

باسمہ تعالیٰ
درس معماری کامپیوٹر
فصل 5 کتاب مانو
سازمان و طراحی کامپیوٹر پایه
دکتر محسن سریانی

سازمان ذخیره شده STORED PROGRAM ORGANIZATION

15 12 11 0

کد عمل	آدرس
--------	------

قالب دستور العمل

15 0

عملوند دودویی

دستور العمل ها (برنامه)

عملوند ها (داده ها)

ثبت پردازندہ (انباره یا AC)

شبکہ کامپیوٹر پاکستان

11 0

PC

11 0

AR

15 0

IR

15 0

TR

15 0

DR

7 0 7 0

OUTR

INPR

15 0

AC

حافظہ

4096 کلمہ 16 بیتی

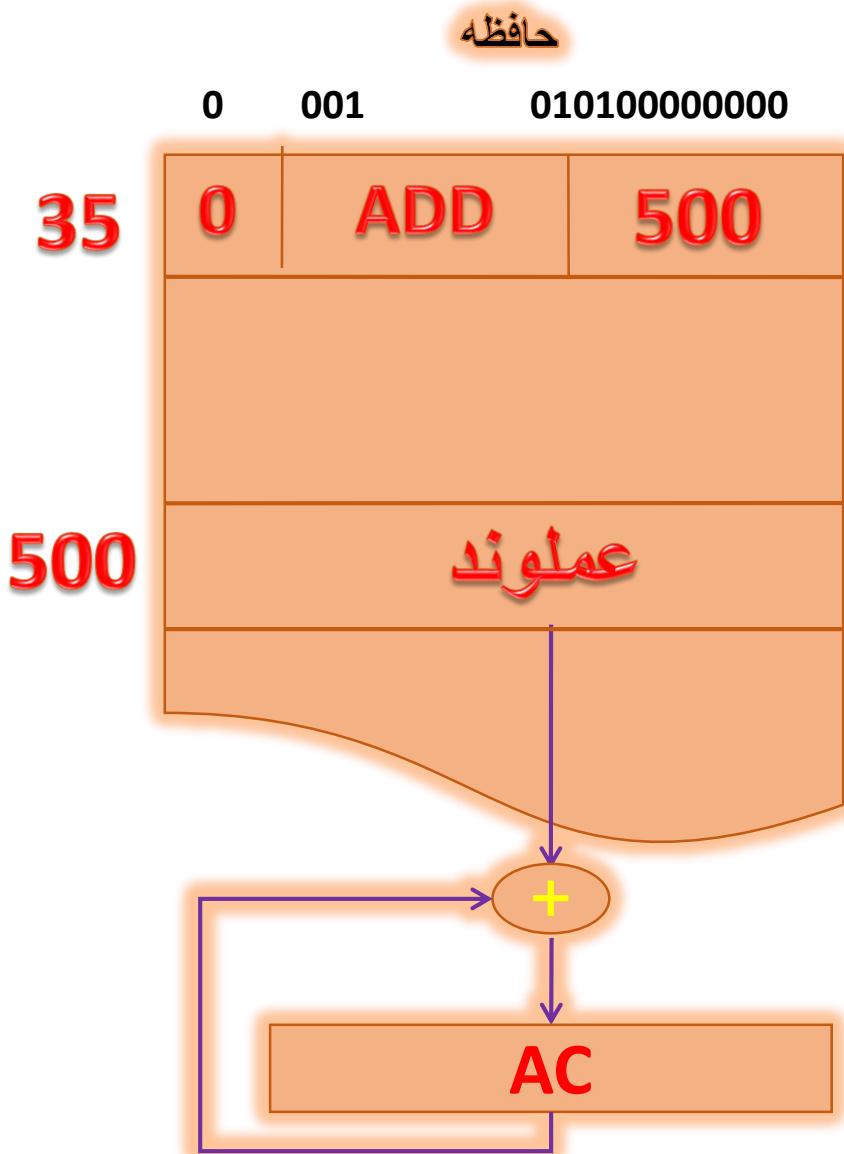
ثبتات های کامپیووتر پایه

- ثبات انباره 16 بیتی است و نتیجه کلیه عملیات حسابی و منطقی و شیفت در این ثبات قرار می گیرد. **AC**
- ثبات شمارنده برنامه 12 بیتی است و در هر لحظه از اجرای برنامه، شماره سطrix از حافظه که دستور العمل بعدی در آن قرار دارد را نشان می دهد. **(Program Counter) PC**
- ثبات داده برای نگهداری عملوند ها (داده ها) استفاده می شود و این ثبات 16 بیتی است. **DR**
- ثبات آدرس 12 بیتی است و آدرس حافظه را نگه می دارد. این ثبات به طور مستقیم به پایه های آدرس حافظه متصل است. **(Address Register) AR**
- هر دستور العملی که در حافظه نوشته شده باشد برای اجرا ابتدا به ثبات IR منتقل و سپس دید می شود. این ثبات 16 بیتی است. **Instruction Register) IR**
- این ثبات 16 بیتی است و برای نگهداری موقت داده ها استفاده می شود. **(Temporary Register) TR**
- این ثبات 8 بیتی است و اطلاعاتی که از صفحه کلید دریافت می شود به این ثبات منتقل و سپس از طریق ALU در AC قرار می گیرد. **:Input Register**
- ثبات خروجی **Output Register**: این ثبات 8 بیتی است و برای ارسال اطلاعات از AC به وسیله خارجی (چاپگر یا مانیتور) استفاده می شود.

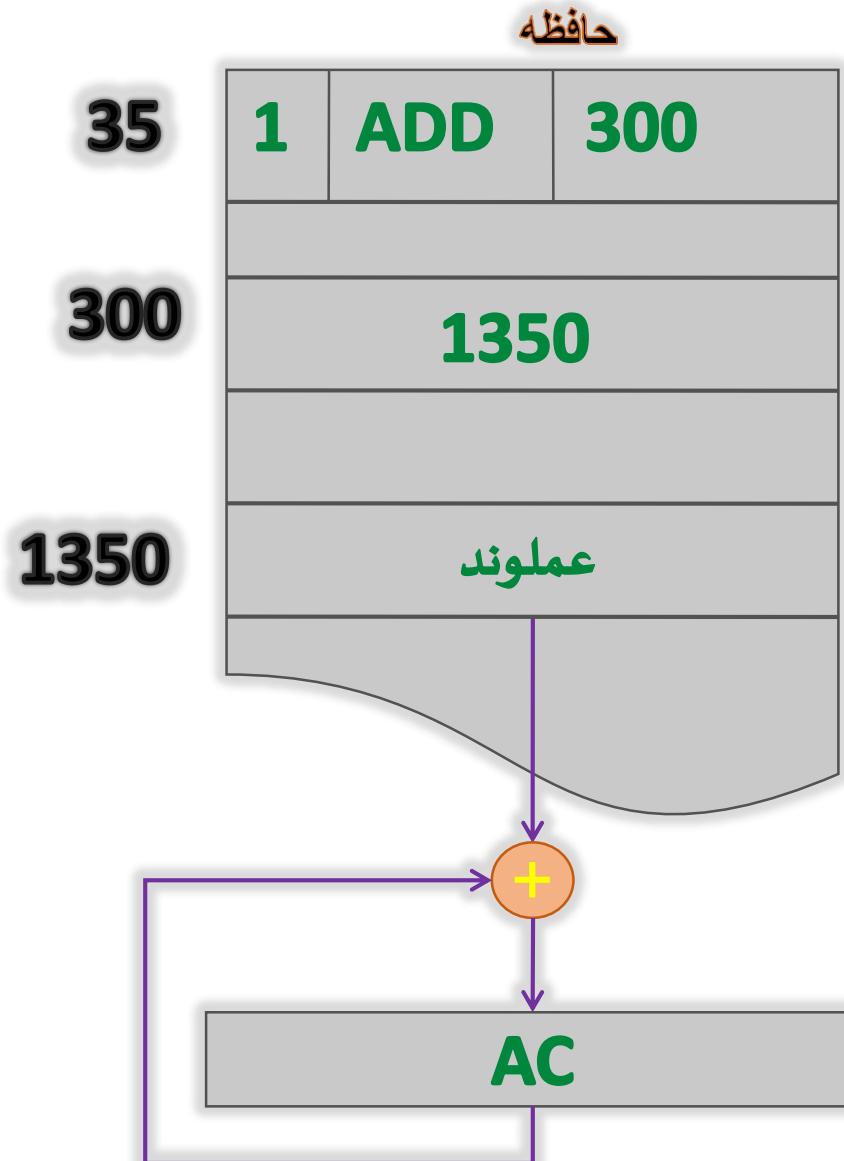
قالب دستورالعملهای حافظه ای در کامپیووتر پایه مانو

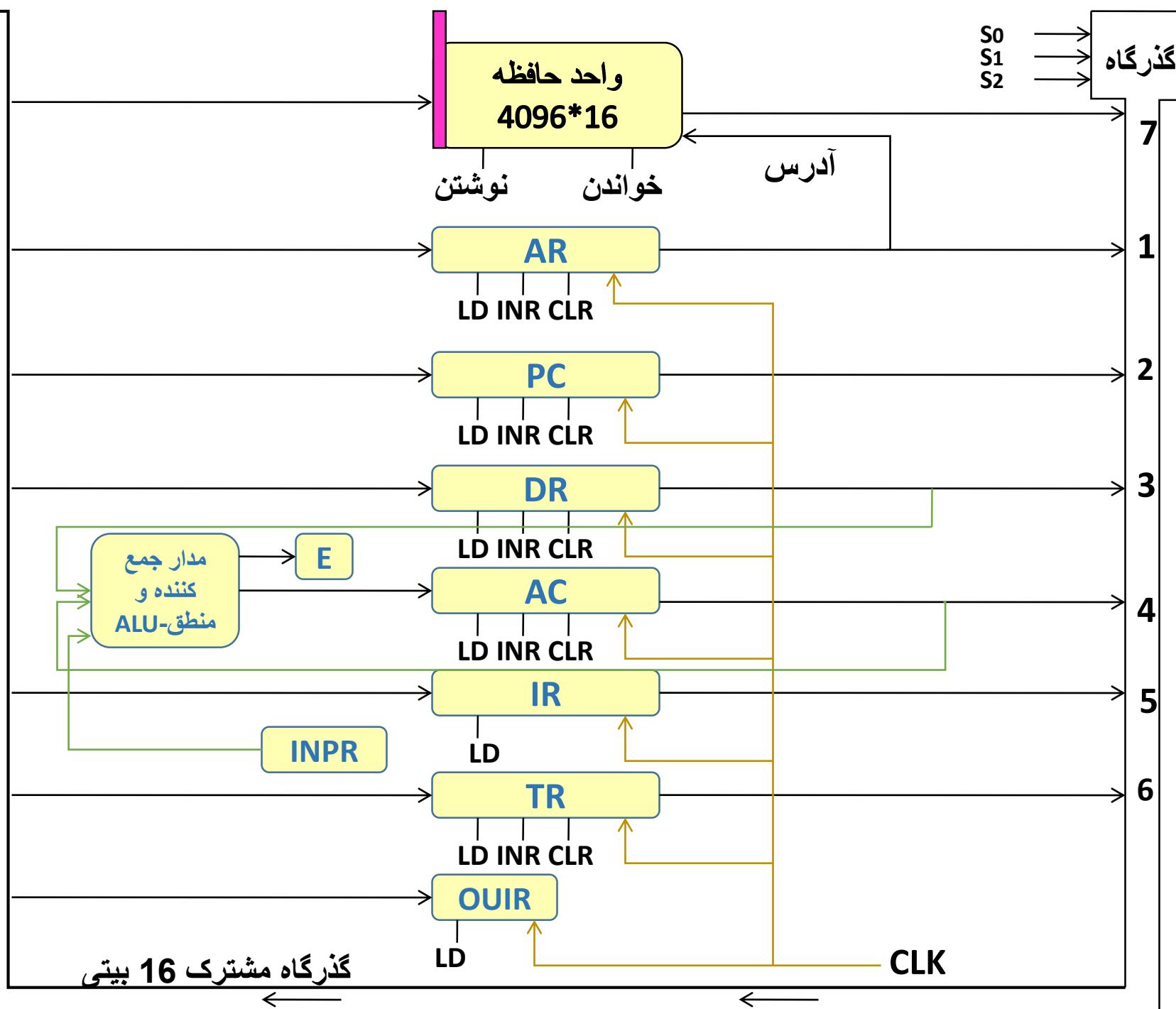
1514	12 11	0
	کد عمل	آدرس

نمایش آدرس مستقیم و غیر مستقیم



نمایش آدرس مستقیم و غیر مستقیم





قالب کنایی دستورالعمل در کامپیوتر پایه

15 14 12 11 0

۱	کد عمل	آدرس
---	--------	------

الف) دستورالعمل های حافظه ای

15 12 11 0

0 1 1 1	عمل ثباتی
---------	-----------

ب) دستورالعمل های ثباتی

15 12 11 0

1 1 1 1	عمل I/O
---------	---------

ج) دستورالعمل های ورودی-خروجی

کد عمل از 000 تا 110

کد عمل برابر ۱,۱۱۱ برابر ۰

کد عمل برابر ۱,۱۱۱ برابر ۱

واحد زمانبندی و کنترل کامپیوتر

همه ریز عملیات در کامپیوتر پایه بوسیله یک واحد کنترل هدایت می شود. فرآمین کنترلی با یک زمان بندی مناسب در فازهای زمانی متوالی به اجزای سیستم صادر می شوند و با آمدن لبه بالارونده پالس ساعت بعدی به اجرا در می آیند. طراحی واحد کنترل به دو صورت انجام می شود:

1. کنترل سخت افزاری
2. کنترل ریز برنامه نویسی شده

در کنترل سخت افزاری سیگنالهای کنترلی توسط گیتهاي واحد کنترل بر اساس وروديها و در فازهای زمانی متوالی تولید می شوند. حسن این روش سرعت بالا است.

در کنترل ریز برنامه نویسی شده، سیگنالهای کنترلی که در هر فاز زمانی باید به سیستم اعمال شود از قبل در یک سطر حافظه **EPROM** ذخیره شده و سپس در زمان فرآخوانی و اجرای هر دستور، سطرهای مربوطه از حافظه خوانده شده و به سیستم اعمال می شوند. حسن این روش قابلیت تغییر و اضافه نمودن دستورات است.

واحد زمانبندی و کنترل کامپیوتر

هر دستوری که از حافظه خوانده شود در ثبات دستور العمل IR قرار میگیرد و 3 بخش دارد:

1. بیت غیر مستقیم

2. کد عمل 3 بیتی

3. بیت های 11-0 برای تعیین آدرس یا نوع عمل ثباتی و I/O

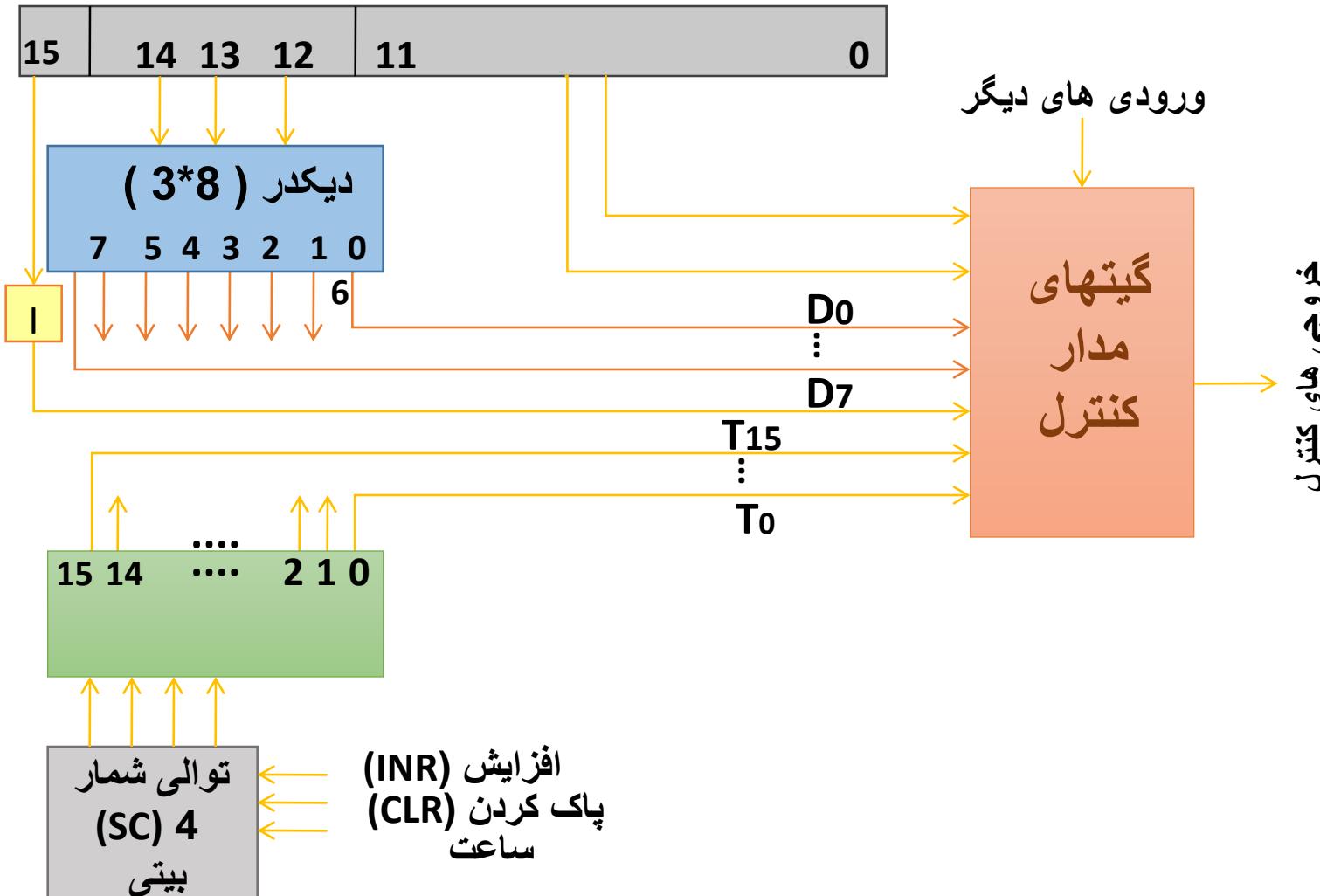
قسمت کد عمل به یک DECODER متصل شده است.

اگر در خروجی این DECODER، D6-D0 تولید شود دستور حافظه ای و

اگر D7 فعال شود دستور غیر حافظه ای است.

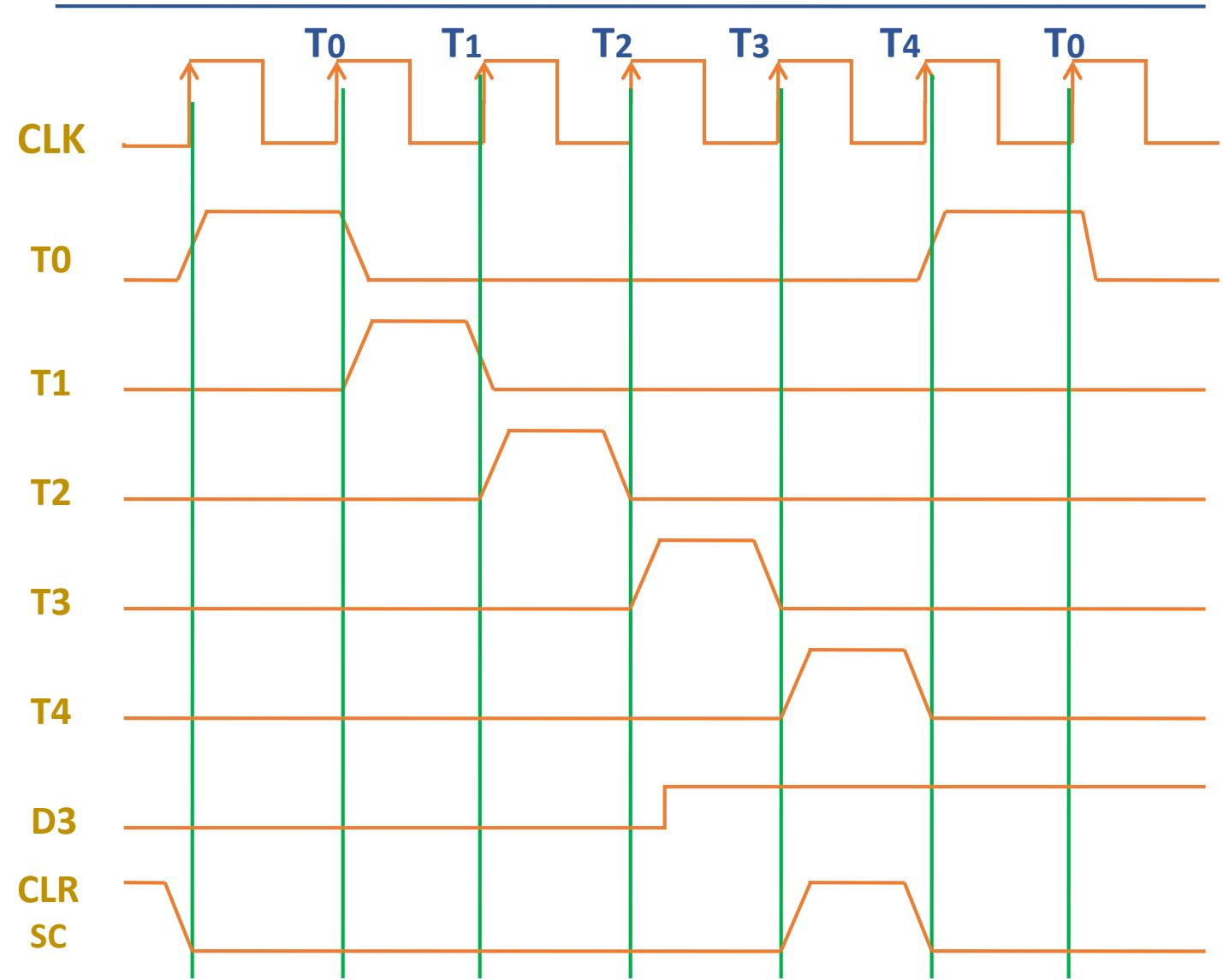
واحد کنترل کامپیوت

ثبتات دستور العمل (IR)



مثال از سیگنال های زمانبندی واحد کنترل

D3T4: SC ←----- 0



سیکل دستورالعمل

در کامپیوتر پایه هر سیکل دستور متشکل از سیکل های زیر است:

1. فرآخوانی یک دستورالعمل از حافظه
2. کردن دستورالعمل Decode.
3. اجرای دستورالعمل

Instruction Fetch, Decode and Execute cycles

هر کدام از این سیکلها از چند فاز زمانی تشکیل شده اند.

برداشت و دیکد

ریز عمل های مربوط به سیکل فرآخوانی و
دیکد را میتوان به صورت زیر نوشت:

۳. دستگاه

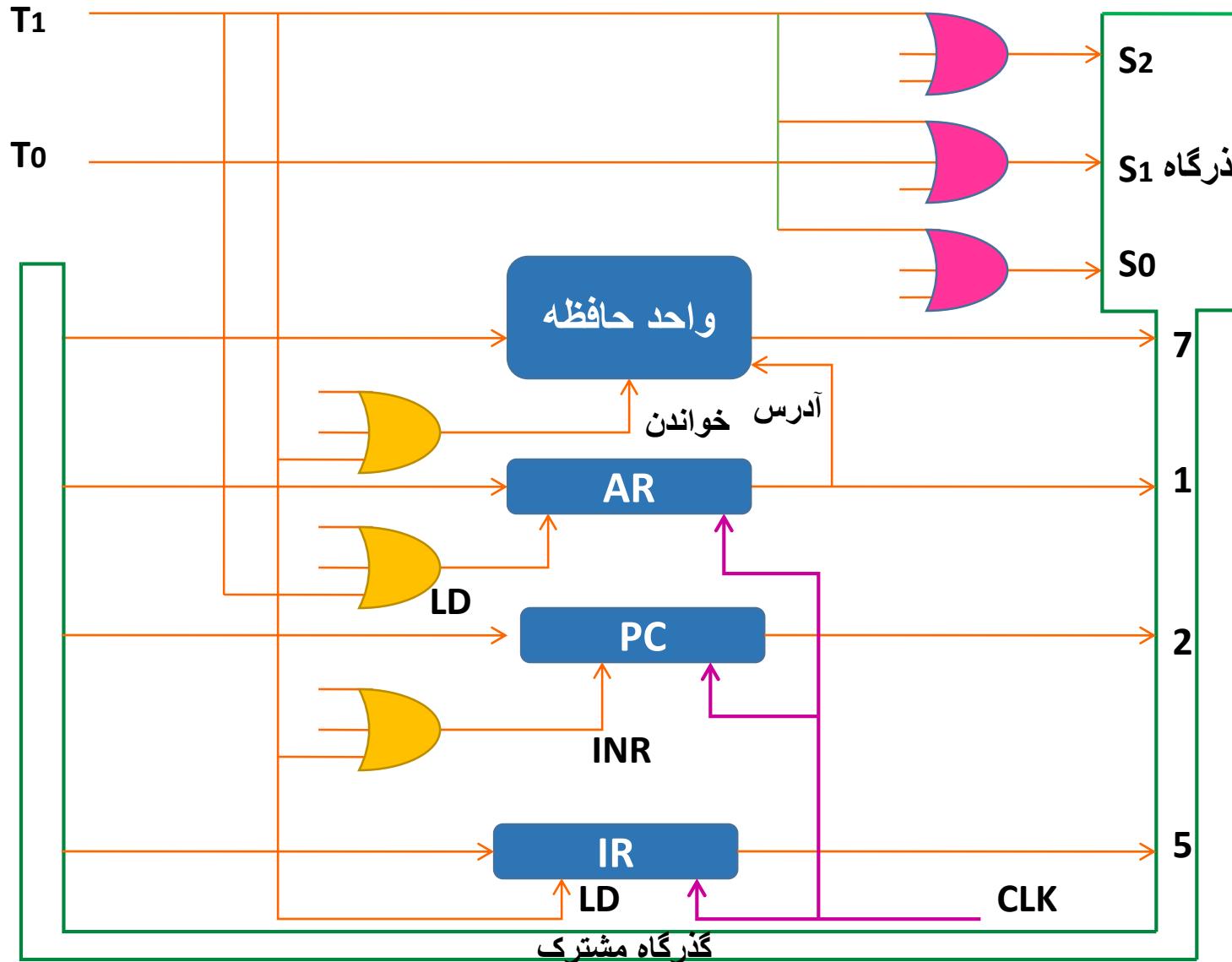
T₀: AR ← PC

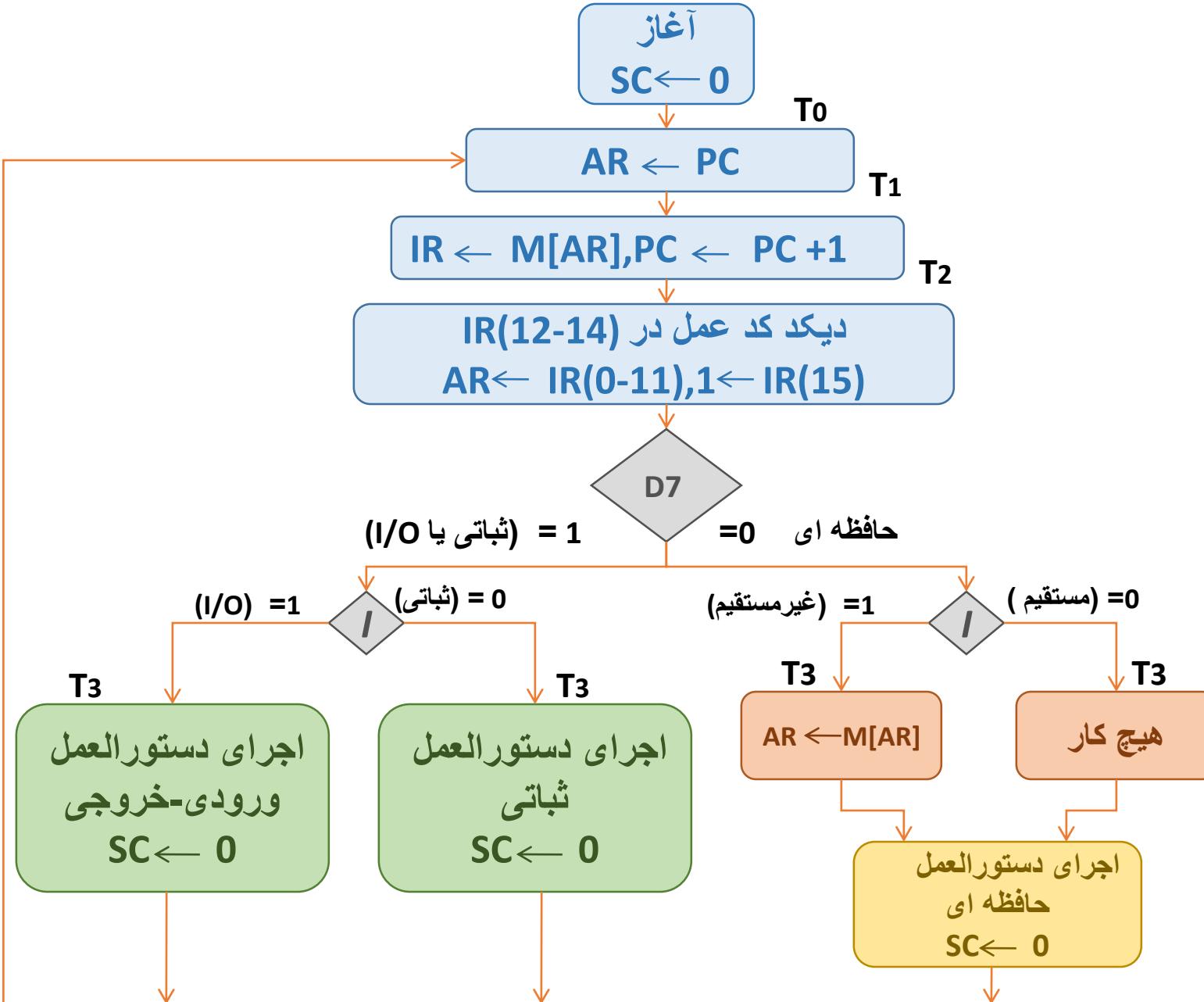
T₁: IR ← M[AR], PC ← PC+1

۴. پردازش

T₂: AR ← IR(0-11), D₀-D₇ ← IR(12-14), I ← IR(15)

نحوه تولید سیگنالهای کنترلی مربوط به فاز فرآخوانی





۳-دستور العمل های ثباتی

D7I'T3

نام	I	کد عمل	نوع عمل ثباتی $IR(i) = Bi$			کد هگزا
CLA	0	111	1000	0000	0000	7800
CLE	0	111	0100	0000	0000	7400
CMA	0	111	0010	0000	0000	7200
CME	0	111	0001	0000	0000	7100
CIR	0	111	0000	1000	0000	7080
CIL	0	111	0000	0100	0000	7040
INC	0	111	0000	0010	0000	7020
SPA	0	111	0000	0001	0000	7010
SNA	0	111	0000	0000	1000	7008
SZA	0	111	0000	0000	0100	7004
SZE	0	111	0000	0000	0010	7002
HLT	0	111	0000	0000	0001	7001

سستورات ثباتی

$r = D7 I'T3$, نوع عمل ثباتی $IR(i) = Bi$

$r: SC \leftarrow 0$

پاک کردن AC	$AC \leftarrow 0$	rB11	CLA
پاک کردن E	$E \leftarrow 0$	rB10	CLE
متمم کردن AC	$AC \leftarrow AC$	rB9	CMA
متمم کردن E	$E \leftarrow E$	rB8	CME
شیفت چرخشی به راست	$AC \leftarrow shr AC .Ac(15) \leftarrow E,E \leftarrow AC(0)$	rB7	CIR
شیفت چرخشی به چپ	$AC \leftarrow shl AC .AC(0) \leftarrow E,E \leftarrow AC(15)$	rB6	CIL
افزایش یک واحد	$AC \leftarrow AC + 1$	rB5	INC
اگر $AC > 0$ یک خط گذرن	If($AC(15)=0$) then ($PC \leftarrow PC + 1$)	rB4	SPA
اگر $AC < 0$ یک خط گذرن	If($AC(15)=1$) then ($PC \leftarrow PC + 1$)	rB3	SNA
اگر $AC = 0$ یک خط گذرن	If ($AC=0$) then ($PC \leftarrow PC + 1$)	rB2	SZA
اگر $E = 0$ یک خط گذرن	If ($E=0$) then ($PC \leftarrow PC + 1$)	rB1	SZE
توقف برنامه	S $\leftarrow 0$ یک فلیپ فلاپ آغاز و پایان است.	rB0	HLT

بستور العمل طای حافظه ای

نماد	دیکدر عمل	توضیح سمبیک
AND	D ₀	$AC \leftarrow AC \wedge M[AR]$
ADD	D ₁	$AC \leftarrow AC + M[AR], E \leftarrow Cout$
LDA	D ₂	$AC \leftarrow M[AR]$
STA	D ₃	$M[AR] \leftarrow AC$
BUN	D ₄	$PC \leftarrow AR$
BSA	D ₅	$M[AR] \leftarrow PC, PC \leftarrow AR + 1$
ISZ	D ₆	$M[AR] \leftarrow M[AR] + 1$ IF $M[AR] + 1 = 0$ then $PC \leftarrow PC + 1$

دستورات عملی حافظه ای

کد شانزده شانزدهی

سمبل

$I=0$

$I=1$

شرح دستور العمل

AND

0xxx

8xxx

AC کردن کلمه حافظه با AND

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 بدون شرط

BSA

5xxx

Dxxx

انشعاب و ذخیره آدرس بازگشت

ISZ

6xxx

Exxx

افزایش و گذر در صورت صفر شدن

قیمت‌وراٽ حافظه‌ای

AC با AND

D₀T₄: DR ← M[AR]

D₀T₅: AC ← DR ^ AC, SC ← 0

AC با ADD

D₁T₄: DR ← M[AR]

D₁T₅: AC ← DR + AC, E ← Cout, SC ← 0

AC: بار کردن LDA

D₂T₄: DR ← M[AR]

D₁T₅: AC ← DR, SC ← 0

: ذخیره کردن AC در حافظه STA

D₃T₄: M[AR] ← AC, SC ← 0

بستورات ارجاع به حافظه

: انشاب بدون شرط **BUN**

D4T4: PC ← AR, SC ← 0

: انشاب با ذخیره آدرس برگشت **BSA**

D5T4: M[AR] ← PC AR ← AR + 1

D5T5: PC ← AR, SC ← 0

: افزایش و گذر در صورت صفر شدن **ISZ**

D6T4: DR ← M[AR]

D6T5: DR ← DR + 1

D6T5: If (DR=0) then PC ← PC + 1, SC ← 0

چکونگی عملکرد BSA

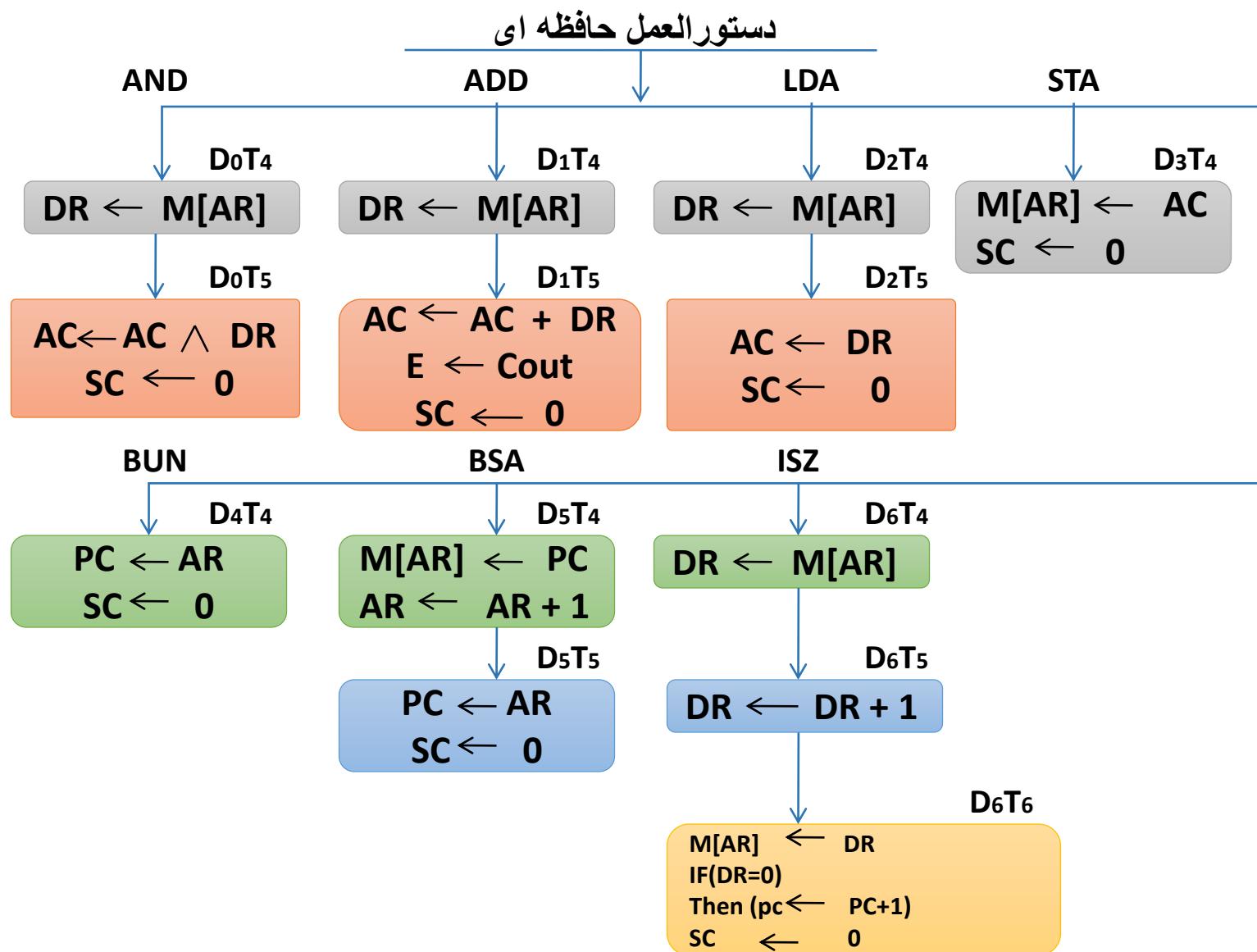


ب) حافظه و PC پس از اجرا



الف) حافظه، AR، PC در زمان T4

فایل چارت ندروه اجرایی دستور العمل حافظه ای از خازن T4

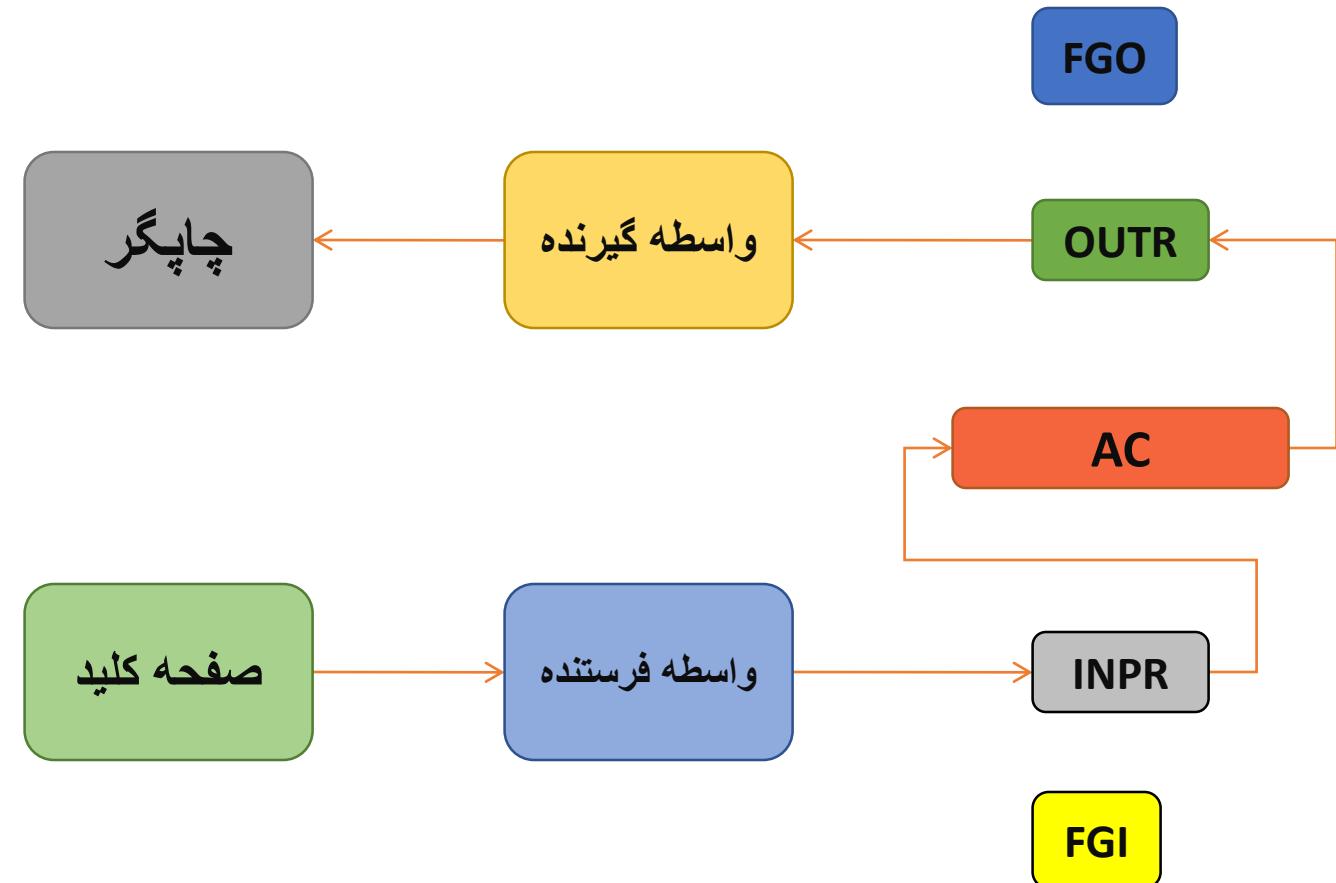


ارتباط ورودی خروجی با پردازنده مانو

پایانه ورودی-خروجی

واسطه ارتباطی سری

ثبتات و فلیپ فلاب های کامپیوتر



I/O چاک فستور العمل

	I	کد عمل یکسان	بیت نوع عمل	کد هگز
INP	1	111	1000 0000 0000	F800
OUT	1	111	0100 0000 0000	F400
SKI	1	111	0010 0000 0000	F200
SKO	1	111	0001 0000 0000	F100
ION	1	111	0000 1000 0000	F080
IOF	1	111	0000 0100 0000	F040

I/O دستور العمل های

کد شانزده شانزدهی

سمبل	I=1	شرح دستور العمل
INP	F800	دربافت کاراکتر ازورودی و انتقال آن به انباره و صفر کردن پرچم
OUT	F400	ارسال کاراکتر از AC به خروجی و صفر کردن پرچم
SKI	F200	گذار مبتنی بر پرچم ورودی
SKO	F100	گذار مبتنی بر پرچم خروجی
ION	F080	فعال کردن وقفه ها
IOF	F040	غیر فعال کردن وقفه ها

دستورات ورودی-خروجی

D7 IT3 = p

p: sc ← 0 (مشترک در همه دستورالعملهای ورودی - خروجی)

INP	pB ₁₁ :	AC (0_7) ← INPR , FGI ← 0	دریافت کارکتر ورودی
OUT	pB ₁₀ :	OUTR ← AC (0_7), FGO ← 0	ارسال کاراکتر خروجی
SKI	pB ₉ :	IF (FGI=1) then (PC ← PC + 1)	کذر بر اساس پرچم ورودی
SKO	pB ₈ :	IF (FGO=1) then (PC ← PC + 1)	کذر بر اساس پرچم خروجی
ION	pB ₇ :	IEN ← 1	روشن کردن فعال ساز وقفه
IOF	pB ₆ :	IEN ← 0	خاموش کردن فعال ساز وقفه

Program Interrupt

- There are two ways of data transfer between I/O and the processor:
- **1- Program Controlled Transfer**
- The processor keeps checking the flag bit, and when it finds it set, it initiates an information transfer (this is sometimes called **Polling**)
- This type of transfer is in-efficient due to the difference of information flow rate between the computer and the I/O device.

Program Interrupt^{cont.}

- The computer is wasting time while checking the flag instead of doing some other useful processing task
- **2- Interrupt controlled transfer**
- An alternative to the program controlled transfer is to let the external device inform the computer when it is ready for the transfer
- This type of transfer uses the interrupt facility

Program Interrupt^{cont.}

- While the computer is running a program, it does not check the flags
- Instead:
 - When a flag is set, the computer is immediately interrupted from proceeding with the current program

Program Interrupt^{cont.}

- The computer stops what it is doing to take care of the input or output transfer
- Then, it returns to the current program to continue what it was doing before the interrupt
- The interrupt facility can be enabled or disabled via a flip-flop called IEN
- The interrupt enable flip-flop IEN can be set and cleared with two instructions (IOF, ION):
 - IOF: IEN $\leftarrow 0$ (the computer cannot be interrupted)
 - ION: IEN $\leftarrow 1$ (the computer can be interrupted)

Program Interrupt^{cont.}

- Another flip-flop (called the interrupt flip-flop **R**) is used in the computer's interrupt facility to decide when to go through the interrupt cycle
- **FGI** and **FGO** are different here compared to the way they acted in an earlier discussion
- So, the computer is either in an Instruction Cycle or in an Interrupt Cycle

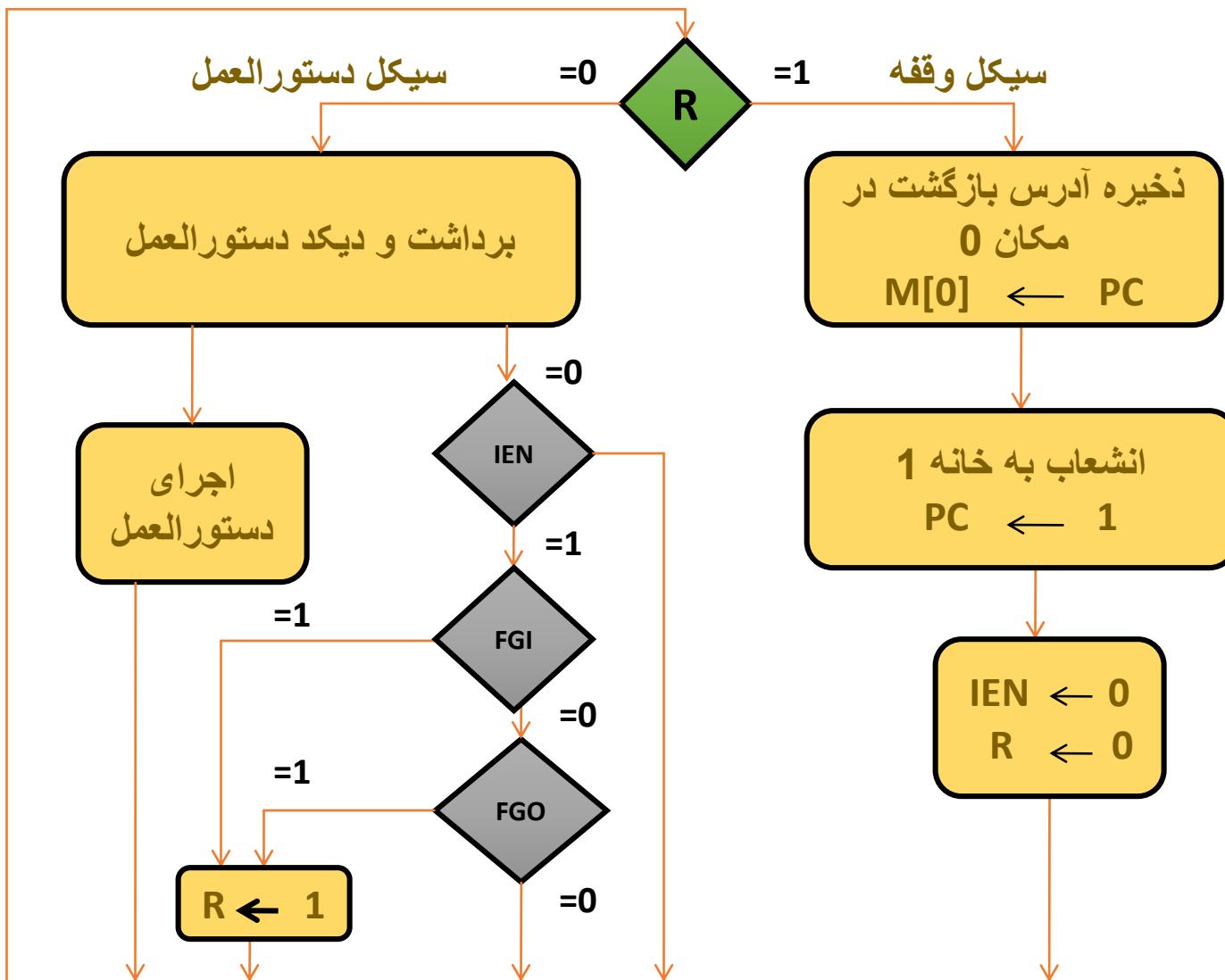
Program Interrupt^{cont.}

- The interrupt cycle is a hardware implementation of a branch and save return address operation (BSA)
- The return address available in PC is stored in a specific location where it can be found later when the program returns to the instruction at which it was interrupted
- This location may be a processor register, a memory stack, or a specific memory location

Program Interrupt^{cont.}

- For our computer, we choose the memory location at address 0 as a place for storing the return address
- Control then inserts address 1 into PC: this means that the first instruction of the interrupt service routine should be stored in memory at address 1, or, the programmer must store a branch instruction at address 1 that sends the control to an interrupt service routine.

فلوچارت سیکل وقفه



خاڙ ڪسٽيٽ ڦانچه ز ڪمپيوٽر

RT₀: AR ← 0 , TR ← PC

RT₁: M[AR] ← TR , PC ← 0

RT₂: PC ← PC+1 , IEN ← 0 , R ← 0 , SC ← 0

فازهای زمانی فرآخوانی و دیگر دستورات عمل

با در نظر گرفتن فلیپ فلاپ R

$R'T_0: AR \leftarrow PC$

$R'T_1: AR \leftarrow M[AR], PC \leftarrow PC+1$

$R'T_2: D_0 \dots D_7 \leftarrow IR(12-14), AR \leftarrow IR(0-11),$
 $I \leftarrow IR(15)$

مثال از سیکل وقفه

حافظه

0	256	
PC=1	0	BUN 1120
255	برنامه اصلی	
256		
1120	برنامه I/O	
1	BUN	0

ب) بعد از سیکل وقفه

حافظه

0	256	
1	0	BUN 1120
255	برنامه اصلی	
1120		
1	برنامه I/O	
1	BUN	0

الف) قبل از وقفه

فلوچارت برای

اعمال کامپیوت

اعار

$SC \leftarrow 0, IEN \leftarrow 0, R \leftarrow 0$

سیکل دستور العمل

0

سیکل وقفه

1

$AR \leftarrow PC$

$AR \leftarrow 0.TR \leftarrow PC$

$IR \leftarrow M[AR], PC \leftarrow PC + 1$

$M[AR] \leftarrow TR.PC \leftarrow 0$

$AR \leftarrow IR(0-11), 1 \leftarrow IR(15)$

$D_0 \quad D_7 \leftarrow \text{Decode IR (12-14)}$

$PC \leftarrow PC + 1, INE \leftarrow 0$

$R \leftarrow 0, SC \leftarrow 0$

$(I/O) = 1$ (ثبتی یا 0)

$I = 0$ (ثبتی)

اجرای دستور العمل
ورودی-
خروجی (I/O)

اجرای دستور العمل
ثبتی

$I = 1$ (غیرمستقیم)

$I = 0$ (مستقیم)

$IEN = 0$

$IEN = 1$

$FGI = 0$

$FGI = 1$

$FOG = 0$

$FOG = 1$

$D_7IT_3 = 1$

$D_7IT_3 = 0$

T₄

اجرای دستور العمل
حافظه ای

$IR \leftarrow M[AR]$

هیچ چیز

$R \leftarrow 1$

مراحل نظریاتی طراحی کامپیوٹر پایه وساخت افزار صورت نیاز

1- حافظه:

$4096 * 16$

2- ثبات ها :

AR, PC, DR, AC, IR, TR, INPR, OUTR

3- فلیپ فلاپ ها

I, S,E, R, FGI, FGO, IEN

4- دیکدرها

$3*8 \longrightarrow$ برای تولید D0-D7

$4*16 \longrightarrow$ دیکدر فاز های زمانی

5- گذرگاه مشترک و مالتی پلکسراها

6- مدار کنترل

خروجی های مدار کنترل منطقی

- 1- سیگنال های کنترلی 8 عدد ثبات
- 2- سیگنال های کنترلی خواندن و نوشتן حافظه
- 3- سیگنال های نشاندن، پاک کردن و مکمل سازی فلیپ
فلاب ها
- 4- سیگنال های S_2 ، S_1 ، S_0 برای انتخاب ثبات برای
گذرگاه
- 5- سیگنال های کنترلی مدار جمع کننده و منطق
وشیفت (ALU)

مدار کنترل گیتی پایه حافظه WRITE و READ

پایه حافظه	ریز عمل	
R	$R'T_1: IR$	← M[AR]
R	$D_7'IT_3: AR$	← M[AR]
W	$RT_1:M[AR]$	← TR
R	$D_0T_4: DR$	← M[AR]
R	$D_1T_4: DR$	← M[AR]
R	$D_2T_4: DR$	← M[AR]
W	$D_3T_4:M[AR]$	← AC
W	$D_5T_4:M[AR]$	← PC
R	$D_6T_4: DR$	← M[AR]
W	$D_6T_6:M[AR]$	← DR

مدار کنترل گیتی پایه های WRITE و READ

فر شیج

$$R = R'T_1 + D_7' IT_3 + T_4 (D_0 + D_1 + D_2 + D_3)$$

$$W = RT_1 + T_4(D_3 + D_5) + D_6T_6$$

جدول 5-6- توابع کنترلی و ریز عملهای کامپیوتر پایه:

Fetch	R'T0:	$AR \leftarrow PC$
	R'T1:	$IR \leftarrow M[AR], PC \leftarrow PC + 1$
Decode	R'T2:	$D0, \dots, D7 \leftarrow \text{Decode IR}(12 \sim 14), AR \leftarrow IR(0 \sim 11), I \leftarrow IR(15)$
Indirect	D7'IT3:	$AR \leftarrow M[AR]$
Interrupt:		
T0'T1'T2'(IEN)(FGI + FGO):	RT0:	$R \leftarrow 1$
	RT1:	$AR \leftarrow 0, TR \leftarrow PC$
	RT2:	$M[AR] \leftarrow TR, PC \leftarrow 0$
		$PC \leftarrow PC + 1, IEN \leftarrow 0, R \leftarrow 0, SC \leftarrow 0$
Memory-Reference:		
AND	D0T4:	$DR \leftarrow M[AR]$
	D0T5:	$AC \leftarrow AC . DR, SC \leftarrow 0$
ADD	D1T4:	$DR \leftarrow M[AR]$
	D1T5:	$AC \leftarrow AC + DR, E \leftarrow Cout, SC \leftarrow 0$
LDA	D2T4:	$DR \leftarrow M[AR]$
	D2T5:	$AC \leftarrow DR, SC \leftarrow 0$
STA	D3T4:	$M[AR] \leftarrow AC, SC \leftarrow 0$
BUN	D4T4:	$PC \leftarrow AR, SC \leftarrow 0$
BSA	D5T4:	$M[AR] \leftarrow PC, AR \leftarrow AR + 1$
	D5T5:	$PC \leftarrow AR, SC \leftarrow 0$
ISZ	D6T4:	$DR \leftarrow M[AR]$
	D6T5:	$DR \leftarrow DR + 1$
	D6T6:	$M[AR] \leftarrow DR, \text{ if}(DR=0) \text{ then } (PC \leftarrow PC + 1), SC \leftarrow 0$

ادامه جدول 5-6- توابع کنترلی و ریز عملهای کامپیووتر پایه

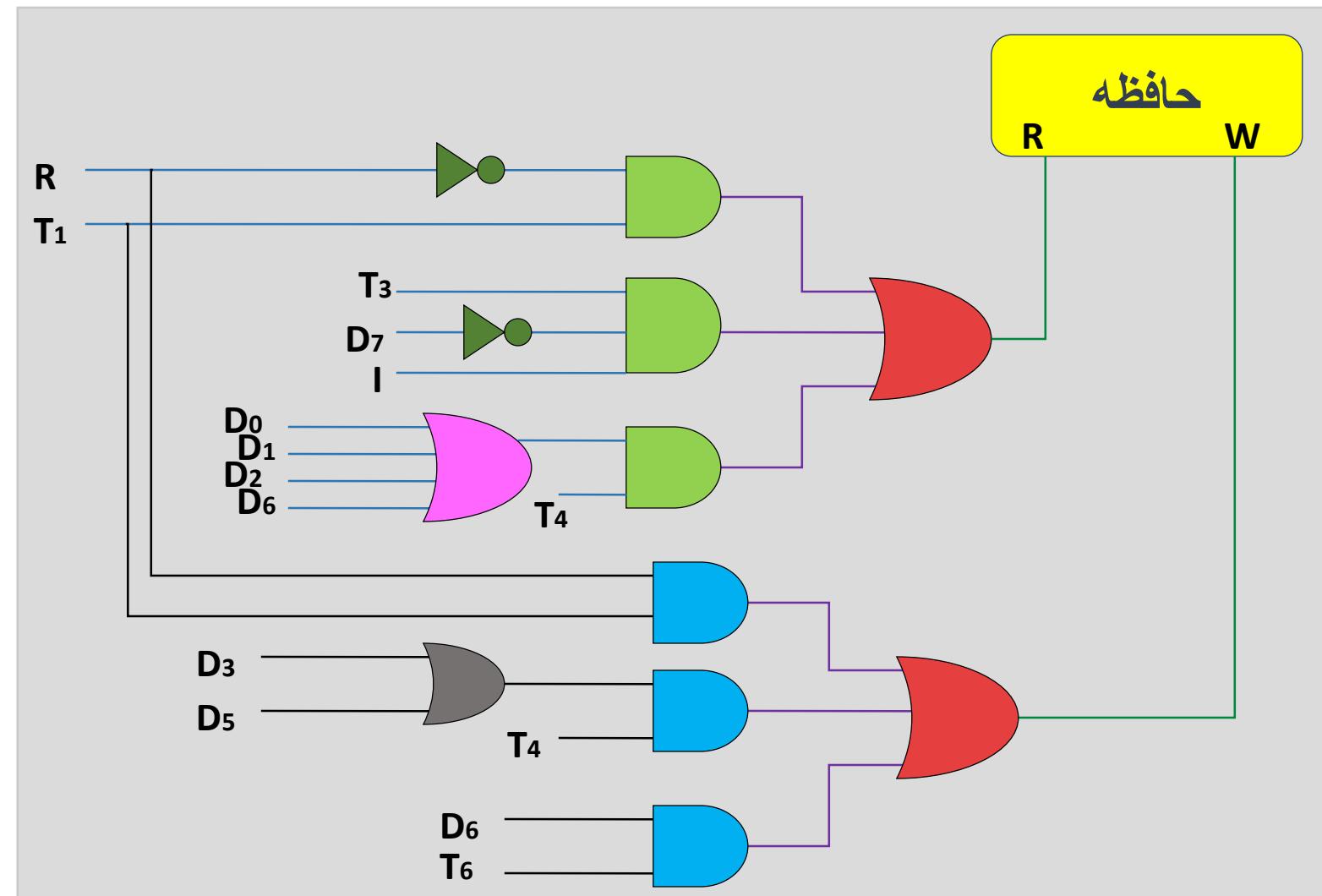
Register-Reference:

	D7IT3 = r	(Common to all register-reference instructions)
	IR(i) = Bi	(i = 0,1,2, ..., 11)
	r:	SC \leftarrow 0
CLA	rB11:	AC \leftarrow 0
CLE	rB10:	E \leftarrow 0
CMA	rB9:	AC \leftarrow AC'
CME	rB8:	E \leftarrow E'
CIR	rB7:	AC \leftarrow shr AC, AC(15) \leftarrow E, E \leftarrow AC(0)
CIL	rB6:	AC \leftarrow shl AC, AC(0) \leftarrow E, E \leftarrow AC(15)
INC	rB5:	AC \leftarrow AC + 1
SPA	rB4:	If(AC(15) = 0) then (PC \leftarrow PC + 1)
SNA	rB3:	If(AC(15) = 1) then (PC \leftarrow PC + 1)
SZA	rB2:	If(AC = 0) then (PC \leftarrow PC + 1)
SZE	rB1:	If(E=0) then (PC \leftarrow PC + 1)
HLT	rB0:	S \leftarrow 0

Input-Output:

	D7IT3 = p	(Common to all input-output instructions)
	IR(i) = Bi	(i = 6,7,8,9,10,11)
	p:	SC \leftarrow 0
INP	pB11:	AC(0-7) \leftarrow INPR, FGI \leftarrow 0
OUT	pB10:	OUTR \leftarrow AC(0-7), FGO \leftarrow 0
SKI	pB9:	If(FGI=1) then (PC \leftarrow PC + 1)
SKO	pB8:	If(FGO=1) then (PC \leftarrow PC + 1)
ION	pB7:	IEN \leftarrow 1
IOF	pB6:	IEN \leftarrow 0

مدار کنترل گیتی پایه فکار WRITE و READ حافظه



پایہ طیار کنٹرول شبیت AR

ریز عمل

L $R'T_0: AR \leftarrow PC$

L $R'T_2: AR \leftarrow IR(0-11)$

L $D'7|T_3: AR \leftarrow M[AR]$

CLR $RT_0: AR \leftarrow 0$

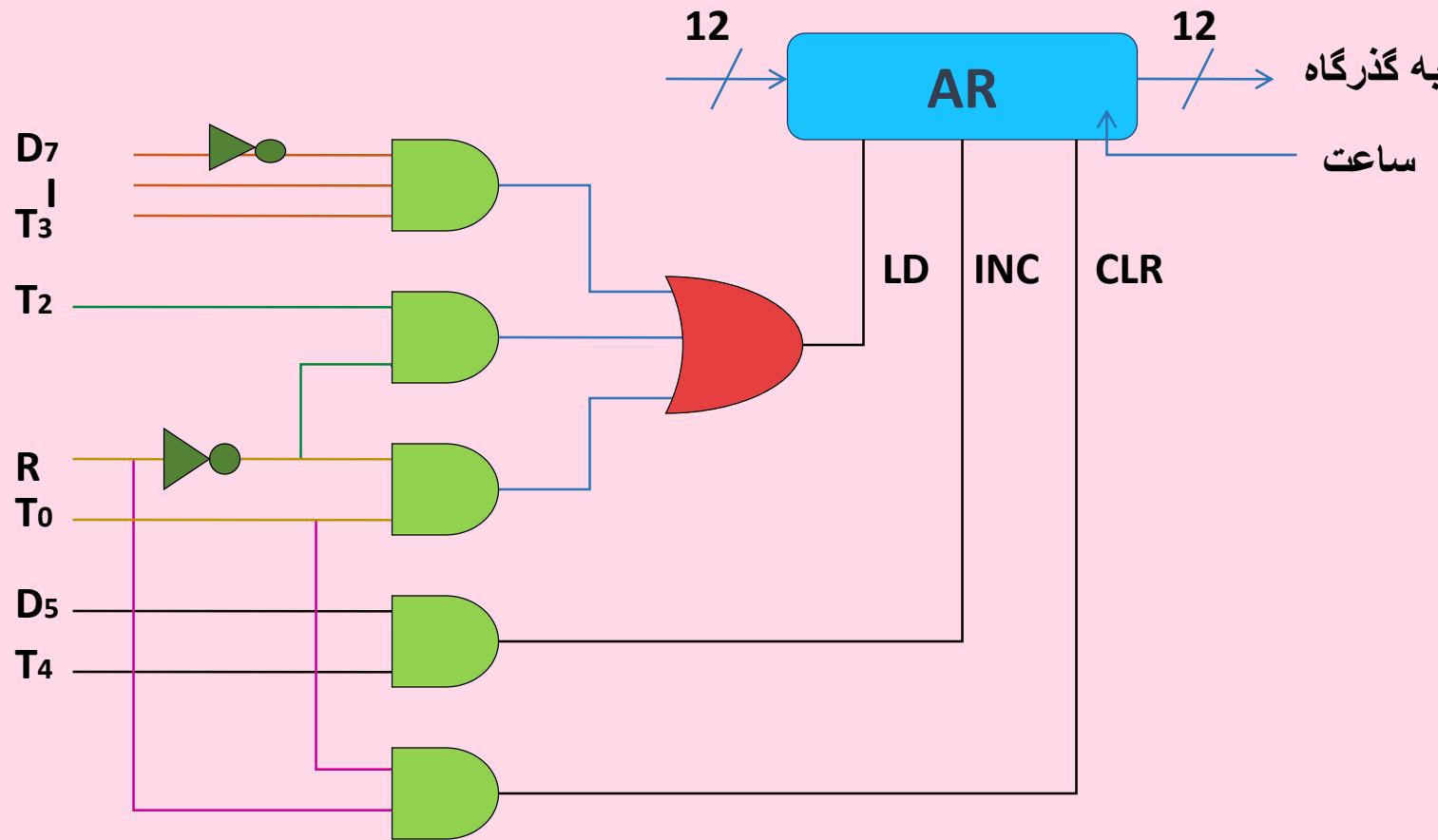
INC $D_5T_4: AR \leftarrow AR + 1$

$LD(AR) = R'T_0 + R'T_2 + D'7|T_3$

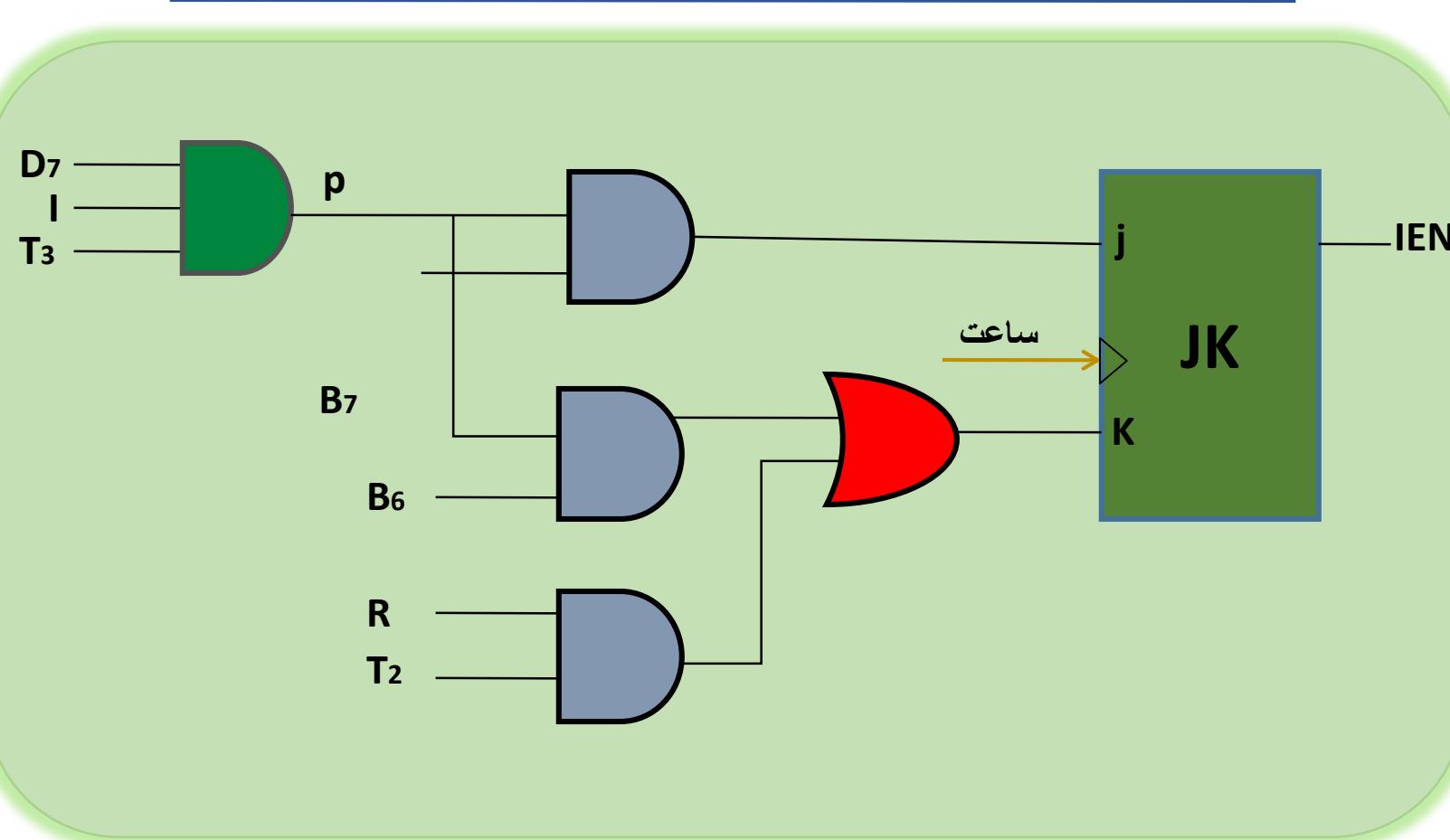
$CLR(AR) = RT_0$

$INC(AR) = D_5T_4$

AR گیت فلای کنترل

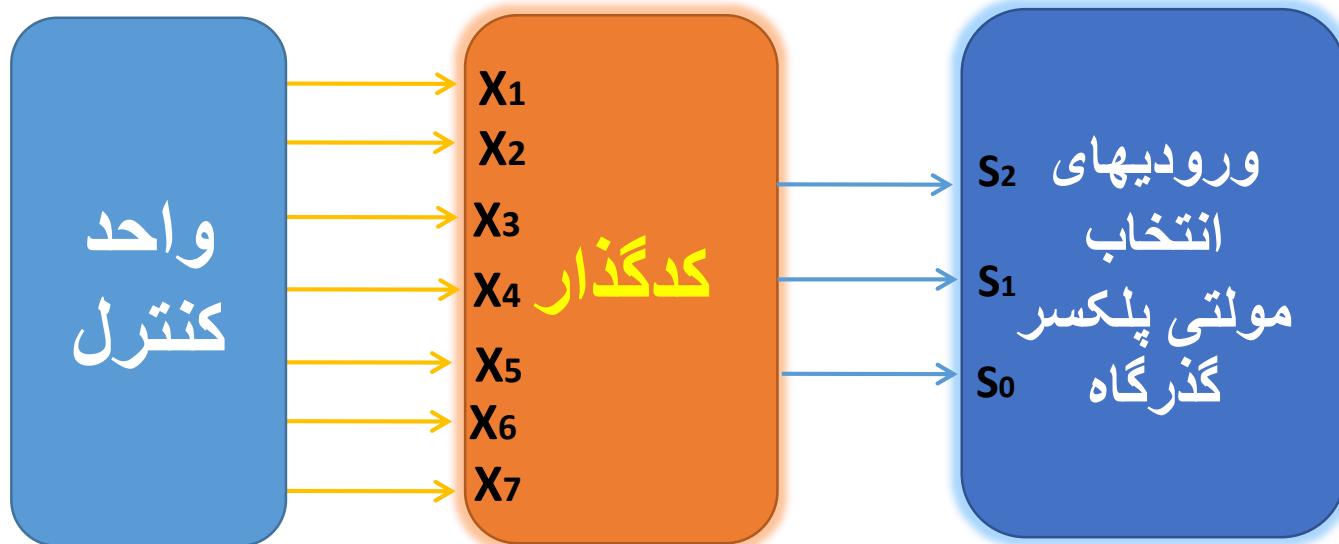


IEN پروتکل های کنترل برای فلیپ فلامپ



PB7: IEN ← 1
PB6: IEN ← 0
RT2: IEN ← 0

کدگذار برای انتخاب گذرگاه



- To select AR on the bus then x_1 must be 1. This happens when:
 - $D_4 T_4: PC \leftarrow AR$
 - $D_5 T_5: PC \leftarrow AR$
- $\Rightarrow x_1 = D_4 T_4 + D_5 T_5$

x_1	x_2	x_3	x_4	x_5	x_6	x_7	S_2	S_1	S_0	selected register
0	0	0	0	0	0	0	0	0	0	none
1	0	0	0	0	0	0	0	0	1	AR
0	1	0	0	0	0	0	0	1	0	PC
0	0	1	0	0	0	0	0	1	1	DR
0	0	0	1	0	0	0	1	0	0	AC
0	0	0	0	1	0	0	1	0	1	IR
0	0	0	0	0	1	0	1	1	0	TR
0	0	0	0	0	0	1	1	1	1	Memory

- For x_7 :
 - $x_7 = R'T_1 + D_7'IT_3 + (D_0 + D_1 + D_2 + D_6)T_4$ where it is also applied to the read input of Memory.

$$S0 = X1 + X3 + X5 + X7$$

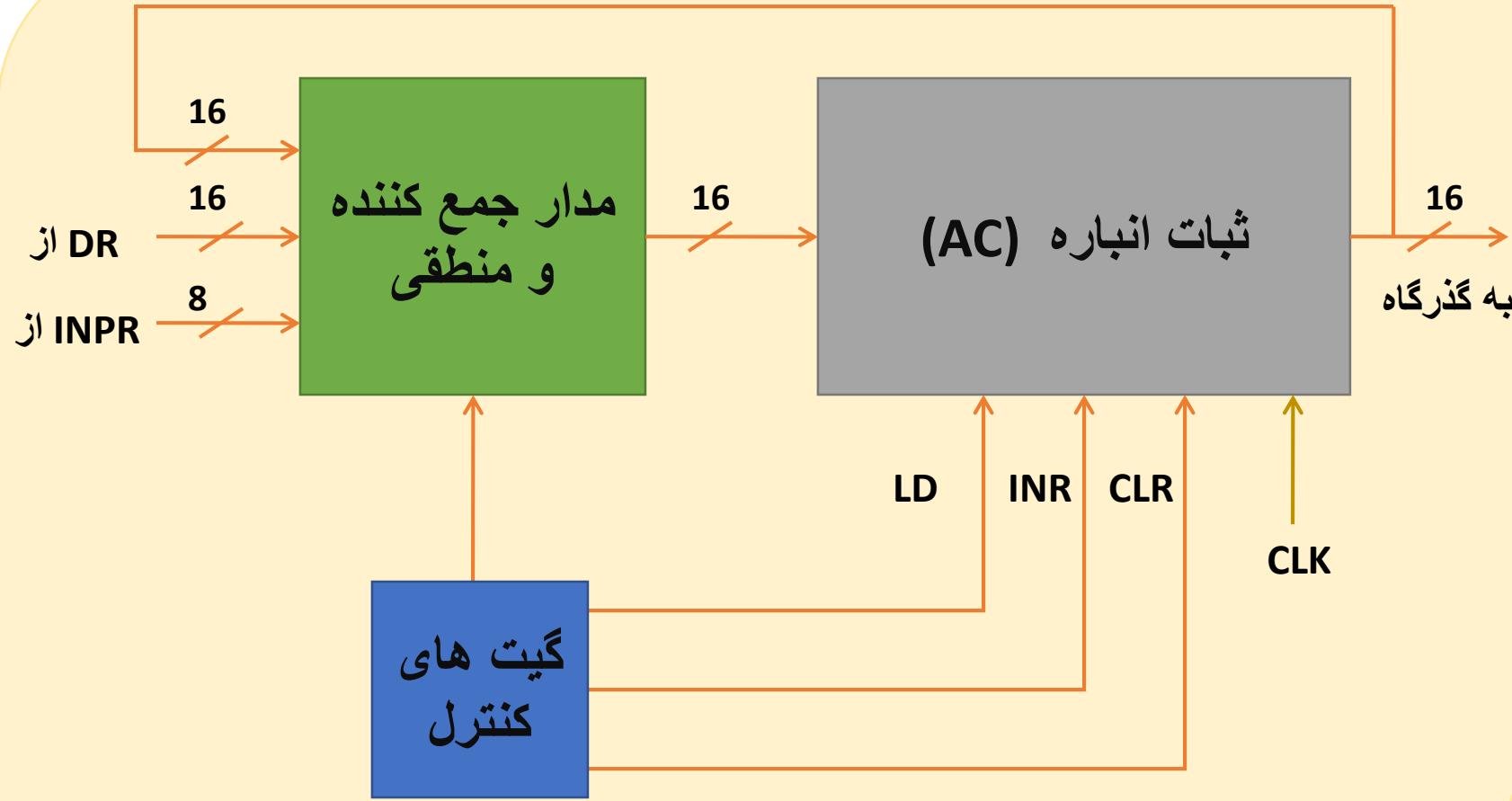
$$S1 = X2 + X3 + X6 + X7$$

$$S2 = X4 + X5 + X6 + X7$$

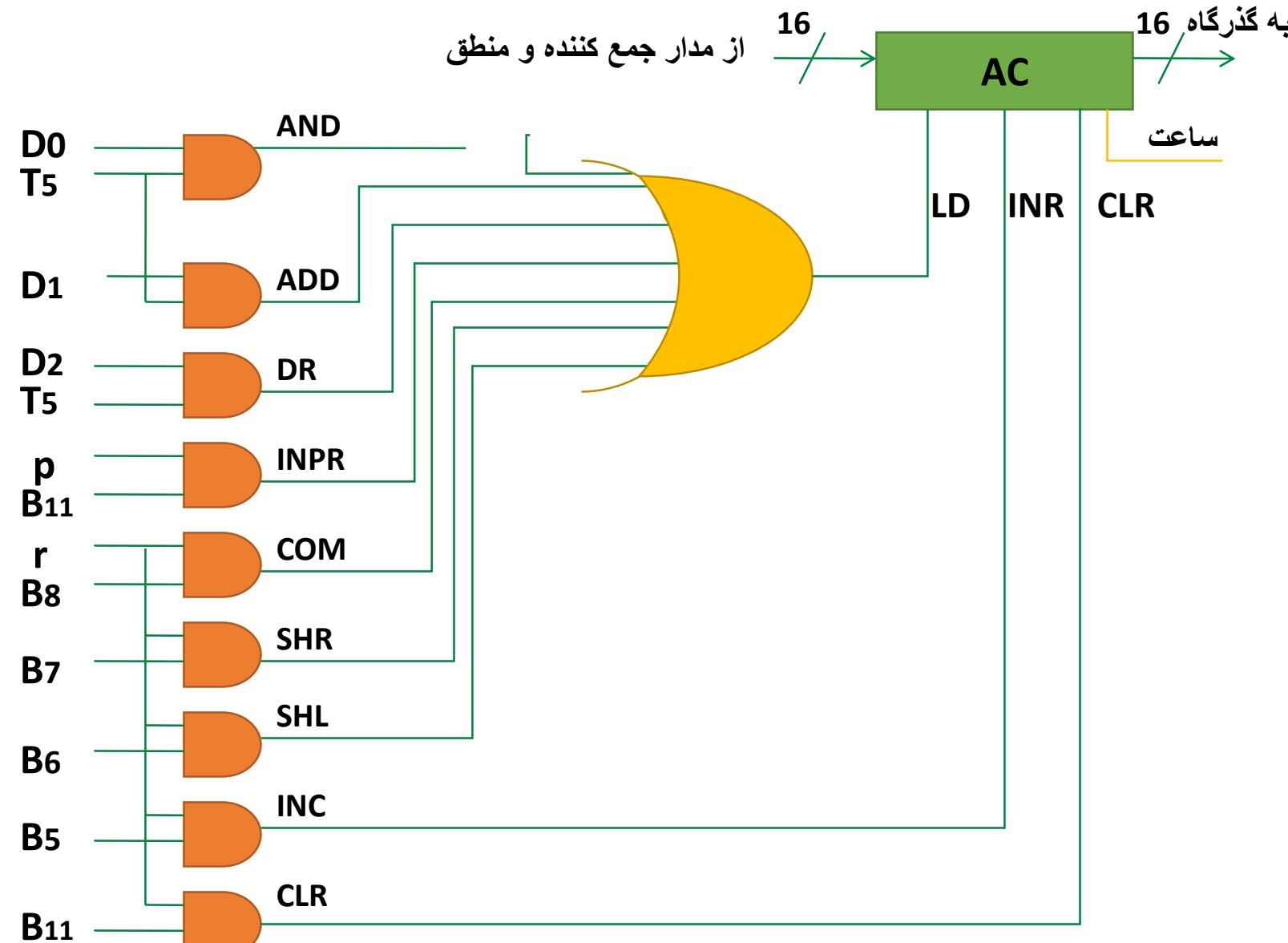
AC دستورات اثربار روش سیگنالهای کنترلی

LD	DR پا AND	$AC \leftarrow AC \wedge DR$	D0T5
LD	DR پا ADD	$AC \leftarrow AC + DR$	D1T5
LD	انتقال از DR	$AC \leftarrow DR$	D2T5
LD	INPR انتقال از	$AC(0-7) \leftarrow INPR$	pB11
LD	متمم	$AC \leftarrow \overline{AC}$	rB9
LD	چرخش به راست	$AC \leftarrow shr AC, AC(15) \leftarrow E$	rB7
LD	چرخش به چپ	$AC \leftarrow shl AC, AC(0) \leftarrow E$	rB6
CLR	پاک کردن	$AC \leftarrow 0$	rB11
INR	افزایش دادن	$AC \leftarrow AC + 1$	rB5

تولید سیگنالهای کنترلی AC



ساختار گیت های مربوط به کنترل در ورودی های AC از CLR، INR، LD



یک طبقه از مدار جمع کننده و منطق

DR(i) AC(i)

