

فرادرس

فراتر از یک کلاس درس
www.faradars.org

معماری کامپیوتر

درس سوم: طراحی کامپیوتر پایه (براساس کتاب مانو)

مدرس:

منوچهر بابایی

کارشناس ارشد مهندسی کامپیوتر (نرم افزار)

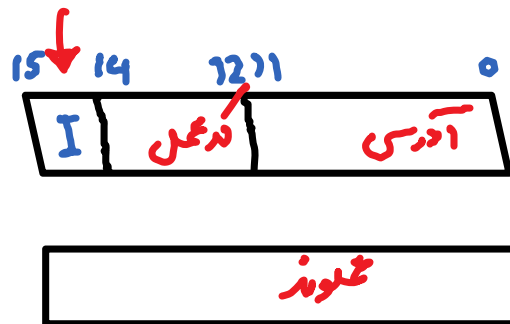
(دستورالعمل - کد دستورالعمل)

❖ دستورالعمل کامپیوتر یک کد دودویی است که رشته ای از ریزاعمال را برای کامپیوتر مشخص می کند.

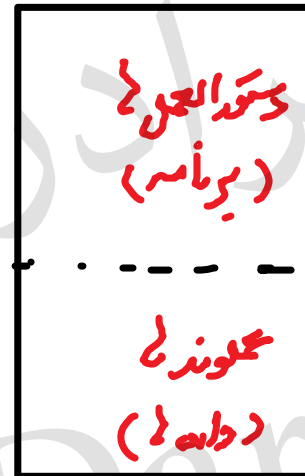
❖ کد دستورالعمل مجموعه ای بیت هاست که منجر به انجام یک عمل خاص می شود.

FaraDars.org

کجوه آدرس دهی



حافظه اصلی

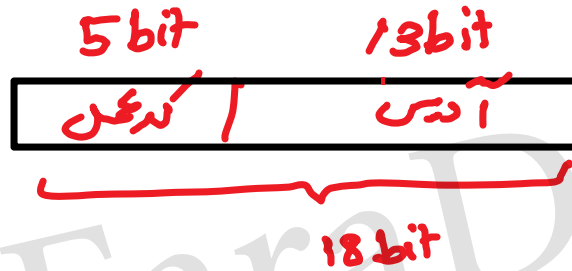


4A23_H
110010100011
بخش آدرس کد عمل

■ مثال (۱): در یک کامپیوتر حافظه 18×8192 داریم. در این کامپیوتر اگر فقط یک روش آدرس دهی وجود داشته باشد، چند نوع دستور میتوان تعریف نمود؟

13 bit = بخش آدرس

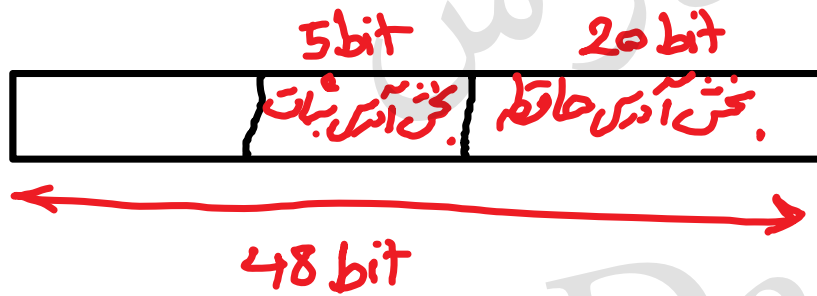
$$8192 = 8 \times 1024 = 2^3 \times 2^{10} = 2^{13}$$



$$2^5 = 32 : \text{تعداد عملیات}$$

پس 32 نوع دستور مختلف داریم.

■ مثال (۲): در یک کامپیوتر حافظه $2^{20} \times 48$ داریم. در این کامپیوتر اگر یکی از عملوندها در حافظه و دیگری در یکی از ۳۲ ثبات موجود، وجود داشته باشد. آنگاه چند نوع دستور میتوان تعریف نمود؟

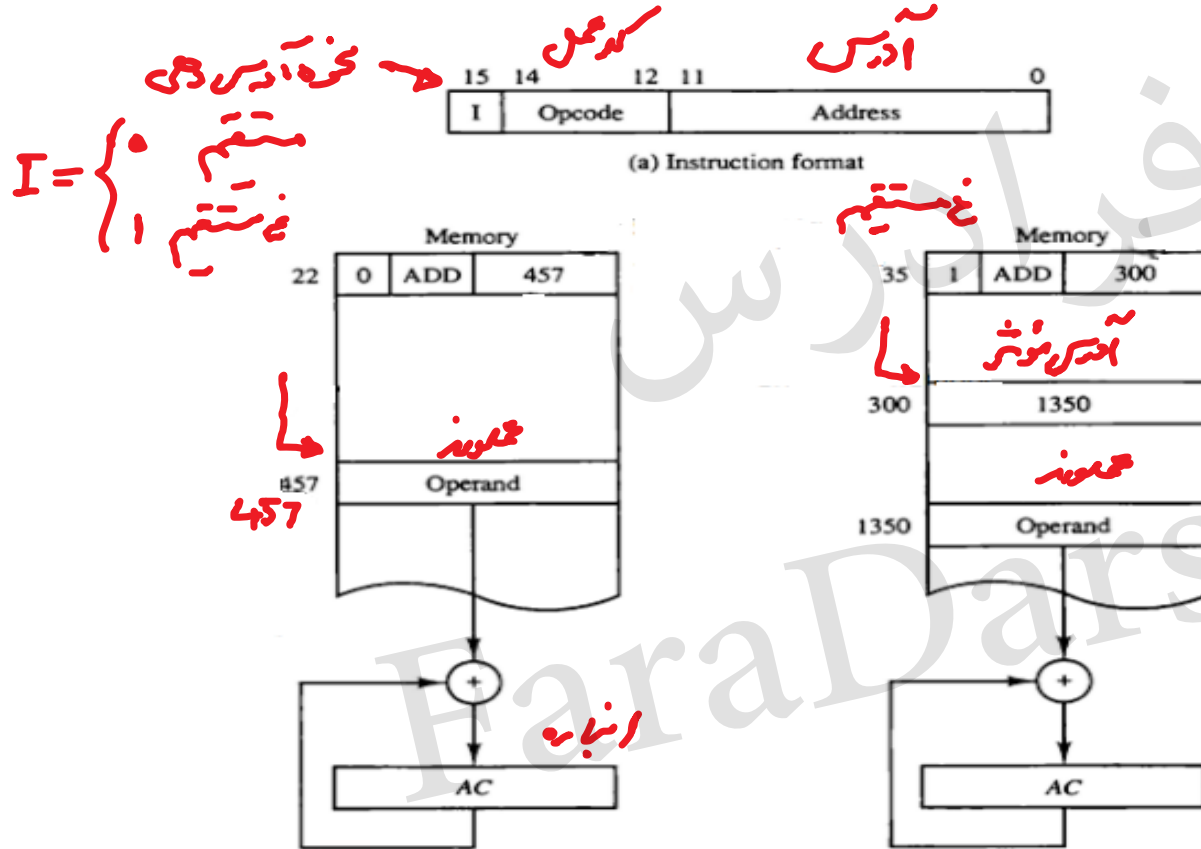


25 bit : آدرس حافظه

5 bit : آدرس ثبات

$$\text{کد عمل} : 48 - 25 = 23$$
$$2^{23} \text{ ، تعداد دستورات}$$

روش های آدرس دهی مستقیم و غیرمستقیم





مثال (۳):

اگر دستور مورد نظر ما در خانه ۳۰۰ حافظه باشد، عملوند و آدرس موثر را در هر دو روش آدرس دهی مستقیم و غیر مستقیم بیابید.

300	I ADD 532
	.
	.
	.
	.
	.
532	1A2B
	.
	.
	.
	.
1A2B	234C

روش آدرس دهی مستقیم
EA : 532
Op : 1A2B

روش آدرس دهی غیر مستقیم
EA : 1A2B
Op : 234C

(ثبات های کامپیوتر)

وظیفه	تعداد بیت	ثبات
نگه داری عملوند حافظه	16	DR
نگه داری آدرس حافظه	12	AR
ثبات پردازنده-نگه داری نتایج میانی	16	AC
نگه داری کد دستور	16	IR
نگه داری آدرس دستور	12	PC
نگه داری داده های موقت	16	TR
نگه داری کاراکتر ورودی	8	INPR
نگه داری کاراکتر خروجی	8	OUTR

■ مثال (۴): یک کامپیوتر دارای حافظه 16×16384 است. اگر هر دستور در دو خانه متوالی حافظه باشد. اندازه ثبات های IR، PC، AC، DR، AR را بیابید.

$$16384 = 16 \times 1024 = 2^4 \times 2^{10} = 2^{14}$$

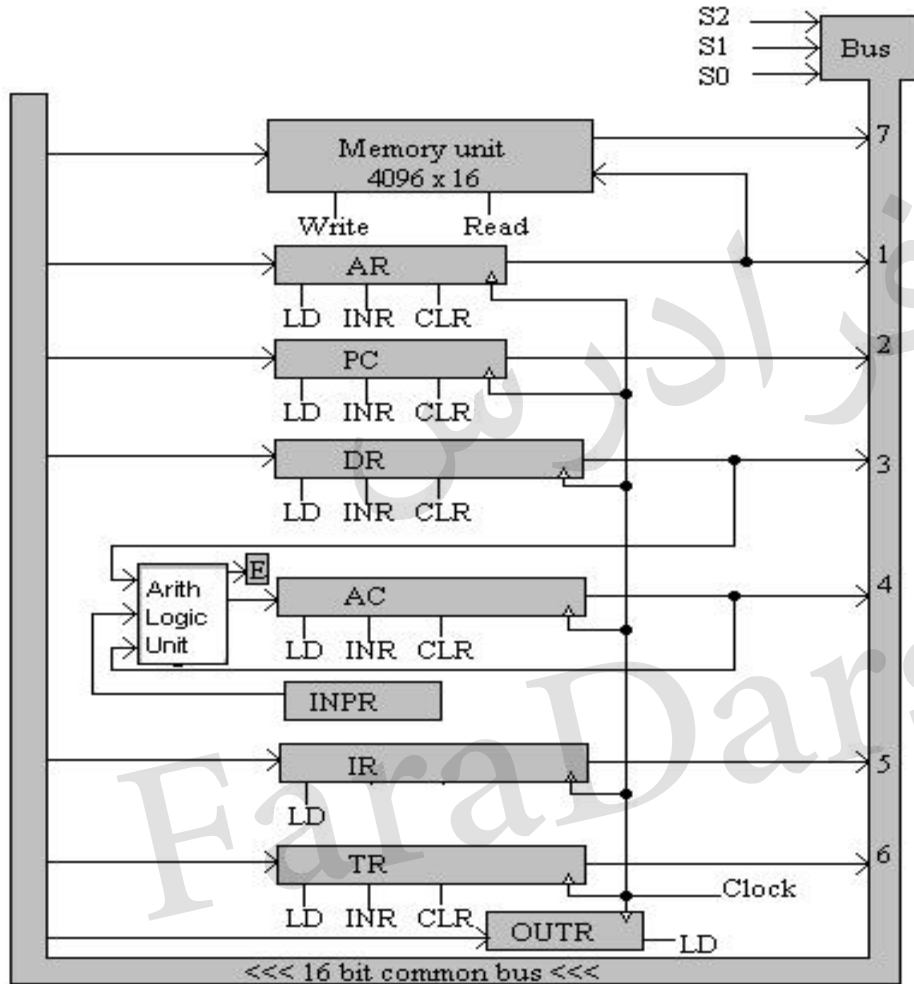
AR : 14 bit

IR : 32 bit

DR : 16 bit

AC : 16 bit

PC : 14 bit



S_2	S_1	S_0	آدرس
0	0	0	—
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	Mem

■ مثال (۵): چرا هر یک از اعمال زیر در سیستم گذرگاه مشترک اسلاید قبل، در یک پالس ساعت قابل اجرا نیستند؟ (تمرین کتاب موریس مانو)

$$a) IR \leftarrow M[PC]$$

$$b) AC \leftarrow AC + TR$$

$$c) DR \leftarrow DR + AC$$

$$d) AC \leftarrow TR$$

$$e) AC \leftarrow M[AR]$$

$$a) AR \leftarrow PC$$

$$IR \leftarrow M[AR]$$

$$b) DR \leftarrow TR$$

$$AC \leftarrow AC + DR$$

$$c) DR \leftarrow AC, AC \leftarrow DR$$

$$AC \leftarrow AC + DR$$

$$DR \leftarrow AC, AC \leftarrow DR$$

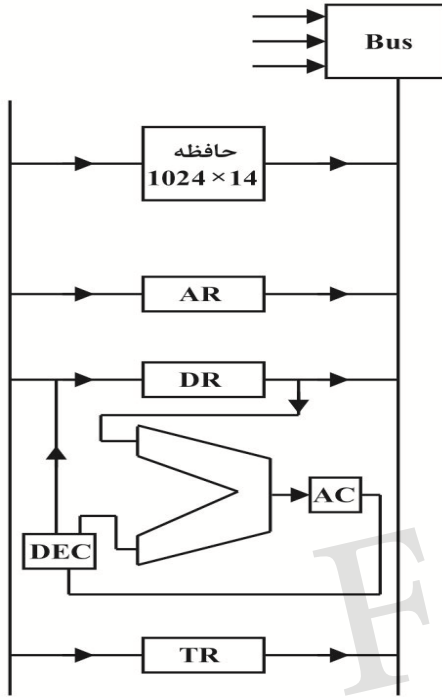
$$d) DR \leftarrow TR$$

$$AC \leftarrow DR$$

$$e) DR \leftarrow M[AR]$$

$$AC \leftarrow DR$$

مثال (۶): شکل زیر ارتباط ثباتها را در یک کامپیوتر با حافظه و واحد محاسبه و منطق نشان می دهد. برای انجام عمل $M[AR] \leftarrow TR + M[AR]$ به چند پالس ساعت نیاز داریم؟



(1) $DR \leftarrow M[AR]$

(2) $AC \leftarrow DR$

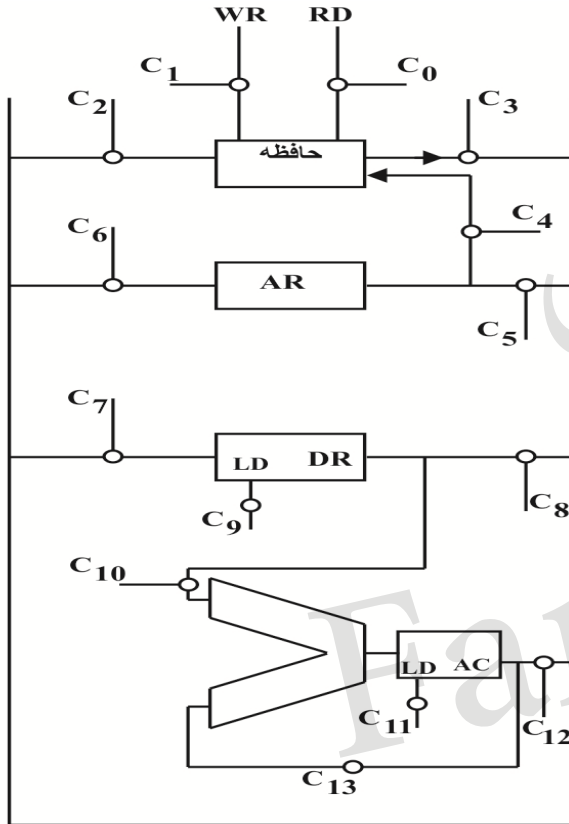
(3) $DR \leftarrow TR$

(4) $AC \leftarrow AC + DR$

(5) $DR \leftarrow AC$

(6) $M[AR] \leftarrow DR$

مثال (۷): در سیستم گذرگاه زیر برای انجام عمل $AC \leftarrow AC + M[AR]$ کدام سیگنال ها باید فعال شوند؟



$C_4 - C_0 - C_3$

$M[AR] \rightarrow \text{Bus}$

$C_7 - C_6$

$DR \leftarrow M[AR]$

$C_{10} - C_{13} - C_{11}$

$AC \leftarrow AC + DR \equiv AC \leftarrow AC + M[AR]$

مثال (۸): در مورد سیستم گذرگاه اسلاید قبل کدام موارد صحیح هستند؟

- (۱) انتقال همزمان AC به DR و برعکس امکانپذیر است ✓
- (۲) برای انتقال AC درون M[AR] به ۳ پالس ساعت نیاز است ✗
- (۳) برای متمم کردن AC به پالس ساعتی نیاز نیست ✗ یک پالس ساعت نیاز است ✓
- (۴) برای محاسبه $AC \leftarrow AC + DR$ فقط یک پالس ساعت نیاز است. ✓

(دستورالعمل های کامپیوتر)

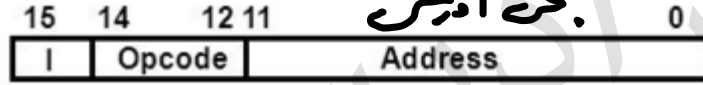
رجوع به حافظه

انواع رجیسترها

Memory-Reference Instructions (OP-code = 000 ~ 110)

خودآدرس دهی

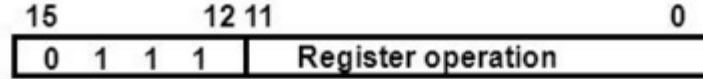
بخش آدرس



حالت های
نشانی
(۱۲-۱۱) { ۰۰۰-۱۱۰
 { ۰۱۱
 { ۱۱۱

رجوع به ثبت

Register-Reference Instructions (OP-code = 111, I = 0)



ورودی خروجی

Input-Output Instructions (OP-code = 111, I = 1)



(دستورالعمل های حافظه ای)

I

0 ۰۰۰

۱ ۰۰۰

op

۰۰۰

۰۰۱

۰۱۰

۰۱۱

۱۰۰

۱۰۱

۱۱۰

Symbol	Hex Code		Description
	I = 0	I = 1	
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load AC from memory
STA	3xxx	Bxxx	Store content of AC into memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero

(دستورالعمل های ثباتی)

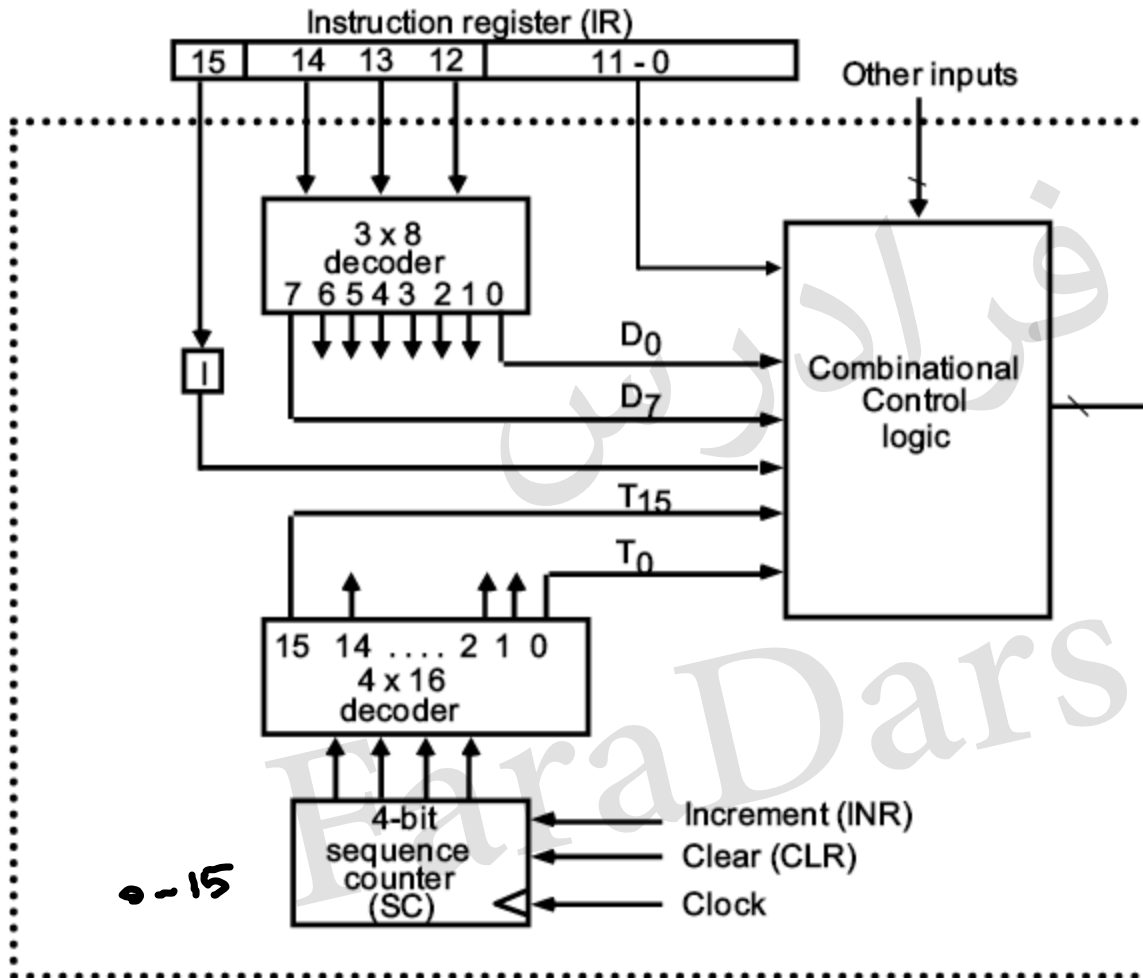
۵۱۱)

Symbol	Hex Code	Description
	I = 0 2xxx	
CLA	7800	Clear AC
CLE	7400	Clear E
CMA	7200	Complement AC
CME	7100	Complement E
CIR	7080	Circulate right AC and E
CIL	7040	Circulate left AC and E
INC	7020	Increment AC
SPA	7010	Skip next instr. if AC is positive
SNA	7008	Skip next instr. if AC is negative
SZA	7004	Skip next instr. if AC is zero
SZE	7002	Skip next instr. if E is zero
HLT	7001	Halt computer

(دستورالعمل های ورودی-خروجی)

 $111 \equiv F$ $p c \leftarrow p c + 1 \quad \text{آنگاه} \quad (FGI = 1) \quad \text{آن}$

Symbol	Hex Code	Description
	FFFF $I = 1$	
INP	F800	Input character to AC
OUT	F400	Output character from AC
SKI	F200	Skip on input flag
SKO	F100	Skip on output flag
ION	F080	Interrupt on $\longrightarrow IEN \leftarrow 1$
IOF	F040	Interrupt off $\longrightarrow IEN \leftarrow 0$



$D_0 - D_8$: حافظه‌های

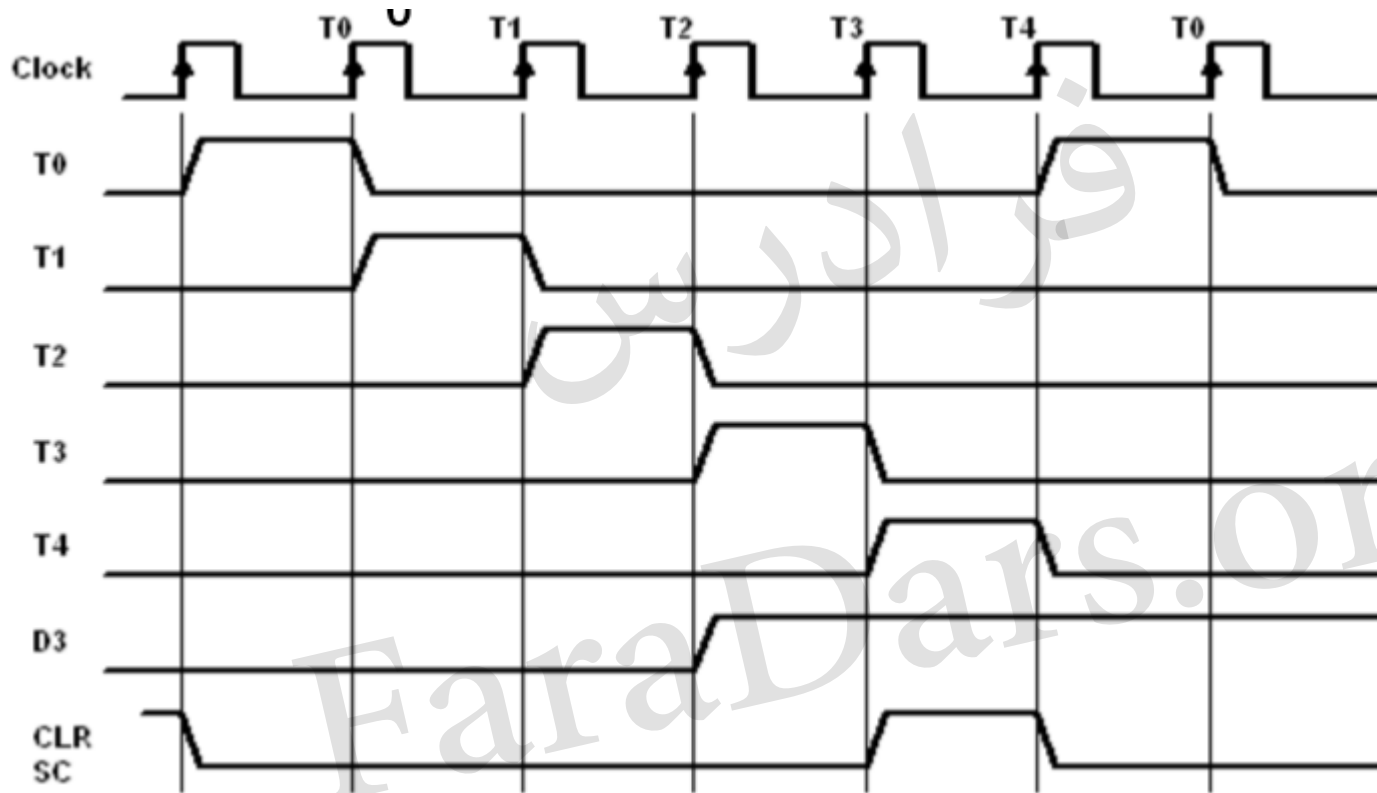
D_7 : I/O - ثابت

$D_7 I$: I/O

$D_7 \bar{I}$: ثابت

Control signals

0-15

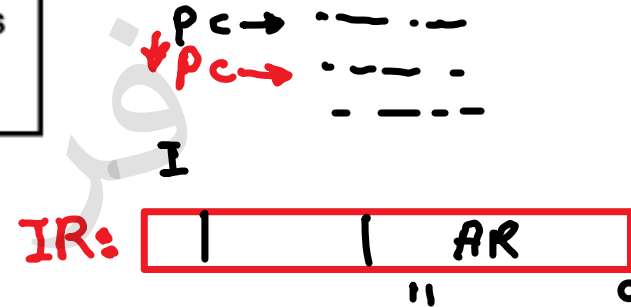
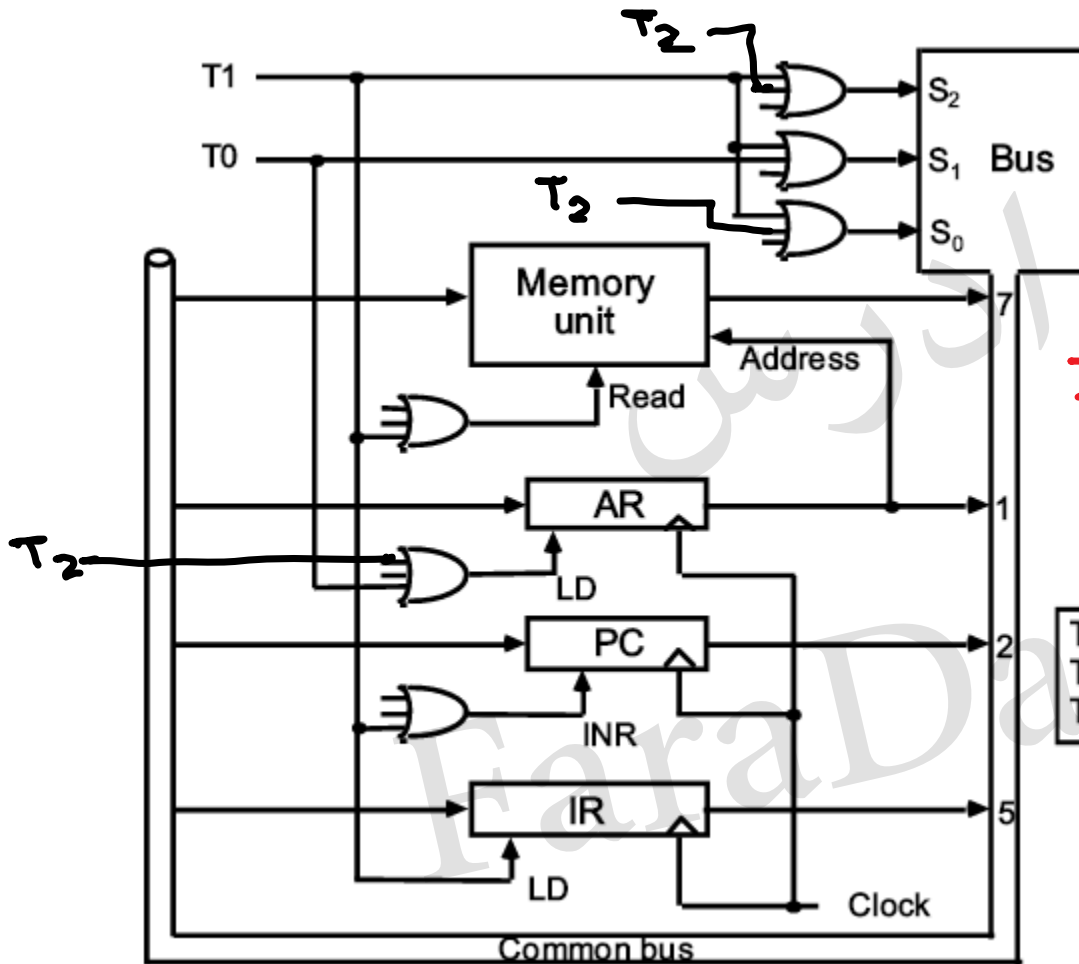


(سیکل دستورالعمل)

- ۱- برداشت یک دستور از حافظه (Fetch)
- ۲- دیکد کردن دستور (Decode)
- ۳- خواندن آدرس موثر (Effective Address)
- ۴- اجرای دستورالعمل (Execute)

FaraDars.org

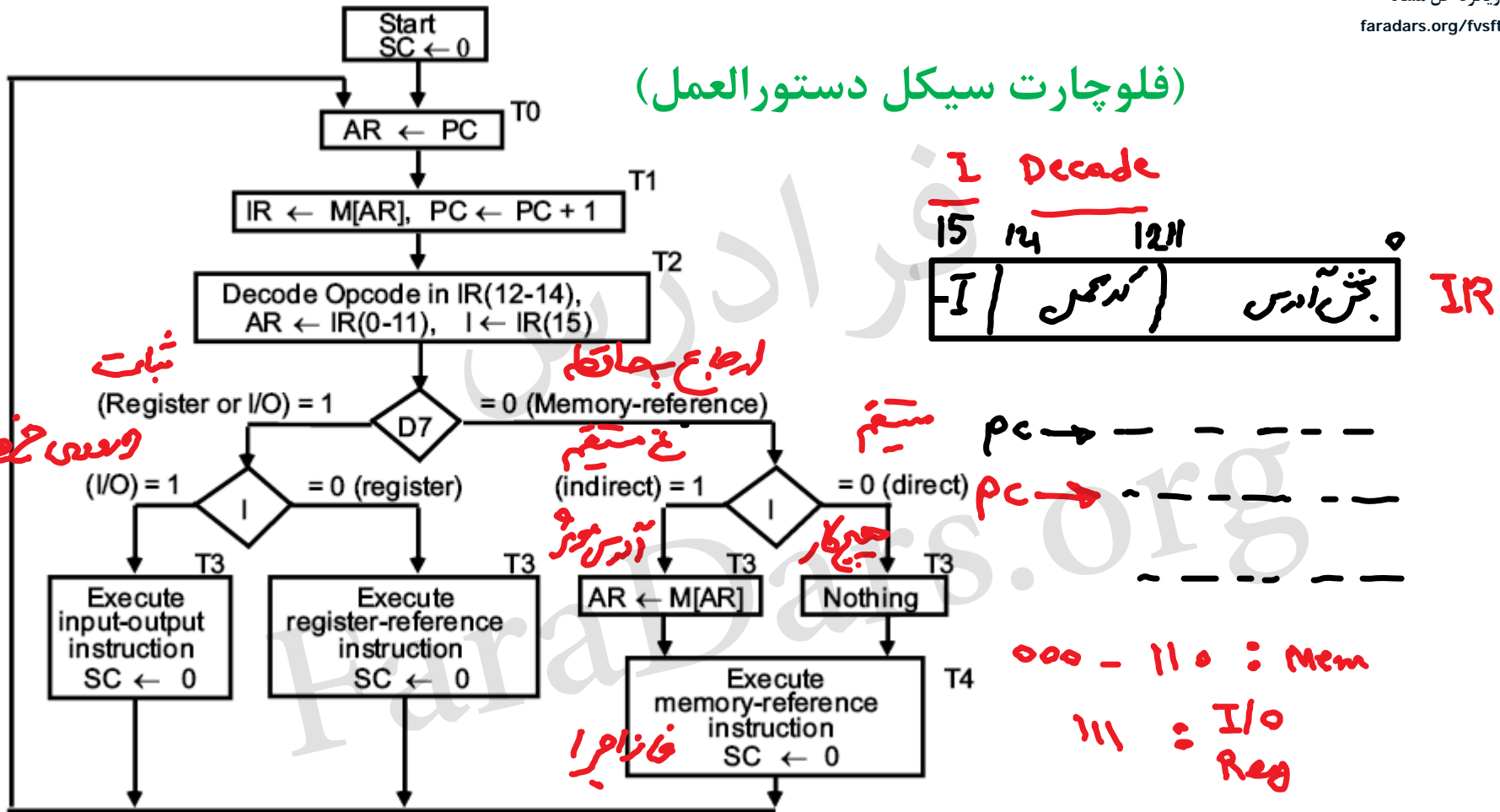
$AR \leftarrow IR$



(فاز برداشت و دیکد)

T0: $AR \leftarrow PC$ ($S_0S_1S_2=010, T0=1$)
T1: $IR \leftarrow M[AR], PC \leftarrow PC + 1$ ($S_0S_1S_2=111, T1=1$)
T2: $D_0, \dots, D_7 \leftarrow \text{Decode } IR(12-14), AR \leftarrow IR(0-11), I \leftarrow IR(15)$

(فلوچارت سیکل دستورالعمل)



(تشریح دستورالعمل های حافظه ای)

۰۰۰ → D_۰

AND to AC

D_۰T_۴: DR ← M[AR]

D_۰T_۵: AC ← AC ∧ DR, SC ← 0

Read operand
AND with AC

ADD to AC

۰۰۱

D_۱T_۴: DR ← M[AR]

D_۱T_۵: AC ← AC + DR, E ← C_{out}, SC ← 0

Read operand
Add to AC and store carry in E



FaraDars.org

(تشریح دستورالعمل های حافظه ای...)

۰۱۵ → D₂

LDA: Load to AC

D₂T₄: DR ← M[AR]

D₂T₅: AC ← DR, SC ← 0

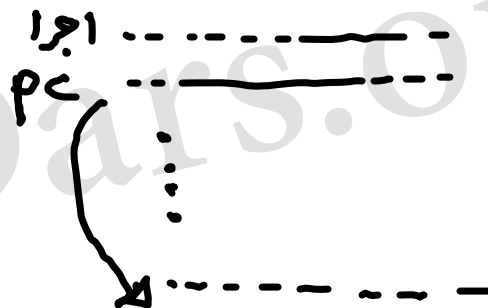
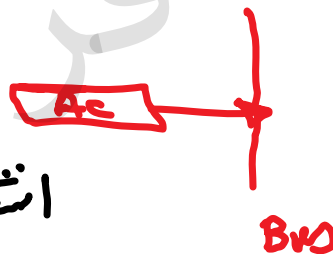
STA: Store AC

D₃T₄: M[AR] ← AC, SC ← 0

BUN: Branch Unconditionally

D₄T₄: PC ← AR, SC ← 0

انتخاب غیر شرطی



۱۰۱

BSA:

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

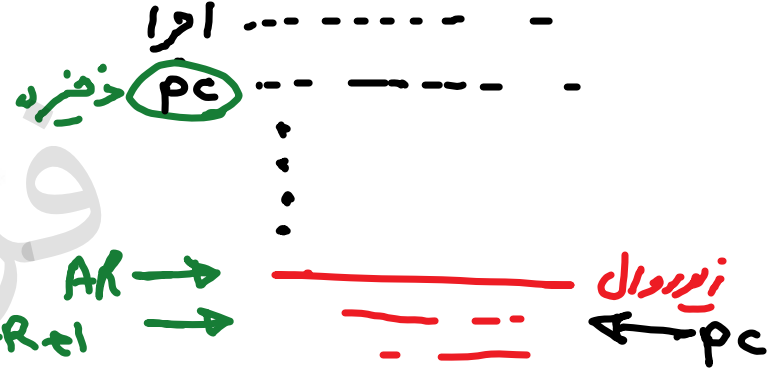
$D_5T_5: PC \leftarrow AR, SC \leftarrow 0$

ISZ: Increment and Skip-if-Zero

$D_6T_4: DR \leftarrow M[AR]$

$D_6T_5: DR \leftarrow DR + 1$

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



(مثال: اجرای دستور BSA)

وضعیت حافظه قبل از اجرای دستور

Memory, PC, AR at time T4

20	0	BSA	135
PC = 21	Next instruction		
AR = 135			
136	Subroutine		
	↓		
	1	BUN	135

زیربروال

اولین دستور عمل
زیربروال

Memory, PC after execution

20	0	BSA	135
21	Next instruction		
135	21		
PC = 136	Subroutine		
	↓		
	1	BUN	135

$PC \leftarrow PC$

$AR \leftarrow 135 + 1$

$PC \leftarrow AR$

EA = 21

غستیم

■ مثال (۱۰): محتویات PC در کامپیوتر پایه 3AF است. محتویات AC هم 7EC3 است. محتویات آدرس 3AF

برابر با 932E میباشد. محتویات حافظه در آدرس 32E برابر 09AC و در آدرس 9AC هم 8B9F است؟

الف) دستورالعملی که بعدا دریافت و اجرا می شود را بیابید؟

ب) محتویات ثبات های PC, AR, DR, AC, TR را بیابید. مقادیر SC, IE در انتهای سیکل دستور را مشخص کنید

Fetch : $AR \leftarrow PC : AR \leftarrow 3AF$

$IR \leftarrow M[3AF] : IR \leftarrow 932E, PC \leftarrow PC + 1$

9	32E
---	-----

حل: ADD و $AR \leftarrow M[32E] : AR \leftarrow 09AC$
 خواندن آدرس: 1001
 کد عمل

الف- جمع (ADD)

مخمويز : 8B9F

$$\begin{array}{r} \\ \\ + \\ \hline \end{array}$$

$$\rightarrow AC \leftarrow 0A62, E = 1$$

$$DR \leftarrow M[AR]$$

$$AC \leftarrow AC + DR$$

$$E \leftarrow Cout$$

$$AC = 0A62, DR \leftarrow 8B9F \cdot PC \leftarrow 3AF + 1 = 3B0, AR = 9AC \quad (ب)$$

$$Sc \leftarrow 0, E = 1$$

■ **مثال (۱۱):** فرض کنید حافظه‌ای داریم که ۱۰۲۴ کلمه ۱۶ بیتی دارد. در این ماشین هر دستور دو کلمه از حافظه را اشغال کرده است. یکی از دستورات این ماشین STA است که مقدار AC را در حافظه ذخیره می‌کند. مراحل واکشی، ترجمه و اجرای این دستور چند دسترسی به حافظه نیاز دارد؟

$$AR \leftarrow PC$$

$$\underline{IR(0-15)} \leftarrow M[AR], \quad PC \leftarrow PC + 1, \quad AR \leftarrow AR + 1$$

$$\underline{IR(16-32)} \leftarrow M[AR]$$

بدون مراجعه به حافظه
آدرس موثر را داریم
→ نحوه آدرس دهی مستقیم

x برای آدرس موثر باید به حافظه مراجعه کنیم
→ نحوه آدرس دهی غیرمستقیم

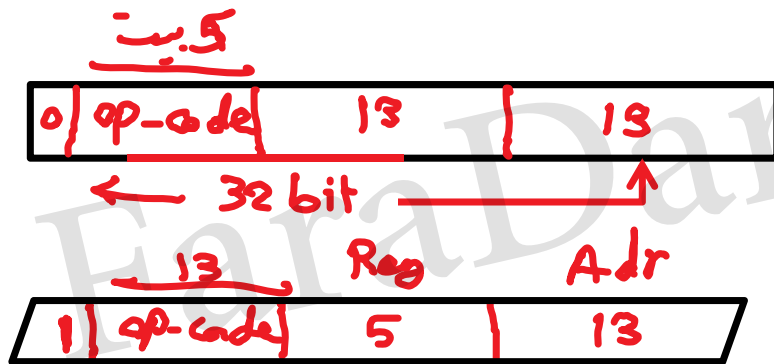
$$M[AR] \leftarrow AC \quad \text{: نحوه اجرا}$$

■ مثال (۱۲): یک سیستم حافظه 32×8192 وجود دارد. و در هر کلمه از حافظه یک دستور وجود دارد. در این ماشین دستورات مراجعه به حافظه با دو عملوند در حافظه و دستورات با یک عملوند در حافظه و یک عملوند در یکی از ۳۲ ثبات وجود دارند. حداکثر دستورات قابل تعریف در این ماشین را بیابید.

$$8192 = 2^{13} \quad 8192 = 8 \times 1024 = 2^3 \times 2^{10}$$

دفعه‌های حافظه ثباتی

نوع رستور { ۰ : دفعه‌های حافظه ثباتی
۱ : دفعه‌های حافظه ثباتی



مقدار دستورات نوع اول : $2^3 = 32$

تعداد دستورات نوع دوم : $2^{13} = 8192$

کل دستورات : $32 + 8192 = 8224$

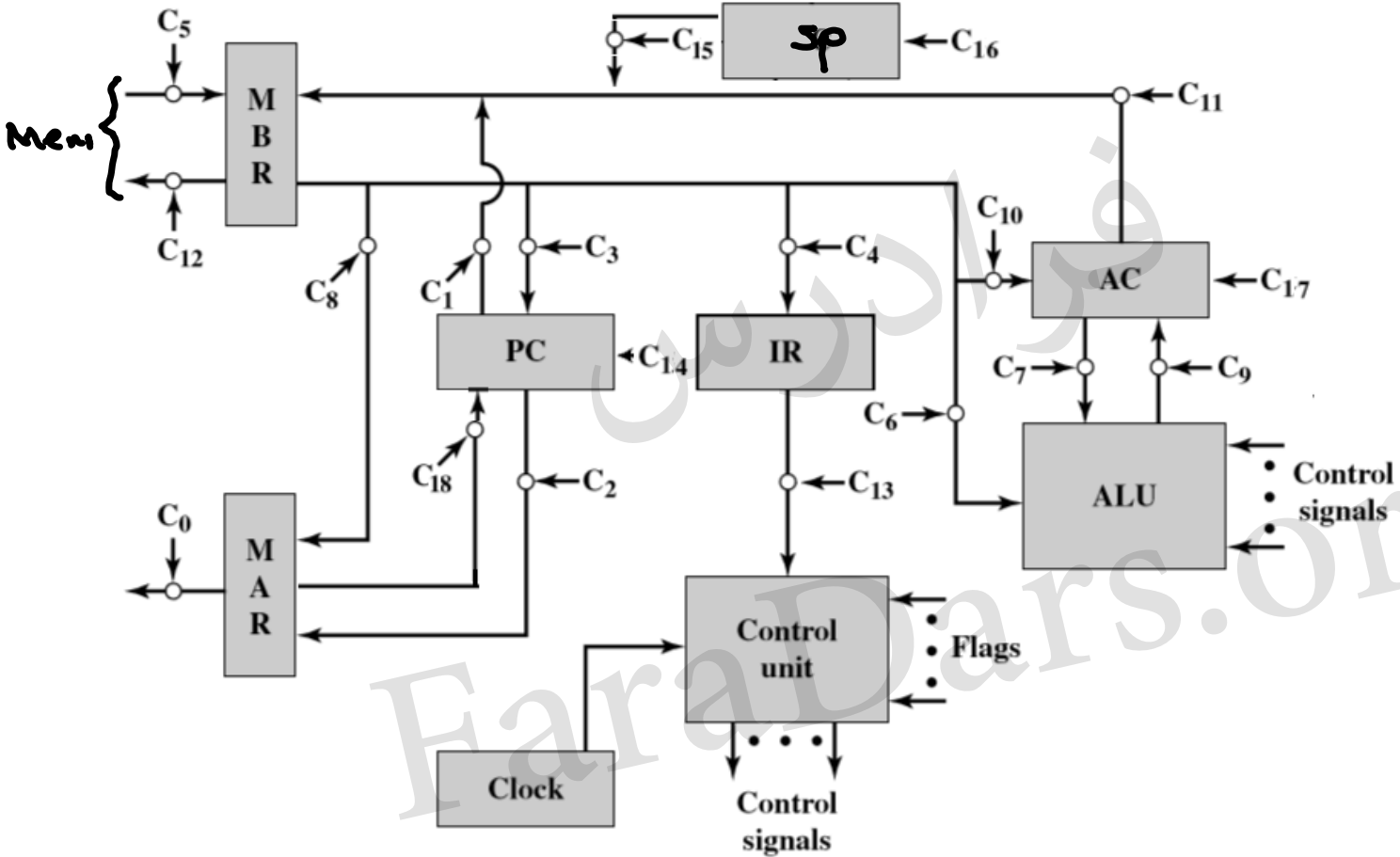
■ مثال (۱۳): در معماری اسلاید بعدی، مراحل واکنشی، دیکود، تبدیل آدرس غیرمستقیم به مستقیم و اجرای

دستورالعمل های

$ISZ \{M[MAR] \leftarrow M[MAR] + 1, \text{if } (M[MAR] == 0) \text{ Then } PC \leftarrow MAR\}$

$BSA \{Push\ PC, PC \leftarrow MAR\}$

را نوشته و مشخص کنید در هر کلاک کدام سیگنال ها باید فعال باشند توجه کنید که SP به عنصر بالای پشته اشاره می کند، و کف پشته دارای آدرس کمتر است. در این معماری برای ذخیره آدرس برگشت از پشته استفاده می شود. هرگونه تبادل داده با حافظه از طریق ثبات MBR و آدرس حافظه توسط ثبات MAR تامین می شود.



signal	op
C_{14}	INC (PC)
C_{17}	INC (AC)
C_{16}	INC (SP)

$AR \leftarrow PC$ $IR \leftarrow M[AR], PC \leftarrow PC + 1$ $AR \leftarrow IR(0-11)$ Fetch : $T_0 : C_2 \leftarrow 1 : MAR \leftarrow PC$ $T_1 : C_5 \leftarrow 1 : MBR \leftarrow M[MAR], C_4 \leftarrow 1 : PC \leftarrow PC + 1$
 $C_0 \leftarrow 1$ $T_2 : C_4 \leftarrow 1 : IR \leftarrow MBR, C_8 \leftarrow 1, MAR \leftarrow MBR$

Decode :

 $T_3 : C_{13} \leftarrow 1 : \text{Decode}(IR(12-4)), I \leftarrow IR(15)$

Eff Address:

 $T_4 I : C_5 \leftarrow 1 : MBR \leftarrow M[MAR]$
 $C_0 \leftarrow 1$ $T_5 I : C_8 \leftarrow 1 : MAR \leftarrow MBR$

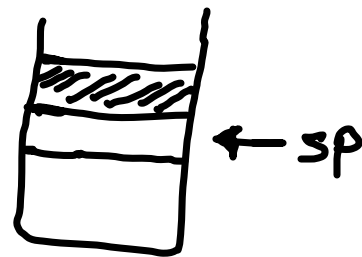
BSA: $C_{10} \leftarrow 1 : SP \leftarrow SP + 1$

$C_1 \leftarrow 1 : MBR \leftarrow PC$

$C_0 \leftarrow 1, C_2 \leftarrow 1 : M[SP] \leftarrow MBR \wedge push PC$

$C_8 \leftarrow 1 : PC \leftarrow MAR$

$C_{15} \leftarrow 1 : MBR \leftarrow SP$
 $C_8 \leftarrow 1 : MAR \leftarrow MBR$



ISZ: $C_0 \leftarrow 1, C_5 \leftarrow 1 : MBR \leftarrow M[MAR]$

$C_{10} \leftarrow 1 : AC \leftarrow MBR$

$C_7 \leftarrow 1 : AC \leftarrow AC + 1$

$C_{11} \leftarrow 1 : MBR \leftarrow AC$

$C_0 \leftarrow 1, C_{12} \leftarrow 1 : M[MAR] \leftarrow MBR$

$C_7 \leftarrow 1 : OR(AC) : C_{15} \leftarrow 1, PC \leftarrow MAR$

$M[MAR] + 1$

I 111
57

(دستورات ورودی-خروجی)

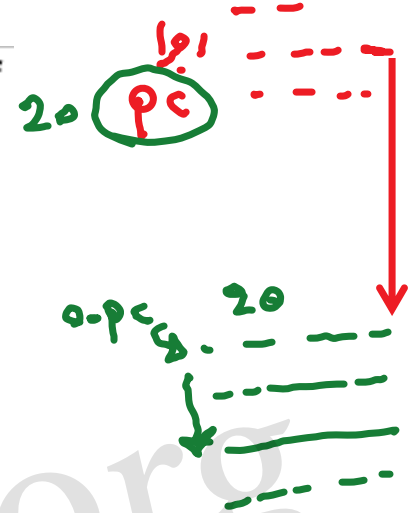
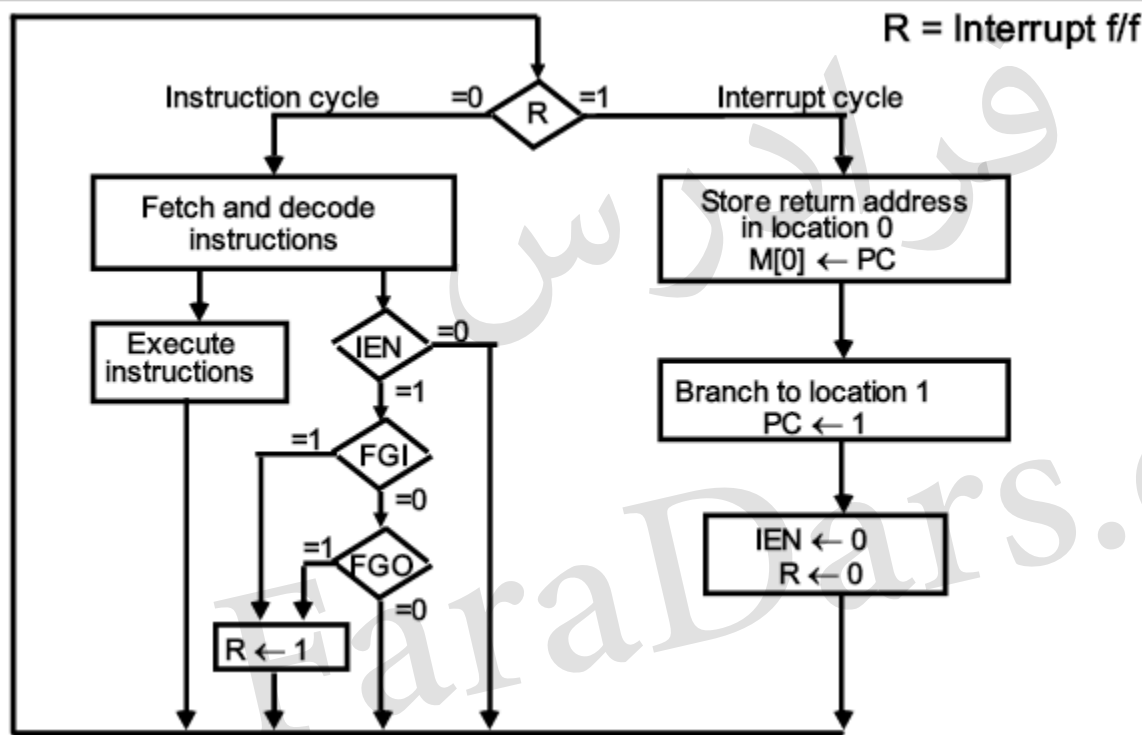


$$D_7IT_3 = p$$

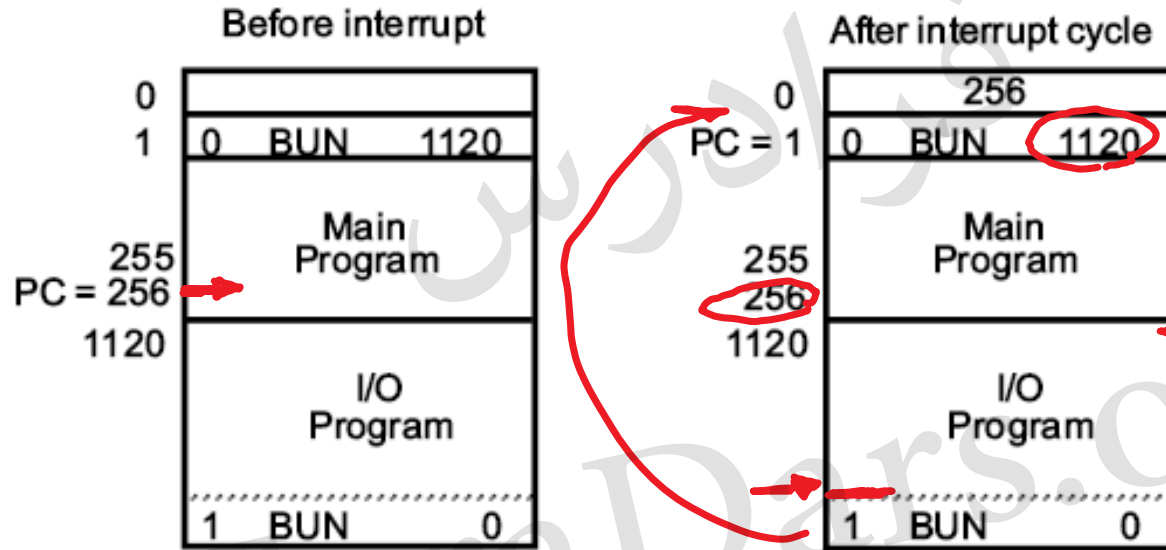
$$IR(i) = B_i, i = 6, \dots, 11$$

INP	p:	$SC \leftarrow 0$	Clear SC
OUT	pB_{11} :	$AC(0-7) \leftarrow INPR, FGI \leftarrow 0$	Input char. to AC
SKI	pB_{10} :	$OUTR \leftarrow AC(0-7), FGO \leftarrow 0$	Output char. from AC
SKO	pB_9 :	if($FGI = 1$) then ($PC \leftarrow PC + 1$)	Skip on input flag
ION	pB_8 :	if($FGO = 1$) then ($PC \leftarrow PC + 1$)	Skip on output flag
IOF	pB_7 :	$IEN \leftarrow 1$	Interrupt enable on
	pB_6 :	$IEN \leftarrow 0$	Interrupt enable off

فلوچارت سیکل وقفه و دستور



(سرویس دهی به وقفه)



اولین دستور
از بدنه وقفه

(سیکل وقفه)

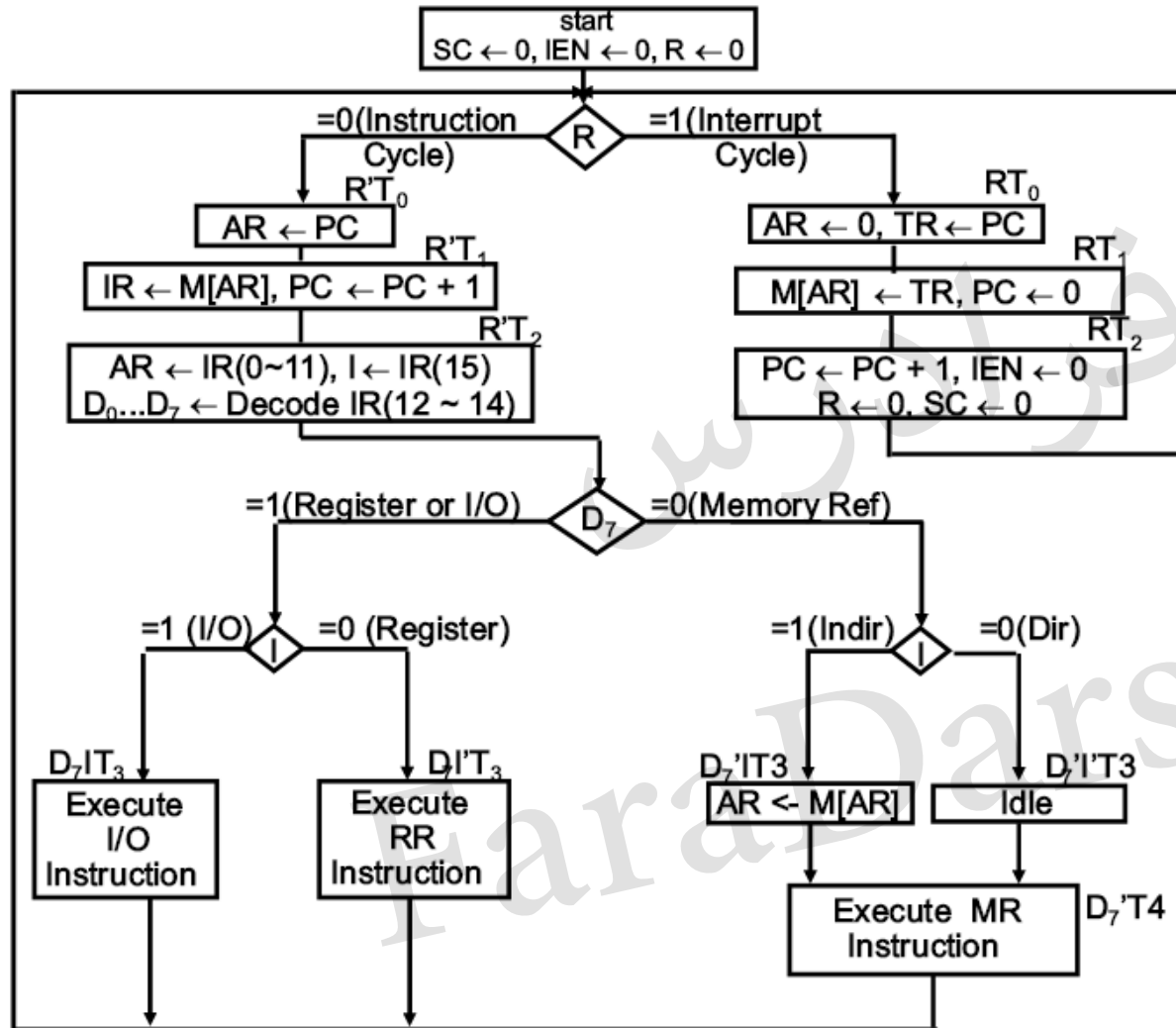
$T_0'T_1'T_2'$ (IEN)(FGI + FGO): $R \leftarrow 1$

RT_0 : $AR \leftarrow 0, TR \leftarrow PC$

RT_1 : $M[AR] \leftarrow TR, PC \leftarrow 0$

RT_2 : $PC \leftarrow PC + 1, IEN \leftarrow 0, R \leftarrow 0, SC \leftarrow 0$

FaraDars.org



دسترز = 0 R
وقتن = 1 R

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 Interrupt	D ₇ 'IT ₃ :	AR ← M[AR]
	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

Register-Reference

	$D_7I'T_3 = r$	(Common to all register-reference instr)
	$IR(i) = B_i$	($i = 0, 1, 2, \dots, 11$)
	$r:$	$SC \leftarrow 0$
CLA	$rB_{11}:$	$AC \leftarrow 0$
CLE	$rB_{10}:$	$E \leftarrow 0$
CMA	$rB_9:$	$AC \leftarrow AC'$
CME	$rB_8:$	$E \leftarrow E'$
CIR	$rB_7:$	$AC \leftarrow shr\ AC, AC(15) \leftarrow E, E \leftarrow AC(0)$
CIL	$rB_6:$	$AC \leftarrow shl\ AC, AC(0) \leftarrow E, E \leftarrow AC(15)$
INC	$rB_5:$	$AC \leftarrow AC + 1$
SPA	$rB_4:$	If($AC(15) = 0$) then ($PC \leftarrow PC + 1$)
SNA	$rB_3:$	If($AC(15) = 1$) then ($PC \leftarrow PC + 1$)
SZA	$rB_2:$	If($AC = 0$) then ($PC \leftarrow PC + 1$)
SZE	$rB_1:$	If($E = 0$) then ($PC \leftarrow PC + 1$)
HLT	$rB_0:$	$S \leftarrow 0$

Input-Output

	$D_7IT_3 = p$	(Common to all input-output instructions)
	$IR(i) = B_i$	($i = 6, 7, 8, 9, 10, 11$)
	$p:$	$SC \leftarrow 0$
INP	$pB_{11}:$	$AC(0-7) \leftarrow INPR, FGI \leftarrow 0$
OUT	$pB_{10}:$	$OUTR \leftarrow AC(0-7), FGO \leftarrow 0$
SKI	$pB_9:$	If($FGI = 1$) then ($PC \leftarrow PC + 1$)
SKO	$pB_8:$	If($FGO = 1$) then ($PC \leftarrow PC + 1$)
ION	$pB_7:$	$IEN \leftarrow 1$
IOF	$pB_6:$	$IEN \leftarrow 0$

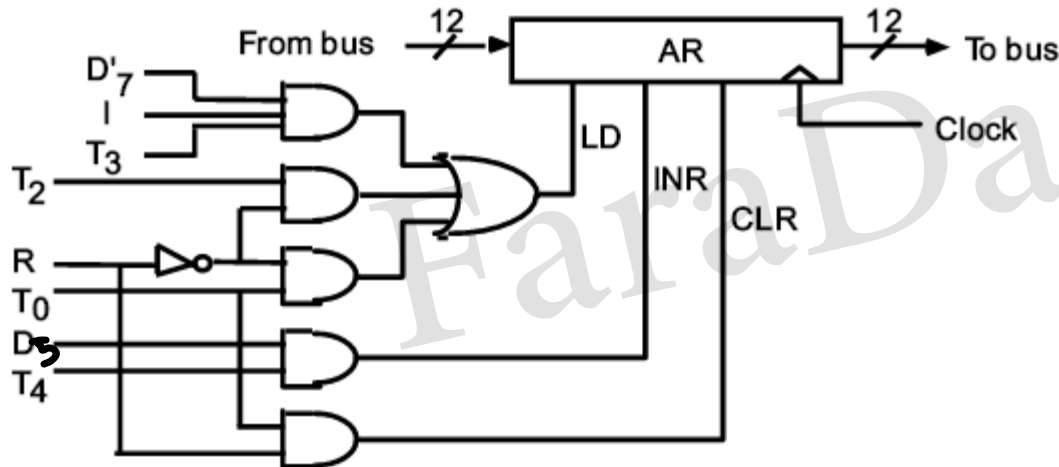
$R'T_0$:	$AR \leftarrow PC$	$LD(AR)$
$R'T_2$:	$AR \leftarrow IR(0-11)$	$LD(AR)$
D'_7IT_3 :	$AR \leftarrow M[AR]$	$LD(AR)$
RT_0 :	$AR \leftarrow 0$	$CLR(AR)$
D_5T_4 :	$AR \leftarrow AR + 1$	$INR(AR)$



$$LD(AR) = R'T_0 + R'T_2 + D'_7IT_3$$

$$CLR(AR) = RT_0$$

$$INR(AR) = D_5T_4$$



pB_7 : $IEN \leftarrow 1$ (I/O Instruction)

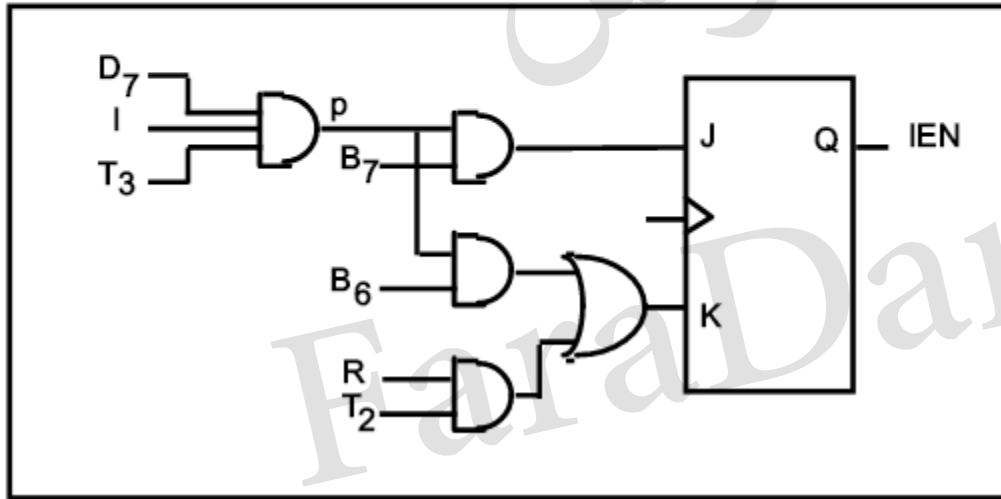
pB_6 : $IEN \leftarrow 0$ (I/O Instruction)

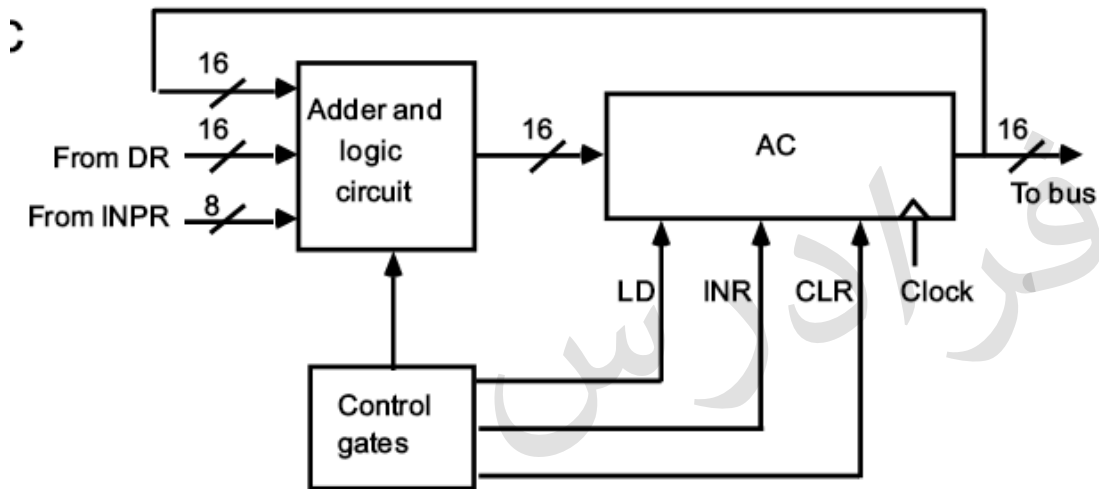
RT_2 : $IEN \leftarrow 0$ (Interrupt)

$p = D_7IT_3$ (Input/Output Instruction)

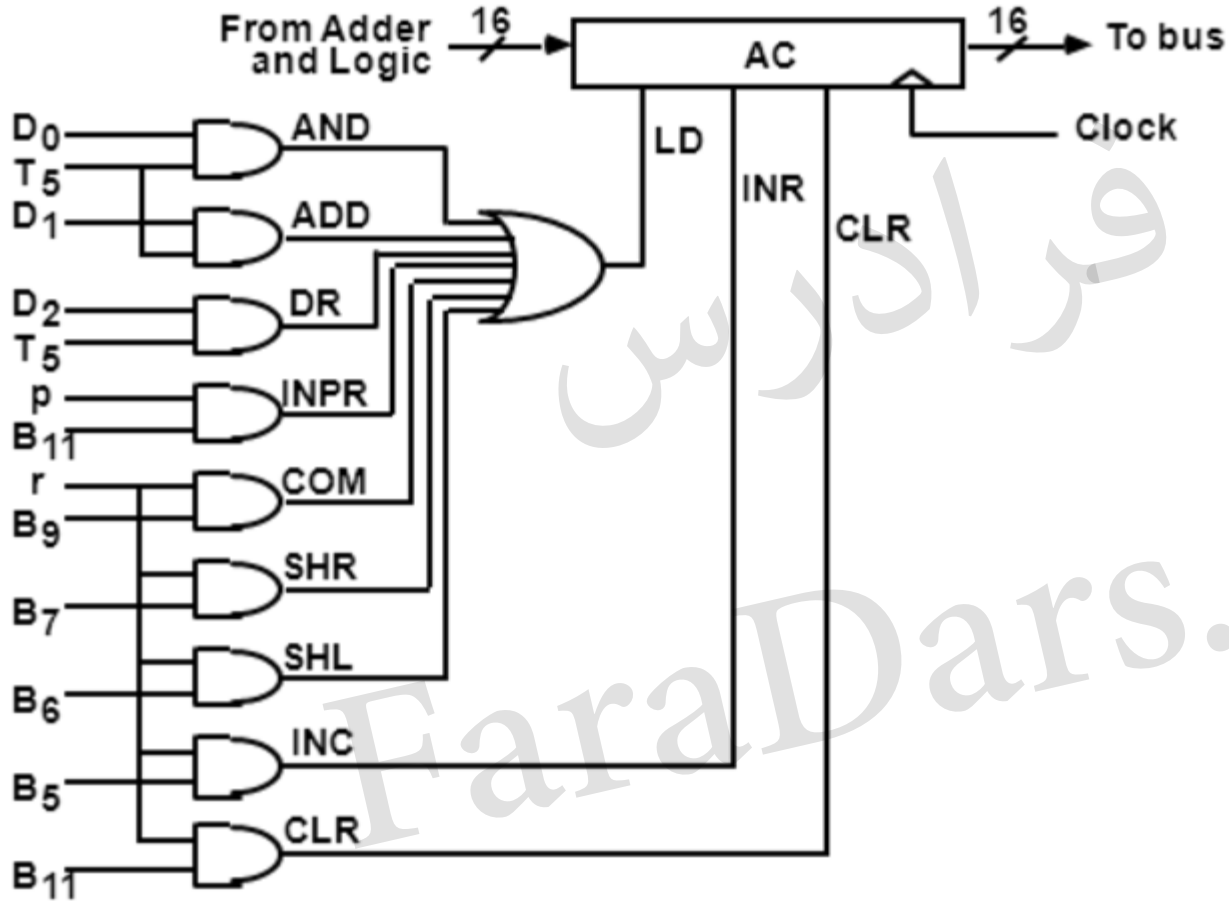
j	k	$Q(t+1)$
0	1	0
1	0	1

$$Q(t+1) = jQ'(t) + k'Q(t)$$





$D_0T_5:$	$AC \leftarrow AC \wedge DR$	AND with DR
$D_1T_5:$	$AC \leftarrow AC + DR$	Add with DR
$D_2T_5:$	$AC \leftarrow DR$	Transfer from DR
$pB_{11}:$	$AC(0-7) \leftarrow INPR$	Transfer from INPR
$rB_9:$	$AC \leftarrow AC'$	Complement
$rB_7:$	$AC \leftarrow shr AC, AC(15) \leftarrow E$	Shift right
$rB_6:$	$AC \leftarrow shl AC, AC(0) \leftarrow E$	Shift left
$rB_{11}:$	$AC \leftarrow 0$	Clear
$rB_5:$	$AC \leftarrow AC + 1$	Increment



این اسلاید ها بر مبنای نکات مطرح شده در فرادرس
«آموزش معماری کامپیوتر با رویکرد حل مساله»
تهیه شده است.

برای کسب اطلاعات بیشتر در مورد این آموزش به لینک زیر مراجعه نمایید
faradars.org/fvsft112