

معماری کامپیوتر

سازمان و طراحی کامپیوتر پایه

فصل پنجم کتاب موریس مانو

محمدعلی شفیعیان

<http://shafieian-education.ir>

زمستان ۹۷

سازمان و طراحی کامپیوتر پایه

کدهای دستورالعمل

ثبات های کامپیوتر

دستورالعمل های کامپیوتر

زمان بندی و کنترل

سیکل دستورالعمل

دستورالعمل های ارجاع به حافظه

ورودی-خروجی و وقفه

تشریح کامل کامپیوتر

طراحی کامپیوتر پایه

طراحی مدار منطقی انباره

مقدمه

- **سازمان کامپیوتر** بوسیله ثبات های داخلی اش، زمانبندی و ساختار کنترل و مجموعه دستوراتی که به کار می برد تعریف می گردد.
- **دستورالعمل:** یک کد دودوئی است که رشته ای از ریز اعمال را برای کامپیوتر مشخص می کند.
- دستورالعمل مجموعه ای از بیت ها است که انجام یک عمل خاص را به کامپیوتر فرمان می دهد.
- کد دستورالعمل معمولاً به دو قسمت تقسیم می شود:
 - بخش عمل: کد عمل را دربر می گیرد و اعمالی مانند جمع، تفریق، ضرب، شیفت و متمم سازی را تعریف می کند.
 - نکته: تعداد بیت های مربوط به کد عمل یک دستور به تعداد کل اعمال موجود در کامپیوتر بستگی دارد.
 - بخش آدرس

3

مقدمه

- دستورالعمل ، عمل و ریز عمل
- **دستورالعمل:** تشکیل شده از کد عمل و آدرس.
- **عمل:** بخشی از دستور (دستورالعمل) ذخیره شده در حافظه است که انجام عملی خاص را به کامپیوتر دستور می دهد.
- **ریز عمل:** واحد کنترل برای هر کد عمل، رشته ای از ریز اعمال لازم را برای تحقق سخت افزاری عمل صادر می کند.

4

مقدمه

• بخش آدرس:

- در کد دستورات عمل نه تنها عمل بلکه ثبات یا حافظه ای که عمل باید روی آنها انجام شود نیز مشخص می شود.
- کلمات حافظه با استفاده از آدرسشان در بخش آدرس دستورات عمل مشخص می شوند.

5

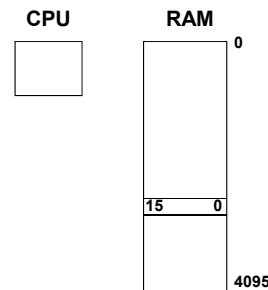
مقدمه

- هر پردازنده طراحی خاص خود (ثبات ها، گذرگاه ها، ریز عمل ها، دستورات عمل های ماشین و ...) را دارد.
- کامپیوترهای مدرن ساختار پیچیده ای دارند.
- در ادامه برای فهم اینکه کامپیوتر چگونه کار می کند از یک مدل ساده شده استفاده شده است.
- این مدل را آقای مانو (Mano) معرفی کرده و نام آنرا کامپیوتر پایه گذاشته است.
- این مدل شبیه کامپیوترهایی است که ۳۰ سال پیش کار می کرده اند.
- از این مدل برای معرفی سازمان پردازنده و ارتباط RTL با سطح بالاتر پردازنده استفاده می کنیم.

6

کامپیوتر پایه

- کامپیوتر پایه دو جز (component) اصلی دارد: پردازنده و حافظه.
- حافظه ۴۰۹۶ کلمه دارد.
- $4096 = 2^{12}$ یعنی به ۱۲ خط آدرس نیاز داریم.
- هر کلمه ۱۶ بیت طول دارد.



7

دستورالعمل ها

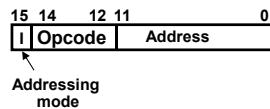
- برنامه : یک دنباله از دستورالعمل ها
- دستورالعمل
- یک گروه از بیت ها که به کامپیوتر اعلام می کنند که یک عمل خاص را انجام دهند.
- (یک دنباله از ریزعمل ها)
- دستورالعمل های یک کامپیوتر به همراه همه داده های لازم در حافظه ذخیره شده اند.
- CPU دستور بعدی را از حافظه می خواند.
- این دستور در یک ثبات به نام IR ذخیره شده است.
- دستورالعمل به دنباله ای از ریزعمل ها تبدیل می شود تا با انجام ریزعمل ها دستورالعمل مورد نظر اجرا شود.

8

فرمت دستورالعمل ها

- یک دستورالعمل اغلب از دو بخش تشکیل شده است.
- کد عمل (*opcode*): عملی را که دستورالعمل باید انجام دهد مشخص می کند.
- آدرس (*address*): ثبات یا مکانی از حافظه که دستورالعمل باید روی آن عمل کند را مشخص می کند.

Instruction Format



- همانطور که گفتیم در کامپیوتر پایه ۱۲ بیت برای آدرس دهی حافظه داریم.
- در کامپیوتر پایه بیت ۱۵ دستورالعمل، مُد آدرس دهی (addressing mode) را مشخص می کند. آن را با I نشان می دهیم.
- صفر: آدرس دهی مستقیم (direct addressing)
- یک: آدرس دهی غیر مستقیم (indirect addressing).
- چون کلمه های حافظه و دستورات و بنابراین ثبات دستورالعمل ۱۶ بیتی هستند، ۳ بیت باقی مانده برای کد دستورالعمل مورد استفاده قرار می گیرد.

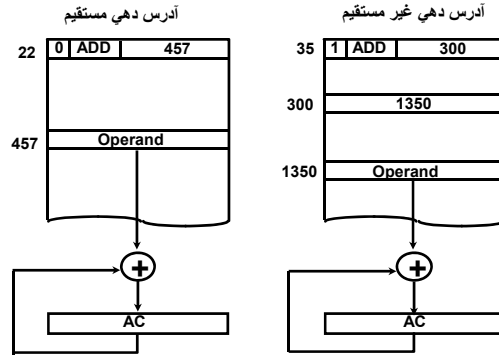
9

مُد آدرس دهی (Addressing Mode)

- ناحیه آدرس یک دستورالعمل می تواند به یکی از دو شکل زیر تفسیر شود:
 - آدرس مستقیم: آدرس عملوند (داده مورد نظر در حافظه)
 - آدرس غیر مستقیم: آدرس آدرس عملوند

10

مُد آدرس دهی (Addressing Mode)



• آدرس مؤثر (EA) Effective Address

- آدرس مؤثر = آدرس عملوند
- در شکل بالا سمت چپ آدرس مؤثر ۴۵۷ است.
- در شکل بالا سمت راست آدرس مؤثر ۱۳۵۰ است.

11

ثبات های پردازنده

- یک پردازنده تعداد زیادی ثبات برای نگهداری دستورالعمل ها، آدرس ها و داده ها و ... دارد.
- ✓ **PC**: شمارنده برنامه (PC=Program Counter) که آدرس دستوری که باید اجرا شود را نگه داری می کند.
- ❖ چون حافظه در کامپیوتر پایه ۴۰۹۶ کلمه دارد پس PC ، ۱۲ بیتی است.
- ✓ **AR**: پردازنده برای آنکه آدرس عملوند را نگه دارد از یک ثبات به نام ثبات آدرس (AR=Address Register) استفاده می کند.
- ❖ چون حافظه در کامپیوتر پایه ۴۰۹۶ کلمه دارد پس AR ، ۱۲ بیتی است.
- ✓ **DR**: پس از آنکه عملوند در حافظه پیدا شد، در آدرس دهی مستقیم یا غیر مستقیم، عملوند به یک ثبات به نام ثبات داده (DR=Data Register) منتقل می شود.
- ✓ **AC**: کامپیوتر پایه یک ثبات همه منظوره به نام انبار (AC=Accumulator) نیز دارد.

12

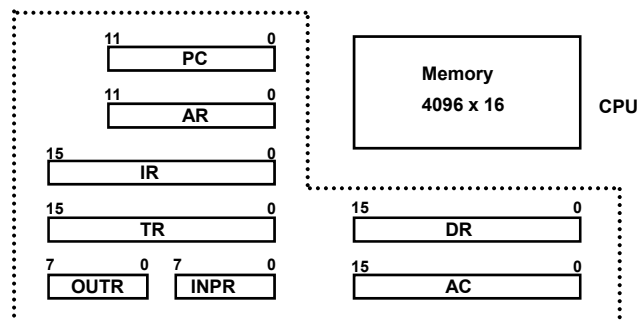
ثبات های پردازنده

✓ **TR:** در کامپیوتر پایه از یک ثبات برای نگهداری داده های میانی یا موقتی استفاده شده است که به این ثبات، ثبات موقتی (TR=Temporary Register) می گویند.

- کامپیوتر پایه یک مدل بسیار ساده ورودی/خروجی دارد.
- دستگاه های ورودی، کاراکترهای ۸ بیتی را به پردازنده می فرستند.
- پردازنده کاراکترهای ۸ بیتی را به دستگاه های خروجی می فرستد.
- **INPR:** ثبات ورودی Input Register (INPR) داده ۸ بیتی را که از دستگاه ورودی رسیده است نگه میدارد.
- **OUTR:** ثبات خروجی Output Register (OUTR) داده ۸ بیتی را که به دستگاه خروجی فرستاده می شود نگه میدارد.

13

ثبات های کامپیوتر پایه



14

لیست ثبات های کامپیوتر پایه

سمبل ثبات	تعداد بیت ها	نام ثبات	وظیفه
DR	16	Data Register	مقدار عملوند را نگه می دارد
AR	12	Address Register	آدرس عملوند را نگه می دارد
AC	16	Accumulator	ثبات همه منظوره
IR	16	Instruction Register	کد عملیات را نگه می دارد
PC	12	Program Counter	آدرس دستورالعمل را نگه می دارد
TR	16	Temporary Register	داده های موقتی را نگه می دارد
INPR	8	Input Register	کاراکتر ورودی را نگه می دارد
OUTR	8	Output Register	کاراکتر خروجی را نگه می دارد

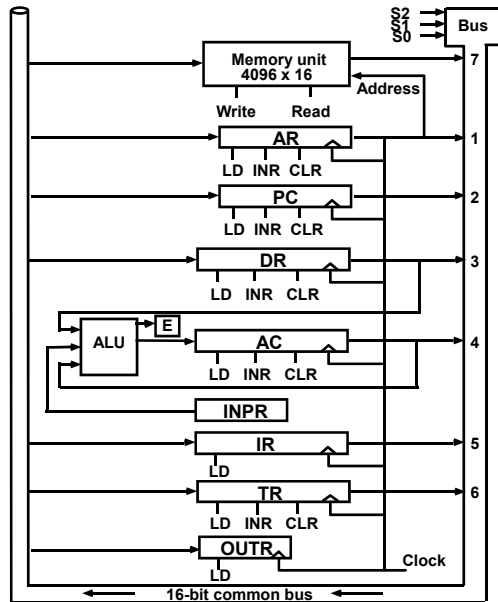
15

سیستم گذرگاه عمومی

- ثبات ها در کامپیوتر پایه با استفاده از گذرگاه به هم متصل شده اند.
- استفاده از گذرگاه نسبت به اتصال مستقیم ثبات ها به هم در سیم بندی صرفه جویی می کند.

16

سیستم گذرگاه عمومی



17

سیستم گذرگاه عمومی

- سه خط کنترل s_0 , s_1 و s_2 کنترل می کنند که مقدار کدام ثبات روی خطوط گذرگاه قرار گیرد.

s_2	s_1	s_0	Register
0	0	0	x
0	0	1	AR
0	1	0	PC
0	1	1	DR
1	0	0	AC
1	0	1	IR
1	1	0	TR
1	1	1	Memory

- برای اینکه محتویات از گذرگاه وارد حافظه یا هر یک از ثبات ها شود
- read حافظه یا load ثبات ها فعال می شود.
- وقتی ثبات های ۱۲ بیتی (AR و PC) روی گذرگاه اطلاعات قرار می دهند، ۴ بیت بارزش تر گذرگاه مقدار صفر می گیرد.

18

دستوالعمل های کامپیوتر پایه

- دستوالعمل های کامپیوتر پایه ۱۶ بیتی هستند.
- فرمت دستوالعمل های کامپیوتر پایه:

15	14	12	11	0
I				کد عمل
آدرس				کد عمل از 000 تا 110

(الف) دستوالعمل های حافظه ای

دستوالعمل های مراجعه به حافظه
(OP-code = 000 ~ 110)

15	12	11	0
0	1	1	1
عمل لباتی			

(ب) دستوالعمل های لبات

دستوالعمل های مراجعه به ثبات
(OP-code = 111, I = 0)

15	12	11	0
1	1	1	1
عمل I/O			

(ج) دستوالعمل های ورودی - خروجی

دستوالعمل های ورودی خروجی
(OP-code = 111, I = 1)

- واحد کنترل کامپیوتر نوع دستوالعمل را با توجه به بیت های مکان های ۱۲ تا ۱۵ دستوالعمل تشخیص می دهد.

19

جدول ۲-۵ دستوالعمل های کامپیوتر پایه

سمبل	کد شانزده شازدهی		شرح
	I = 0	I = 1	
AND	0xxx	8xxx	AND کردن کلمه حافظه با AC
ADD	1xxx	9xxx	جمع کردن کلمه حافظه با AC
LDA	2xxx	Axxx	بار کردن کلمه حافظه در AC
STA	3xxx	Bxxx	ذخیره محتوای AC در حافظه
BUN	4xxx	Cxxx	انشعاب نامشروط
BSA	5xxx	Dxxx	انشعاب و ضبط آدرس بازگشت
ISZ	6xxx	Exxx	افزایش و گذر در صورت نتیجه صفر
CLA	7800		پاک کردن AC
CLE	7400		پاک کردن E
CMA	7200		منتم کردن AC
CME	7100		منتم کردن E
CIR	7080		چرخش AC و E به راست
CIL	7040		چرخش AC و E به چپ
INC	7020		افزایش AC
SPA	7010		گذر از دستور بعدی اگر AC مثبت باشد
SNA	7008		گذر از دستور بعدی اگر AC منفی باشد
SZA	7004		گذر از دستور بعدی اگر AC صفر باشد
SZE	7002		گذر از دستور بعدی اگر E صفر باشد
HLT	7001		توقف کامپیوتر
INP	F800		دریافت کاراکتر و انتقال آن به AC
OUT	F400		برداشتن کاراکتر از AC و انتقال آن به خروجی
SKI	F200		گذر مبنی بر پرچم ورودی
SKO	F100		گذر مبنی بر پرچم خروجی
ION	F080		فعال کردن وقفه ها
IOF	F040		غیرفعال کردن وقفه ها

دستوالعمل های کامپیوتر پایه

دستور ۷

دستور ۱۲

دستور ۶

جمعاً ۲۵ دستور

20

کامل بودن مجموعه دستورالعملها

هر کامپیوتر باید مجموعه‌ای از دستورالعملها را داشته باشد که کاربر بتواند برای محاسبه هر تابعی که محاسبه پذیر بودن آن محرز است، برنامه‌ای به زبان ماشین بنویسد.

- نوع دستورالعملها:
 - حسابی، منطقی و جابجایی
 - ADD, CMA, INC, CIR, CIL, AND, CLA-
- تبادل اطلاعات
 - تبادل اطلاعات با حافظه و ثباتهای کامپیوتر
 - LDA, STA-
- دستورالعملهای کنترلی
 - دستورالعملهای کنترل برنامه و واریسی شرایط
 - BUN, BSA, ISZ-
- دستورالعملهای ورودی خروجی
 - ورودی خروجی
 - INP, OUT -

21

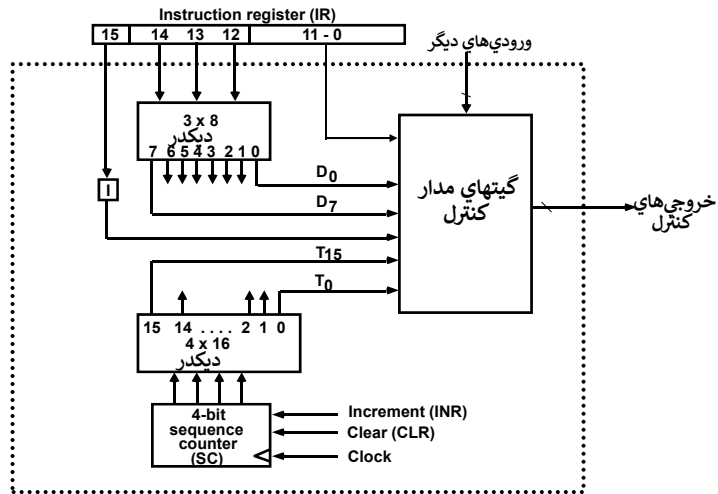
واحد کنترل

- زمانبندی همه ثبات ها در کامپیوتر پایه بوسیله یک مولد پالس ساعت کنترل می شود.
- پالس هاس ساعت حالت ثبات ها را تغییر نمی دهند مگر اینکه ثبات توسط یک سیگنال کنترل فعال شود.
- واحد کنترل به دو طریق قابل ساخت می باشد:
 - *کنترل سخت افزاری*
 - واحد کنترل از مدارهای ترکیبی و ترتیبی ساخته شده است که کار آنها تولید سیگنالهای کنترلی است.
 - *کنترل ریز برنامه نویسی شده*
 - یک حافظه کنترلی درپروسور وجود دارد که شامل ریز عملیاتی است که سیگنالهای کنترلی لازم را تولید می کند.
- ما یک واحد کنترلی سخت افزاری را برای کامپیوتر پایه ارائه و استفاده می کنیم.

22

زمان بندی و کنترل

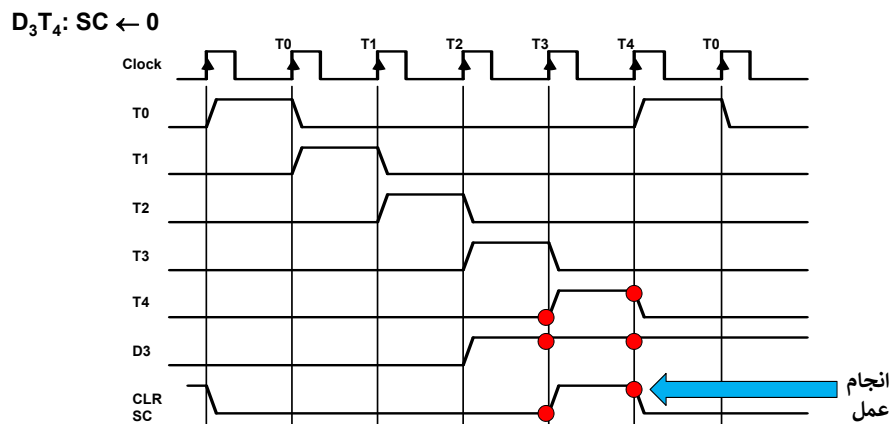
واحد کنترل در کامپیوتر پایه



23

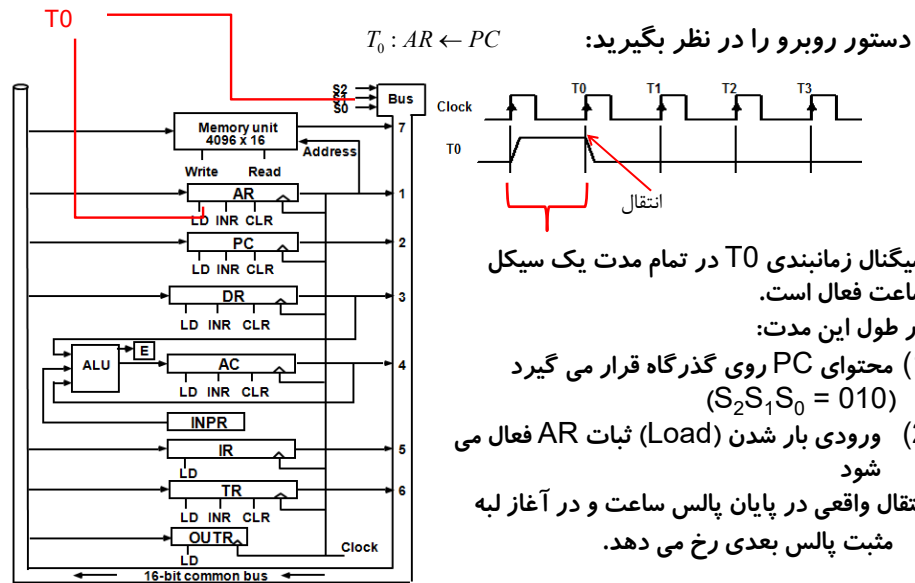
سیگنالهای زمان بندی

- بوسیله دنباله شمار ۴ بیتی و دیکدر ۱۶*۴ تولید می شود.
- SC می تواند افزایش یافته یا پاک شود.
- بعنوان مثال: T₀, T₁, T₂, T₃, T₄, T₀, T₁, ...
- فرض: در زمان T₄, SC پاک می شود اگر خروجی دیکدر D₃ فعال باشد.



24

مثال



25

سیکل دستورالعمل

• در کامپیوتر پایه هر سیکل دستورالعمل متشکل از فازهای زیر است:

- (1) برداشت دستور از حافظه (fetch)
- (2) دیکد کردن دستور
- (3) خواندن آدرس موثر
- (4) اجرای دستورالعمل

• فاز اول و دوم: برداشت و دیکد دستور

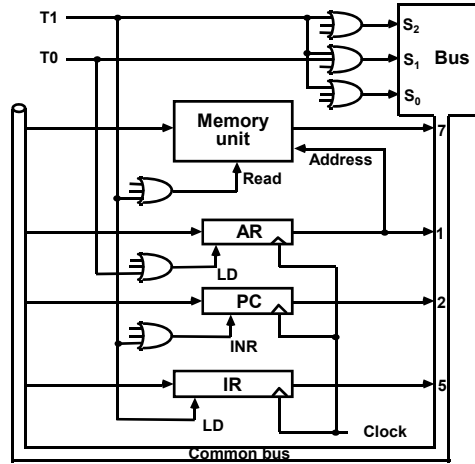
$T_0: AR \leftarrow PC$	$(S_0S_1S_2=010, T_0=1)$
$T_1: IR \leftarrow M[AR], PC \leftarrow PC + 1$	$(S_0S_1S_2=111, T_1=1)$
$T_2: D_0, \dots, D_7 \leftarrow \text{Decode } IR(12-14), AR \leftarrow IR(0-11), I \leftarrow IR(15)$	

26

برداشت و دیکد (واکشی و کدگشایی)

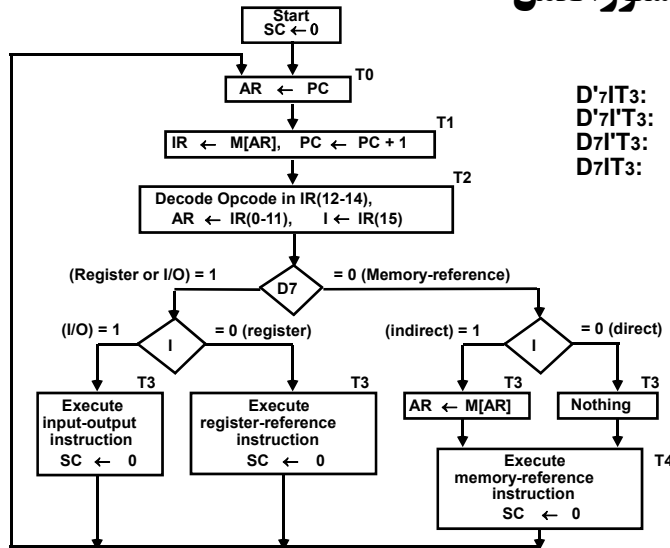
• Fetch and Decode

T0: $AR \leftarrow PC$ ($S_0S_1S_2=010, T0=1$)
 T1: $IR \leftarrow M[AR], PC \leftarrow PC + 1$ ($S0S1S2=111, T1=1$)
 T2: $D0, \dots, D7 \leftarrow \text{Decode } IR(12-14), AR \leftarrow IR(0-11), I \leftarrow IR(15)$



27

تعیین نوع دستورالعمل



D⁷I³: $AR \leftarrow M[AR]$
 D⁷I¹T³: هیچ کار
 D⁷I¹T³: اجرای دستورالعمل ثباتی
 D⁷I¹T³: اجرای دستورالعمل ورودی / خروجی

28

دستورالعمل های ثابتی

جدول ۳-۵ اجرای دستورالعمل های ثابتی

مشترک در همه دستورالعمل های ثابتی $D_7I_7T_3 = r$		
بیتی در IR (0-11) که عمل را مشخص می کند $IR(i) = B_i$		
r :	$SC \leftarrow 0$	پاک کردن SC
CLA rB_{11} :	$AC \leftarrow 0$	پاک کردن AC
CLE rB_{10} :	$E \leftarrow 0$	پاک کردن E
CMA rB_9 :	$AC \leftarrow \overline{AC}$	متعم کردن AC
CME rB_8 :	$E \leftarrow \overline{E}$	متعم کردن E
CIR rB_7 :	$AC \leftarrow shr AC, AC(15) \leftarrow E, E \leftarrow AC(0)$	شیفت چرخشی به راست
CIL rB_6 :	$AC \leftarrow shl AC, AC(0) \leftarrow E, E \leftarrow AC(15)$	شیفت چرخشی به چپ
INC rB_5 :	$AC \leftarrow AC + 1$	افزایش AC
SPA rB_4 :	If $(AC(15) = 0)$ then $(PC \leftarrow PC + 1)$	گذر در صورت مثبت بودن
SNA rB_3 :	If $(AC(15) = 1)$ then $(PC \leftarrow PC + 1)$	گذر در صورت منفی بودن
SZA rB_2 :	If $(AC = 0)$ then $(PC \leftarrow PC + 1)$	گذر در صورت صفر بودن AC
SZE rB_1 :	If $(E = 0)$ then $(PC \leftarrow PC + 1)$	گذر در صورت صفر بودن E
HLT rB_0 :	$S \leftarrow 0$	توقف کامپیوتر

29

دستورالعمل های ثابتی

• دستور AND کردن با محتوای AC

$D_0T_4: DR \leftarrow M[AR]$
 $D_0T_5: AC \leftarrow AC \wedge DR, SC \leftarrow 0$

• دستور ADD کردن با محتوای AC

$D_1T_4: DR \leftarrow M[AR]$
 $D_1T_5: AC \leftarrow AC + DR, E \leftarrow C_{out}, SC \leftarrow 0$

• بار کردن AC (LDA)

$D_2T_4: DR \leftarrow M[AR]$
 $D_2T_5: AC \leftarrow DR, SC \leftarrow 0$

دستورالعمل های ثباتی

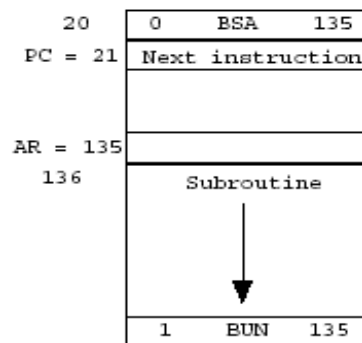
• ذخیره کردن AC در حافظه (STA)

$D_3T_4: DR \leftarrow M[AR] \leftarrow AC, SC \leftarrow 0$

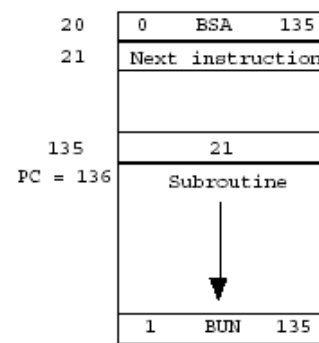
• انشعاب بدون شرط (BUN)

$D_4T_4: PC \leftarrow AR, SC \leftarrow 0$

دستورالعمل های ثباتی



Memory, PC, and AR at time T_4



Memory and PC after BSA execution

BSA:

$D_5T_4: M[AR] \leftarrow PC, AR \leftarrow AR + 1$
 $D_5T_5: PC \leftarrow AR, SC \leftarrow 0$

دستورالعمل های ثباتی

ISZ: Increment and Skip-if-Zero

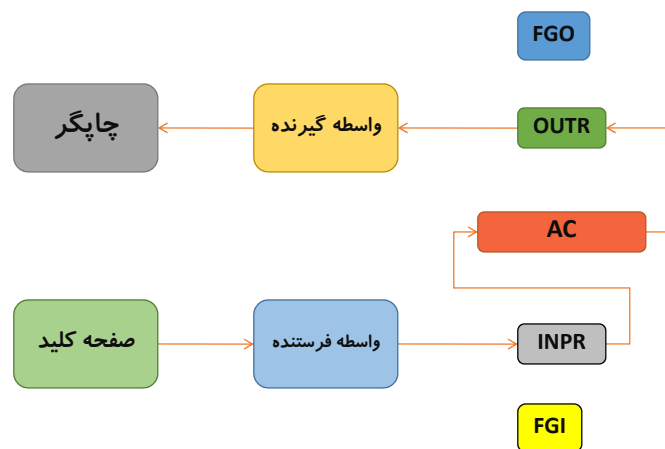
D_6T_4 : $DR \leftarrow M[AR]$

D_6T_5 : $DR \leftarrow DR + 1$

D_6T_4 : $M[AR] \leftarrow DR$, if $(DR = 0)$ then $(PC \leftarrow PC + 1)$, $SC \leftarrow 0$

دستورهای ورودی و خروجی

ثبات ها و فلیپ فلاپ های کامپیوتر واسطه ارتباطی سری پایانه ورودی-خروجی



دستورهای ورودی و خروجی

$D7 \text{ IT}_3 = p$ (مشترک در همه دستورالعملهای ورودی - خروجی)

$IR(i) = B_i$ بیتهای در (IR(0-11) که دستورالعمل را مشخص می کند.

INPR	pB₁₁:	$AC(0_7) \leftarrow INPR, FGI \leftarrow 0$	دریافت کاراکتر ورودی
OUTR	pB₁₀:	$OUTR \leftarrow AC(0_7), FGO \leftarrow 0$	ارسال کاراکتر خروجی
SKI	pB₉:	$IF (FGI=0) \text{ then } (PC \leftarrow PC + 1)$	گذر بر اساس پرچم ورودی
SKO	pB₈:	$IF (FGO=0) \text{ then } (PC \leftarrow PC + 1)$	گذر بر اساس پرچم خروجی
ION	pB₇:	$IEN \leftarrow 1$	روشن کردن فعال ساز وقفه
IOF	pB₆:	$IEN \leftarrow 0$	خاموش کردن فعال ساز وقفه

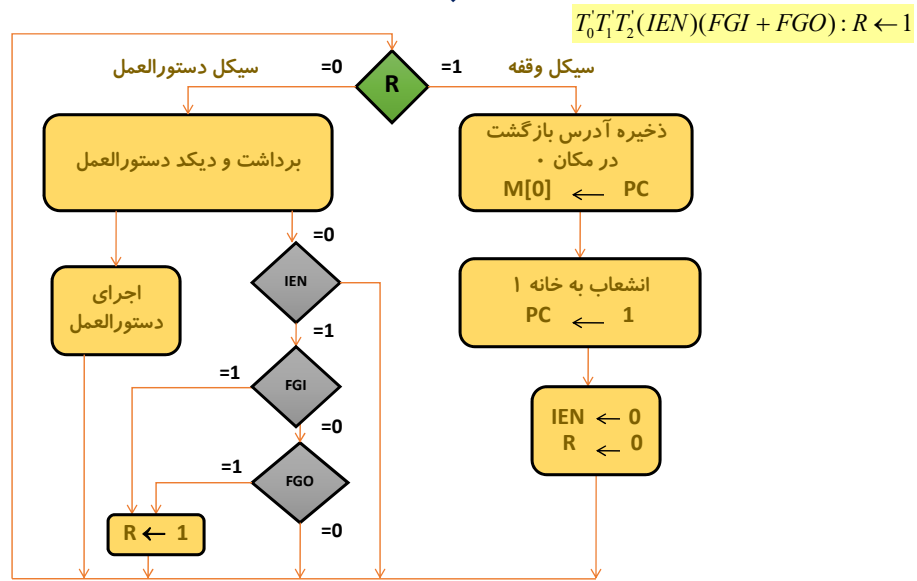
وقفه

RT₀: $AR \leftarrow 0, TR \leftarrow PC$

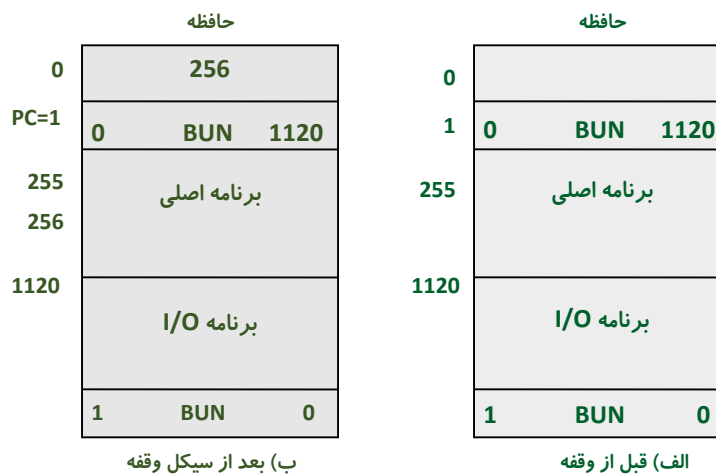
RT₁: $M[AR] \leftarrow TR, PC \leftarrow 0$

RT₂: $PC \leftarrow PC+1, IEN \leftarrow 0, R \leftarrow 0, SC \leftarrow 0$

فلوجارت سيكل وقفه



مثالی از سيكل وقفه



لیست کلی ریز عملیات‌ها

Fetch	R'T ₀ :	AR ← PC
	R'T ₁ :	IR ← M[AR], PC ← PC + 1
Decode	R'T ₂ :	D0, ..., D7 ← Decode IR(12 ~ 14), AR ← IR(0 ~ 11), I ← IR(15)
Indirect	D ₇ 'IT ₃ :	AR ← M[AR]
Interrupt	T ₀ 'T ₁ 'T ₂ '(IEN)(FGI + FGO):	R ← 1
	RT ₀ :	AR ← 0, TR ← PC
	RT ₁ :	M[AR] ← TR, PC ← 0
	RT ₂ :	PC ← PC + 1, IEN ← 0, R ← 0, SC ← 0
Memory-Reference		
AND	D ₀ T ₄ :	DR ← M[AR]
	D ₀ T ₅ :	AC ← AC ∧ DR, SC ← 0
ADD	D ₁ T ₄ :	DR ← M[AR]
	D ₁ T ₅ :	AC ← AC + DR, E ← C _{out} , SC ← 0
LDA	D ₂ T ₄ :	DR ← M[AR]
	D ₂ T ₅ :	AC ← DR, SC ← 0
STA	D ₃ T ₄ :	M[AR] ← AC, SC ← 0
BUN	D ₄ T ₄ :	PC ← AR, SC ← 0
BSA	D ₅ T ₄ :	M[AR] ← PC, AR ← AR + 1
	D ₅ T ₅ :	PC ← AR, SC ← 0
ISZ	D ₆ T ₄ :	DR ← M[AR]
	D ₆ T ₅ :	DR ← DR + 1
	D ₆ T ₆ :	M[AR] ← DR, if(DR=0) then (PC ← PC + 1), SC ← 0

لیست کلی ریز عملیات‌ها

Register-Reference	D ₇ 'IT ₃ = r	(Common to all register-reference instr)
	IR(i) = B _i	(i = 0,1,2, ..., 11)
	r:	SC ← 0
CLA	rB ₁₁ :	AC ← 0
CLE	rB ₁₀ :	E ← 0
CMA	rB ₉ :	AC ← AC'
CME	rB ₈ :	E ← E'
CIR	rB ₇ :	AC ← shr AC, AC(15) ← E, E ← AC(0)
CIL	rB ₆ :	AC ← shl AC, AC(0) ← E, E ← AC(15)
INC	rB ₅ :	AC ← AC + 1
SPA	rB ₄ :	If(AC(15) = 0) then (PC ← PC + 1)
SNA	rB ₃ :	If(AC(15) = 1) then (PC ← PC + 1)
SZA	rB ₂ :	If(AC = 0) then (PC ← PC + 1)
SZE	rB ₁ :	If(E=0) then (PC ← PC + 1)
HLT	rB ₀ :	S ← 0
Input-Output	D ₇ 'IT ₃ = p	(Common to all input-output instructions)
	IR(i) = B _i	(i = 6,7,8,9,10,11)
	p:	SC ← 0
INP	pB ₁₁ :	AC(0-7) ← INPR, FGI ← 0
OUT	pB ₁₀ :	OUTR ← AC(0-7), FGO ← 0
SKI	pB ₉ :	If(FGI=1) then (PC ← PC + 1)
SKO	pB ₈ :	If(FGO=1) then (PC ← PC + 1)
ION	pB ₇ :	IEN ← 1
IOF	pB ₆ :	IEN ← 0

— *Jordan Peterson*

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

<http://shafieian-education.ir>