معماری کامپیوتر

کنترل ریزبرنامه‌نویسی‌شده

فصل هفتم کتاب موریس مانو

محمدعلی شفیعیان

http://shafieian-education.ir

بهار ۹۸

کنترل ریزبرنامه‌نویسی‌شده

مقدمه

حافظه واحد کنترل

تولیدکننده آدرس بعدی

زیرروال (سابروتین)

نگاشت دستورالعمل

مثال‌هایی از کنترل میکروپروگرام

2

مقدمه

- کنترل سخت‌افزاری زمانی استفاده می‌شود که سیستم پیچیدگی کمی داشته باشدو تعداد ریزعمل‌ها محدود باشد.

- کنترل ریزبرنامه‌نویسی (میکروپروگرام) زمانی استفاده می‌شود که سیستم دارای پیچیدگی زیاد باشد.
  - در این حالت، سیگنال‌های کنترلی بیت‌های ذخیره شده در یک حافظه کنترلی هستند که به قسمت‌های مختلف ارسال می‌شوند.
  - مزیت عمده این روش، انعطاف‌پذیری آن است زیرا با تغییر در سخت‌افزار و سیستم کنترل تنها کافی است برنامه ذخیره شده در حافظه کنترلی را تغییر داد.

3

مقدمه

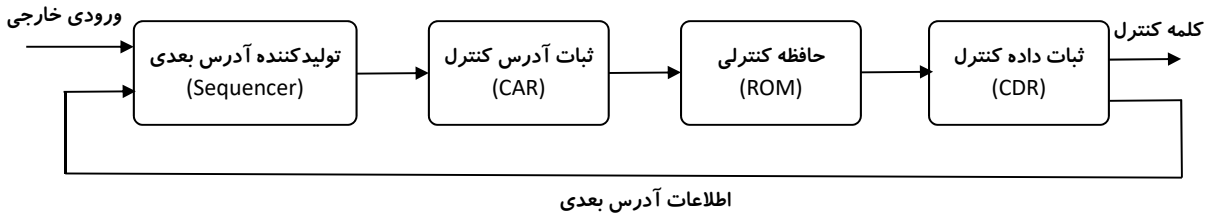
- کلمه کنترل (Control Word) : یک خانه از حافظه کنترلی است.

- ریزدستور (Micro-instruction): هر کلمه کنترل دربرگیرنده یک ریزدستور است که اجرای یک یا چند ریزعمل را به عهده دارد.

- ریزبرنامه (Micro-program): مجموعه‌ای از ریزدستورات به‌طور متوالی را ریزبرنامه می‌گویند. در مجموع ریزبرنامه‌ها وظیفه کنترل سیستم دیجیتالی را بر عهده دارد. عموماً این برنامه نیازی به تغییر نداشته و بنابراین در حافظه‌های ROM ذخیره می‌شوند.

4

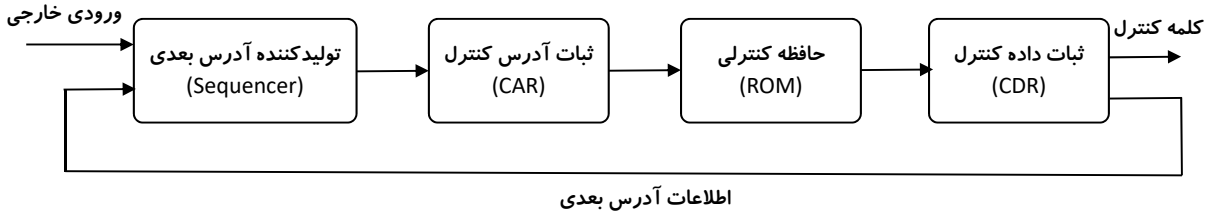
حافظه کنترل



- هر دستور ماشین که اجرا می‌شود باعث اجرای رشته‌ای از ریزعمل‌ها می‌شود تا مراحل مختلف اجرای یک دستور (Fetch، Addressing، Execute) اجرا شود.
- بعد از طراحی و تشخیص بیت‌های کنترلی که به‌صورت ریزدستور در حافظه کنترلی قرار می‌گیرند آنچه حائز اهمیت است، توالی اعمال ریزدستورات است.
  - این توالی باید به‌گونه‌ای باشد که فازهای مختلف اجرای دستور، متوالیاً اجرا شوند.

5

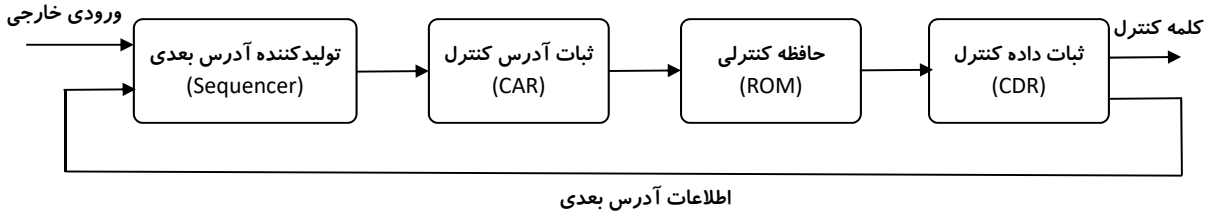
حافظه کنترل



- حافظه کنترل (ROM) حاوی ریزدستورات (اطلاعات) کنترلی است. حافظه کنترل جرا از حافظه اصلی و بخشی از سیستم کنترل می‌باشد.
- تولیدکننده آدرس بعدی (Sequencer) آدرس ریزدستور بعدی حافظه را تولید می‌کند. در ساده‌ترین شکل، این آدرس، آدرس ریزدستور فعلی بعلاوه یک است، اما حالاتی نیز وجود دارد که آدرس ریزدستور بعدی در مکانی دورتر از مکان فعلی است که در این حالت ریزدستور فعلی در تولید آدرس ریزدستور بعدی دخیل است.

6

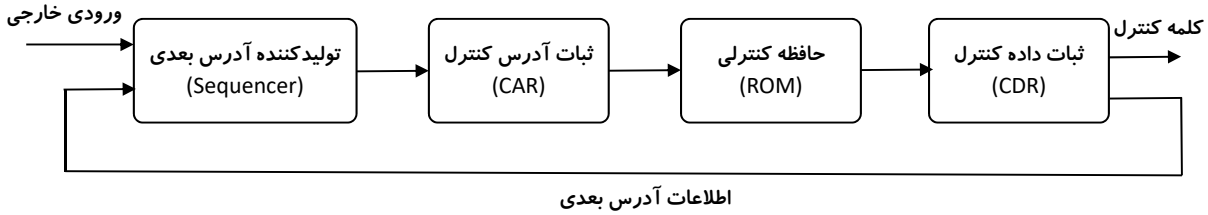
حافظه کنترل



- **ثبات آدرس کنترل (CAR) آدرس ریزدستورات مورد نظر برپا اعمال به سیستم است.**
- **ثبات داده کنترل (CDR) حاوی ریزدستور می‌باشد. در برخی از سیستم‌ها از CDR صرف‌نظر می‌شود.**

7

حافظه کنترل

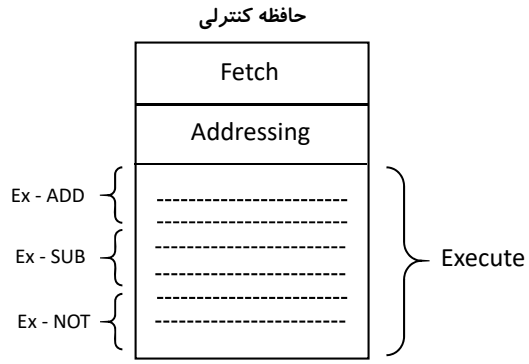


- **روال (Routine): به مجموعه‌ای از ریزدستورات که یک عمل خاص را انجام می‌دهند روال یا روتین گفته می‌شود.**
- **روتین‌های Fetch و Addressing در یک محل خاص از حافظه کنترلی هستند ولی برای اجرای هر دستور بایستی یک روتین خاص اجرا شود.**

8

چگونگی عمل کنترل ریزبرنامه‌نویسی

- وقتی که کامپیوتر روشن می‌شود ابتدا یک آدرس اولیه در CAR بار می‌شودو معمولاً این آدرس، آدرس اولین خانه روال fetch است.



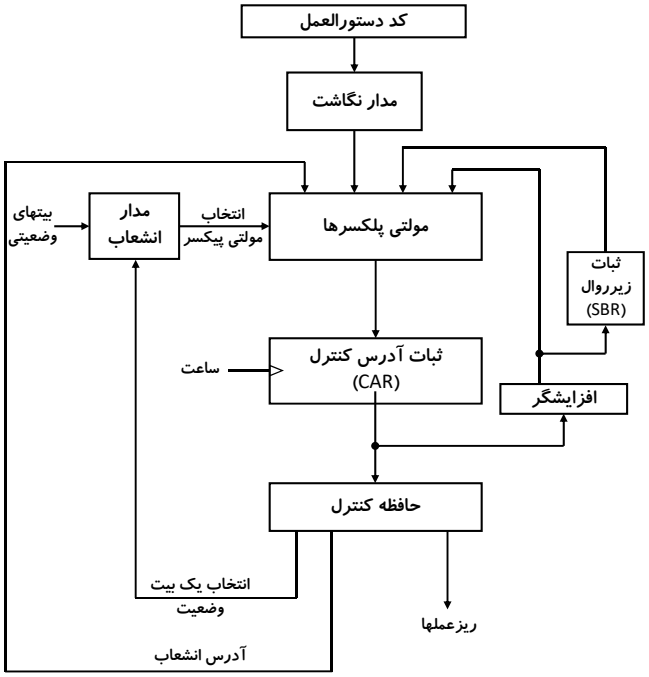
9

چگونگی عمل کنترل ریزبرنامه‌نویسی

- در پایان روال fetch دستورالعمل در IR قرار گرفته است.
- سپس حافظه کنترلی بایستی روال Addressing را اجرا کند. این عمل از طریق یک ریزدستور پرش انجام می‌شود. پس از انجام روال Addressing آدرس نیز در AR قرار گرفته است.
- با توجه به بخش Opcode دستور در IR، یک زیروال خاص برای Execute اجرا می‌شود.

10

تولیدکننده آدرس بعدی



- **نگاشت** (Mapping)
- **افـزایش** CAR بـرای اجـرای ریزدستورات متوالی
- **پرش** شرطی یا غیرشرطی (بـا توجـه بـه بیتهـای وضعیت)
- امکان فراخوانی و بازگشت از **زیرروال**

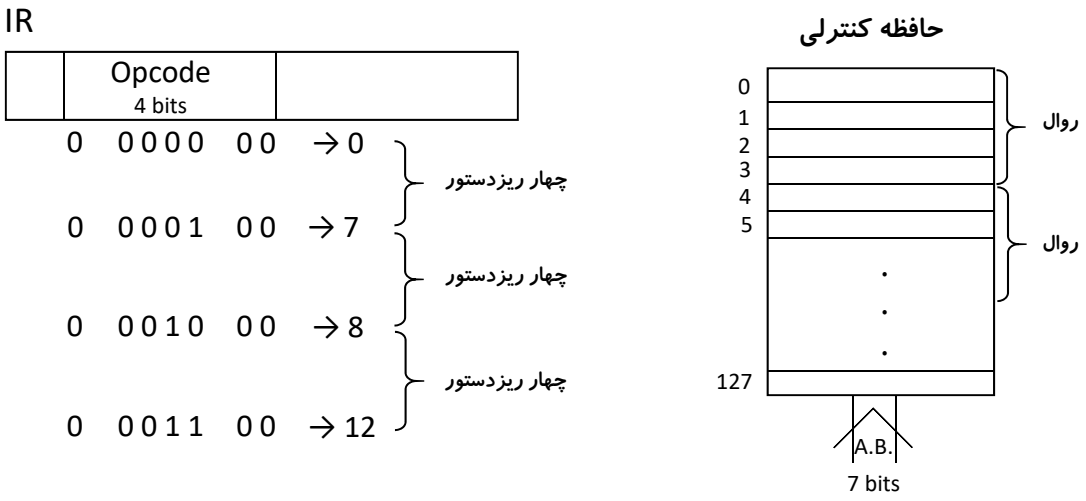
11

نگاشت ( )

- تبدیل بیتهای Opcode بـه **آدرس خاصـی از حافظـه کنترلـی** کـه در بردارنـده **روال** مربوط به آن opcode است.
- وقتی به روال مورد نظر دست یافتیم با **افزایش** CAR آدرس ریزدستورات متـوالی از یک روال ایجاد می‌شود.
- برای دستیابی به مکان ذخیره ریزدستور یک opcode خـاص، بایسـتی opcode بـه یک **آدرس در حافظه کنترلی** تبدیل شود.

12

نگاشت ( )



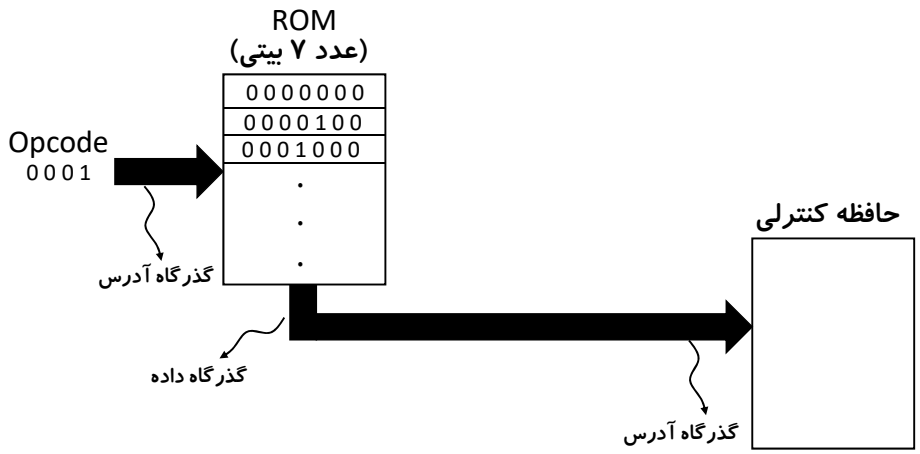
13

نگاشت ( )

- روش دیگر برای نگاشت، استفاده از حافظه ROM است.
- در این حالت بیت های opcode به عنوان خطوط آدرس حافظه ROM استفاده می‌شوند.
- در این حالت طول هر کلمه حافظه ROM هفت بیت است، بنابراین با آمدن هر opcode معین، یک کلمه هفت بیتی که آدرس حافظه کنترلی است در خروجی ROM ظاهر می‌شود و به حافظه کنترلی ارسال می‌شود.

14

نگاشت ( )



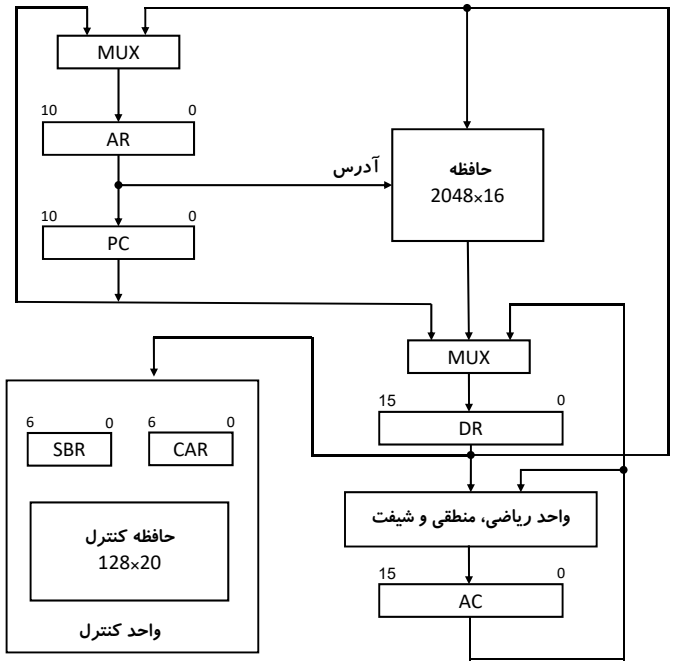
15

زیرروال

- زیرروال برنامه‌هایی هستند که روال‌های دیگر از آنها برای انجام یک کار خاص استفاده می‌کنند، لذا زیرروال ها باعث صرفه جویی در تعداد ریزدستورات می‌شود.
- زیرروال‌ها از هر کجای برنامه اصلی میکروپروگرام می‌تواند فراخوانی شود.
- ریزبرنامه هایی که زیرروال بکار می‌برند باید امکاناتی برای ذخیره آدرس برگشت به برنامه اصلی ریزبرنامه داشته باشند. برای این کار می توان آدرس برگشت را در ثبات زیرروال SBR قرار داد.
- بهترین وسیله برای ذخیره آدرس برگشت برای مراجعه به برنامه اصلی و ذخیره اطلاعات دیگر، استفاده از ثبات‌هایی است که به‌صورت حافظه پشته، LIFO، سازمان‌دهی شده باشند.

16

مثالی از ریزبرنامه

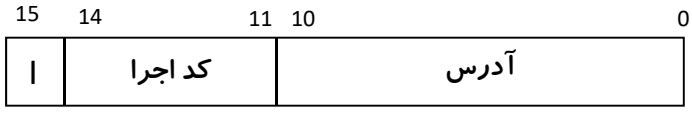


- نقل و انتقال اطلاعات بین ثبات‌ها در پردازنده، به‌جای استفاده از گذرگاه، از طریق مالتی‌پلکسر انجام می‌شود.

17

مثالی از ریزبرنامه

- فرمت دستور



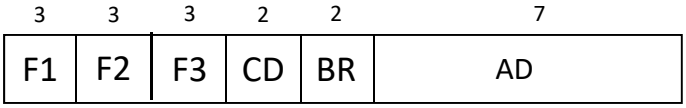
- چهار دستور کامپیوتر

نماد	کد اجرا	توضیح
ADD	0000	AC ← AC + M[EA]
BRANCH	0001	اگر (AC<0) آنگاه (PC ← EA)
STORE	0010	M[EA] ← AC
EXCHANGE	0011	AC ← M[EA] و M[EA] ← AC

18

مثالی از ریزبرنامه

- فرمت ریزدستور



F1، F2 و F3: میدان‌های ریزعمل

CD: شرایط انشعاب

BR: میدان انشعاب

AD: میدان آدرس

19

مثالی از ریزبرنامه

- سمبل‌ها و کدهای باینری برای میدانهای ریزدستورالعمل‌ها

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

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

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

20

مثالی از ریزبرنامه

- سمبل‌ها و کدهای باینری برای میدانهای ریزدستورالعمل‌ها

CD	شرط	سمبل	توضیح
00	1 = همیشه	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

21

مثالی از ریزبرنامه

- سمبل‌ها و کدهای باینری برای میدانهای ریزدستورالعمل‌ها

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	$CAR(2\text{-}5) \leftarrow DR(11\text{-}14), CAR(0,1,6) \leftarrow 0$

22

ریزدستورالعمل‌های سمبلیک

- هر ریزدستورالعمل سمبلیک شامل ۵ فیلد می‌باشد:
  - فیلد سرفصل (عنوان) ممکن است خالی باشد یا اینکه یک آدرس سمبلیک را مشخص کند که در انتهای آن دونقطه، : ، قرار می‌گیرد.
  - فیلد ریزعمل‌ها مه ممکن است شمل یک، دو یا سه سمبل باشد که با کاما از هم جدا شده‌اند.
  - فیلد CD دارای یکی از حروف U، I، S و Z است.
  - فیلد BR دارای یکی از چهار سمبل JMP، CALL، RET و MAP است.
  - فیلد AD مشخص‌کننده مقدار آدرس ریزدستورات است.

```
FETCH:    PCTAR          U          JMP          NEXT
```

23

ریزدستورالعمل‌های سمبلیک

- تعیین مقدار آدرس ریزدستورات در فیلد AD به یکی از سه راه زیر انجام می‌شود:
  - به صورت آدرس سمبلیک که باید به فرم سرفصل نیز وجود داشته باشد.
  - به فرم سمبلیک NEXT که نشان‌دهنده مراجعه به آدرس بعدی است.
  - زمانی که فیلد BR شامل سمبل RET یا MAP است فیلد AD خالی است و با برنامه اسمبلر به هفت عدد صفر تبدیل می‌شود.

24

ریزدستورالعمل‌های سمبلیک

- از شبه‌دستور ORG برای تعریف مبدأ یا اولین آدرس یک روال ریزبرنامه استفاده می‌شود.

- مثلاً سمبل ORG 64 به اسمبلر اطلاع می‌دهد که ریزدستورالعمل بعدی را در حافظه کنترل در آدرس‌دهی ۶۴ قرار دهد که معادل آدرس باینری 1000000 است.

25

روال واکشی دستورالعمل

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       JMP
```

26

روال واکشی دستورالعمل

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

F1	F2	F3	CD	BR	AD

آدرس باینری F	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

27

عنوان	ریزعمل‌های	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	

ریزروال	آدرس		ریزدستورالعمل دودویی					
	دهدهی	دودویی	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

28

__ *Claire Cook*

برای کسب اطلاعات بیشتر در مورد این درس می‌توانید به وب سایت آموزشی در لینک زیر مراجعه نمایید

http://shafieian-education.ir

30