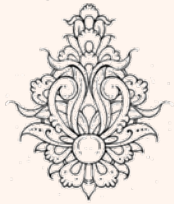
زبان ماشین و اسمبلی
(۰۰۵-۱۱-۱۳)
بخش سوم

انواع ماشین و
شیوه‌های نشانی‌دهی

دانشگاه شهید بهشتی
دانشکده‌ی مهندسی برق و کامپیوتر
زمستان ۱۳۹۳
احمد محمودی ازناوه

فهرست مطالب

- انواع دستورالعمل بر حسب تعداد عملوندها
- انواع ماشین‌ها
- سایر شیوه‌های نشانی‌دهی
- انواع پردازنده



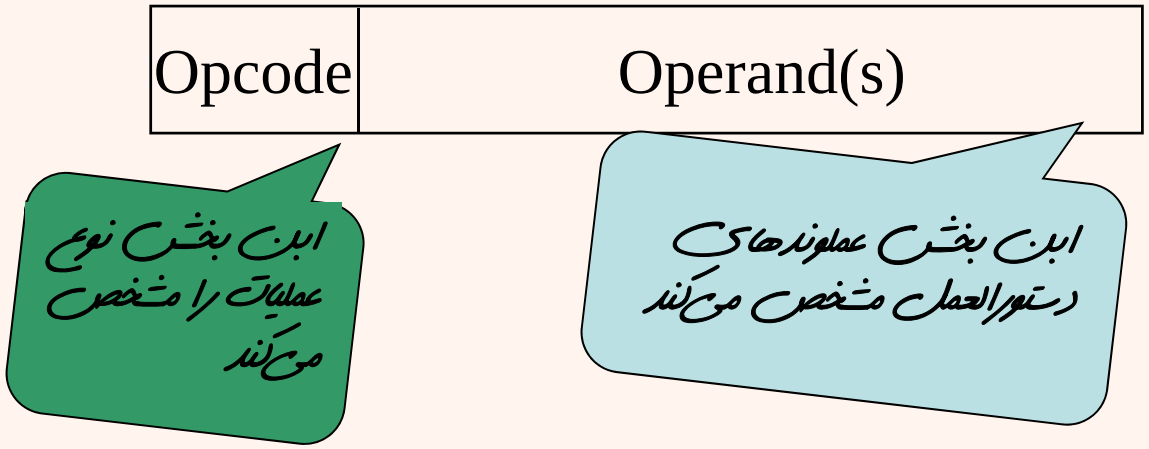
Instruction code

قالب دستورالعمل‌ها

- در ساده‌ترین حالات می‌توان، قالب یک دستورالعمل را به صورت زیر تصور کرد:

operational code

قالب کلی دستورالعمل

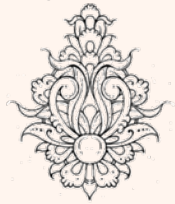


انوع مختلف دستور بر اساس تعداد عملوند

- بر اساس تعداد عملوندها، می‌توان دستورها را تقسیم‌بندی کرد: (منظور تعداد عملوندهایی است که در کد دستور بیان می‌شوند)
  - به عنوان مثال در MIPS

تعداد عملوند

- در MIPS دستورهای محاسباتی دارای دو و سه عملوند هستند.
- سخت‌افزارهای مختلف، دستورهای محاسباتی را به صورت‌های دیگری نیز پیاده‌سازی کرده‌اند. در ادامه نگاهی به شکل‌های مختلف دستورهای محاسباتی خواهیم انداخت.
- نخستین نمونه، ماشینی است که تعداد عملوندهای دستورهای محاسباتی صفر است.



دستورات بدون عملوند (صفر آدرسه)

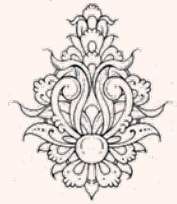
مثال: ارزیابی عبارت

$(a + b) \times (c - d)$

Push a	Push b	Add	Push d	Push c	Subtract	Multiply
a	b	a + b	d	c	c – d	Result
	a		a + b	d	a + b	
				a + b		

Reverse Polish string: a b + d c – ×

postfix notation



stack machine

zero-address machines

ماشین پشته‌ای

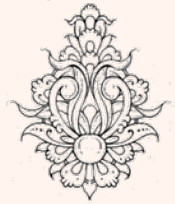
- به چنین ماشینی stack machine می‌گویند.
- مزایا:
  - در این ماشین، طول کد دستورالعمل کاهش می‌یابد، اما با توجه به نیاز به بارگذاری داده روی پشته تعداد دستورها افزایش پیدا می‌کند. در این حالت اثر کاهش طول کد دستورالعمل اثر بیشتری دارد و باعث می‌شود حجم object code به شدت کاهش یابد.
  - این ماشین داری کامپایلر و مفسر ساده‌تری خواهد بود.
- معایب:
  - مراجعه به حافظه افزایش می‌یابد.
  - مفسر آن کندتر است.

دستورات تک‌آدرسه

accumulator machine

one-address machines

- در این نوع دستورات یکی از عملوندها به صورت ضمنی مورد استفاده قرار می‌گیرد:
  - مثال:
    - `add X`
    - `accumulator ← accumulator + X`
- انباره(accumulator): ثباتی که نتیجه‌ی میانی همه‌ی محاسبات در آن قرار می‌گیرد.
- در این ماشین مقصد(منبع) دستورهای خواندن(نوشتن) از(در) حافظه انباره است.



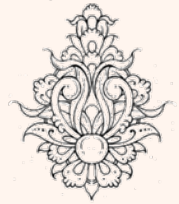
دستورهای دو آدرسه و سه‌آدرسه

- این شیوه در پردازنده‌های امروزی معمول‌تر است:
- مثال:
  - در خانواده‌ی x86

- `add Y, X` $Y \leftarrow X + Y$
  - در خانواده‌ی MIPS و ARM

- `add Z, Y, X` $Z \leftarrow X + Y$

- در برخی معماری‌ها هر دو عملوند باید ثبات باشد(مانند MIPS)، در برخی یکی از عملگرها می‌تواند حافظه باشد(مانند x86)



تمرین کلاسی

- عبارت c=a+b را به زبان اسمبلی ماشین‌های مختلف بیان شده بنویسید.

stack machine

```
push a
push b
add
pop c
```

accumulator machine

```
load a
add b
store c
```

load-store(register-register)

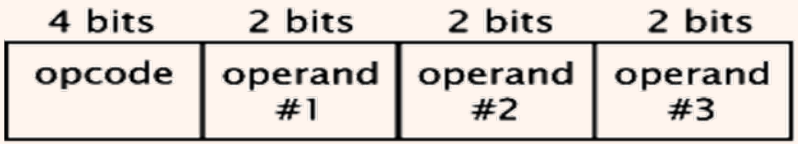
```
load $s1,a
load $s2,b
add $s3,$s1,$s2
store c,$s3
```

```
move $s1,a
add $s2,$s1,b
move c,$s2
```

load-store(register-memory)

در ادامه مروری خواهیم داشت بر انواع شیوه‌های نشانی دهی عملوندها

قالب دستور

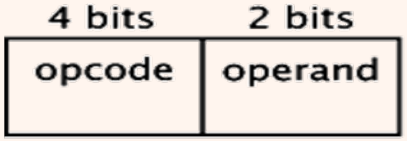


ADD *A,B,C* (*A=B+C*) 1010 00 01 10

(a)

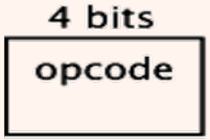
MOVE *A,B* (*A=B*) 1000 00 01
ADD *A,C* (*A=A+C*) 1010 00 10

(b)



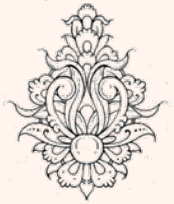
LOAD *B* (*Acc=B*) 0000 01
ADD *C* (*Acc=Acc+C*) 1010 10
STORE *A* (*A=Acc*) 0001 00

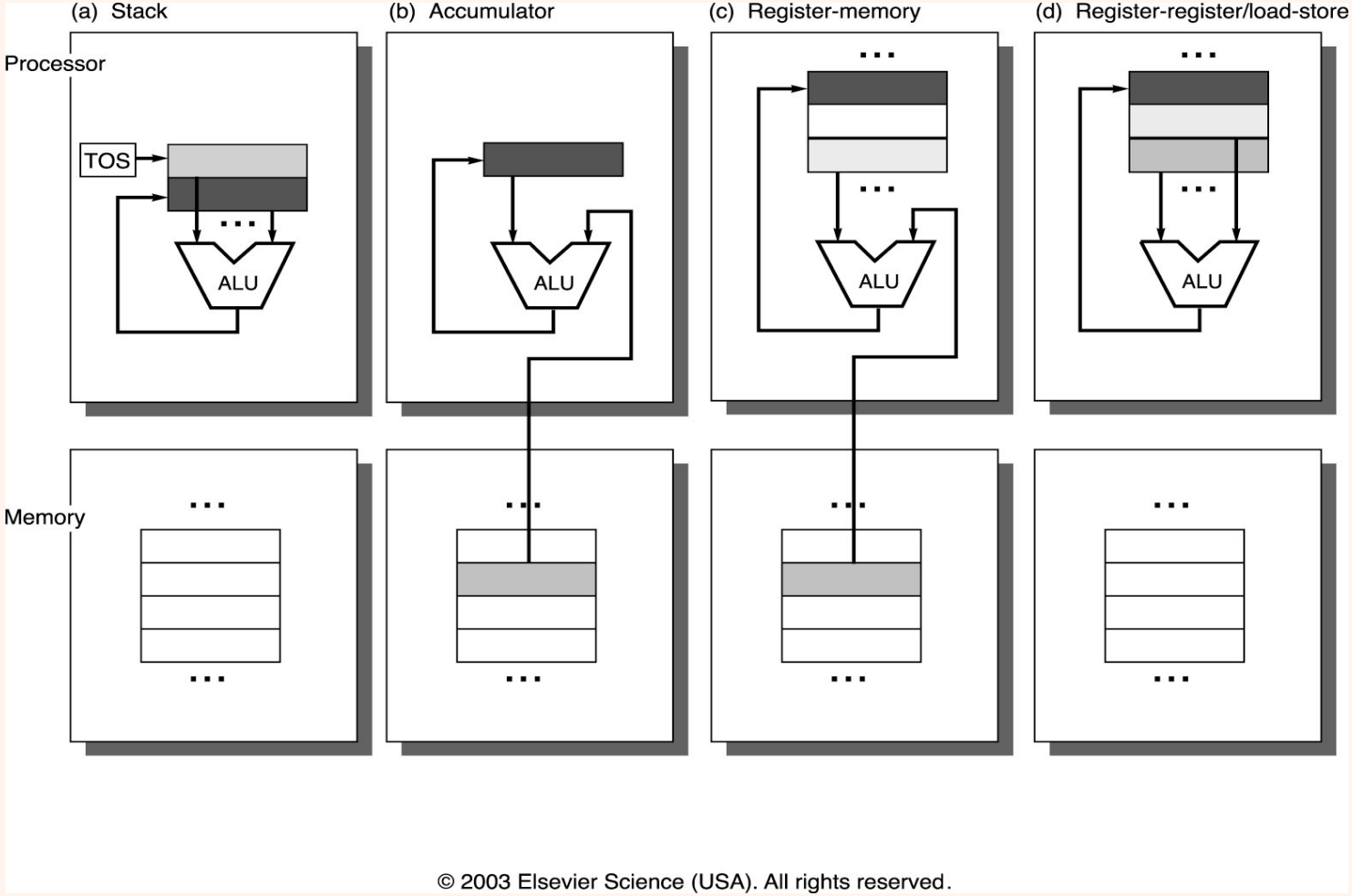
(c)



PUSH *B* (Stack=*B*) 0101
PUSH *C* (Stack=*C,B*) 0110
ADD (Stack=*B+C*) 1010
POP *A* (*A*=stack) 1100

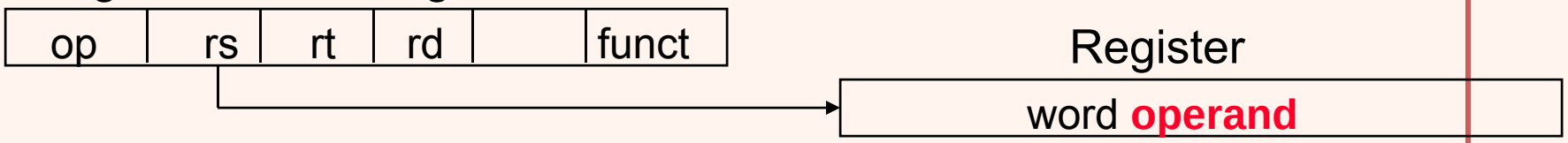
(d)





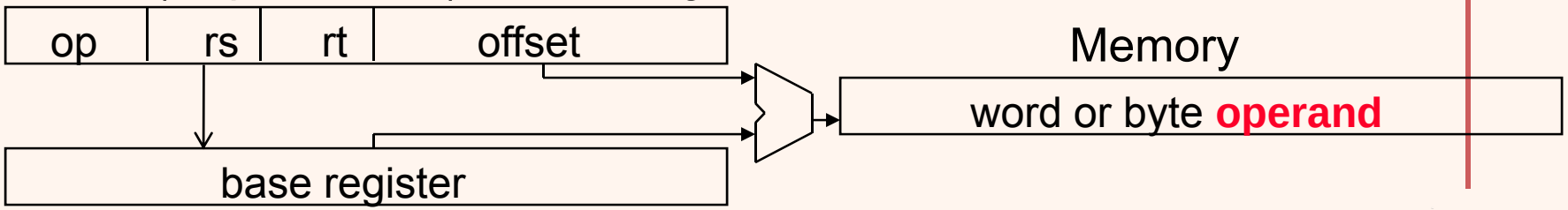
مرور شیوه‌های نشانی‌دهی مطرح شده

1. Register addressing



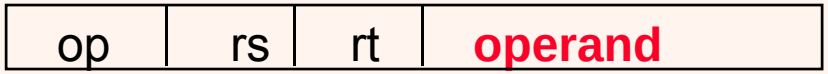
در این حالت عملوند در یک ثبات قرار دارد، ما تنها آدرس ثبات را مشخص می‌کنیم.

2. Base (displacement) addressing



عملوند در حافظه است، اما آدرس واقعی به دو قسمت شکسته شده است: پایه و جابجایی (آفست)

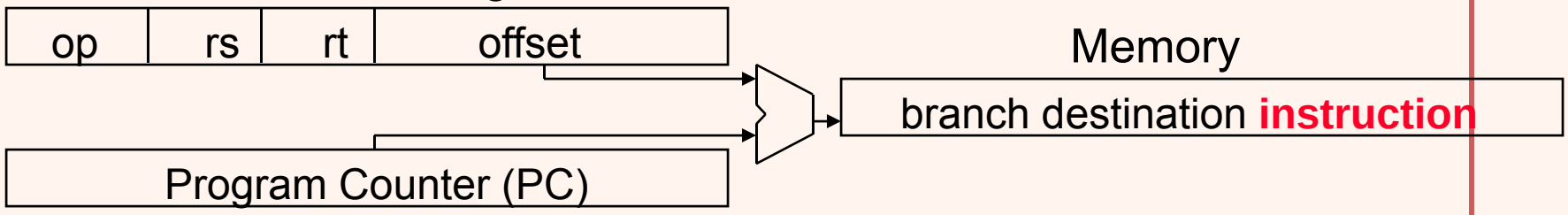
3. Immediate addressing



خود عملوند ذکر شده است

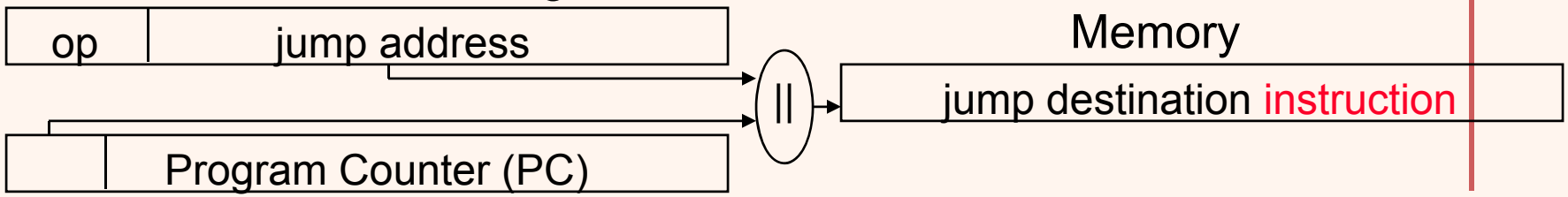
مرور شیوه‌های نشانی‌دهی مطرح شده

4. PC-relative addressing



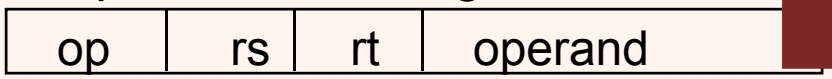
عملوند یک آدرس است که به صورت نسبی (نسبت به **PC**) بیان می‌شود.

5. Pseudo-direct addressing



عملوند یک آدرس است که بخشی از آن به صورت مستقیم آمده است و بخش دیگر با کمک **PC** تامین می‌شود.

6. Implied addressing

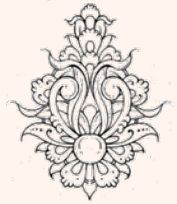


عملوند در دستور مطرح نمی‌شود.

علاوه بر شیوه‌های نشانی دهی که در **MIPS** استفاده می‌شود، شیوه‌های دیگری نیز وجود دارد

انواع شیوه‌های نشانی‌دهی

- گاهی در منابع مختلف برای شیوه‌های نشانی‌دهی یکسان نام‌های متفاوتی مطرح شده است و حتی نام یکسان برای دو شیوه‌ی متفاوت مورد استفاده قرار گرفته است.
- در برخی دستورالعمل، باید نشانی بخشی از حافظه که حاوی دستورالعمل است، مشخص شود(عملوند آدرس دستور است) و در برخی دیگر عملوند آدرس داده و یا خود داده است.

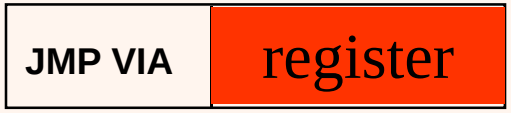


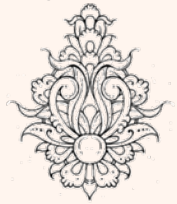
انواع نشانی‌دهی دستورها

- مطلق(مستقیم):
  - در MIPS چنین دستوری وجود ندارد.

- شبه‌مستقیم:
  - دستورهای jmp و jal
- نسبی (نسبت به PC)
  - دستورهای be و bne

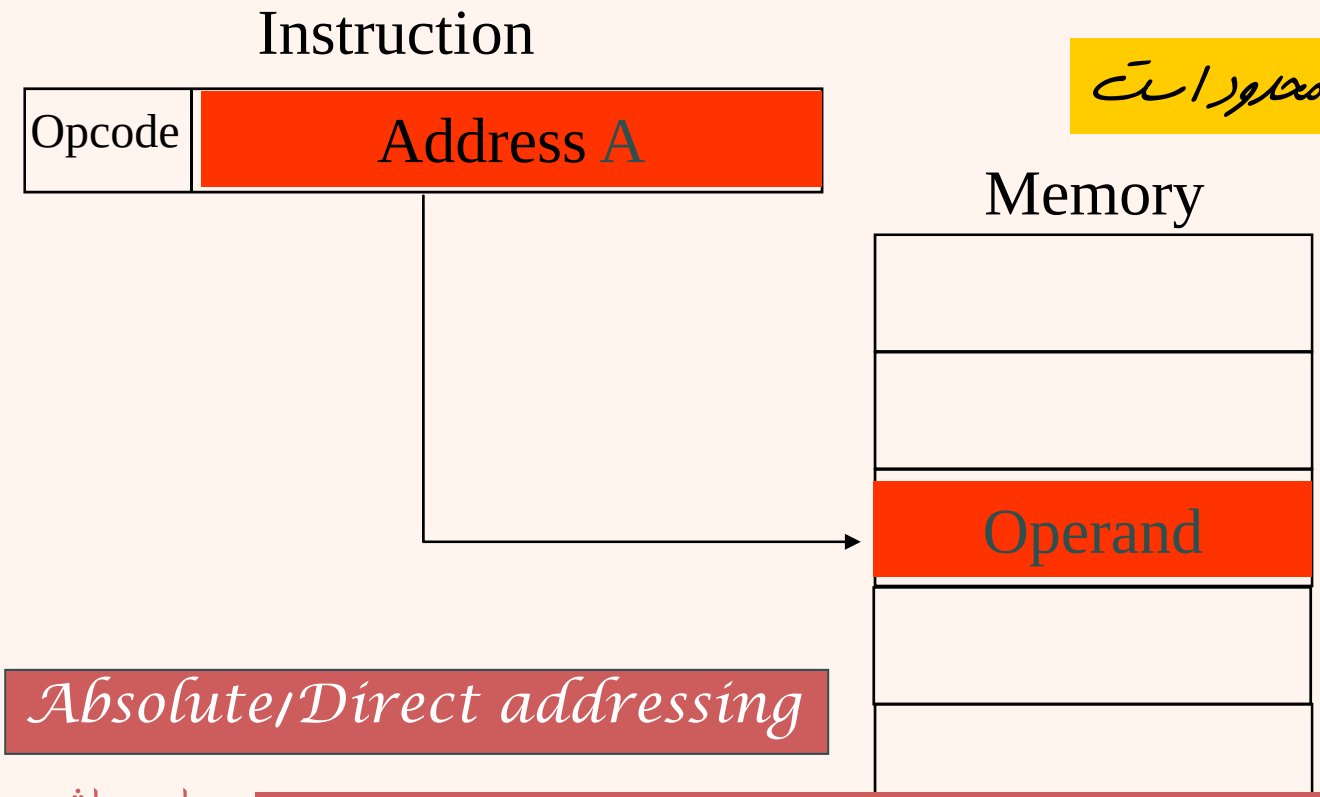
- غیرمستقیم ثباتی:
  - دستور jr





نشانی‌دهی مستقیم داده

- در فیلد آدرس آدرس خانه‌ی حافظه آورده می‌شود.
- در MIPS به دلیل محدودیت کد دستورالعمل چنین شیوه‌ای وجود ندارد(برخلاف x86).



فضای آدرس‌دهی محدود است

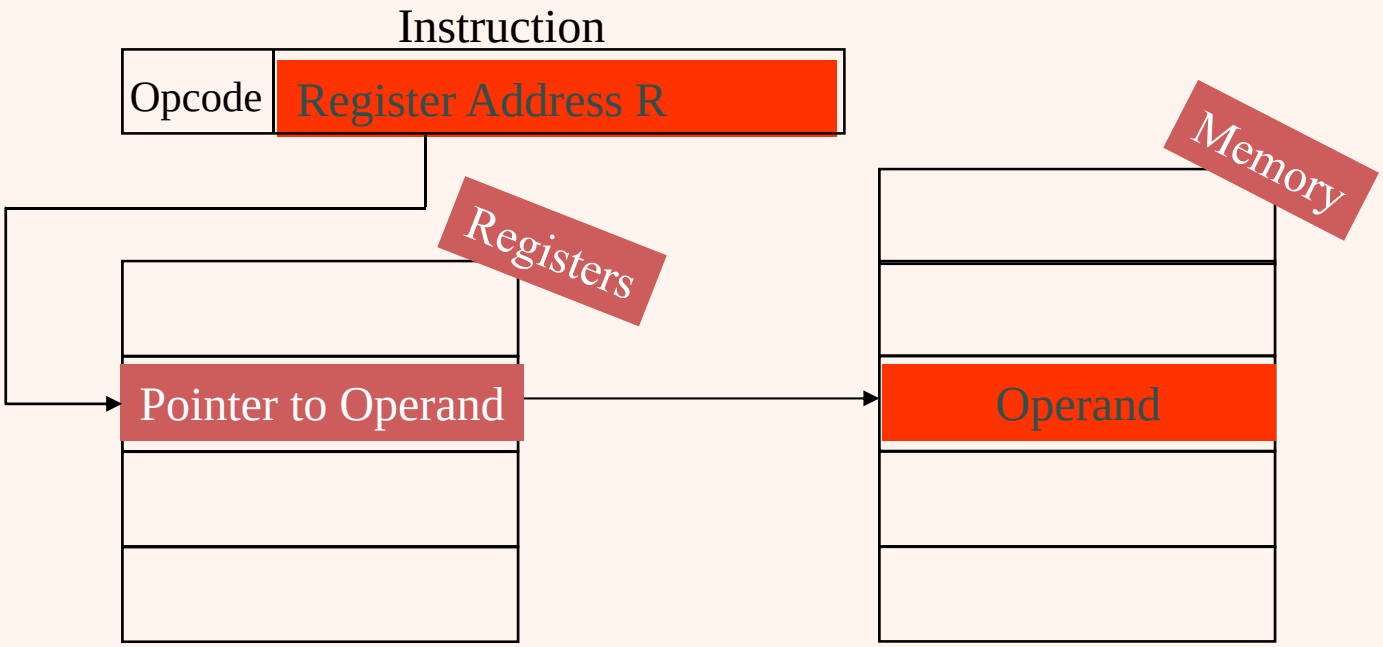
$EA = A$

Absolute/Direct addressing

نشانی‌دهی غیرمستقیم از طریق ثبات

Register Indirect Addressing

در این شیوه آدرس خانه‌ی حافظه در یک ثبات قرار دارد، این حالت در عمل حالت خاصی از آدرس‌دهی پایه است.



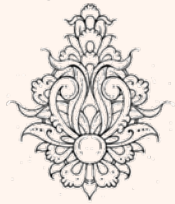
$EA = (R)$

انواع نشانی‌دهی (ادامه...)

Indexed Addressing

- آدرس‌دهی شاخص:
  - شبیه آدرس‌دهی بر اساس آدرس پایه است، با این تفاوت که بخش آدرس، ابتدای آدرس بخشی از حافظه را نشان می‌دهد در حالی که بخش شاخص اختلاف از آن بخش را نشان می‌دهد.
  - به بخش شاخص یک ثبات اختصاص داده شده است.
  - با افزودن یک scale به این نوع آدرس‌دهی می‌توان شیوه‌های آدرس‌دهی پیچیده‌تری ایجاد کرد.

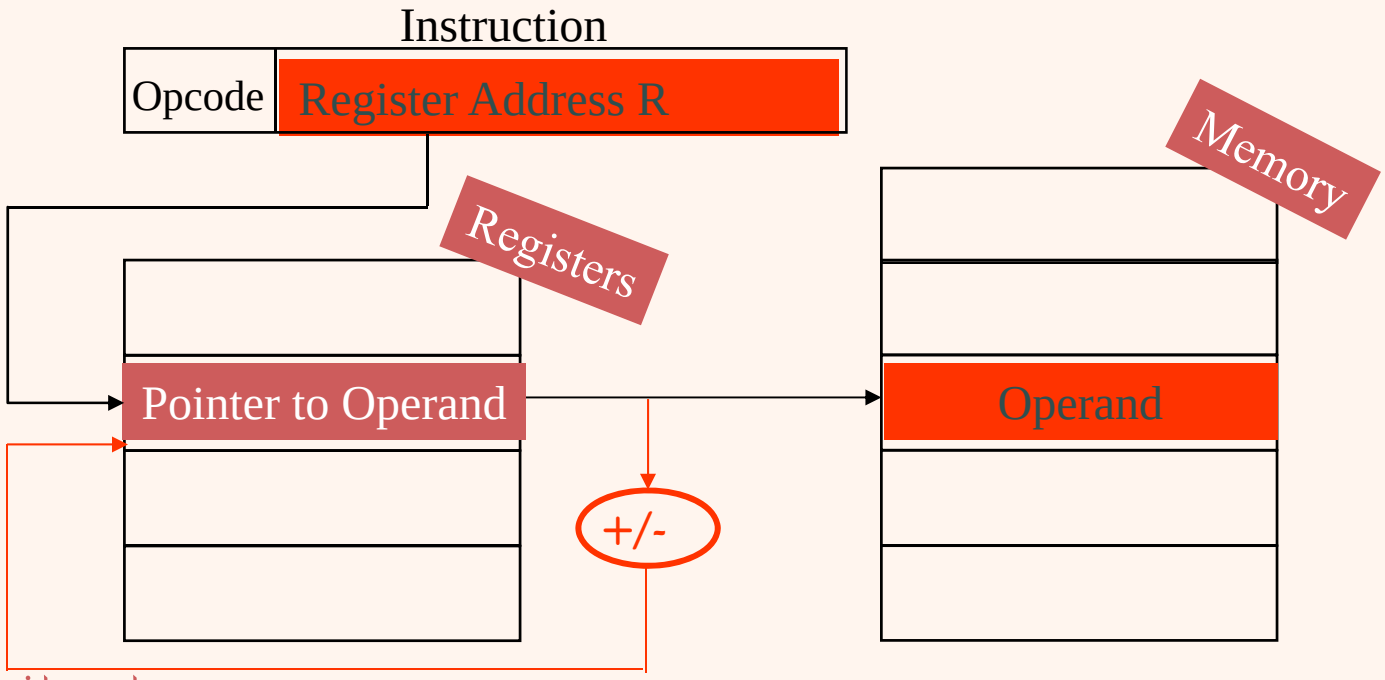
reg ← base+index×scale



شیوه‌ی کاهشی/افزایشی

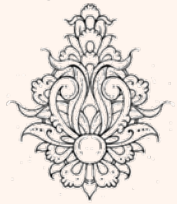
- در این شیوه‌ی محتوای ثبات هر بار یکی افزوده می‌شود.

Auto-increment Auto-decrement



انواع معماری

- معماری RISC یا مجموعه دستورالعمل‌های محدود، نوعی معماری کامپیوتری است که از دستورهای ساده، پرکاربرد و ساده استفاده شده است. برای انجام عملیات پیچیده‌تر ترکیبی از دستورها مورد استفاده قرار می‌گیرد.
- معماری CISC یا مجموعه دستورالعمل‌های پیچیده، نوعی معماری کامپیوتری است که در آن یک دستورالعمل در عمل شامل چندین دستورالعمل سطح پایین دیگر است. (مانند خواندن از حافظه، عملیات حسابی و ذخیره در حافظه)



CISC در مقابل RISC

RISC

- تعداد دستورات کم
- طول دستورات ثابت
- زمان اجرای ثابت
- هزینه‌ی پایین
- تنها دستورات خواندن و نوشتن به حافظه دسترسی دارند.
- همه‌ی عملوندها ثبات‌های پردازنده هستند.
- مودهای آدرس محدود
- واحد کنترل به صورت سیم‌بندی

CISC

- دارای دستورات پیچیده و متنوع
  - حتی دستوراتی که کم‌تر به کار می‌روند
- طول دستورات متغیر
- میکروکدهای پیچیده
- بیشتر دستورات به حافظه دسترسی دارند.
- مودهای آدرس‌دهی بسیار متنوع هستند.

CISC در مقابل RISC (ادامه...)

- در عمل مرزهای بین این دو در حال محو شدن هستند.
- پردازنده‌های جدید از خصوصیات هر دو بهره می‌گیرند.
- با این وجود، برای سیستم‌های درون‌کار (توکار) پردازنده‌ی RISC ترجیح داده می‌شوند.