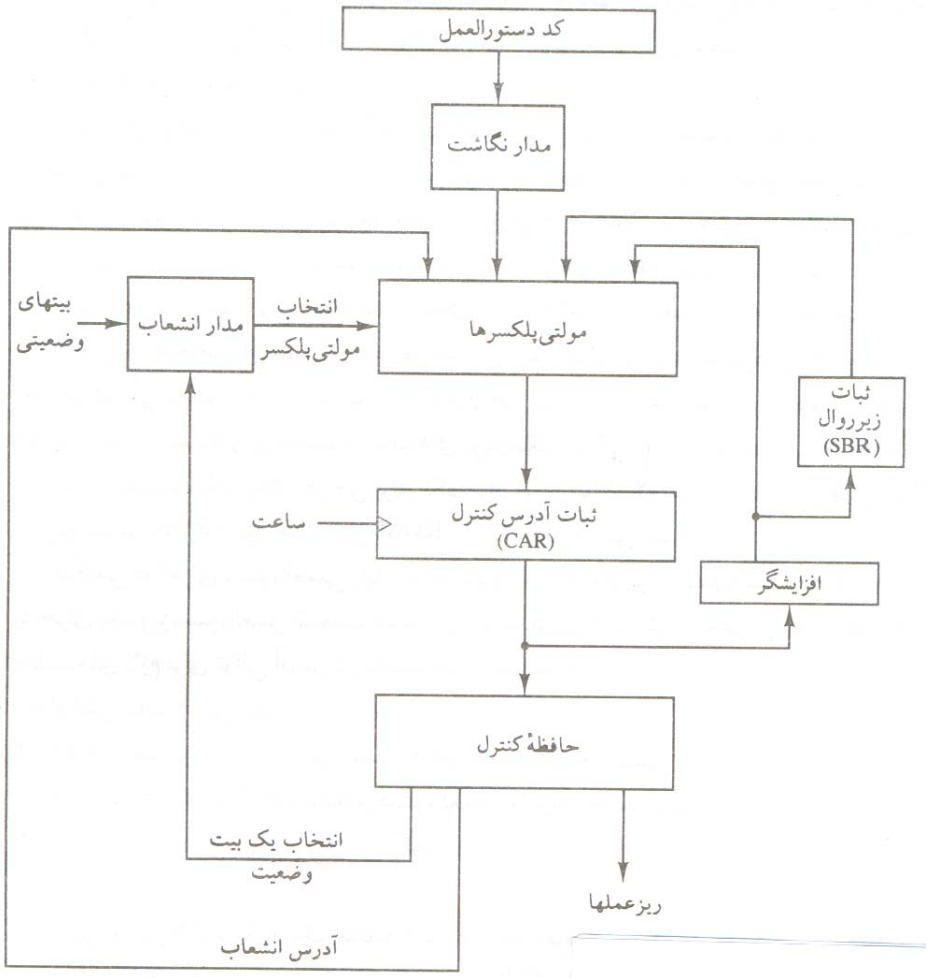
انشعاب غیر شرطي و یا شرطي

فرآیند نگاشت (تبدیل) از بیت هاي دستورالعمل به آدرس حافظه کنترل

امکاناتي براي فراخواني زیرروال و بازگشت از آن

WWW.PNU-CLUB.COM

شکل **2-7** بلاک دیاگرام یک حافظه کنترل و سخت افزار لازم براي انتخاب آدرس ریزدستورالعمل بعدي را نشان مي دهد . ریز دستورالعمل واقع در حافظه کنترل متشکل از مجموعه اي از بیت هاست که ریزاعمال را در ثبات هاي کامپیوتر موجب مي شود و نیز مجموعه اي از بیت هاي دیگر براي تهیه آدرس بعدي است . دیاگرام چهار مسیري که توسط آنها ثبات آدرس کنترل **(CAR)** آدرس را دریافت مي کند نشان مي دهد .



شکل **2-7** : انتخاب آدرس برای حافظه کنترل

WWW.PNU-CLUB.COM

✓ انشعاب شرطي :

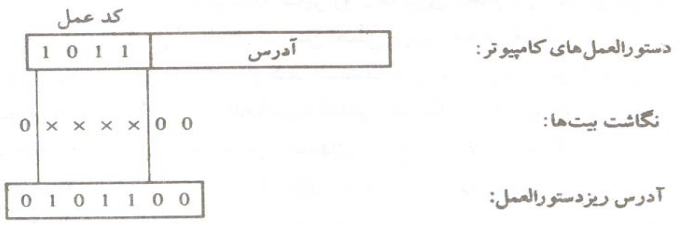
منطق انشعاب شكل **2-7** قابليت تصميم گيري را در واحد كنترل فراهم مي اورد . حالت وضعيت سيستم ، بيت هاي خاصي هستند كه اطلاعات پارامتري همچون ، نقلي خروجي يك جمع كننده ، بيت علامت يك عدد ، بيت هاي روش **(Mode bit)** در يك دستورالعمل ، شرايط وضعيتي ورودي يا خروجي را فراهم مي كنند. اطلاعات موجود در اين بيت ها قابل تست بوده و تصميم اتخاذ شده مي تواند بر اساس حالت آنها يعني **1** يا **0** باشد. بيت هاي وضعيت ، همراه با ميداني در ريزدستورالعمل كه آدرس انشعاب را مشخص مي نمايد ، تصميمات انشعاب هاي شرطي را كه در مدار انشعاب توليد مي شود كنترل مي نمايند .

ريزدستورالعمل انشعاب غيرشرطي را مي توان با باركردن آدرس انشعاب از حافظه كنترل به ثبات آدرس كنترل ايجاد كرد . اين كار را مي توان با ثابت نگه داشتن مقدار يك بيت وضعيت در ورودي مولتي پلكسر در منطق **1** انجام داد . ارجاع به اين بيت توسط خطوط انتخاب از حافظه كنترل موجب مي شود تا آدرس انشعاب در ثبات آدرس كنترل بدون هر شرطي بار شود .

✓ نگاشت دستورالعمل :

نوع خاصي از انشعاب هنگامي است كه يك ريزدستورالعمل حكم كند كه به اولين كلمه در حافظه كنترل كه روال ريزبرنامه مربوط به يك دستورالعمل در آنجا واقع است انشعاب شود . بيت هاي وضعيت براي اين نوع انشعاب ، بيت هاي بخش كد عمليات در دستورالعمل هستند . مثلا" ، كامپيوتري با قالب ساده اي براي دستورالعمل ، مطابق شكل **3-7** كد عمل چهاربيتي دارد كه مي تواند تا **16** دستورالعمل متمايز را تشخيص دهد . فرض كنيد حافظه كنترل **128** كلمه داشته باشد ، در نتيجه به آدرس هفت بيتي نياز خواهد داشت . براي هر كد عمليات يك روال ريزبرنامه در حافظه وجود دارد كه دستورالعمل را اجرا مي كند . نمونه ساده اي از فرآيند نگاشت كه كد عمليات **4** بيتي را به آدرس **7** بيتي براي حافظه كنترل تبديل مي كند در شكل **3-7** نشان داده شده است . اين نگاشت بدين صورت انجام مي شود كه يك **0** در پرارزش ترين مكان آدرس قرارداده مي شود ، چهار بيت كد عمليات منتقل مي گردد و دو بيت كم ارزش تر ثبات آدرس كنترل پاك مي شوند . لذا براي هر دستورالعمل كامپيوتر مي توان يك روال ريزبرنامه با ظرفيت چهار ريزدستورالعمل داشت . اگر روال به بيش از چهار ريزدستورالعمل نياز داشته باشد ، مي توان از آدرس هاي **1000000** تا **1111111** استفاده كرد و اگر كمتراز چهار ريز عمل به كار رود ،از مكان هاي حافظه بلا استفاده مي توان براي روال هاي ديگر استفاده كرد .

WWW.PNU-CLUB.COM



شکل **3-7** : نگاشت (تبديل) کد دستورالعمل به آدرس ريزدستورالعمل

✓ زير روال ها :

زيرروال ها برنامه هايي هستند که به وسيله ساير روال ها براي انجام کار خاصي مورد استفاده قرار مي گيرند . زير روال را مي توان از هر نقطه اي در متن اصلي ريزبرنامه فراخواني کرد . غالبا" بسياري از ريزبرنامه ها حاوي بخش هاي يکساني از کد هستند . مي توان ريزدستورالعمل را با به کارگيري زيرروال هايي که بخش هاي مشترکي از ريزکد ها دارند کاهش داد .

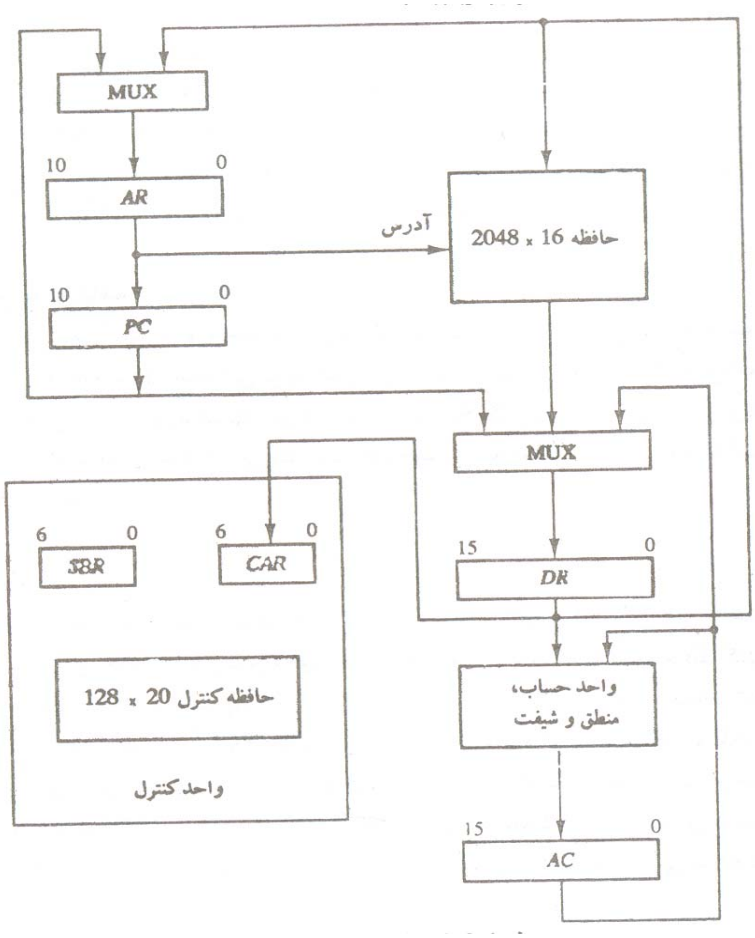
✓ آرايش کامپيوتر :

کامپيوتر مورد نظر از دو واحد حافظه تشکيل شده است : يکي حافظه اصلي براي ذخيره دستورالعمل ها و داده ها ، و ديگري حافظه کنترل براي ذخيره ريزبرنامه . چهار ثبات به واحد پردازنده و دو ثبات به واحد کنترل اختصاص يافته اند . ثبات هاي پردازنده عبارتند از شمارنده برنامه ، **PC** ، ثبات آدرس **AR** ، ثبات داده **DR** و ثبات انباره **AC** .

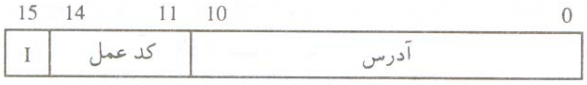
انتقال اطلاعات بين ثبات هاي پردازنده از طريق مولتي پلکسر صورت مي گيرد و نه با استفاده از گذرگاه مشترک . **DR** مي تواند اطلاعات را از **AC** ، **PC** و يا حافظه دريافت کند . **PC** فقط از **AR** اطلاعات دريافت مي کند . واحد حساب ، منطق و شيفت ريزاعمال را بر روي داده هاي دريافتي از **AC** و **DR** انجام داده و حاصل را در **AC** قرار مي دهد . دقت کنيد که حافظه آدرس خود را از **AR** دريافت مي کند . داده ورودي که در حافظه نوشته مي شود از طريق **DR** وارد ، و داده خوانده شده از حافظه هم فقط به **DR** مي رود .

شکل **5-7** ب چهار دستور از شانزده دستور العمل هاي ارجاع به حافظه را نشان مي دهد . دستورالعمل **ADD** عملوند موجود در آدرس موثر را با محتواي **AC** جمع مي کند . دستورالعمل **BRANCH** ، انشعابي را به آدرس موثر موجب مي شود به شرطي که عملوند درون **AC** منفي باشد . اگر **AC** منفي نباشد برنامه با دستورالعمل بعدي ادامه مي يابد . **AC** در صورتي منفي است که بيت علامت (بيت منتها اليه سمت چپ ثبات)**1** باشد . دستورالعمل **STORE** محتواي **AC** را به کلمه اي از حافظه که آدرس موثر آن را مشخص مي کند انتقال مي دهد . دستورالعمل **EXCHANGE** داده هايي را بين **AC** و کلمه حافظه مشخص شده به وسيله آدرس موثر تعويض مي کند .

WWW.PNU-CLUB.COM



شکل **7-4** : پیکربندی سخت افزار کامپیوتر



(الف) قالب دستورالعمل

توضیح	کد عمل	سمبل
$AC \leftarrow AC + M[EA]$	0000	ADD
If $(AC<0)$ then $(PC \leftarrow EA)$	0001	BRANCH
$M[EA] \leftarrow AC$	0010	STORE
$AC \leftarrow M[EA],\ M[EA] \leftarrow AC$	0011	EXCHANGE

EA آدرس مؤثر است

(ب) چهار دستورالعمل کامپیوتر

شکل **7-5** : دستورالعمل های کامپیوتر

WWW.PNU-CLUB.COM

✓ قالب ريز دستورالعمل ها :

قالب ريزدستورالعمل ها براي حافظه كنترل در شكل **7-6** نشان داده شده است . **20** بيت هر دستورالعمل به چهار بخش عملياتي تقسيم شده است . سه ميدان **F1** ، **F2** و **F3** ريزاعمال را براي كامپيوتر مشخص مي كنند . ميدان **CD** بيت هاي شرطي وضعيت را انتخاب مي نمايد . ميدان **BR** نوع انشعاب به كار رفته را معين مي كند . ميدان **AD** آدرس انشعاب را داراست . ميدان آدرس هفت بيت عرض دارد ، زيرا حافظه كنترل داراي $128 = 2^7$ كلمه است .

3	3	3	2	2	7
F1	F2	F3	CD	BR	AD

F1، F2، F3: ميدان‌هاى ريزعمل

CD : شرط انشعاب

BR : ميدان انشعاب

AD : ميدان آدرس

شكل **7-6** : قالب كد دستورالعمل (**20** بيت)

✓ ريز دستورالعمل هاي سمبليك :

هر سطر از ريزبرنامه زبان اسمبلي ، يك ريزدستورالعمل سمبليك را تعريف مي كند . هر دستورالعمل سمبليك به پنج ميدان تقسيم مي شود : عنوان ، ريزاعمال ، **CD** ، **BR** ، **AD** . اين ميدان ها اطلاعات زير را مشخص مي نمايند .

ميدان عنوان مي تواند خالي و يا يك آدرس سمبليك باشد . هر عنوان با يك علامت نقل قول (:) خاتمه مي يابد .

ميدان ريزعمل ها از يك ، دو و يا سه سمبل تعريف شده در جدول **7-1** تعريف مي شود و به وسيله كاما از هم جدا مي شوند . نمي توان بيش از يك سمبل از هر ميدان **F** داشت هرگاه ريزدستورالعمل هيچ عملي نداشته باشد از سمبل **NOP (NO OPERATION)** استفاده مي شود . اسمبلر اين سمبل را به **9** عدد **0** ترجمه مي كند .

ميدان **CD** داراي يك حرف **U** ، **I** ، **S** و يا **Z** است .

ميدان **BR** حاوي يكي از چهار سمبل تعريف شده در جدول **7-1** است .

ميدان **AD** مقدار ميدان آدرس ريزدستورالعمل را به يكي از سه طريق زير مشخص مي كند .

(الف) با يك آدرس سمبليك ، كه بايد به صورت عنوان هم وجود داشته باشد .

(ب) با سمبل **NEXT** براي مشخص كردن آدرس متوالي بعدي .

(ج) اگر ميدان **BR** حاوي سمبل **RET** ، يا **MAP** باشد ، ميدان **AD** خالي گذاشته مي شود و اسمبلر آن را به هفت عدد صفر تبديل مي نمايد .

WWW.PNU-CLUB.COM

F1	ریزعمل	سمبل
000	هیچکار	NOP
001	$AC \leftarrow AC + DR$	ADD
010	$AC \leftarrow 0$	CLRAC
011	$AC \leftarrow AC + 1$	INCAC
100	$AC \leftarrow DR$	DRTAC
101	$AR \leftarrow DR(0\text{–}10)$	DRTAR
110	$AR \leftarrow PC$	PCTAR
111	$M[AR] \leftarrow DR$	WRITE

F2	ریزعمل	سمبل
000	هیچکار	NOP
001	$AC \leftarrow AC - DR$	SUB
010	$AC \leftarrow AC \vee DR$	OR
011	$AC \leftarrow AC \wedge DR$	AND
100	$DR \leftarrow M[AR]$	READ
101	$DR \leftarrow AC$	ACTDR
110	$DR \leftarrow DR + 1$	INCDR
111	$DR(0\text{-}10) \leftarrow PC$	PCTDR

F3	ریزعمل	سمبل
000	هیچکار	NOP
001	$AC \leftarrow AC \oplus DR$	XOR
010	$AC \leftarrow \overline{AC}$	COM
011	$AC \leftarrow \text{shl } AC$	SHL
100	$AC \leftarrow \text{shr } AC$	SHR
101	$PC \leftarrow PC + 1$	INCPC
110	$PC \leftarrow AR$	ARTPC
111	Reserved	

CD	شرط	سمبل	توضیح
00	همیشه = 1	U	انشعاب غیرشرطی
01	$DR(15)$	I	بیت آدرس غیرمستقیم
10	$AC(15)$	S	بیت علامت AC
11	$AC = 0$	Z	مقدار صفر در AC

BR	سمبل	عملکرد
00	JMP	$CAR \leftarrow AD$ اگر شرط برابر 1 باشد $CAR \leftarrow CAR + 1$ اگر شرط برابر صفر باشد
01	CALL	$CAR \leftarrow AD,\ SBR \leftarrow CAR + 1$ اگر شرط برابر 1 باشد $CAR \leftarrow CAR + 1$ اگرشرط برابر 0 باشد
10	RET	$CAR \leftarrow SBR$ بازگشت از زیرروال tine)
11	MAP	$CAR(2\text{–}5) \leftarrow DR(11\text{–}14),\ CAR(0,1,6) \leftarrow 0$

جدول **1-7** : سمبل ها و کدهای دودویی برای میدان های ریزدستورالعمل ها

✓ روال برداشت : (Fetch Roution)

حافظه کنترل داراي **128** کلمه است و هر کلمه **20** بيت دارد . براي ريزبرنامه نويسي حافظه كنترل ، لازم است تا مقادير بيتي هر **128** كلمه را معين كنيم . **64** كلمه اول (آدرس**0** تا **63**) به وسيله روال هاي **16** دستورالعمل اشغال

WWW.PNU-CLUB.COM

مي شوند . **64** كلمه آخر را مي توان براي هر هدف ديگري به كار برد . آدرس **64** شروع مناسبي براي روال برداشت (يا واكشي) است . ريزدستورالعمل هاي لازم براي اين روال عبارتند از :

AR ← PC
DR ← M[AR] , PC ← PC + 1
AR ← DR(0 – 10) , CAR (2 – 5) ← DR (11 – 14) , CAR (0 , 1 , 6) ← 0

```
            ORG 64
FETCH :     PCTAR          U    JMP    NEXT
            READ , INCPC   U    JMP    NEXT
            DRTAR          U    MAP
```

آدرس دودويي	F1	F2	F3	CD	BR	AD
1000000	110	000	000	00	00	1000001
1000001	000	100	101	00	00	1000010
1000010	101	000	000	00	11	0000000

✓ ريزبرنامه سمبليك :

```
INDRCT :  READ    U    JUMP    NEXT
          DRTAR   U    RET
```

عنوان	ريزعمل‌هاى	CD	BR	AD
	ORG 0			
ADD:	NOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	ADD	U	JMP	FETCH
	ORG 4			
BRANCH:	NOP	S	JMP	OVER
	NOP	U	JMP	FETCH
OVER:	NOP	I	CALL	INDRCT
	ARTPC	U	JMP	FETCH
	ORG 8			
STORE:	NOP	I	CALL	INDRCT
	ACTDR	U	JMP	NEXT
	WRITE	U	JMP	FETCH
	ORG 12			
EXCHANGE:	NOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	ACTDR, DRTAC	U	JMP	NEXT
	WRITE	U	JMP	FETCH
	ORG 64			
FETCH:	PCTAR	U	JMP	NEXT
	READ, INCPC	U	JMP	NEXT
	DRTAR	U	MAP	
INDRCT:	READ	U	JMP	NEXT
	DRTAR	U	RET	

جدول **2-7** : بخشى از برنامه سمبليک

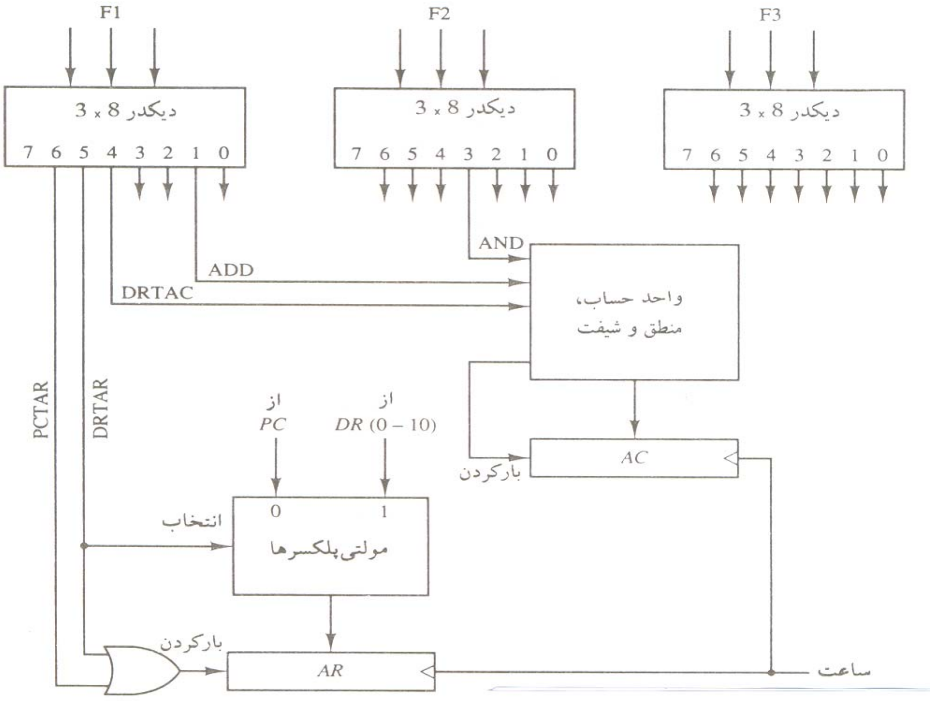
WWW.PNU-CLUB.COM

✓ ريز برنامه دودويي :

ريزروال	آدرس دهدهی	آدرس دودويي	ريزدستورالعمل دودويی F1	F2	F3	CD	BR	AD
ADD	0	0000000	000	000	000	01	01	1000011
	1	0000001	000	100	000	00	00	0000010
	2	0000010	001	000	000	00	00	1000000
	3	0000011	000	000	000	00	00	1000000
BRANCH	4	0000100	000	000	000	10	00	0000110
	5	0000101	000	000	000	00	00	1000000
	6	0000110	000	000	000	01	01	1000011
	7	0000111	000	000	110	00	00	1000000
STORE	8	0001000	000	000	000	01	01	1000011
	9	0001001	000	101	000	00	00	0001010
	10	0001010	111	000	000	00	00	1000000
	11	0001011	000	000	000	00	00	1000000
EXCHANGE	12	0001100	000	000	000	01	01	1000011
	13	0001101	001	000	000	00	00	0001110
	14	0001110	100	101	000	00	00	0001111
	15	0001111	111	000	000	00	00	1000000
FETCH	64	1000000	110	000	000	00	00	1000001
	65	1000001	000	100	101	00	00	1000010
	66	1000010	101	000	000	00	11	0000000
INDRCT	67	1000011	000	100	000	00	00	1000100
	68	1000100	101	000	000	00	10	0000000

جدول **3-7** : بخشی از ریزبرنامه دودويي برای حافظه کنترل

✓ طراحي واحد كنترل :



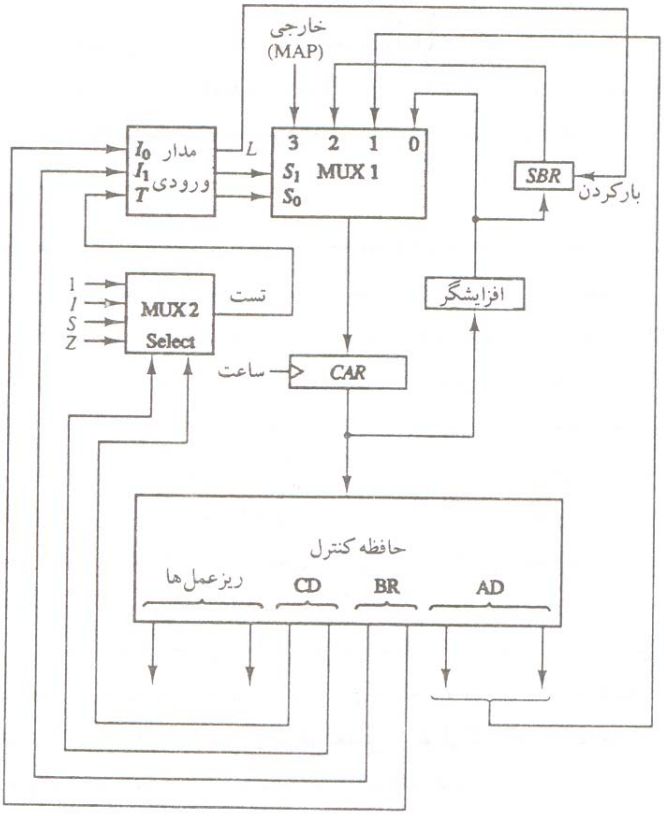
شکل **7-7** : دیکدر میدان های ریزعمل

WWW.PNU-CLUB.COM

✓ توالي گر ريزبرنامه :

اجزاء اصلي واحد كنترل ريزبرنامه نويسي شده حافظه كنترل و مدارات اتخاب آدرس بعدي هستند . بخش انتخاب آدرس ، توالي گر ريزبرنامه خوانده مي شود . يك توالي گر ريزبرنامه را مي توان با اجزاء ديجيتال ، مناسب يك كاربرد خاص ساخت . با اين وجود ، همان طور كه واحدهاي **ROM** بزرگي به شكل بسته هاي مدار مجتمع وجود دارند ، توالي گر هاي همه منظوره هم وجود دارند كه براي ساخت واحدهاي كنترل ريزبرنامه نويسي شده مناسب اند . براي تضمين پذيرش وسيع ، بايد سازمان داخلي مدار مجتمع توالي گر به صورتي باشد كه بتوان آن را در گستره وسيعي از كاربردها مورد استفاده قرار داد .

هدف از به كارگيري توالي گر ريزبرنامه ، ارائه آدرس به حافظه كنترل است به طوري كه يك ريزدستورالعمل از آن خوانده شده و اجرا گردد . مدار آدرس بعدي توالي گر ، منبع آدرس بعدي ، كه در ثبات آدرس كنترل ذخيره خواهد شد را مشخص مي كند . انتخاب منبع آدرس به وسيله بيت هاي اطلاعات آدرس بعدي كه توالي گر آنها را از ريزدستورالعمل جاري دريافت مي كند هدايت مي شود .



شكل **7-8** : توالی گر ریزبرنامه برای حافظه کنترل

WWW.PNU-CLUB.COM

جدول درستي مدار منطقي ورودي در جدول **7-4** نشان داده شده است . ورودي هاي I_0 ، I_1 همان بيت هاي ميدان **BR** هستند . تابع مشخص شده براي هر وارده از جدول **7-1** گرفته شده است . مقادير بيت هاي S_1 و S_0 با توجه به عمل مورد نظر و مسيري كه انتخاب لازم را برقرار مي كند تعيين مي شود . اگر شرط بيت وضعيت در حين اجراي ريزدستورالعمل فراخواني **(BR = 01)** برقرار باشد **(T = 1)** ، مقدار افزايش يافته **CAR** در ثبات زيرروال بار مي شود . با استفاده از اين جدول مي توان توابع ساده شده را براي مدارمنطقي ورودي به دست آورد .

$$S_1 = I_1$$
$$S_0 = I_1 I_0 + I'_1 T$$
$$L = I'_1 I_0 T$$

اين مدار را مي توان با سه گيت **AND** ، يك **OR** و يك معكوس كننده ساخت .

BR ميدان		ورودی I_1	I_0	T	MUX 1 S_1	S_0	بارکردن SBR L
0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0
0	1	0	1	0	0	0	0
0	1	0	1	1	0	1	1
1	0	1	0	×	1	0	0
1	1	1	1	×	1	1	0

جدول **7-4** : جدول درستی مدارمنطقی ورودی توالی شمار ریزبرنامه

WWW.PNU-CLUB.COM

باشگاه دانشجویان دانشگاه پیام نور

www.pnu-club.com

- ارایه جدیدترین **اخبار و اطلاعیه های** مختلف مرتبط با دانشگاه پیام نور
- ارایه جدیدترین نمونه **سوالات امتحانی** در تمامی رشته های پیام نور بصورت رایگان
- بانک جامع نمونه سوالات در چندین نیمسال برای تمامی رشته ها **بصورت رایگان**
- ارایه جدیدترین **مقالات و مطالب علمی مرتبط با رشته های** تحصیلی پیام نور
- بحث و گفتگو و تبادل نظر **و پاسخ به سوالات** مختلف پیام نور و رشته های تحصیلی پیام نور
- ارایه جدید ترین **جزوات و منابع کارشناسی و کارشناسی ارشد** رشته های تحصیلی پیام نور
- ارایه جدیدترین **نرم افزار های کامپیوتری** و مباحث گوناگون آموزشی و علمی و سرگرمی و.....
- و ارایه هر آنچه که مرتبط با پیام نور و سایر موضوعات **تخصصی و عمومی مجاز و قابل بحث**

تهیه کننده و گردآورنده:

BORNA66

Power By WWW.Borna66.IR