

سپهر مقیسه
روح الله احمدزاده

9831103

9831001

پیش گزارش 11

کامپیوتر پایه

کامپیوتر پایه از دو بخش اصلی تشکیل شده: حافظه و پردازنده
نحوه کار کامپیوتر به این صورت است که دستورالعمل های ذخیره شده در حافظه را
میخواند و آنها را اجرا میکند.
پردازنده این دستور ها را طبق الگوریتم فون نیومن در 6 گام تکرار شونده انجام میدهد:

1: واکنشی دستورالعمل (instruction fetch)

2: رمزگشایی دستورالعمل (instruction decode)

3: خواندن عملگرها (operands fetch)

4: اجرا (execute)

5: ذخیره نتیجه (result write back)

6: رفتن به دستور بعد (تکرار از 1)

قبل از طراحی اجزای پردازنده نیاز داریم مجموعه دستورالعمل های مورد استفاده را داشته
باشیم تا قالب دستورالعمل ها را از روی آن بسازیم
یک نمونه از ISA (مجموعه دستورالعمل ها) میتواند جدول زیر باشد:

<i>Symbol</i>	<i>Hex Code</i>		<i>Description</i>
	<i>I = 0</i>	<i>I = 1</i>	
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
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
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
IOF	F040		Interrupt off

دستورات بخش اول تک ادرسی هستند یعنی نیاز به یک OPERAND دارند

دستورات بخش دوم ثباتی یا صفر ادرسی هستند و در همان ثبات اتفاق می افتند

دستورات بخش سوم برای دستگاه های ورودی و یا خروجی (I/O) هستند

برای اجرای الگوریتم فون نیومن در پردازنده نیاز به یک سری ثبات داریم تا چیزهایی مثل دستورالعمل، آدرس حافظه، شماره ثبات، program counter، stack pointer، نتیجه محاسبات و ... را

دو نوع ثبات در طراحی وجود دارد :

1- ثبات عام منظوره : ثبات هایی که برای راحتی انتقال داده ها وجود دارند و عمدتاً در هنگام استفاده و ریختن داده ها از آن ها استفاده میکنیم. تعداد آن در شروع طراحی مشخص میشود

2- ثبات خاص منظوره : که چند نوع مانند (IR، AR، PC، AC و ...) داریم که بعداً به آنها میپردازیم و هر کدام یک وظیفه مشخص دارند و فقط برای آن منظور استفاده میشوند . اندازه هر کدام از این ثبات ها با توجه به کاربرد آنها مشخص میشود

جدول زیر اندازه ثبات های خاص منظوره پردازنده ی کامپیوتری با حافظه ی $16 * 4096$ را نشان میدهد:

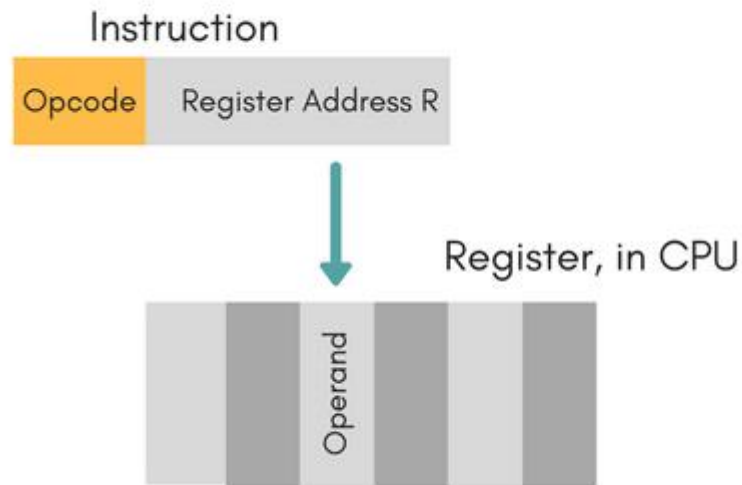
Register	Symbol	Number of bits	Function
Data register	DR	16	Holds memory operand
Address register	AR	12	Holds address for the memory
Accumulator	AC	16	Processor register
Instruction register	IR	16	Holds instruction code
Program counter	PC	12	Holds address of the instruction
Temporary register	TR	16	Holds temporary data
Input register	INPR	8	Carries input character
Output register	OUTR	8	Carries output character

دستور ها میتوانند عملیات یکسانی را در انواع مختلف آدرس دهی انجام دهند.

انواع مختلف آدرس دهی :

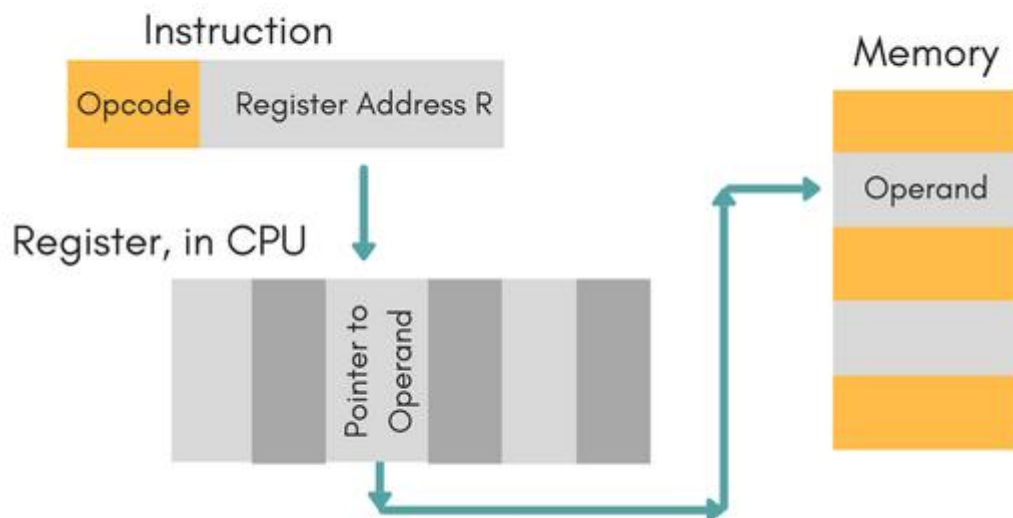
1- حالت بلافاصل : به اینصورت که دیتای مورد نظر را مستقیماً از روی دستور میخوانیم

2- حالت ثباتی : در این حالت operand در ثبات ذخیره شده و با دادن شماره آن در دستورالعمل به آن دسترسی پیدا میکنیم :



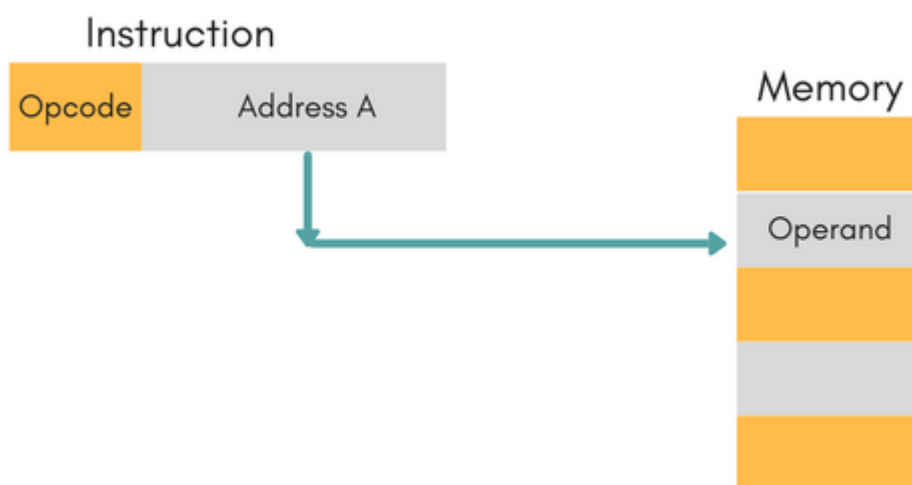
3-حالت ثباتی غیر مستقیم :

در این حالت در ثبات ادرس خانه مورد نظر حافظه ثبت شده و با دادن ادرس خانه مورد نظر در دستورالعمل میتوان به آن دسترسی پیدا کرد:



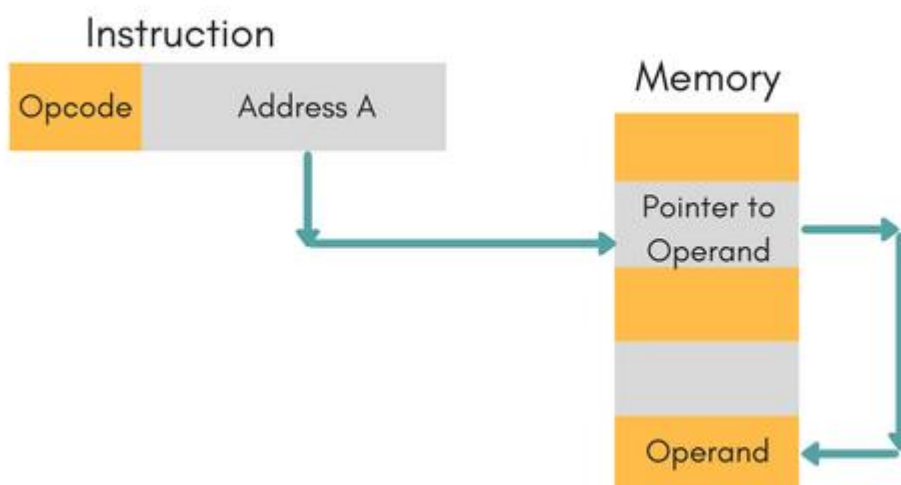
4-حالت ادرس دهی مستقیم:

در این حالت آدرس خانه حافظه در دستور العمل است :



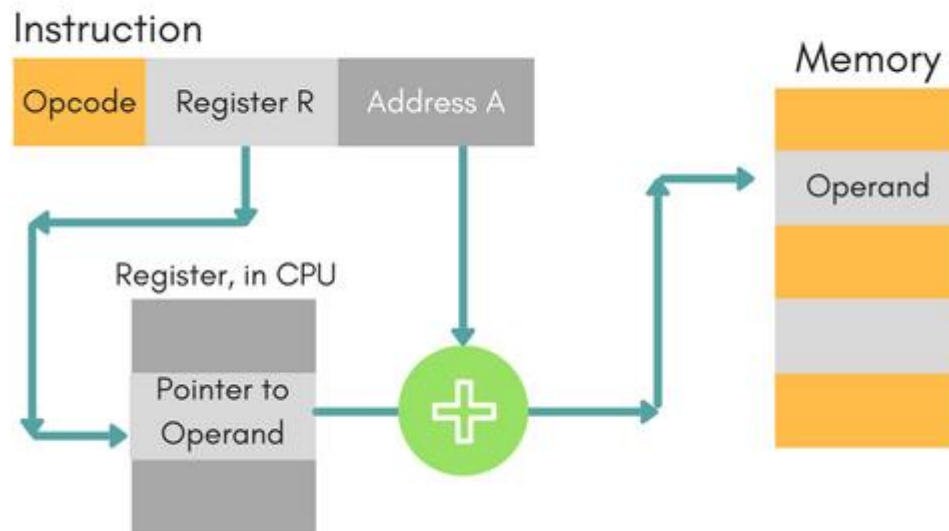
5-حالت آدرس دهی غیر مستقیم:

در این حالت نیز آدرس یک خانه حافظه در دستور العمل است ولی آن خانه خود پوینتر به یک خانه دیگر در حافظه است که مقدار مدنظر را دارد:



6-حالت جایگذاری:

در این حالت دستور العمل دارای یک ادرس ثابت و یک ادرس ثابت است که و هربار با ادرس ثبت شده در یکی از خانه های ثابت جمع میشود و به یک خانه حافظه اشاره میکند :



*حالت های دیگر نیز از همین حالت ها الگو میگیرند

:Address bus

قسمتی است که تمام component های پردازنده را به هم وصل میکند و امکان انتقال داده را از خانه ای در حافظه و یک ثابت به خانه ای دیگر و ثباتی دیگر فراهم میکند.

نکته ای که در استفاده از Bus باید به آن توجه داشت این است که هیچگاه نباید چندین داده همزمان روی آن ریخته شود. و باید با استفاده از پایه های load دقت کنیم که در هر زمانی داده ی روی Bus در کدام قسمت پردازنده ریخته شود.

شکل زیر شمایی کلی از مسیر داده و common bus را نشان میدهد:

COMMON BUS SYSTEM

