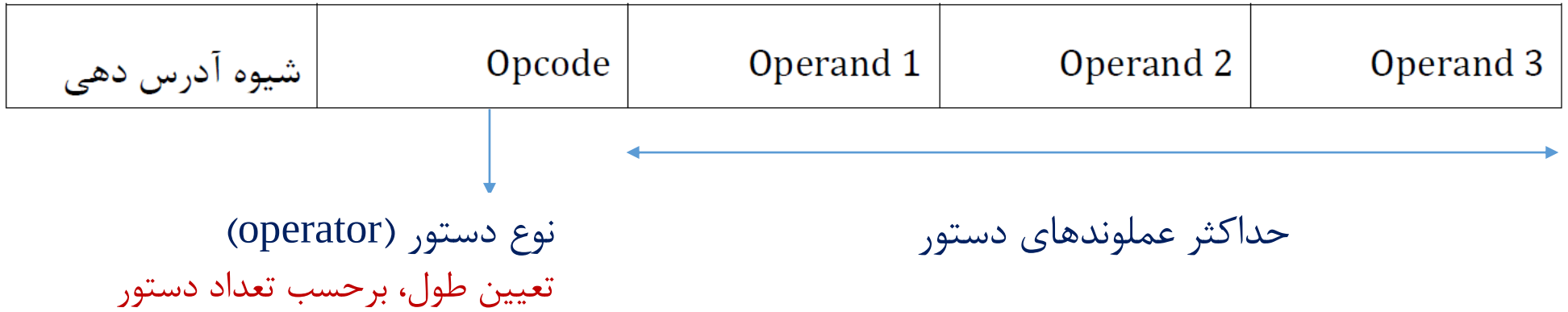
بنام خداوند جان و خرد

معماری کامپیوتر

جلسه نوزدهم: واحد کنترل (Control Unit)-۲

قالب دستورات

- در طراحی پردازشگر قالب دستورات می‌بایست مشخص شده باشد
  - قالب دستورات ISA



شیوه‌های آدرس‌دهی

۱- آدرس‌دهی ضمنی

۲- آدرس‌دهی بلافصل

۳- آدرس‌دهی حافظه‌ای مستقیم (دستورات حافظه‌ای)

۴- آدرس‌دهی حافظه‌ای غیرمستقیم

۵- آدرس‌دهی ثباتی مستقیم (دستورات ثباتی)

۶- آدرس‌دهی ثباتی غیرمستقیم

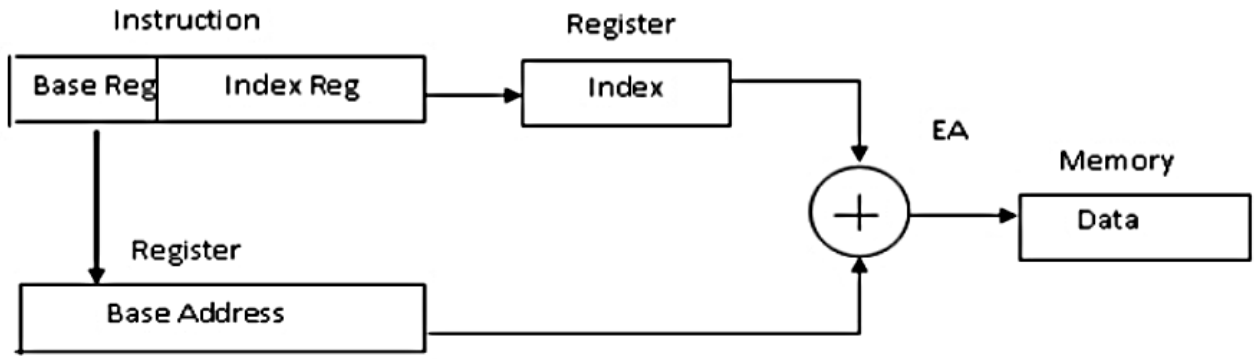
۷- آدرس‌دهی با ثبات پایه

۸- آدرس‌دهی شاخص‌دار

شیوه‌های آدرس‌دهی

9. آدرس‌دهی شاخص‌دار با ثبات پایه (based indexed addressing):

- برای کار با آرایه‌های دوبعدی مناسب است
- آدرس شروع و اندیس آرایه هردو قابل تغییر هستند (ترکیب دو حالت قبل)
- Add BX[SI]

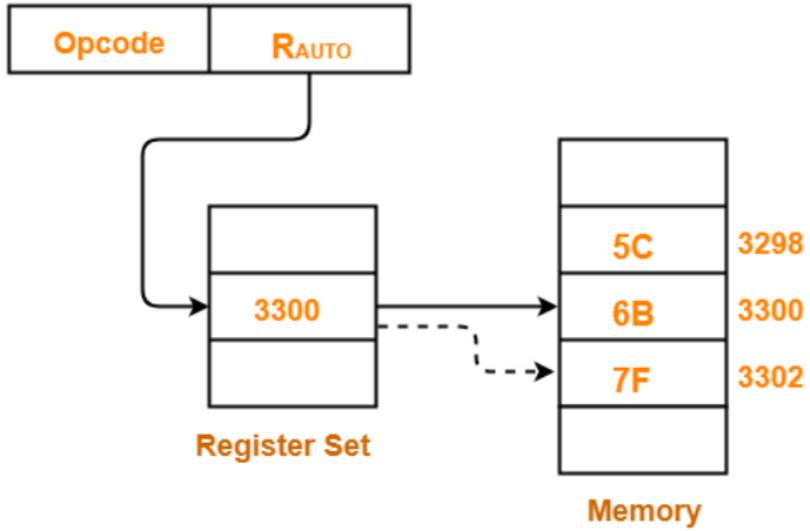


Effective address = [Base register] + [Index register]

شیوه‌های آدرس‌دهی

10. آدرس‌دهی خودافزایشی (Auto increment):

- دستوراتی که اتوماتیک، مقدار یک ثبات را زیاد می‌کنند مانند شمارنده حلقه
- گام افزایش به سایز عملوند بستگی دارد
- ابتدا داده از حافظه برداشته شده و سپس مقدار در ثبات به‌روز می‌شود



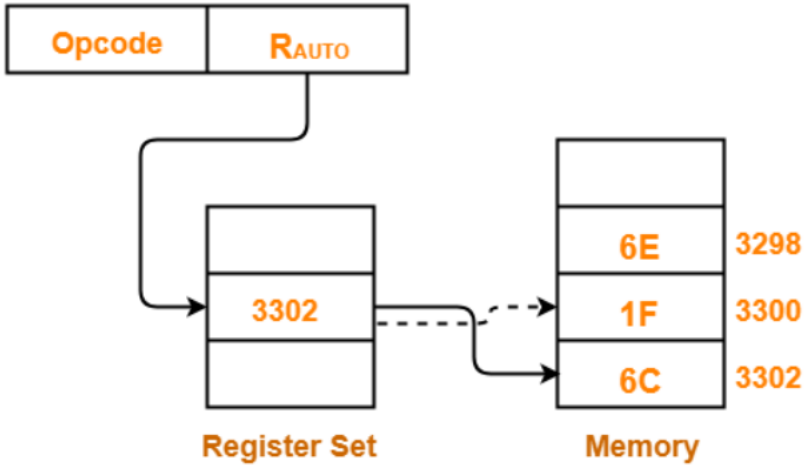
Effective Address of the Operand

= Content of Register

شیوه‌های آدرس‌دهی

10. آدرس‌دهی خودکاهشی (Auto decrement):

- دستوراتی که اتوماتیک، مقدار یک ثبات را کم می‌کنند (مانند اشاره‌گر پشته‌ها)
- گام کاهش به سایز عملوند بستگی دارد
- ابتدا مقدار موجود در ثبات کاهش داده شده و به آدرس به‌روز شده می‌رویم



Effective Address of the Operand

= Content of Register – Step Size

شیوه‌های آدرس‌دهی

- در پردازنده‌های RISC (هدف این درس)
  - استفاده از شیوه‌های آدرس‌دهی memory direct/indirect و register direct/indirect
  - ثابت بودن تعداد بیت‌های opcode و operand
- در پردازنده‌های CISC
  - از شیوه‌های آدرس‌دهی پیچیده‌تر هم استفاده می‌شود
- در همه روش‌های آدرس‌دهی
  - مبادله‌ای بین فضای آدرس‌دهی و تعداد مکان‌های پوشش داده شده در حافظه وجود دارد

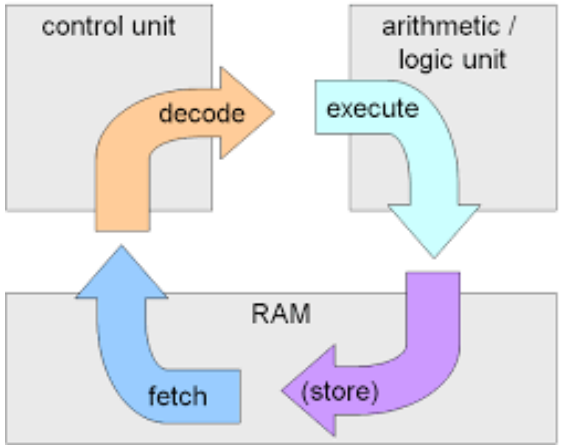
طراحی واحد پردازشگر مرکزی

- طراحی واحد پردازشگر شامل مراحل زیر است:
  - طراحی Data Path: اجزای داخلی پردازنده برای نگهداری اطلاعات، انتخاب یا انتقال اطلاعات
    - مدیریت و کنترل فرایند حرکت داده
      - ALU، memory، Reg. Files، Bus و ...(المان‌هایی که برروی داده‌ها کار می‌کنند)
  - طراحی Control Unit: اجزای داخل پردازنده برای کنترل، انتخاب عملیات و اجرای دستورالعمل
    - ترتیب اجرا، طراحی مدار کنترلی و اجرای الگوریتم فون نیومن
    - Decoder، sequence counter، Mux و سخت‌افزار کنترل‌کننده
    - تعیین عملیات برای مسیر داده

طراحی واحد پردازشگر مرکزی

- چرخه اجرای یک دستور در پردازنده (instruction cycle): الگوریتم Von Neumann

1. خواندن دستورالعمل (instruction fetch)
2. کدگشایی دستورالعمل (instruction decode)
3. خواندن عملوندها (operands fetch)
4. اجرای دستور (execute)
5. نوشتن نتیجه اجرا در مقصد دستورالعمل (write back)
6. افزایش دادن شمارنده برنامه (program counter)
7. بازگشت به مرحله ۱



طراحی واحد پردازشگر مرکزی

- اجرای الگوریتم نیومن از ابتدا تا انتها: Instruction Cycle
- هرچه سرعت اجرای الگوریتم نیومن بیشتر باشد
  - سرعت اجرای دستور بیشتر می‌شود
  - کارایی پردازشگر افزایش می‌یابد
  - برای افزایش سرعت از روش‌های موازی‌سازی در اجرا استفاده می‌شود
    - اجرای مراحل به‌صورت هم‌زمان

طراحی واحد پردازشگر مرکزی

- برای اجرای چرخه فون نیومن ناچار به اجرای دنباله در سخت‌افزار هستیم
  - رعایت ترتیب در سخت‌افزار
    - در هر مرحله بخشی از قسمت‌های سخت‌افزار بلااستفاده می‌مانند
    - با زیاد شدن عمق یک مرحله، اجرای برنامه کندتر می‌شود
  - تعبیه واحد sc (sequence counter)
    - شمارنده‌ای که مشخص می‌کند در کدام مرحله الگوریتم برای اجرای هر دستور هستیم
    - با افزایش شمارنده، اجرای دستور وارد مراحل مختلف می‌شود
    - این شمارنده تعداد کلاک موردنیاز برای اجرای یک دستور را برمی‌گرداند

طراحی واحد پردازشگر مرکزی کامپیوتر پایه

- هدف ما، طراحی پردازنده RISC ساده و پایه است (کامپیوتر پایه)
  - قالب دستورالعمل از دو بخش opcode و operand تشکیل شده است
    - تعداد بیت‌های مربوط به این دو بخش را در پردازنده RISC ثابت درنظر می‌گیریم
  - سه نوع دستورالعمل با کدهای عملیاتی مجزا درنظر می‌گیریم: حافظه‌ای، ثباتی و I/O
    - دستورات حافظه‌ای: داده‌های داخل حافظه با دو شیوه آدرس‌دهی مستقیم و غیرمستقیم
    - دستورات ثباتی: شیوه آدرس‌دهی ثباتی دارند
    - دستورات I/O: برقراری ارتباط با ورودی و خروجی

طراحی واحد پردازشگر مرکزی کامپیوتر پایه

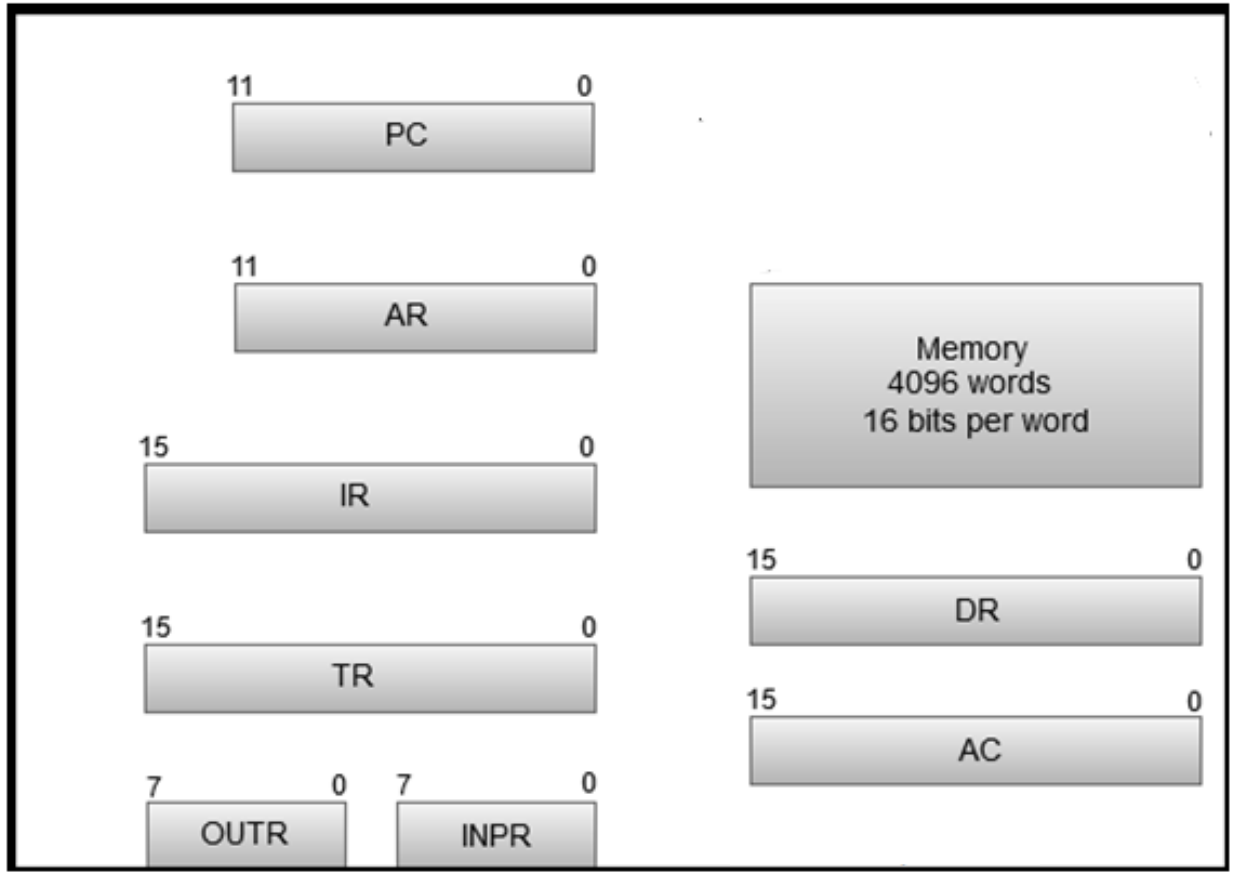
- برای طراحی پردازشگر برحسب instruction set به حافظه و ثبات نیاز است:
  - ساختن مسیر حرکت و کنترل داده‌ها
  - اجزای اصلی موردنیاز برای اجرای هر دستور
    - حافظه دستورالعمل‌ها برای نگهداری دستورات
    - مشخصه و آدرس دستورالعمل جاری (ثبات‌ها)
    - واحد جمع‌کننده با هدف پیمایش آدرس‌ها و اجرای همه آن‌ها

طراحی واحد پردازشگر مرکزی کامپیوتر پایه

- ثبات‌های اصلی و موردنیاز برای اجرای دستورات:
  - Program Counter (PC): نگهداری آدرس دستورالعمل جاری که نوبت fetch شدن آن است
  - Instruction Register (IR): نگهداری دستوری که بتازگی fetch شده است
  - Address Register (AR): نگهداری آدرسی از حافظه که داده در آن است
  - Data Register (DR): نگهداری داده‌ها
  - Temp Register (TR): نگهداری نتایج میانی (ثبات کمکی)
  - Accumulator (AC): نگهداری خروجی ALU (ثبات بهتر است چون ممکن است نتیجه میانی باشد)
    - ورودی های ALU ثبات‌های DR و AC
  - Input Register/Output Register (IR/OR): نگهداری ورودی و خروجی

ثبات‌های اصلی و موردنیاز

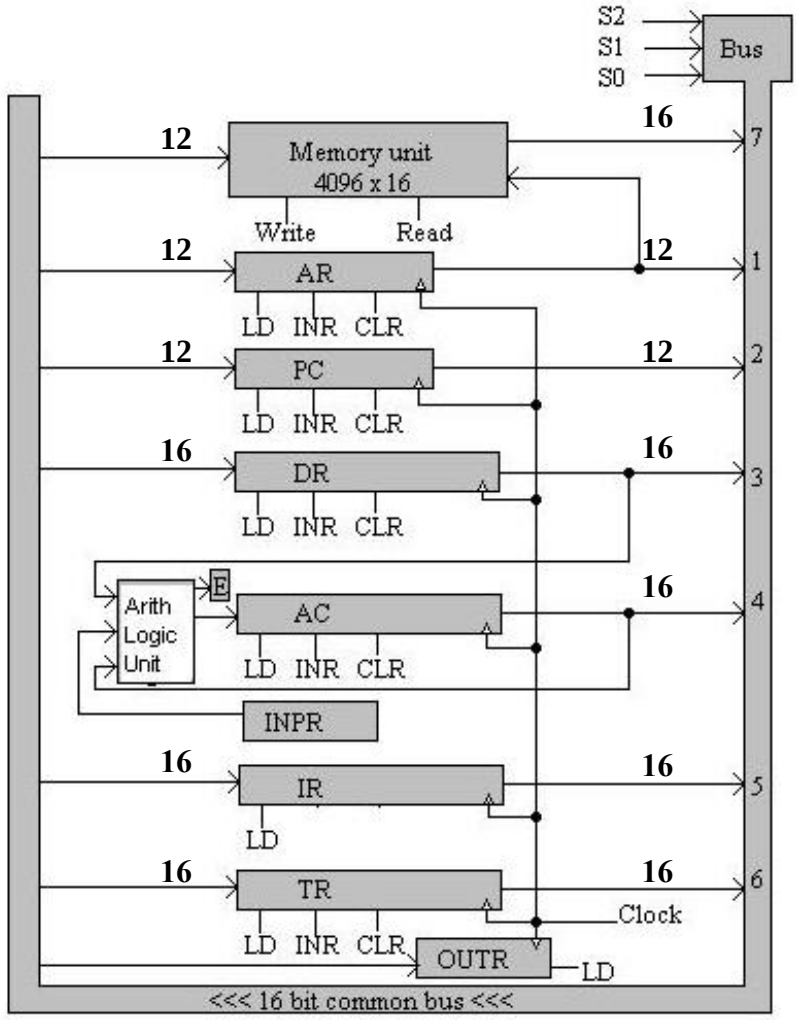
- طراحی پردازنده ۱۶ بیتی از نوع RISC



اتصال حافظه و ثبات‌ها

- اتصال حافظه و ثبات‌ها در شکل‌دهی مسیر داده
- برای ارتباط دادن ثبات‌ها به یکدیگر و اتصالشان به حافظه شیوه‌های متفاوتی وجود دارد
  - اتصال باس یا نقطه به نقطه (point to point)
  - متداول‌ترین روش استفاده از باس است
  - باس به‌اندازه عرض حافظه (در اینجا ۱۶ بیتی) درنظر گرفته می‌شود
  - حافظه به‌صورت نوبتی با ثبات‌ها در ارتباط قرار می‌گیرد
    - در هر لحظه ثبات آدرس یا ثبات داده به حافظه وصل است.

اتصال حافظه و ثبات‌ها



- اتصال به‌صورت باس
  - همه ثبات‌ها پایه load و clk دارند
    - مقدار روی باس وارد کدام ثبات شود
  - مدیریت باس با MUX سه‌تایی
  - باس ۱۶ بیتی (common bus)
    - اگر ۱۲ بیت روی آن باشد: آدرس
    - اگر ۱۶ بیت روی آن باشد: داده