



دانشکده مهندسی کامپیوتر  
دانشگاه صنعتی امیرکبیر

# فصل دوم

## طراحی واحد پردازش گر مرکزی ریزپردازنده ۸۰۸۶

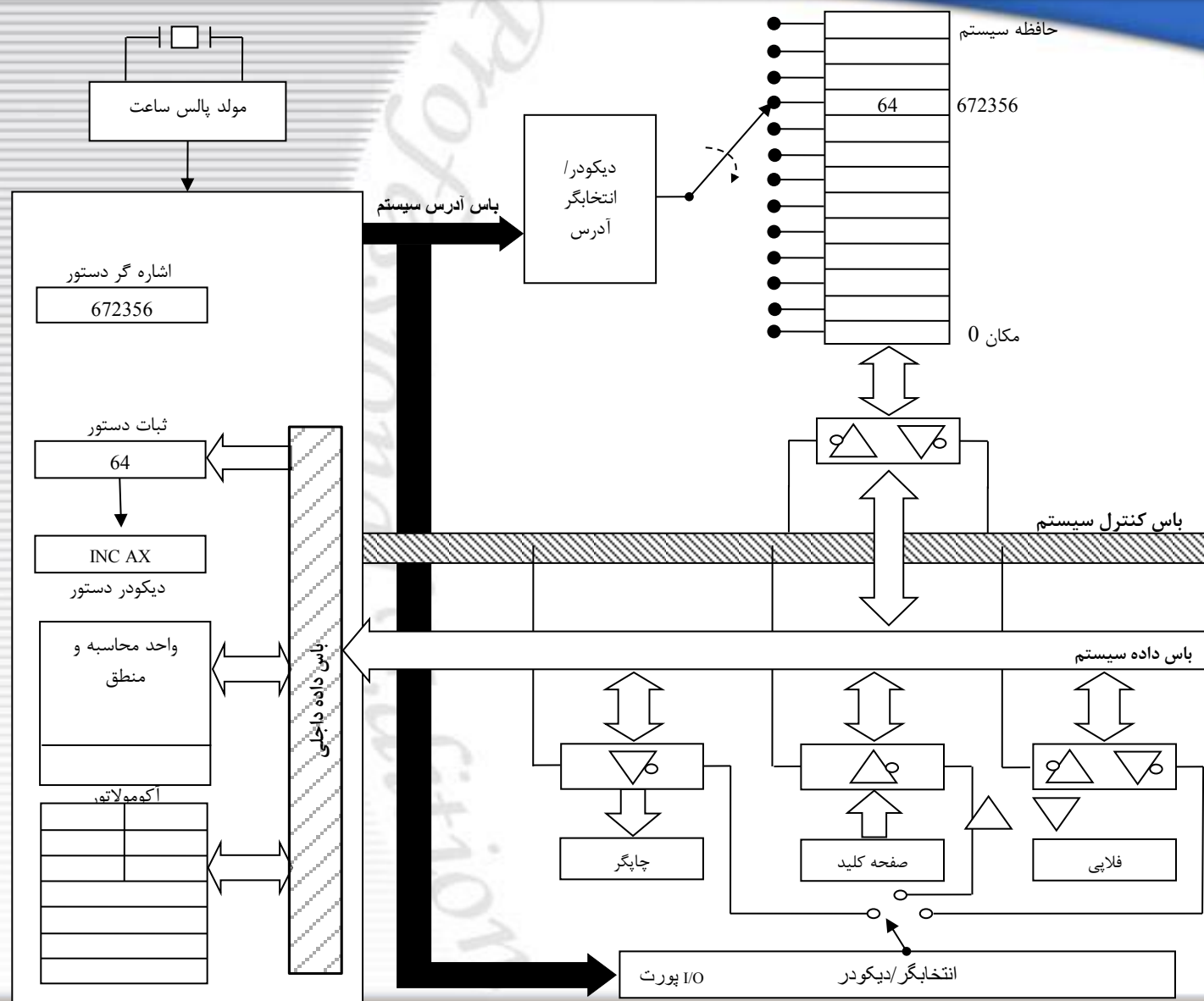
ریزپردازنده ۱  
محمد مهدی همایون پور

## سرفصل مطالب

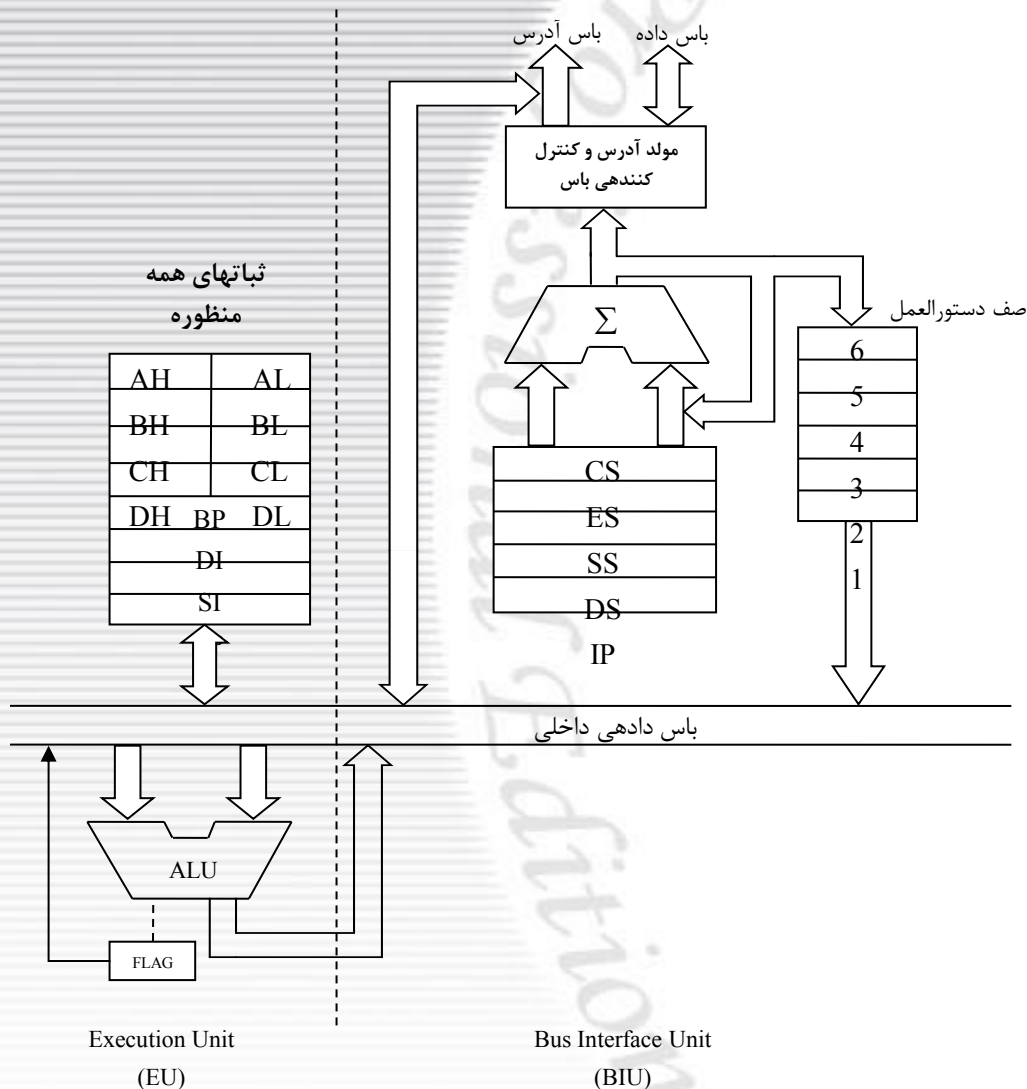
---

- بلوک دیاگرامی یک ریزپردازنده فرضی با برنامه‌ی ذخیره شده در حافظه
- ساختار CPU در ۸۸ / ۸۰۸۶
- واکنشی و اجرای دستورالعمل‌ها در ریزپردازنده‌ها
- آشنایی با انواع سیکل‌های باس در ۸۰۸۶
- آشنایی با پایه‌های ۸۰۸۶
- طراحی سخت‌افزار پایه برای ریزپردازنده ۸۰۸۶
- تولید سیگنال بازنشانی و پالس ساعت ۸۰۸۶
- واحد CPU تراشه ۸۰۸۶ برای مد حداقل

# بلوک دیاگرامی یک ریزپردازنده فرضی با برنامه‌ی ذخیره شده در حافظه



# ساختار CPU در ۸۸ / ۸۰۸۶



اجزاء CPU:

• واحد BIU

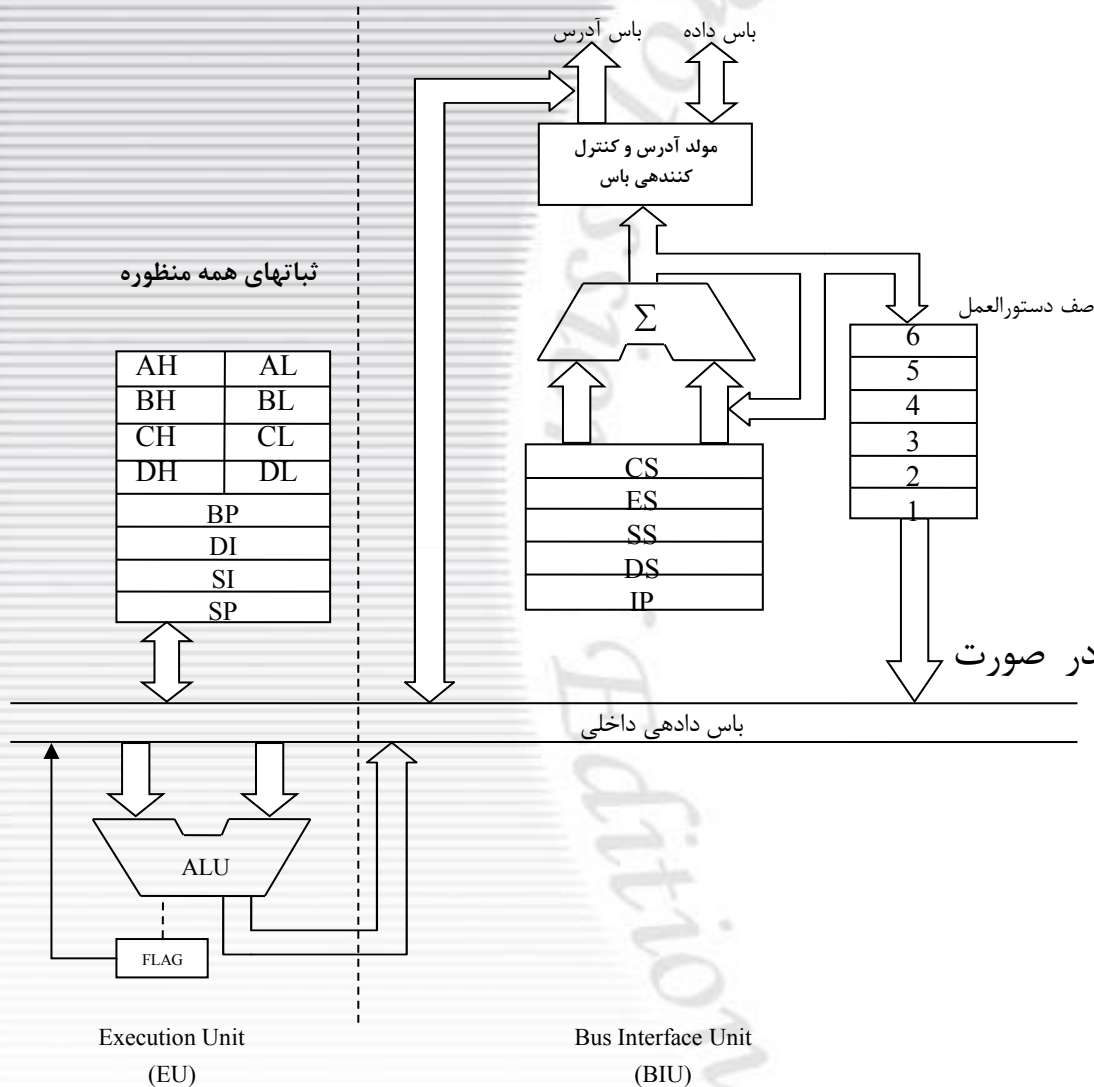
• واحد EU

شکل ۱

# ساختار CPU در ۸۸ / ۸۰۸۶

## • وظایف واحد BIU:

- تولید آدرس‌های فیزیکی
- گذاشتن آنها روی باس آدرس
- تولید سیگنال‌های کنترلی
- انتقال دستورالعمل‌ها به درون ریزپردازنده
- گذاشتن دستورالعمل‌ها در صف دستورالعمل
- بازنشانی (ریست کردن) صف دستورالعمل لزوم



# ساختار CPU های ۸۰۸۶ / ۸۸

## • وظایف واحد EU:

• کد دستورالعمل‌های برنامه و نیز داده‌ها را از **BIU** دریافت می‌کند

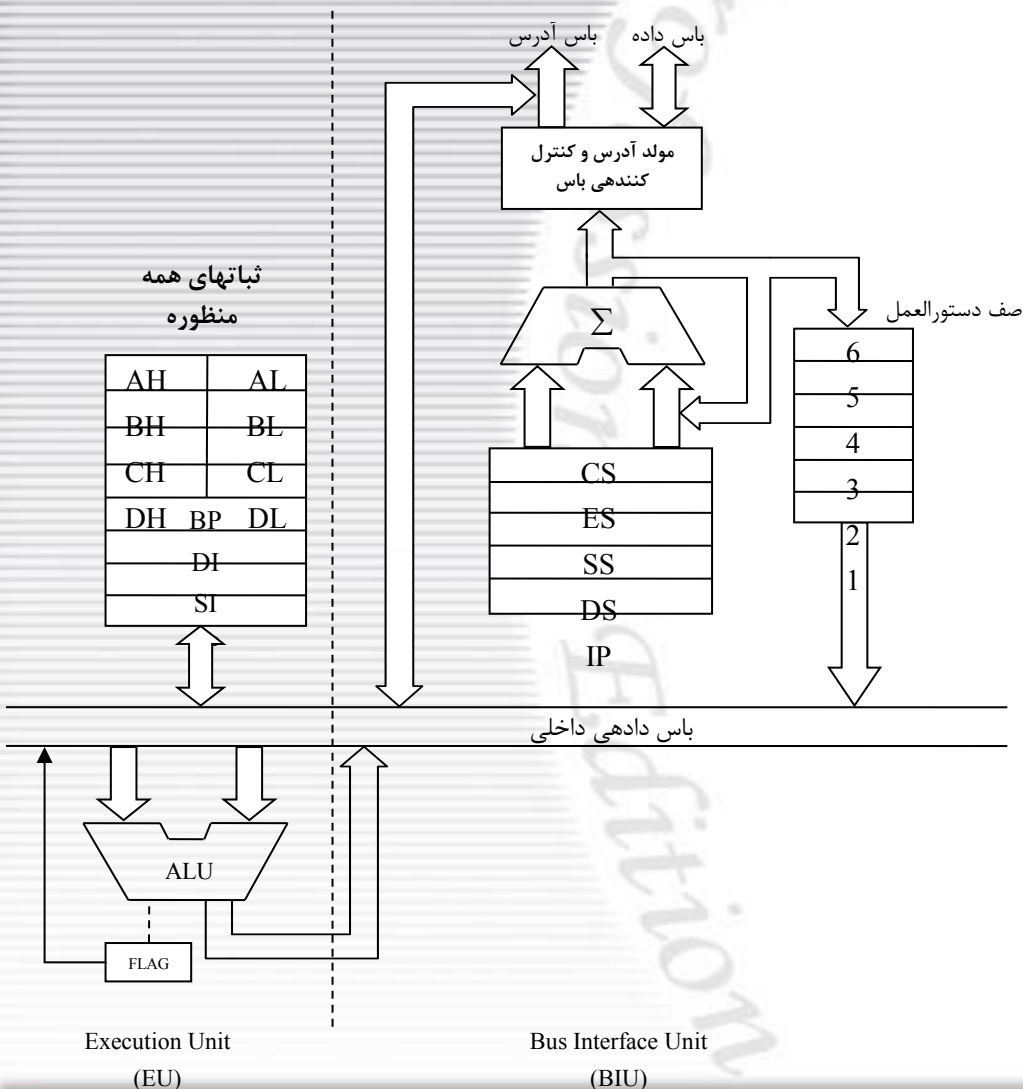
• دستورات را اجرا می‌کند

• نتایج را در ثبات‌های عمومی ذخیره می‌کند

• با برگرداندن این نتایج به واحد **BIU**، می‌توان آن را در مکان‌های حافظه یا پورت‌های خروجی نوشت.

• واحد **EU** هیچ ارتباطی با باس‌های سیستم ندارد و همه داده‌ها را از طریق **BIU** دریافت می‌کند یا می‌فرستد.

• با اجرای دستورالعمل‌ها پرچم‌ها را متاثر یا از وضعیت پرچم‌ها در اجرای دستورات شرطی استفاده می‌کند.



## ثبات‌های ۸۰۸۶

برنامه نویس ۸۸/۸۰۸۶ باید با ثبات‌های متنوع به کار رفته در واحدهای BIU و EU آشنا باشد.  
این ثبات‌ها در چند گروه قرار داده شده‌اند.

نام ثبات	نام کلی	بایت کم ارزش	بایت پر ارزش	نام گروه
Accumulator	AX	AL	AH	گروه ثبات‌های داده
Base	BX	BL	BH	
Count	CX	CL	CH	
Data	DX	DL	DH	
Stack pointer	-	SP		گروه ثبات‌های اشاره‌گر و اندیس
Base pointer	-	BP		
Source index	-	SI		
Destination index	-	DI		
Instruction pointer	-	IP		

## ثبات‌های ۸۰۸۶

گروه	ES	-	Extra
ثبات‌های	CS	-	Code
سگمنت	DS	-	Data
	SS	-	Stack
پرچم کنترل و وضعیت	FLAGS H	FLAGS L	-
			Status and Control flags



# ثبات‌های سگمنت پیش فرض و جایگزین

آدرس منطقی =  $(\text{Segment Register})0H + \text{آدرس فیزیکی}$  (شامل ۲۰ بیت)

آدرس فیزیکی در جدول زیر ارمشخص شده است.

آدرس منطقی	سگمنت‌های جایگزین	سگمنت پیش فرض	نوع رجوع به حافظه
IP	ندارد	CS	واکشی دستورالعمل
SP	ندارد	SS	عملیات پشته
آدرس موثر	CS, ES, SS	DS	داده عمومی
SI	CS, ES, SS	DS	منبع رشته
DI	ندارد	ES	مقصد رشته
آدرس موثر	CS, ES, SS	DS	BX بعنوان اشاره‌گر
آدرس موثر	CS, ES, DS	SS	BP بعنوان اشاره‌گر

- مثلاً اگر ثبات‌های  $IP=1000H$  و  $CS=B3FFH$  باشند، آدرس فیزیکی ساخته شده در BIU برابر است با  $B3FF0H + 1000H = B4FF0H$ .
- مثلاً اگر ثبات‌های  $SPI=5000H$  و  $SS=5000H$  باشند، آدرس فیزیکی ساخته شده در BIU برابر است با  $50000H + 5000H = 55000H$ .
- مثلاً اگر ثبات‌های  $SI=2000H$  و  $DS=A3FFH$  باشند، آدرس فیزیکی ساخته شده در BIU برابر است با  $A3FF0H + 2000H = A5FF0H$ .

# آدرس موثر

- آدرس‌های موثر به ترتیب زیر قابل تولید هستند:

آدرس موثر					
مود آدرس دهی	آدرس موثر				
مد آدرس دهی	جابجایی		ثبات پایه		ثبات شاخص
ثباتی غیر مستقیم	ندارد ندارد	+	BP یا BX ندارد	+	ندارد DI یا SI
شاخص دار	۱۲۸- تا ۱۲۷+	+	ندارد	+	DI یا SI
آدرس پایه	۱۲۸- تا ۱۲۷+	+	BP یا BX	+	ندارد
آدرس پایه و شاخص دار	ندارد	+	BP یا BX	+	DI یا SI
آدرس پایه و شاخص دار به همراه جابجایی	۱۲۸- تا ۱۲۷+	+	BP یا BX	+	DI یا SI

توصیف	عملیات سمبولیک	سگمنت مربوطه	عبارت یادآور	کد شیء (Object Code)	مد آدرس دهی
منبع داده درون دستورالعمل است	$AH \leftarrow 10H; AL \leftarrow 00$	کد	<code>MOV AX, 1000H</code>	<b>B8 00 10</b>	بلافصل
منبع و مقصد داده، ثباتهای CPU است.	$DX \leftarrow CX$	درون CPU	<code>MOV DX, CX</code>	<b>8B D1</b>	ثبات
آدرس حافظه درون دستورالعمل است.	$AH \leftarrow [1000H]$	داده	<code>MOV AH, [MEMBDS]</code>	<b>8A 26 00 10</b>	مستقیم
آدرس حافظه در یک ثبات شاخص یا اشاره گر قرار دارد.	$AL \leftarrow [SI]; AH \leftarrow [SI+1]$ $IP \leftarrow [DI+1:DI]$ $[BP] \leftarrow [BP]+1$ $[BX+1:BX] \leftarrow [BX+1:BX]-1$	داده داده پشته داده	<code>MOV AX, [SI]</code> <code>JMP [DI]</code> <code>INC BYTE PTR [BP]</code> <code>DEC WORD PTR [BX]</code>	<b>8B 04</b> <b>FF 25</b> <b>FE 46 00</b> <b>FF 0F</b>	غیر مستقیم ثباتی
آدرس حافظه مجموع ثبات شاخص و یک جابجایی درون دستور است.	$AL \leftarrow [SI+6]; AH \leftarrow [SI+7]$ $IP \leftarrow [DI+7:DI+6]$	داده داده	<code>MOV AX, [SI+6]</code> <code>JMP [DI+6]</code>	<b>8B 44 06</b> <b>FF 65 06</b>	شاخص دار
آدرس حافظه مجموع ثباتهای پایه (BP یا BX) و یک جابجایی درون دستور است.	$AL \leftarrow [BP+2]; AH \leftarrow [BP+3]$ $IP \leftarrow [BX+3:BX+2]$	پشته داده	<code>MOV AX, [BP+2]</code> <code>JMP [BX+2]</code>	<b>8B 46 02</b> <b>FF 67 02</b>	آدرس پایه
آدرس حافظه مجموع ثباتهای شاخص و ثبات پایه است.	$AL \leftarrow [BX+SI]$ $AH \leftarrow [BX+SI+1]$	داده	<code>MOV AX, [BX+SI]</code>	<b>8B 00</b>	آدرس پایه و شاخص دار
		داده	<code>JMP [BX+DI]</code>	<b>FF 21</b> <b>FE 02</b> <b>FF 0B</b>	
		پشته	<code>INC BYTE PTR [BP+SI]</code>		
		پشته	<code>DEC WORD PTR [BP+DI]</code>		
آدرس حافظه مجموع ثبات شاخص، پایه و جابجایی درون دستور است.	$AL \leftarrow [BX+SI+5]$ $AH \leftarrow [BX+SI+6]$	داده	<code>MOV AX, [BX+SI+5]</code>	<b>8B 40 05</b>	آدرس پایه و شاخص دار به همراه جابجایی
		داده	<code>JMP [BX+DI+5]</code>	<b>FF 61 05</b>	
		پشته	<code>INC BYTE PTR [BP+SI+5]</code>	<b>FE 42 05</b>	
		پشته	<code>DEC WORD PTR [BP+DI+5]</code>	<b>FF 4B 05</b>	
آدرس حافظه می‌دهد، ثبات SI درون سگمنت داده است، و آدرس حافظه‌ی مقصد، ثبات DI در سگمنت اضافه است.	$[ES:DI] \leftarrow [DS:SI]$ اگر $DF=0$ سپس: $SI \leftarrow SI+1; DI \leftarrow DI+1$ در غیر اینصورت: $SI \leftarrow SI-1; DI \leftarrow DI-1$	داده، اضافی	<code>MOVS</code>	<b>A4</b>	رشته

## ثبات پرچم‌ها در ۸۰۸۶

• جدول زیر چگونگی تعریف بیت‌ها برای ثبات ۱۶ بیتی پرچم را نشان می‌دهد.

• شش بیت از این ثبات شاخص‌های وضعیت هستند که ویژگی‌های نتیجه آخرین محاسبات ریاضی و منطقی انجام شده را بیان می‌کند:

Flags H								Flags L							
X	X	X	X	OF	DF	IF	TF	SF	ZF	X	AF	X	PF	X	CF

عملکرد	نام پرچم	بیت
پرچم <b>Carry</b> : اگر بر بیت پر ارزش نتیجه، <b>Carry</b> یا <b>Borrow</b> اتفاق افتد، این پرچم 1 شده و در غیر اینصورت 0 خواهد بود.	CF	0
پرچم <b>Parity</b> : این پرچم 1 می شود اگر تعداد بیت‌های 1 در هشت بیت مرتبه پایین نتیجه، زوج باشد. در غیر اینصورت 0 می شود.	PF	2
اگر از چهار بیت کم ارزش <b>AL</b> ، <b>Carry</b> یا <b>Borrow</b> اتفاق افتد، مقدار این پرچم برابر 1 و گرنه 0 می شود.	AF	4
پرچم <b>Zero</b> : اگر نتیجه صفر باشد، این پرچم 1 و گرنه 0 می شود	ZF	6
پرچم <b>Sign</b> : این پرچم مقدار پرارزش ترین بیت نتیجه را می گیرد. (بیت علامت)	SF	7
پرچم <b>Single-step</b> : وقتی این پرچم 1 باشد، بعد از اجرای دستورالعمل بعدی، یک وقفه <b>single-step</b> اتفاق می افتد. با به وجود آمدن وقفه ی <b>single-step</b> ، این بیت 0 می شود	TF	8
پرچم <b>Interrupt-enable</b> : وقتی این پرچم 1 شود، وقفه های قابل <b>mask</b> شدن باعث می شوند که CPU، کنترل برنامه را به مکان بردار وقفه منتقل کند.	IF	9
پرچم <b>Direction</b> : 1 بودن این پرچم، موجب می گردد که دستورات رشته ای به طور خودکار ثبات اندیس مربوطه را کاهش دهند، اگر 0 باشد، افزایش خودکار صورت می گیرد.	DF	10
پرچم <b>Overflow</b> : اگر نتیجه محاسبات علامت دار انجام شده قابل قرار گرفتن در تعداد بیت‌های عملوند مقصد نباشد (سرریز رخ دهد)، این پرچم 1 می شود.	OF	11

## پرچم‌ها

- مقدار 1 در پرچم TF, عملکرد پردازنده را در مد تک‌گامی (single step) قرار می‌دهد.
- این مد برای عیب‌یابی برنامه‌ها بسیار مفید است.
- در اینصورت کنترل برنامه بعد از اجرای هر دستورالعمل، به مکان خاصی از حافظه که برنامه نویس قبلاً آدرس آن را مشخص کرده است منتقل می‌شود.
- معمولاً برنامه‌ای برای نمایش دادن همه ثبات‌ها و بیت‌های CPU در آن مکان ذخیره شده است.
- بدین ترتیب بعد از اجرای هر دستور برنامه‌نویس می‌تواند مقادیر ثبات‌ها و پرچم‌ها را بررسی کند.

## پرچم ها

• هنگامی که بیت پرچم IF (پرچم وقفه) مقدار 1 داشته باشد، خط ورودی مربوط به وقفه‌های خارجی (INTR) فعال می‌شود.

• پرچم DF با دستورات "انتقال بلوکی" (که انتقال رشته ای "String" نیز نامیده می‌شوند) به کار می‌رود.

• هنگامی که پرچم DF در وضعیت 1 قرار دارد، اشاره گر حافظه بلوکی به صورت خودکار کاهش می‌یابد و اگر در وضعیت 0 باشد افزایش می‌یابد.

MOVSB ;  $(ES(0)+DI) \leftarrow (DS(0)+SI)$   
; Increment or decrement SI  
; Increment or decrement DI

# واکشی و اجرای دستورالعمل

اجرا	واکشی	اجرا	واکشی	اجرا	واکشی	
------	-------	------	-------	------	-------	--

زمان →

(الف)

BIU

واکشی	واکشی	واکشی	واکشی	واکشی	واکشی	خواندن داده	واکشی*	واکشی*	واکشی*	واکشی	واکشی	
-------	-------	-------	-------	-------	-------	-------------	--------	--------	--------	-------	-------	--

EU

اجرا	انتظار	اجرا <sup>+</sup>	اجرا	اجرا	اجرا	انتظار	اجرا	اجرا	اجرا <sup>+</sup> *	انتظار	اجرا	
------	--------	-------------------	------	------	------	--------	------	------	---------------------	--------	------	--

\*: این بایت‌ها دور ریخته می‌شوند.

<sup>+</sup>: این دستور درخواست داده‌ای را دارد که در صف موجود نیست.

<sup>+</sup>\*: دستور پرش اتفاق افتاده است.

(ب)



# واکشی و اجرای دستورالعمل

دو حالت وجود دارند که واحد EU را به مد انتظار می‌برند:

- اولین حالت زمانی اتفاق می‌افتد که دستور اجراشونده نیاز به دسترسی به یک خانه‌ی حافظه برنامه دارد که در صف موجود نیست.

- حالت دوم زمانی اتفاق می‌افتد که EU بخواهد یک دستور پرش (jump) را اجرا کند. در این حالت کنترل برنامه به یک آدرس جدید که در ترتیب قبلی قرار ندارد، منتقل می‌شود.

یک حالت وجود دارد که باعث می‌شود واکشی دستورات در BIU به تعویق افتد و آن زمانی است که یک دستور کند در EU اجرا می‌شود.

- مثل دستور AAM در ۸۰۸۶ که به ۸۳ پالس کلاک نیاز دارد تا کامل شود. بعد از واکشی دستور، صف کاملاً پر می‌شود و لذا در حین اجرای این دستور عملیات واکشی به حالت تعویق در می‌آید.

# واکشی و اجرای دستورالعمل

مزیت های ساختار خط لوله ای:

- سرعت اجرای بیشتر دستورالعمل ها

- از آنجا که همواره چند دستور بعدی در صف قرار دارد واحد BIU می تواند با سرعت نسبتا کمتری به حافظه مراجعه کند.

- لذا می توان از حافظه های کندتری استفاده کرد بدون اینکه عملکرد کلی سیستم متاثر شود.

## انواع سیکل‌های باس (سیکل ماشین)

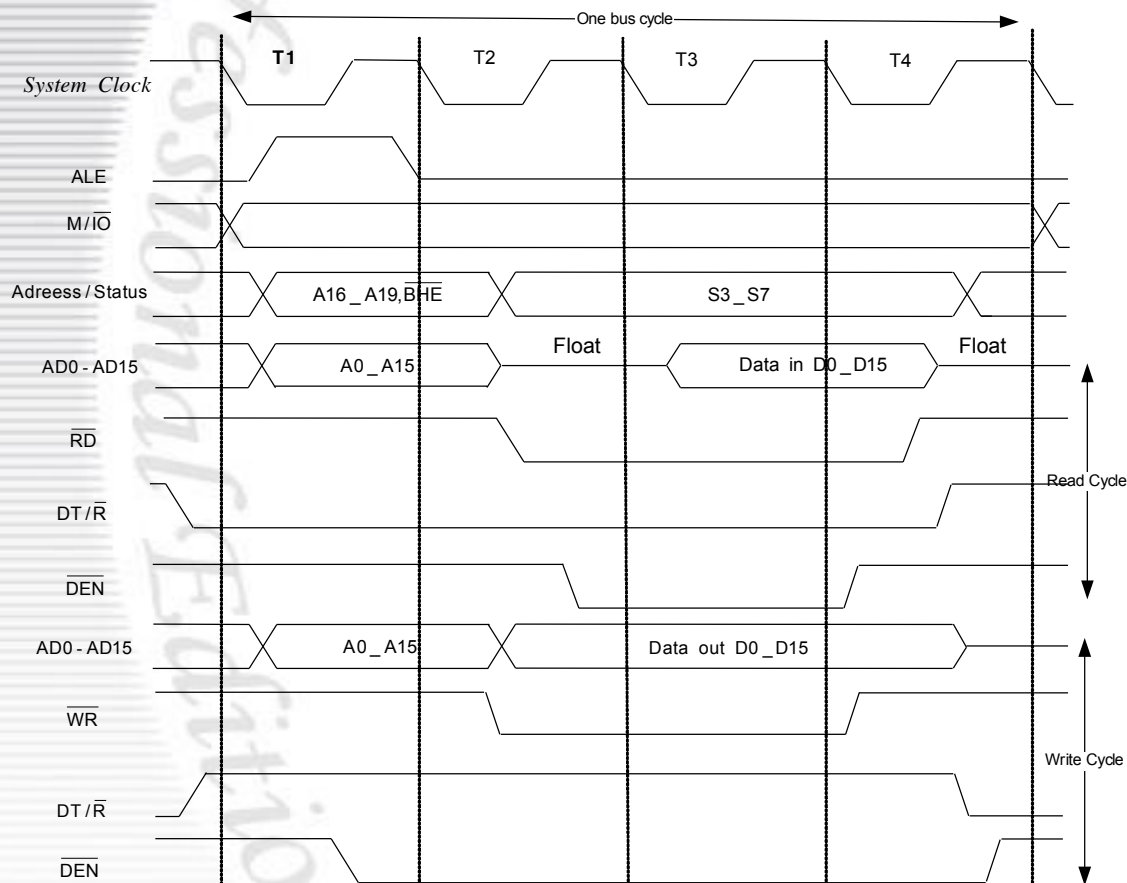
---

### انواع سیکل‌های باس

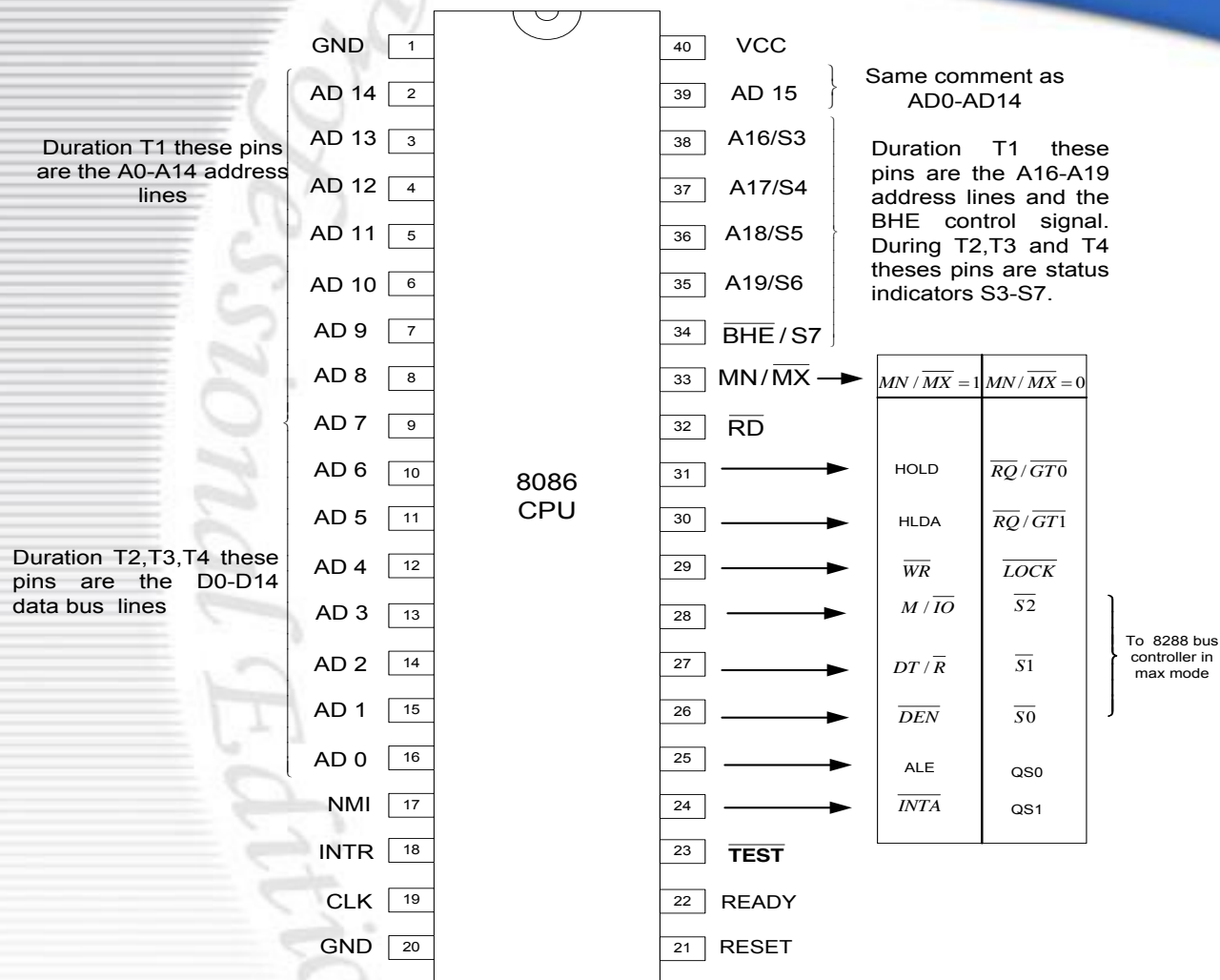
- سیکل خواندن از حافظه (MR)
- سیکل نوشتن در حافظه (MW)
- سیکل خواندن از I/O (IOR)
- سیکل نوشتن در I/O (IOW)
- سیکل باس بیکار (عملیات داخل CPU انجام می‌شود که نیازی به دسترسی به باس ندارد)

# زمان‌بندی سیکل‌های باس

زمان‌بندی سیکل‌های باس خواندن و نوشتن برای حافظه و IO در ریزپردازنده ۸۰۸۶



# توصیف پایه‌های ۸۰۸۶



شکل ۲

## توصیف پایه‌های ۸۰۸۶

- ویژگی جالب توجه ۸۰۸۶، توانایی کارکرد در دو مد کاری حداقل و حداکثر است.
- مد حداقل برای یک سیستم ساده تک‌پردازنده‌ای بر روی یک مدار چاپی در نظر گرفته شده است.
- در حالیه مد حداکثر برای یک سیستم پیچیده‌تر که بردهای جداگانه‌ای برای واحدهای حافظه و I/O دارند مطلوب است.
- مد حداکثر پردازنده‌های کمکی همچون ۸۰۸۷ (NDP) مخصوص انجام عملیات ریاضی خاص و ۸۰۸۹ (IOP) که مخصوص مدیریت و ارتباط با وسایل جانبی چون دیسک سخت است را پشتیبانی می‌کند.

NDP: Numerical Data Processor

IOP: I/O Processor

## توصیف پایه‌های ۸۰۸۶

---

در ۸۰۸۶ از **مالتی پلکس زمانی** برای رفع مشکل کمبود پایه‌ها استفاده شده است.

پایه‌های داده (به نام AD0-AD15) در حین پالس ساعت **T1** حاوی **خطوط آدرس** هستند.

پایه‌های داده در پالس های **T2 تا T4** حاوی **خطوط داده** هستند.

مدار دی‌مالتی پلکس ویژه‌ای برای استخراج داده و آدرس به طور مجزا لازم است.

## توصیف پایه‌های ۸۰۸۶

**باس داده AD0-AD15:** این ۱۶ پایه باس داده دو جهت CPU را تشکیل می‌دهند. این خطوط تنها در بازه زمانی سیکل‌های حالت T2 تا T4 معتبر هستند. در زمان سیکل حالت T1 این خطوط ۱۶ بیت کم ارزش آدرس حافظه یا پورت را نگه می‌دارند.

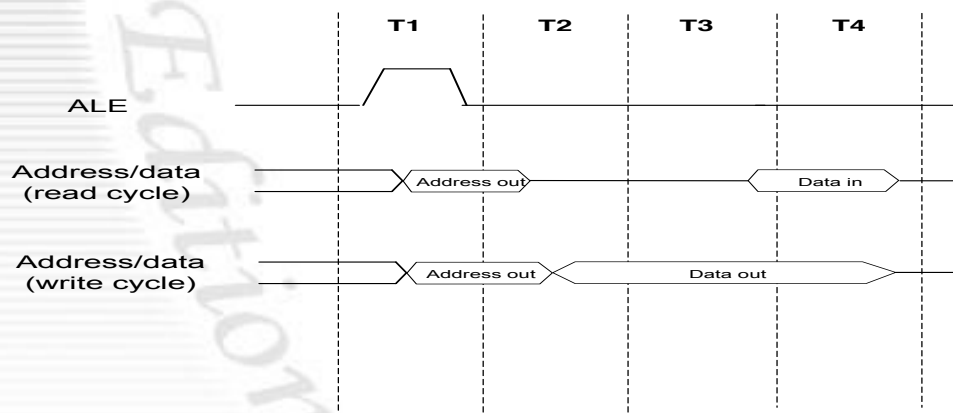
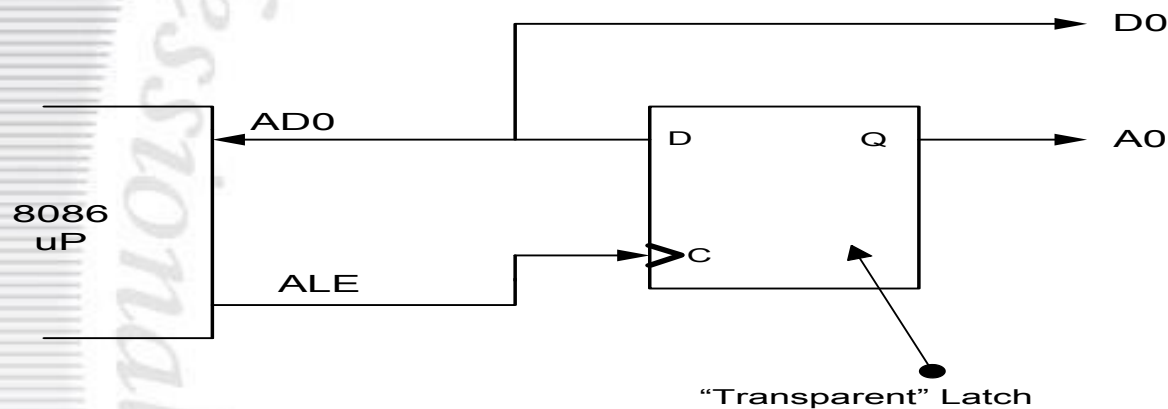
**باس آدرس AD0-AD15 و A16/S3-A19/S6:** این خطوط مربوط به ۲۰ بیت آدرس CPU هستند و به پردازنده اجازه دسترسی به ۱۰۴۸۵۷۶ مکان حافظه (1MB) را می‌دهند. این خطوط خروجی تنها در زمان سیکل حالت T1 معتبر بوده و در زمان سیکل‌های T2 تا T4 به خطوط داده و وضعیت سوئیچ می‌کنند.

**ALE (Address Latch Enable):** سیگنال خارج شده بر این پایه را می‌توان برای دی‌مالتی‌پلکس کردن خطوط آدرس، داده و وضعیت بر AD0-AD15 ، A16/S3-A19/S6 بکار برد. هر سیکل با یک پالس ALE در حین سیکل حالت T1 شروع می‌شود. تقریباً نزدیک به انتهای T1، که ALE از سطح بالا به پایین می‌رود، ۲۰ بیت آدرس معتبر هستند. لذا این سیگنال را می‌توان به عنوان پالس لچ‌کننده آدرس بکار برد.



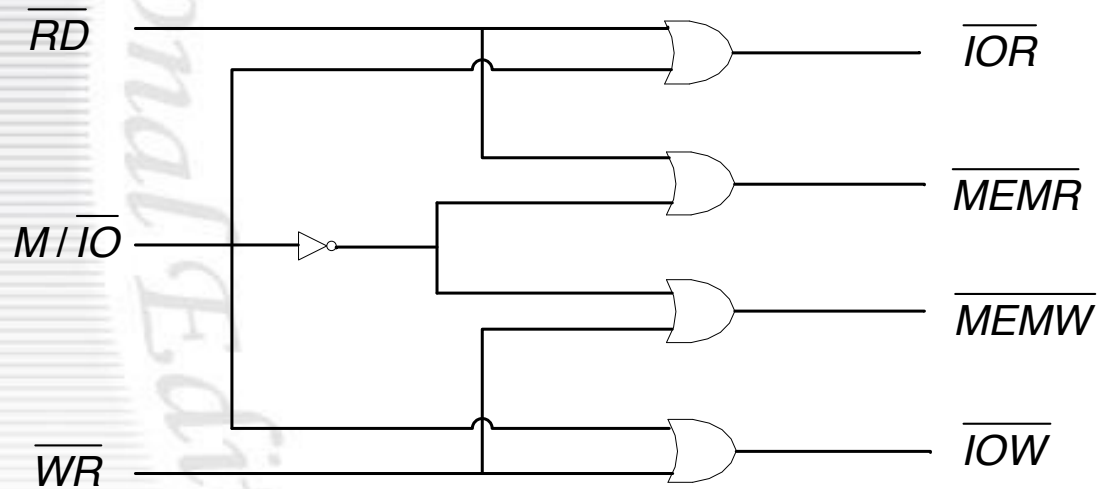
# دی مالتی پلکس کردن باس آدرس/داده ۸۰۸۶/۸۰۸۸

در شکل زیر نحوه لچ کردن یک خط آدرس/داده (خط AD0)، در زمانی که آدرس بر روی خط قرار دارد نشان داده شده است.

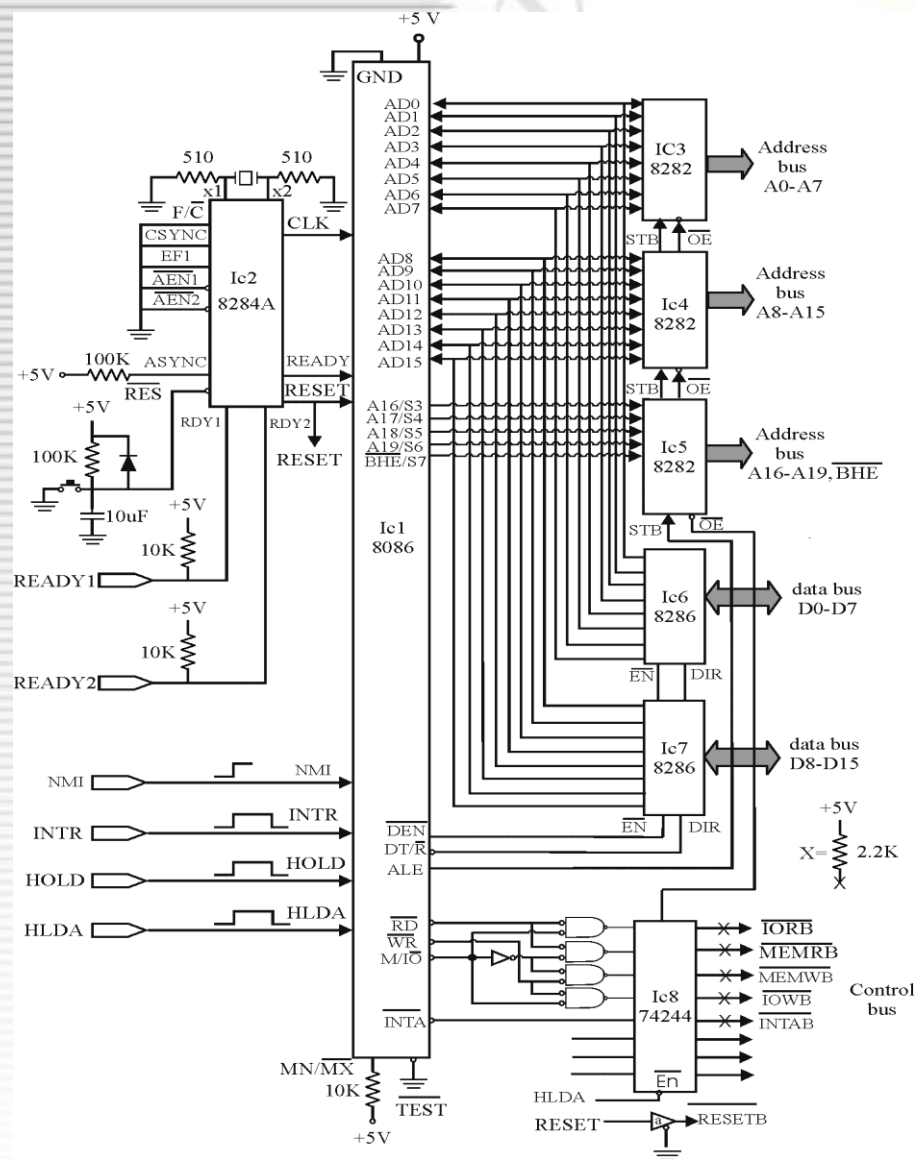


## تولید سیگنال‌های دسترسی به حافظه و ورودی/خروجی

- تولید چهار سیگنال مورد نیاز برای کنترل حافظه و ورودی-خروجی به منظور نوشتن و خواندن از آنها بصورت زیر صورت می‌گیرد:



# سخت افزار مبتنی بر ۸۰۸۶



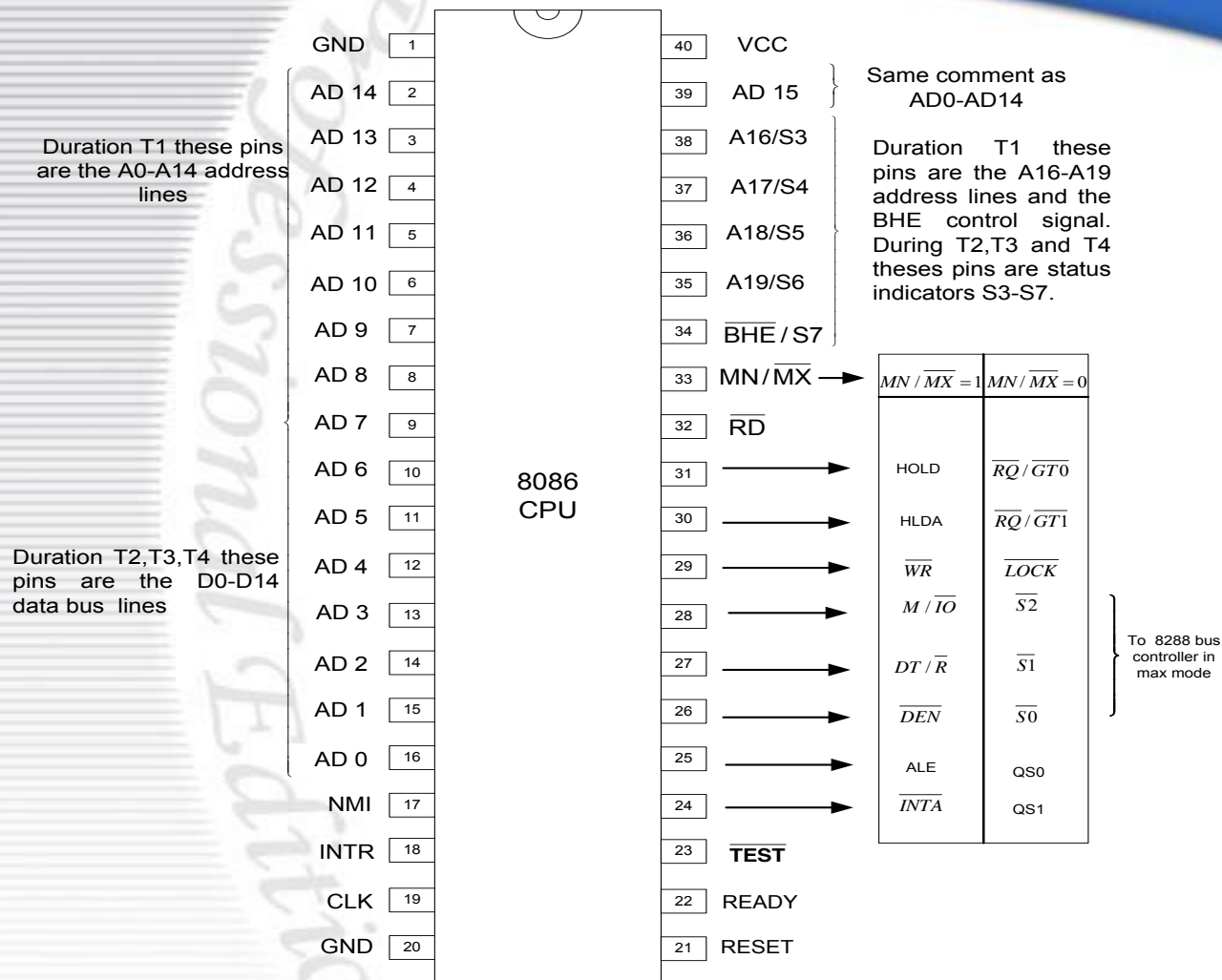
- نحوه اتصال گذرگاه آدرس/داده به لچ‌های دی‌مالتی پلکسر و جداسازی اطلاعات آدرس

## توصیف پایه‌های ۸۰۸۶

- پایه‌های وضعیت **S3** و **S4** نشان دهنده آن هستند که در یک لحظه زمانی خاص چه ثبات سگمنتی توسط واحد **BIU** برای تولید آدرس فیزیکی در حال استفاده است.
- کارکرد پایه‌های **S5**، **S6** و **S7** نیز در جدول زیر مشخص شده‌اند:

S4	S3	دسترسی سیکل باس
0	0	سگمنت اضافی
0	1	سگمنت پشته
1	0	سگمنت کد (یا هیچکدام)
1	1	سگمنت داده
S5, IF (پرچم فعال سازی وقفه). S6, 0 (بیان می‌کند که باس در اختیار ۸۰۸۶ است). S7, (بیت زاپاس است و کاربردی ندارد).		

# توصیف پایه‌های ۸۰۸۶



شکل ۲

## توصیف پایه‌های ۸۰۸۶

---

توصیف پایه **Bus high enable**

**Low** شدن این پایه، دسترسی به بانک بالا (بانک آدرس‌های فرد) را فراهم می‌کند.

## توصیف پایه‌های ۸۰۸۶

### پایه Data transmit/receive ( $\overline{DT/R}$ ):

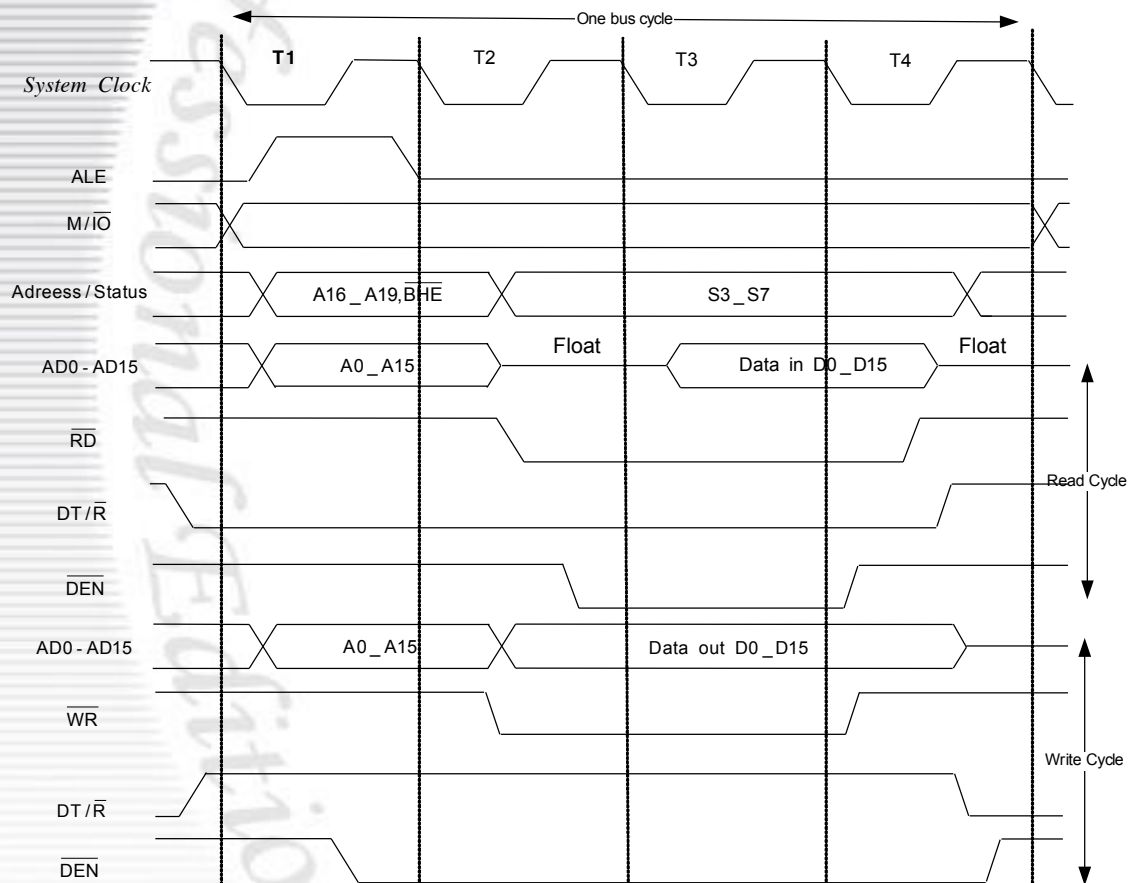
- این سیگنال برای کنترل جهت انتقال داده در بافرهایی که ممکن است به باس داده سیستم متصل شده باشند، بکار می‌رود.
- اگر این پایه در سطح پایین (low) قرار گیرد، عملیات خواندن و اگر در سطح بالا قرار گیرد بیانگر عملیات نوشتن خواهد بود.
- بافرهای دو جهته برای این منظور بکار گرفته می‌شوند.

### پایه Data enable ( $\overline{DEN}$ ):

- این سیگنال به همراه  $\overline{DT/R}$  برای فعال کردن مجموعه‌ای از بافرهای متصل شده به باس داده سیستم بکار می‌رود.
- این سیگنال از درگیری در باس که ممکن است به خاطر تلاش همزمان دو مدار برای نوشتن در یک خط باس به وجود آید جلوگیری می‌کند.
- بافرهای باس داده تا سیکل حالت T2 غیرفعال خواهند بود و بعد از آن خطوط آدرس/داده درگیر، حامل آدرس حافظه یا پورت نخواهند بود.

# زمان بندی سیکل های باس

زمان بندی سیکل های باس خواندن و نوشتن برای حافظه و IO در ریزپردازنده ۸۰۸۶





## توصیف پایه‌های ۸۰۸۶

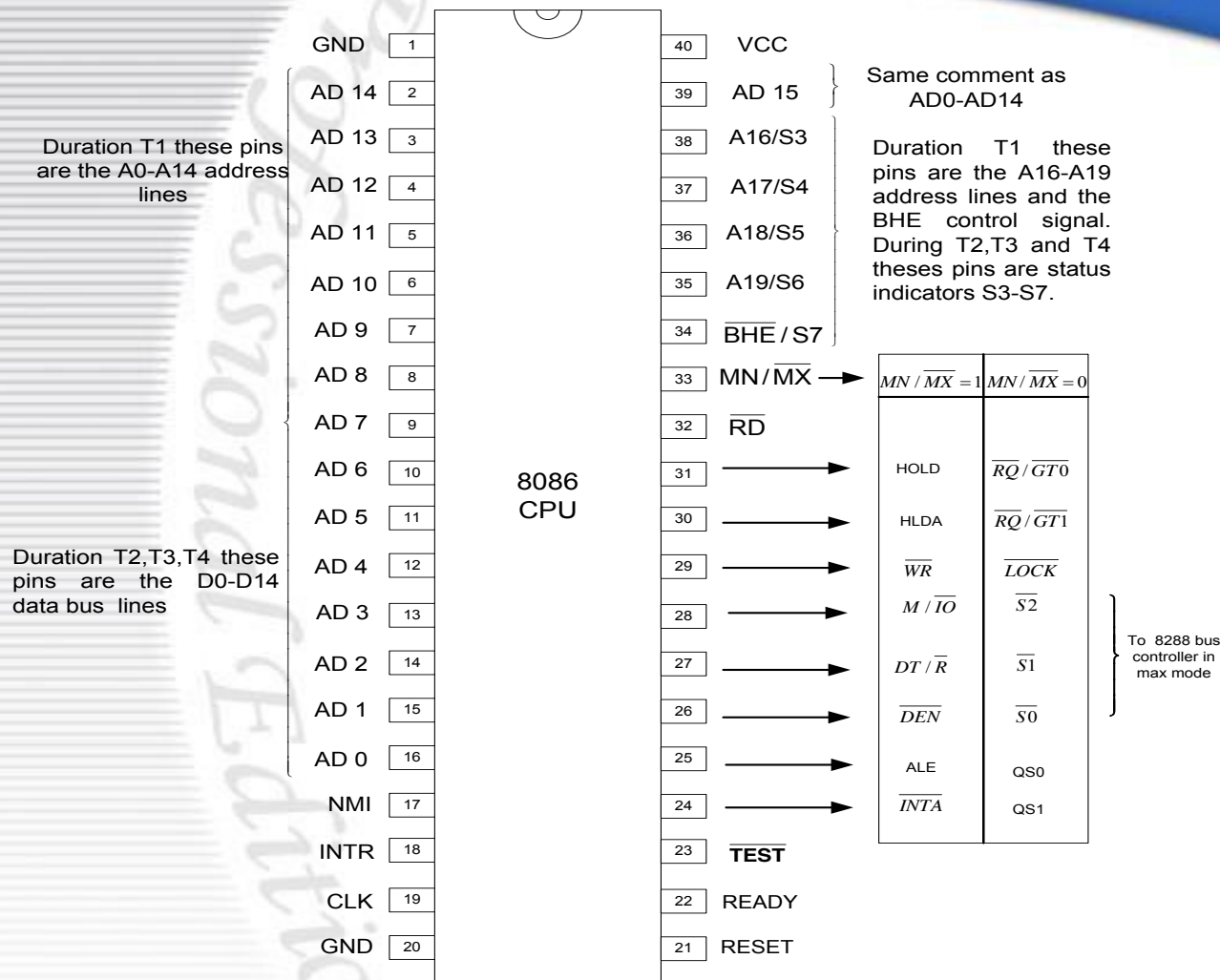
پایه  $MN/\overline{MX}$  برای تعیین حالت حداقل/حداکثر (Minimum/maximum mode)

- ۸۰۸۶ می‌تواند در یکی از مدهای حداقل یا حداکثر کار کند.
- عملکرد پایه‌های ۲۴ تا ۳۲ بسته به مد کاری ریزپردازنده دارند و با تغییر سطح سیگنال اِعمالی به پایه  $MN/\overline{MX}$  عملکرد این پایه ها تغییر می‌کند.

### پایه بازنشانی (Reset):

- زمانی که یک پالس با سطح بالا به این ورودی اعمال شود، ۸۰۸۶ فعالیت جاری خود را خاتمه داده و یک عمل بازنشانی را اجرا می‌کند.
- با هر بازنشانی وضعیت کار قبلی ریزپردازنده از بین خواهد رفت.
- بازنشانی معمولاً زمانی که ریزپردازنده روشن می‌شود یا زمانی که دچار خطا شده است، انجام می‌گردد.

# توصیف پایه‌های ۸۰۸۶

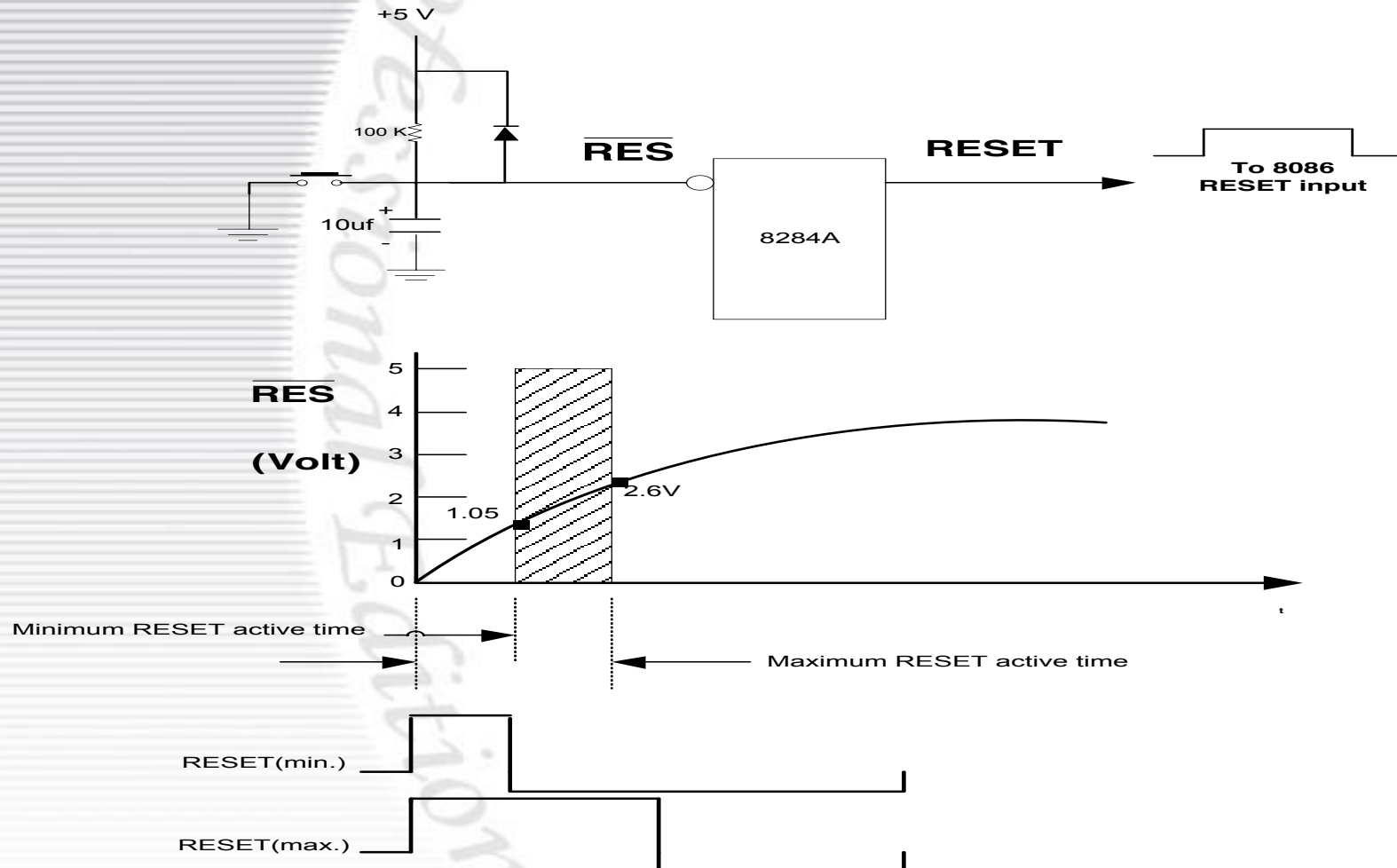


شکل ۲

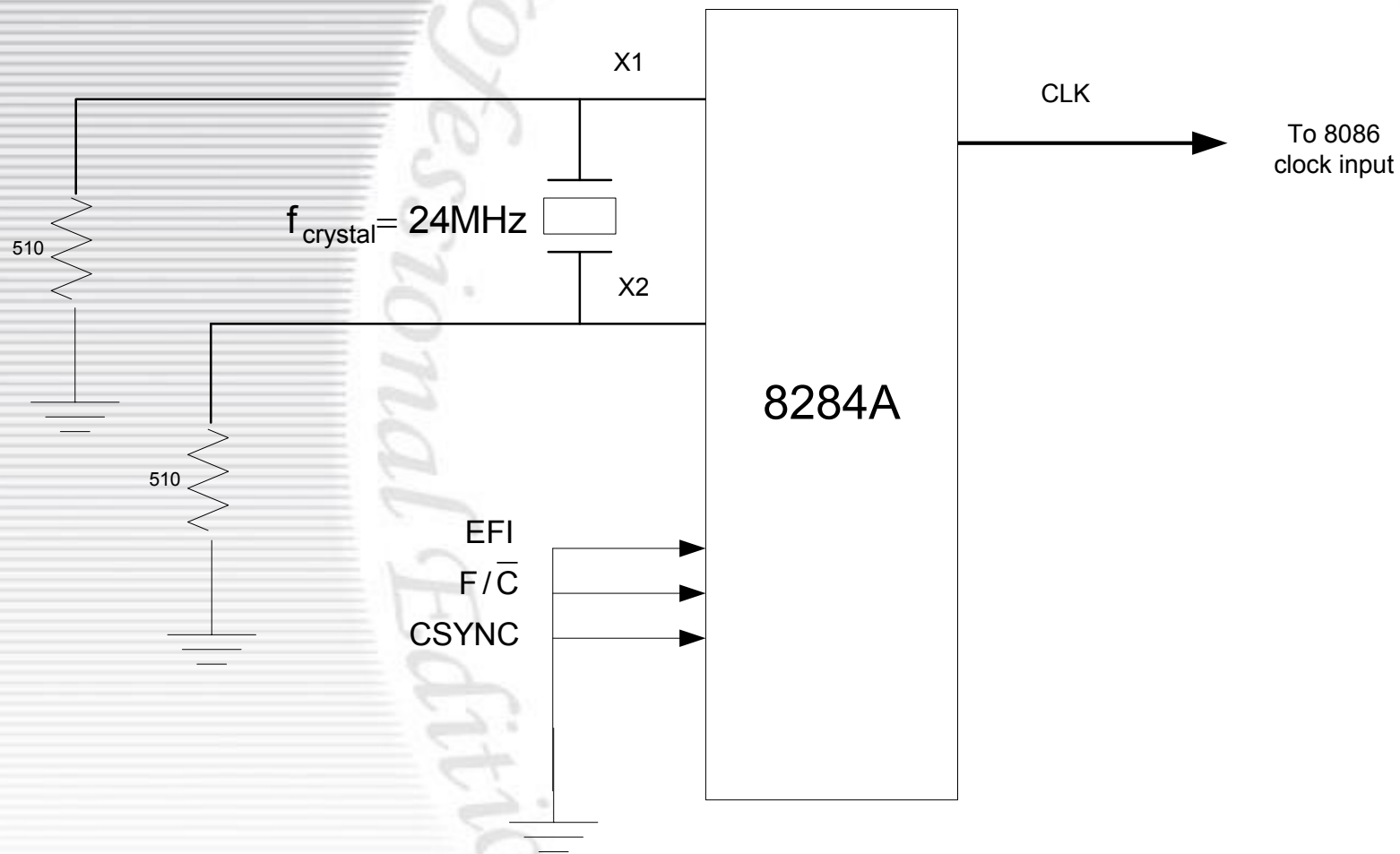
## وضعیت ریزپردازنده بعد از بازنشانی

CPU	محتوا
پرچم ها	پاک می شوند
ثبات اشاره گر دستورالعمل (IP)	0000H
ثبات CS	FFFFH
ثبات DS	0000H
ثبات SS	0000H
ثبات ES	0000H
صف	تهی

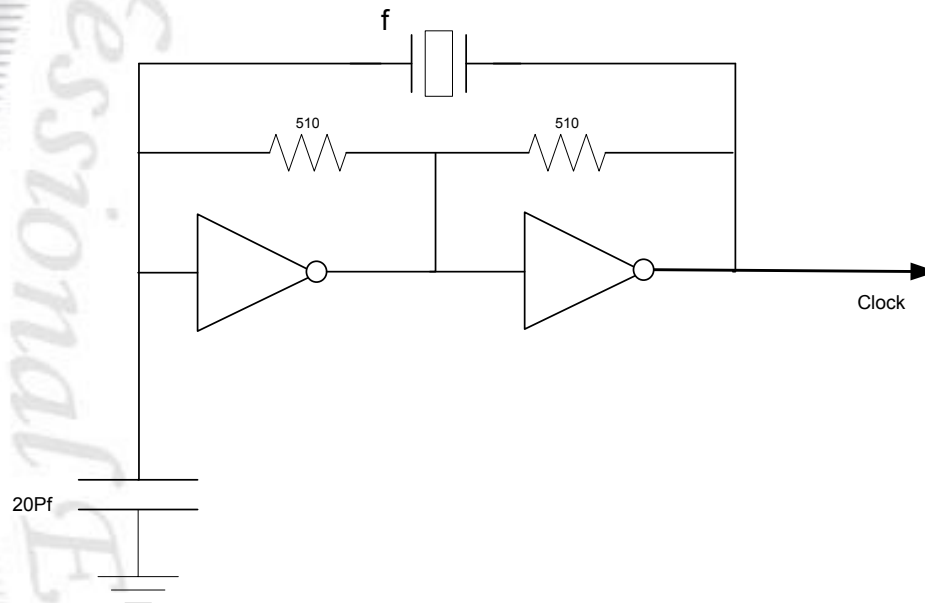
# بازنشانی ۸۰۸۶ با کمک تراشه 8284A



## تولید سیگنال ساعت توسط تراشه 8284A



# تولید سیگنال ساعت خارجی توسط گیت NOT



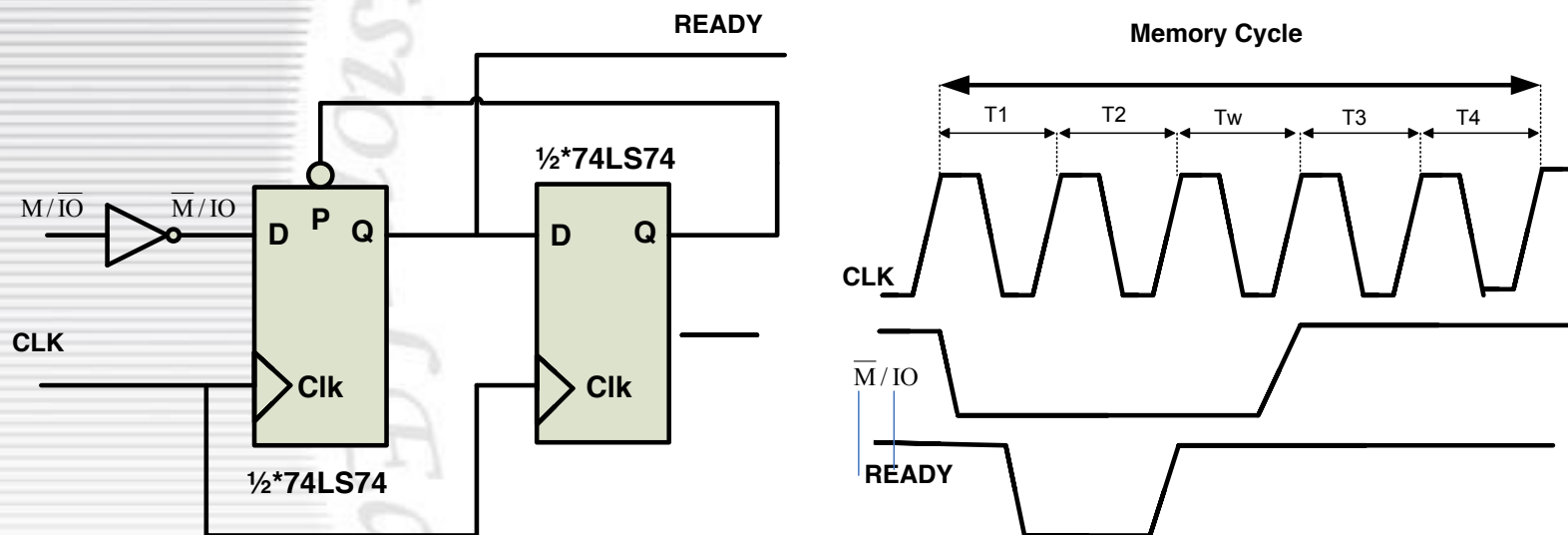
## توصیف پایه‌های ۸۰۸۶

### پایه READY:

- سیگنال READY در لبه بالا رونده T2 نمونه‌برداری می‌شود. اگر این سیگنال در سطح پایین بود (آماده نبودن) یک سیکل حالت T3 دیگر اضافه می‌شود.
- این سیکل آنقدر تکرار می‌شود تا سرانجام سیگنال READY در سطح بالا قرار گیرد. سیگنال READY را معمولاً حافظه‌ها و ورودی/خروجی‌های کندی که نمی‌توانند داده را به سرعت زمان‌بندی CPU آماده کنند بکار می‌گیرند.

## مدار تولید سیگنال READY در زمان مراجعه به حافظه

مدار مورد استفاده برای فعال کردن سر READY ریزپردازنده برای مدت ۲ سیکل ساعت:



در صورت نیاز به low بودن سیگنال READY برای مدت ۲ سیکل ساعت، باید یک عدد فلیپ فلاپ بعد از فلیپ فلاپ سمت راست اضافه گردد و خروجی آن به Preset فلیپ فلاپ اول متصل شود.



## توصیف پایه‌های ۸۰۸۶

- پایه‌های وقفه **Interrupt** (  $\overline{INTA}$ , NMI, INTR ):• INTR و NMI درخواست‌های وقفه سخت‌افزاری هستند که دقیقاً مثل وقفه‌های نرم‌افزاری عمل می‌کنند.
- NMI حساس به لبه بالا رونده است و INTR حساس به سطح بالا است.
- ورودی INTR را می‌توان با تغییر دادن مقدار بیت پرچم IF پوشاند (غیر فعال کرد) ولی ورودی NMI قابل پوشش نبوده و لذا همواره باید سرویس‌دهی شود و معمولاً این وقفه برای مسائل حیاتی سیستم مثل بروز خطا در تغذیه توان سیستم یا سیستم حافظه بکار می‌رود.
- پایه  $\overline{INTA}$  یک پایه خروجی است و زمانی که وقفه INTR پذیرفته شود Low می‌شود.

## توصیف پایه‌های ۸۰۸۶

وقتی **NMI** فعال می‌شود، کنترل برنامه به طور خودکار به مکانی از حافظه که آدرس آن در محدوده آدرس **00008-0000BH** قرار دارد منتقل می‌شود.

وقتی **INTR** فعال می‌شود، یک سیکل تایید وقفه انجام می‌شود. این کار شبیه خواندن از حافظه است ولی با این تفاوت که به جای سیگنال  $\overline{RD}$ ، سیگنال  $\overline{INTA}$  فعال می‌شود.

**CPU** بعد از آن انتظار دارد یک عدد ۸ بیتی که بیانگر نوع وقفه است بر بیت های کم ارزش باس داده قرار گیرد. بعد از آن کنترل برنامه به مکانی از حافظه که آدرس آن در مکان " $4 * \text{عدد بیانگر نوع وقفه}$ " تا " $4 * \text{عدد بیانگر نوع وقفه} + 3$ " قرار دارد منتقل می‌شود.

## توصیف پایه‌های ۸۰۸۶

$\overline{Test}$  : این ورودی با دستور WAIT بکار می‌رود.

هنگام مواجهه با دستور WAIT، اگر ورودی  $\overline{Test}$  در سطح بالا باشد، اجرای برنامه‌ها به حالت تعویق درآمده و CPU وارد مد بیکار می‌شود.

تنها زمانی که این ورودی به سطح پایین برگردد، اجرای برنامه با دستور بعد از WAIT ادامه می‌یابد.

معمولا راه‌اندازی ورودی  $\overline{Test}$  را تراشه کمکی ۸۰۸۷ که مخصوص انجام محاسبات ریاضی پیچیده است (NDP) انجام می‌دهد. لذا دستور WAIT بعد از دستوراتی به کار می‌رود که در آنها به داده‌هایی ارجاع داده می‌شود که ۸۰۸۷ بر آنها کار کرده است.

با بررسی این سیگنال از دسترسی CPU به نتیجه حافظه، قبل از آنکه NDP عملیات محاسباتی را به پایان رساند، جلوگیری به عمل می‌آید.

## توصیف پایه‌های ۸۰۸۶

(HOLD and HLDA): HOLD یک ورودی است که در سطح بالا فعال شده و باعث می‌شود CPU همه خطوط باس خود را در وضعیت مدار باز قرار دهد. این کار ارتباط CPU را از حافظه و I/O قطع می‌کند و پردازشگر دومی می‌تواند به آنها دسترسی داشته باشد.

این کار را "دسترسی مستقیم به حافظه (DMA)" گویند. سیگنال کنترلی خروجی HLDA درخواست DMA را برای کنترل کننده DMA تایید می‌کند.

(VCC and GND): ۸۰۸۶ به یک تغذیه ۵ ولتی نیاز دارد و دارای دو پایه زمین می‌باشد. مصرف توان آن به ازای سرعت‌های مختلف تراشه در جدول ۱ آمده است.

Hold and hold acknowledge  
Power and Ground

## توصیف پایه‌های ۸۰۸۶

---

### پایه LOCK:

- قراردادن پیشوندی بنام Lock در مقابل یک دستورالعمل موجب می شود که در زمان اجرای آن دستورالعمل، پایه خروجی به همان نام LOCK فعال شده و به ریزپردازنده ها و وسایل بیرونی بفهماند که در زمان اجرای آن دستورالعمل تقاضای باس ننمایند.

## مشخصات سرعت و توان مصرفی نسخه‌های مختلف ۸۰۸۶

پردازنده	$f_{max}$ (MHz)	$I_{cc}(max)$ mA	توان تلفاتی W
۸۰۸۶	۵	۳۴۰	۱.۷
۸۰۸۶-۲	۸	۳۵۰	۱.۷۵
۸۰۸۶-۱	۱۰	۳۶۰	۱.۸
۸۰۸۸	۵	۳۴۰	۱.۷
۸۰۸۸-۲	۸	۳۵۰	۱.۷۵
P۸۰۸۸	۵	۲۵۰	۱.۲۵

جدول ۱

