



هسته پردازنده مرکزی

در میکروکنترلر ATmega16

فهرست مطالب

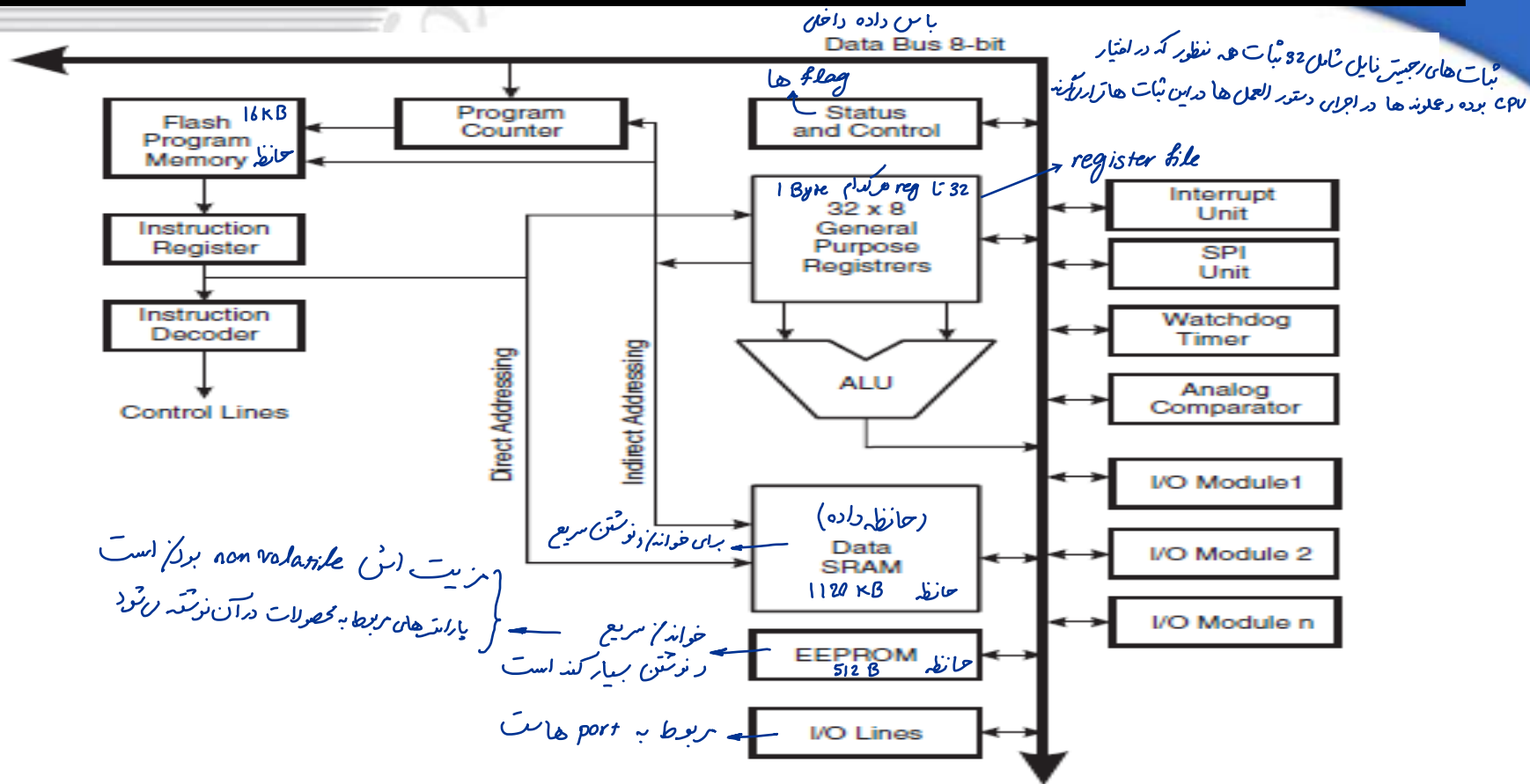
- هسته پردازنده مرکزی میکروکنترلر ATmega16
 - حافظه پشته و ثبات اشاره گر به پشته
 - ثبات وضعیت
 - فایل ثبات همه منظوره
 - واحد ALU
 - زمانبندی اجرای دستورالعمل ها
 - بازنشانی و کنترل وقفه ها

هسته پردازنده مرکزی میکروکنترلر ATmega16

• وظیفه اصلی هسته واحد پردازشگر مرکزی در این میکروکنترلر تضمین اجرای صحیح برنامه است.

• بنابراین واحد پردازشگر مرکزی می‌بایست قادر باشد که به حافظه‌ها دسترسی داشته، محاسبات را انجام دهد، اجزاء مختلف میکروکنترلر را کنترل کند و قف‌ها را مدیریت کند.

هسته پردازنده مرکزی میکروکنترلر ATmega16



دیاگرام بلوکی معماری میکروکنترلر ATmega16

هسته پردازنده مرکزی میکروکنترلر ATmega16

• میکروکنترلر ATmega16 از معماری هاروارد با حافظه‌های جداگانه و گذرگاه‌های متفاوت برای برنامه و داده استفاده می‌کند.

• هنگامی که یک دستورالعمل اجرا می‌شود، دستورالعمل بعدی از حافظه برنامه پیش‌واکشی می‌شود.

دستوری که در کلاک قبل از t_{clock} می‌باشد

• این مفهوم موجب می‌شود که بتوان هر دستورالعمل را در یک پالس ساعت اجرا نمود. حافظه برنامه از نوع حافظه فلش با قابلیت برنامه‌ریزی درون سیستمی است.

اگر دستورات در یک clock pulse انجام می‌شود

هسته پردازنده مرکزی میکروکنترلر ATmega16

register file

- فایل ثبات با امکان دسترسی سریع، شامل ۳۲ ثبات ۸ بیتی همه منظوره با امکان دسترسی در یک چرخه ساعت است.
- این مهم امکان اجرای عملیات ALU در تنها یک چرخه ساعت را فراهم می سازد.
- در یک عملیات ALU نمونه، در یک چرخه ساعت، دو عملوند از فایل ثبات خارج شده، عملیات روی آنها اجرا و نتیجه در فایل ثبات ذخیره می شود.

هسته پردازنده مرکزی میکروکنترلر ATmega16

- جریان اجرای برنامه توسط پرش‌های شرطی و غیرشرطی و دستورالعمل‌های صدازدن زیرروال‌ها برقرار می‌شود. این مهم باعث می‌شود که مستقیماً بتوان به تمام فضای حافظه دسترسی داشت.

با دستورات *Jump*، *Call* و *return* می‌توان حافظه عوض کرد. *Call vs. Jump* بعد از *Jump* برخواهیم تست ← با استفاده از *stack* و در *Call* در *subroutine*؛ *RET* داریم

- اغلب دستورالعمل‌های ATmega16 دارای یک قالب کلمه ۱۶ بیتی هستند.

- هر آدرس مربوط به حافظه برنامه شامل یک دستورالعمل ۱۶ بیتی و یا ۳۲ بیتی است.

هسته پردازنده مرکزی میکروکنترلر ATmega16

• فضای حافظه برنامه فلش به دو قسمت شامل قسمت برنامه راه انداز و قسمت برنامه کاربردی تقسیم می شود.

Boot

Application

• هر دو قسمت دارای بیت های قفل برای محافظت نسبت به نوشتن و خواندن هستند.

• دستورالعمل SPM که امکان نوشتن در قسمت برنامه کاربردی حافظه فلش را فراهم می کند، باید در قسمت برنامه راه انداز قرار داشته باشد.

ثبات وضعیت

- ثبات وضعیت اطلاعاتی در خصوص نتیجه آخرین عملیات محاسباتی اجرا شده را در خود نگه می‌دارد.
- این اطلاعات می‌توانند در جریان برنامه در دستورات شرطی استفاده شوند.
- لازم بذکر است که ثبات وضعیت هنگام ورود به روال وقفه ذخیره و یا هنگام خروج از آن بطور اتوماتیک بازیابی نمی‌شود و اینکار باید توسط نرم‌افزار انجام شود.

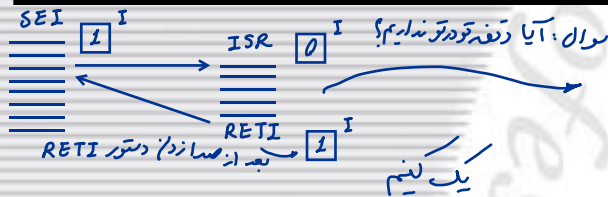
e.g. (اگر تغییر صورت دتر)

set carry flag
clear carry flag

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

دقت براساسی

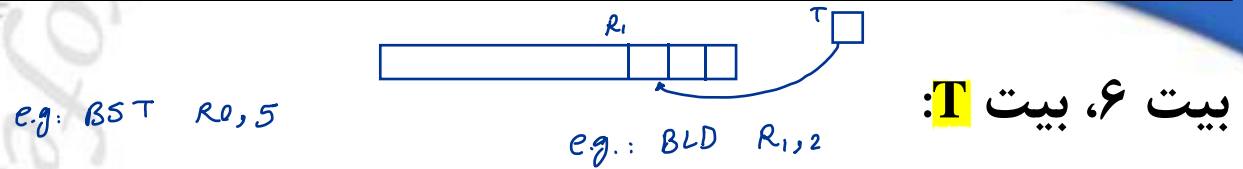
ثبات وضعیت



بیت ۷، بیت I (بیت فعال سازی وقفه):

- برای فراهم ساختن امکان پذیرش وقفه‌ها می‌بایست این بیت را فعال نمود.
 - اگر این بیت غیر فعال شود هیچ وقفه‌ای پذیرفته نخواهد شد.
 - بعد از وقوع هر وقفه، این بیت بطور خودکار صفر می‌شود و پس از اجرای دستورالعمل **RETI** دوباره فعال می‌شود تا امکان پذیرش وقفه‌های بعدی فراهم گردد.
- Return from Interrupt*
- این بیت می‌تواند مستقیماً توسط دستورالعمل‌های **SEI** و **CLI** فعال و غیرفعال شود.

ثبات وضعیت



- دستورالعمل‌های **BLD** (Bit Load) و **BST** (Bit Store) این بیت را به عنوان مقصد و مبدا برای بیتی که عملیات روی آن انجام گرفته بکار می‌برند.

- یک بیت از یک ثبات در مجموعه ثبات‌های عمومی همه منظوره می‌تواند توسط دستورالعمل **BST** به درون بیت T کپی شود.

بیت شماره b از رجیستر Rr را در flag T قرار بده

BST	Rd, b	$T \leftarrow Rr(b)$	Bit Store from Register to T
-----	-------	----------------------	------------------------------

- یک بیت در T می‌تواند توسط دستورالعمل **BLD** در یک بیت در یک ثبات در مجموعه ثبات‌های عمومی کپی شود.

ثبات وضعیت

بیت ۵، بیت H یا بیت نیمه نقلی (Half Carry):

- این بیت نشاندهنده رخداد یک نیمه توازن (۲ بر یک ایجاد شده بعد از بیت چهارم) در بعضی از عملیات‌های محاسباتی است.

- این بیت برای عملیات‌های بی‌سی‌دی مفید است.

بیت ۴، بیت S یا بیت علامت:

$$S = V \oplus N$$

این بیت بیانگر یاء مانع جمع بین پرچم‌های N و V می‌باشد.

ثبات وضعیت

بیت ۳، بیت ^{overflow} V یا سرریز:
این بیت، سرریز را در عملیات محاسباتی مکمل ۲ پشتیبانی می کند.

Overflow occurs when the carry flag and the overflow flag are different.

بیت ۲، بیت N یا منفی:
این بیت منفی بودن نتیجه عملیات محاسباتی یا منطقی را نشان می دهد.

ثبات وضعیت

بیت ۱، بیت Z یا بیت صفر:

• این بیت نشان‌دهنده صفر شدن نتیجه یک عملیات محاسباتی و یا منطقی است.

بیت ۰، بیت C یا بیت نقلی:

• این بیت نشان‌دهنده رخداد بیت نقلی (۲ بر یک بعد از بیت هشتم) در عملیات‌های محاسباتی و یا منطقی است.

فایل ثبات همه منظوره

- مجموعه ثبات‌های عمومی برای اجرای دستورالعمل‌های RISC تقویت شده میکروکنترلرهای AVR بهینه شده است.
 - واحد محاسباتی-منطقی عملیات منطقی و محاسباتی را بین ثبات‌های عمومی و یا بین ثبات و یک مقدار ثابت پشتیبانی می‌نماید.
 - عملیات ساده‌ی تک‌ثباتی نیز در ALU انجام می‌گیرد.
- دو بک‌تی هم داریم.

فایل ثبات همه منظوره

این ثبات‌ها با شماره‌های **R1** الی **R32** در شکل زیر دیده می‌شوند:

ساختار مجموعه
ثبات‌های عمومی
همه منظوره

General
Purpose
Working
Registers

7	0	Addr.
R0		\$00
R1		\$01
R2		\$02
...		
R13		\$0D
R14		\$0E
R15		\$0F
R16		\$10
R17		\$11
...		
R26		\$1A
R27		\$1B
R28		\$1C
R29		\$1D
R30		\$1E
R31		\$1F

X-register Low Byte
X-register High Byte
Y-register Low Byte
Y-register High Byte
Z-register Low Byte
Z-register High Byte

6 تای آخر خاص منظوره ← رتبه‌ها در نشان آدرس گذاشته
گرفته شوند 3 رجیستر 2 بایتی (X, Y, Z)

فایل ثبات همه منظوره

- اغلب دستورالعمل‌هایی که روی ثبات‌های عمومی انجام می‌شوند، به تمام ثبات‌ها دسترسی مستقیم داشته و اغلب آنها دستورالعمل‌هایی هستند که در یک چرخه ساعت انجام می‌شوند.

- به هر ثبات یک آدرس در فضای حافظه داده اختصاص داده شده است که موجب می‌شود این ثبات‌ها مستقیماً به اولین ۳۲ محل در فضای حافظه داده کاربر تصویر شوند.

- چون این ثبات‌ها در درون واحد پردازشگر مرکزی قرار دارند و بطور فیزیکی در فضای **SRAM** پیاده‌سازی نشده‌اند، این ساختار حافظه انعطاف و سرعت بالایی را در دسترسی به ثبات‌ها فراهم می‌کند.

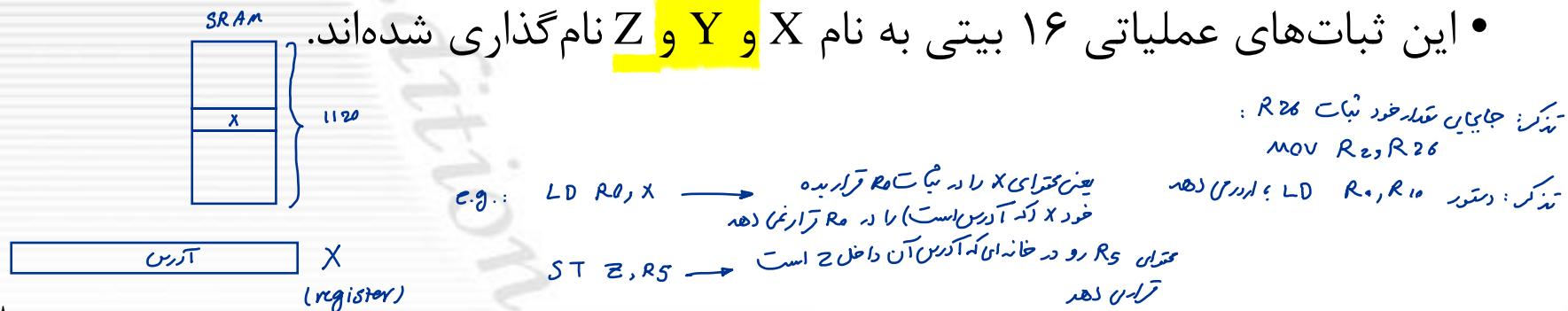
- واحد محاسبات و منطق ALU، بطور مستقیم با تمام ۳۲ ثبات مجموعه ثبات‌های عمومی همه منظوره در ارتباط است.

فایل ثبات همه منظوره، ثبات های X، Y و Z

• ۶ ثبات از بین ۳۲ ثبات مجموعه ثبات‌های عمومی یعنی ثبات های R26 تا R31 به عنوان ۳ جفت ثبات ۱۶ بیتی هستند که به صورت اشاره گر برای **آدرس دهی غیرمستقیم فضای داده** استفاده می‌شوند.

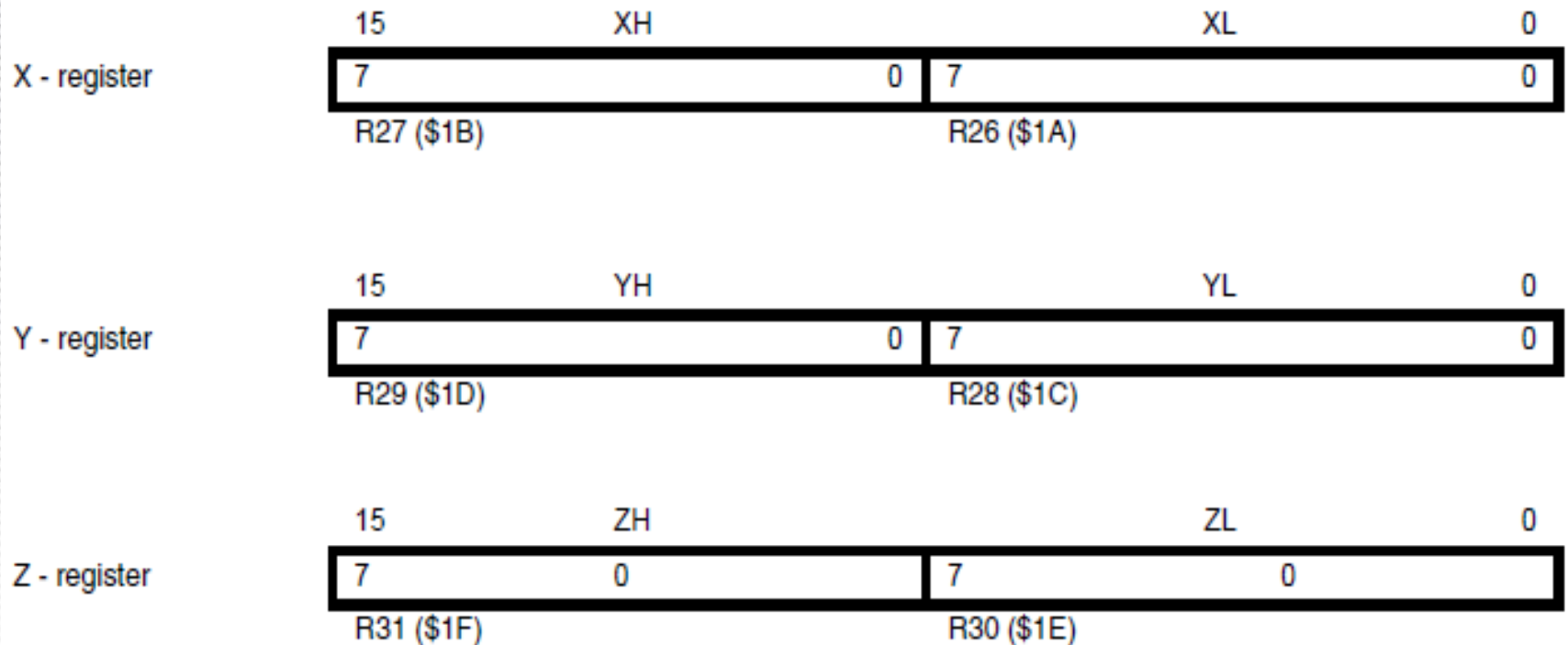
• این ثبات‌ها موجب کارایی بیشتر در محاسبات آدرس می‌شوند.

• یکی از این ۳ اشاره گر به منظور اشاره گر آدرس برای مراجعه به **جداول جستجو** استفاده می‌شود.



فایل ثبات همه منظوره، ثبات های X، Y و Z

شکل های زیر این ثبات ها و موقعیت آنها را در مجموعه ثبات های عمومی همه منظوره نشان می دهند



ثبات های X، Y و Z

واحد ALU

- عملیات‌های واحد ALU به سه دسته اصلی شامل عملیات ریاضی، منطقی و عملیات روی بیت‌ها تقسیم می‌شوند.
- بعضی از انواع میکروکنترلرهای ATmega16، یک ضرب‌کننده با امکان انجام ضرب علامت‌دار و بدون علامت را پشتیبانی می‌کنند.

واحد ALU

• دسترسی تمامی دستورالعمل‌های مربوط به **عملیات بین ثبات‌ها** به صورت مستقیم و در تنها یک چرخه دستورالعمل انجام می‌شود.

در این بین ۵ دستورالعمل محاسباتی - منطقی استثنا می‌باشند. این دستورالعمل‌ها **SBCI, SUBI, CPI, ANDI** و **ORI** هستند که عملیات را بین **مقادیر ثابت** و **ثبات‌ها** انجام می‌دهند.

↓
Immediate

ORI R0, 50H

• استثناء دیگر دستورالعمل **LDI** است که به منظور بار کردن بی‌درنگ **مقادیر ثابت** به کار می‌رود.

e.g.: *LDI R0, 50H*

• این دستورالعمل‌ها بر روی نیمه دوم ثبات‌ها (**R16...R31**) در مجموعه ثبات‌های عمومی اجرا می‌شوند.

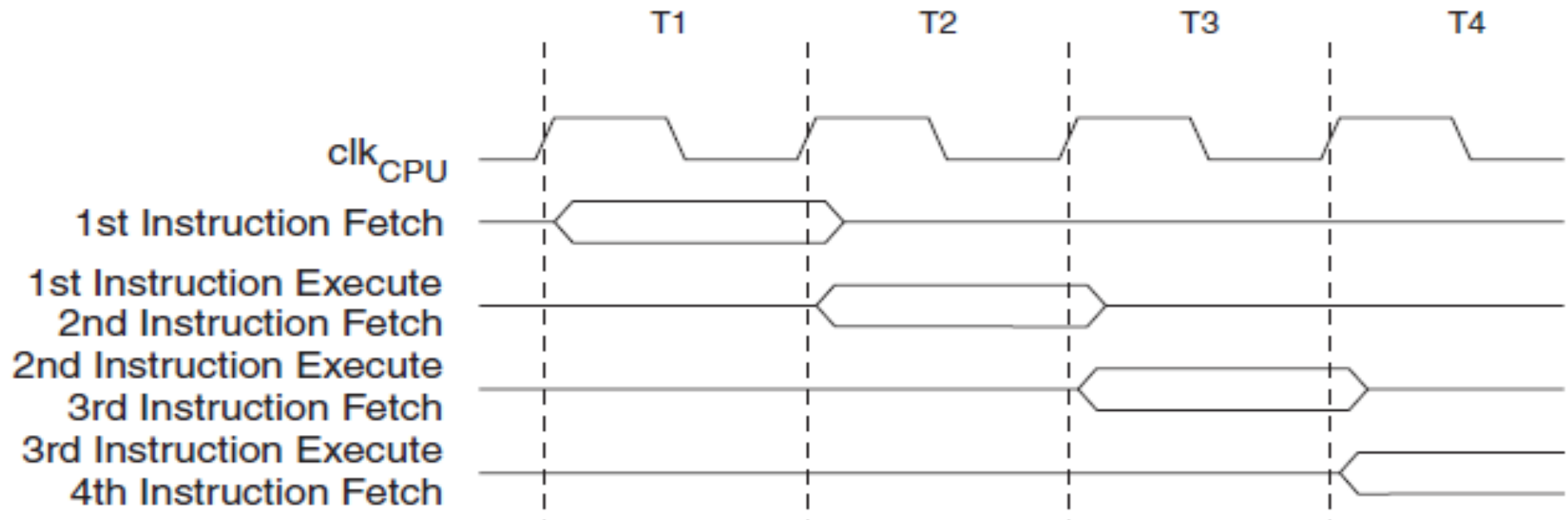
• دستورالعمل‌های **SBC, SUB, CP, AND** و **OR** و تمامی دیگر دستورالعمل‌ها که مابین **دو ثبات** و یا **یک ثبات** انجام می‌شوند، بر روی تمام ثبات‌های عمومی (**R0...R31**) اجرا می‌گردند.

زمانبندی اجرای دستورالعمل‌ها

- چند اسلاید بعد مفاهیم زمانبندی دسترسی عمومی را برای اجرای دستورالعمل‌ها ارائه می‌نماید.
- میکروکنترلر ATmega16 توسط مولد ساعت راه‌اندازی می‌شود.
- این ساعت مستقیماً از منبع ساعت تراشه تامین می‌شود. هیچگونه تقسیم در ساعت داخلی صورت نمی‌گیرد.

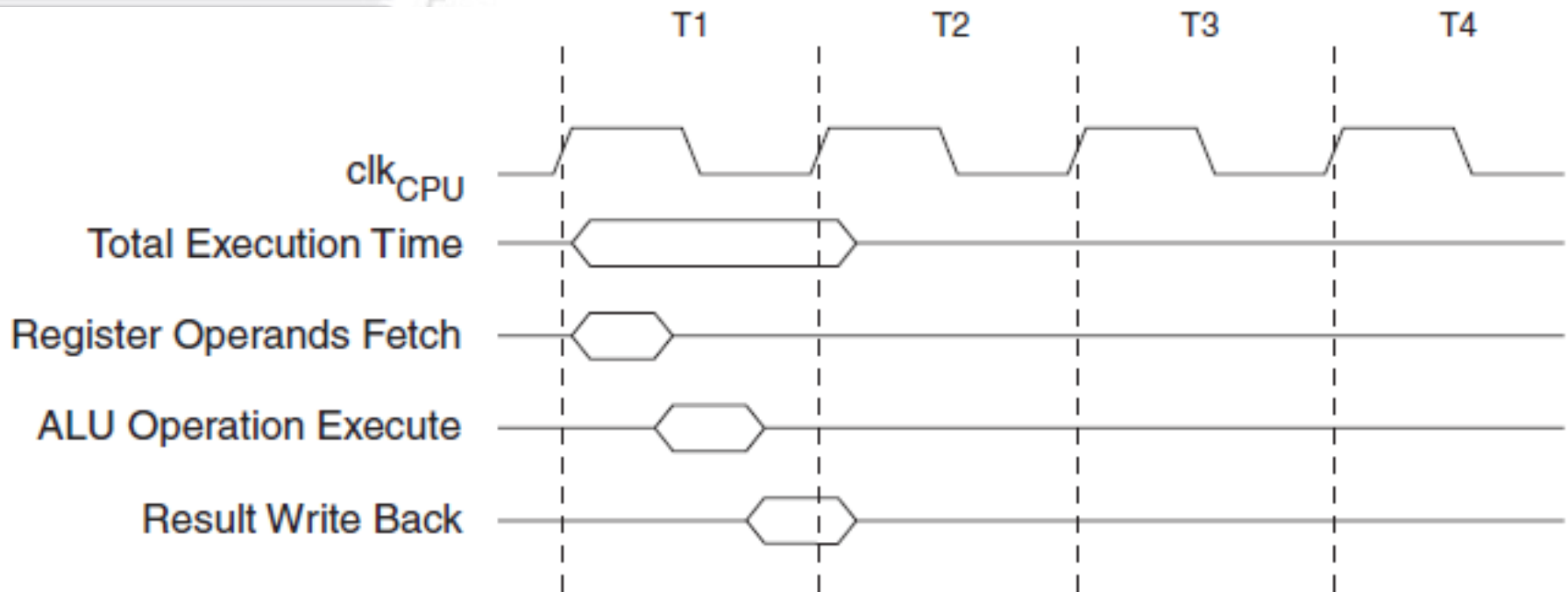
زمانبندی اجرای دستورالعمل‌ها

- شکل زیر چگونگی انجام واکشی و اجرای موازی دستورالعمل‌ها را منطبق بر معماری **هاروارد** و مفهوم فایل ثبات با دسترسی سریع نشان می‌دهد.
- این مفهوم اساسی معماری **خط لوله** برای دستیابی به سرعت ۱ میلیون دستورالعمل در ثانیه بازاء سیگنال ساعت ۱ مگاهرتز است.



زمانبندی اجرای دستورالعمل‌ها

- شکل زیر مفهوم زمان‌بندی داخلی برای فایل ثابت را نشان می‌دهد.



واکشی موازی دستورالعمل‌ها و اجرای دستورالعمل‌ها

زمانبندی اجرای دستورالعمل‌ها

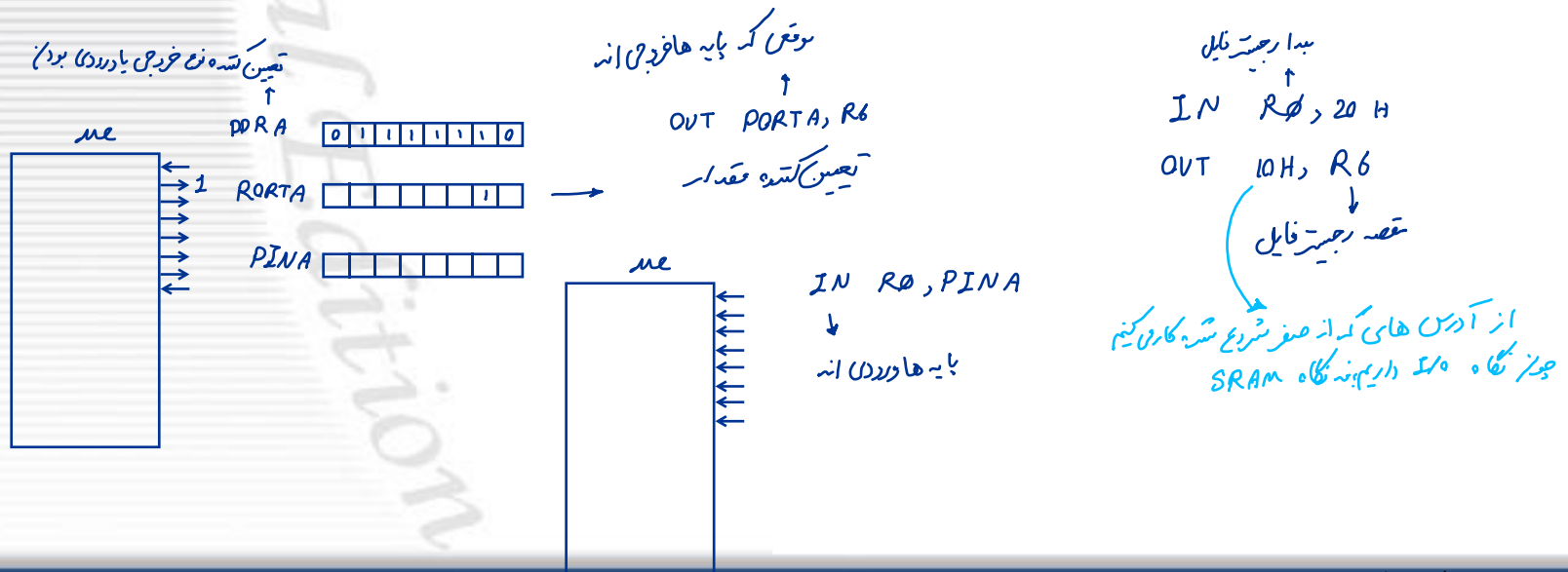
- یک عملیات ALU با استفاده از دو عملوند ثباتی در یک چرخه ساعت اجراء و نتایج در ثبات مقصد ذخیره می‌شود.

- این بدین معنی است که در طول یک چرخه ساعت واحد، یک عملیات توسط ALU انجام می‌گیرد که شامل خارج شدن دو عملوند از مجموعه ثبات‌های عمومی، انجام عملیات در درون ALU بر روی عملوندها و نهایتاً ذخیره نتیجه در درون مجموعه ثبات‌های عمومی می‌باشد.

فضای حافظه I/O

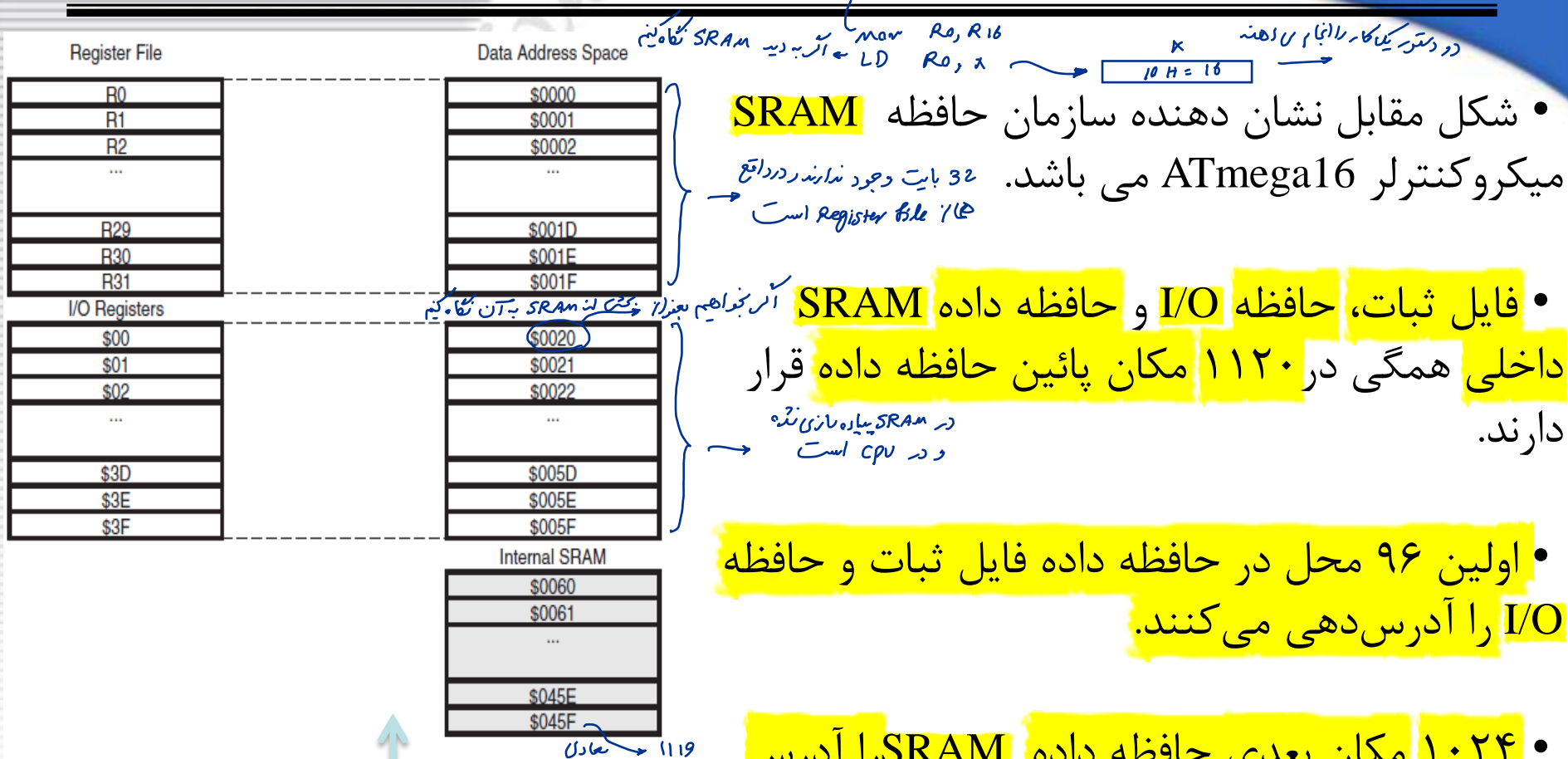
• فضای حافظه I/O شامل ۶۴ آدرس برای عملیات‌های مرتبط با ماژول‌های داخلی میکروکنترلر و ثبات‌های کنترلی آنهاست.

• حافظه I/O می‌تواند مستقیماً مورد دسترسی قرار گیرد و یا به عنوان محل‌های داده که بعد از فایل ثبات قرار دارند محسوب شود (محدوده \$5F - \$20). ☹



حافظه داده SRAM، فایل ثبات و حافظه I/O

آر.بی. دی. حافظه reg نگاه کنیم



• شکل مقابل نشان دهنده سازمان حافظه SRAM میکروکنترلر ATmega16 می باشد. 32 بیت وجود ندارند در واقع 15/16 Register File است

• فایل ثبات، حافظه I/O و حافظه داده SRAM اگر نخواهیم بعد از SRAM پتان نگاه کنیم داخلی همگی در 1120 مکان پائین حافظه داده قرار دارند. در SRAM پیاده سازی نشده و در CPU است

• اولین 96 محل در حافظه داده فایل ثبات و حافظه I/O را آدرس دهی می کنند.

• 1024 مکان بعدی حافظه داده SRAM را آدرس دهی می کنند.

mov Rd, Rr, Copy Register Rd ← Rr
LD Rd, x ; Load Indirect Rd ← (x)

فضای حافظه داده و ثبات های عمومی همه منظوره

پشته و اشاره گر پشته

- اشاره گر پشته قابل دسترسی بصورت نوشتنی/خواندنی در فضای I/O هستند.
- پشته در حافظه SRAM قرار دارد و در نتیجه اندازه پشته به اندازه SRAM و مقداری از آن که به عنوان پشته در نظر گرفته شده است، بستگی دارد.
- حافظه پشته برای ذخیره داده های موقتی، آدرس بازگشت از زیرروال و بازگشت از سرویس وقفه استفاده می شود.
- حافظه پشته می تواند از بزرگترین آدرس حافظه داده SRAM شروع شده و به آدرس های کوچکتر ادامه یابد.

هسته پردازنده مرکزی میکروکنترلر ATmega16

- هنگامی که وقفه‌ای اتفاق می‌افتد و یا فراخوانی یک زیرروال صورت می‌گیرد، مقدار آدرس بازگشت موجود در شمارنده برنامه (PC) در پشته ذخیره می‌شود.
- پشته عملاً در حافظه داده عمومی SRAM قرار دارد و در نتیجه اندازه پشته به اندازه کل SRAM و سایر کاربردهای آن محدود می‌شود.
- کلیه برنامه‌های کاربر باید اشاره گر پشته را در روالِ بازنشانی و قبل از اجرای هر زیر روال یا وقفه مقداردهی اولیه کنند.

SPH

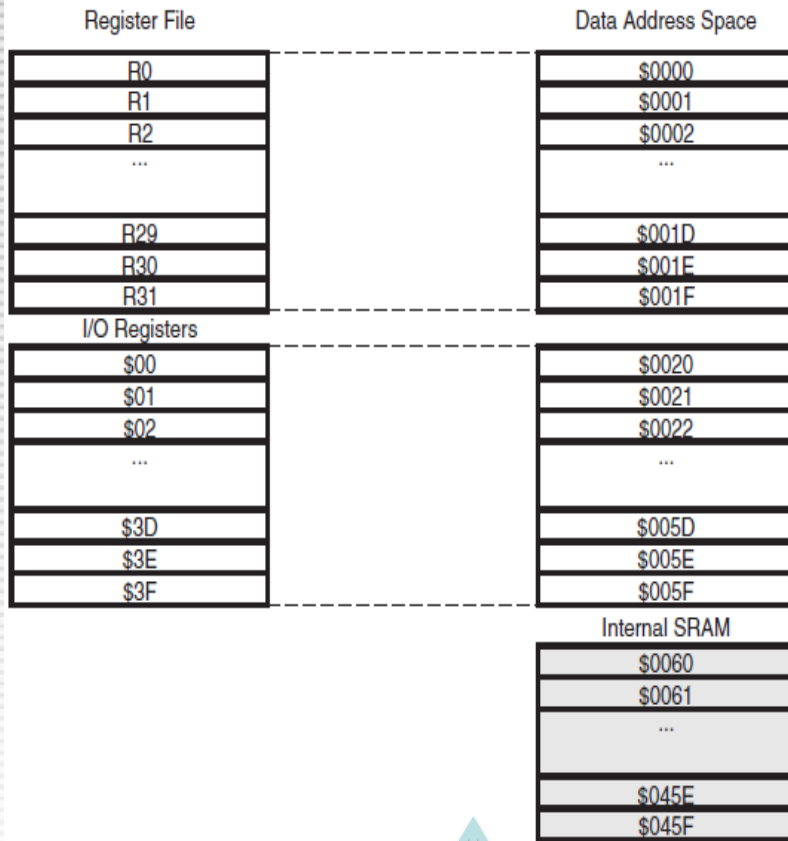
SPL

حافظه پشته و ثبات اشاره گر به پشته

- اشاره گر ۱۶ بیتی پشته که همیشه به بالای پشته اشاره می کند، قابل خواندن و نوشتن است و در فضای ورودی/خروجی قرار دارد.
- در میکروکنترلر ATmega16، اشاره گر پشته باید بالای آدرس \$60 (بعد از فضای I/O) باشد.
- شکل زیر اشاره گر پشته را نشان می دهد:

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

حافظه داده SRAM، فایل ثبات و حافظه I/O



• شکل مقابل نشان دهنده سازمان حافظه SRAM میکروکنترلر ATmega16 می باشد.

• اشاره گر پشته باید بالای آدرس \$60 (بعد از فضای I/O) در بخشی در ۱۰۲۴ آخر حافظه داده SRAM واقع شود.

فضای حافظه داده و ثبات‌های عمومی همه منظوره