

معماری کامپیوتر

جلسه بیستم: واحد کنترل (Control Unit) - ۳



- واحد ISA به عنوان واسط سخت افزار و نرم افزار
- قالب دستورالعمل ها و بخش های آدرس
 - تعداد عملوندها و شیوه های آدرس دهی
- اصول اولیه طراحی پردازنده کامپیوتر پایه
 - واحد کنترل (چرخه اجرای دستور)، مسیر داده
- ساختار حافظه و ثبات های کامپیوتر پایه
 - آشنایی با ثبات های در نظر گرفته شده و هدف هریک

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 بر روی 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 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 حساب و منطق با هدف پیمایش آدرس‌ها و اجرای همه آن‌ها

طراحی واحد پردازشگر مرکزی کامپیوتر پایه

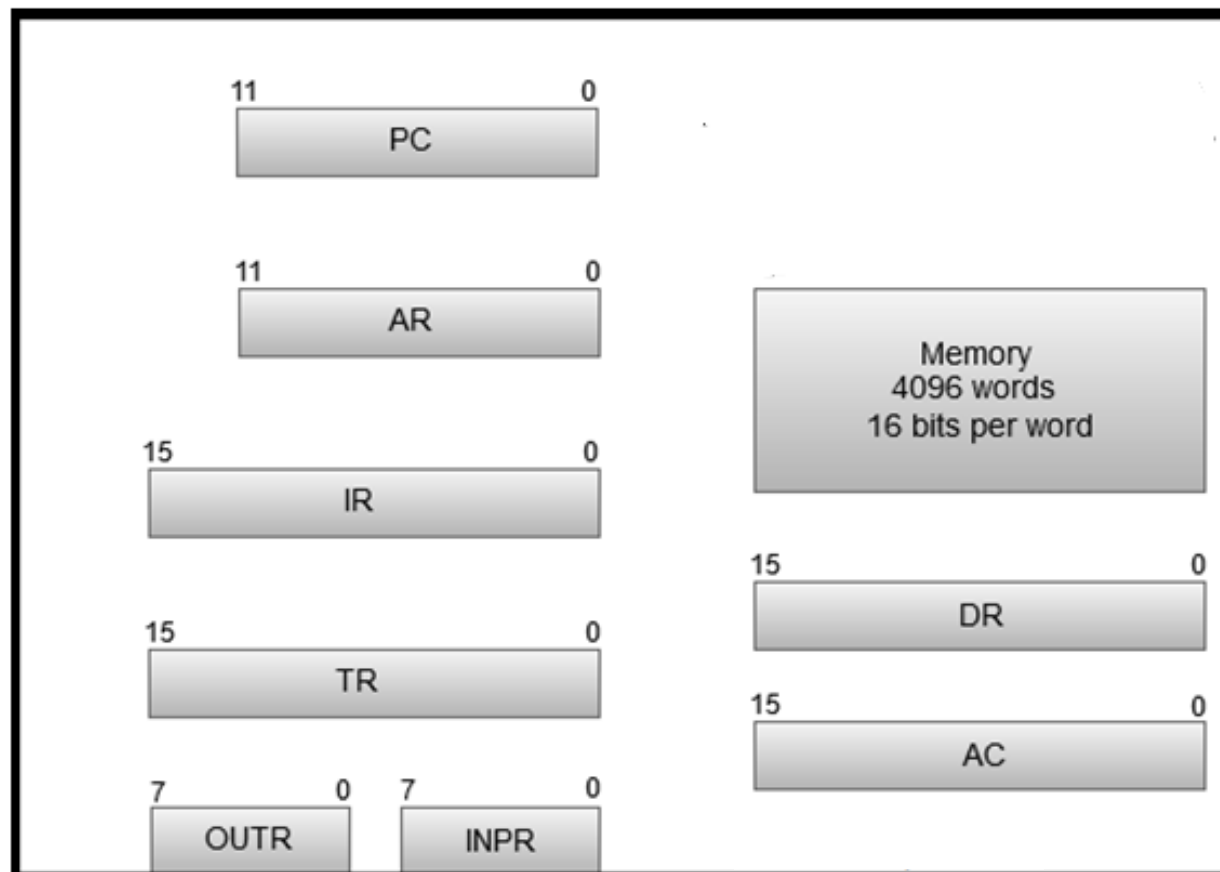


- ثبات‌های اصلی و موردنیاز برای اجرای دستورات:
- Program Counter (PC): نگهداری آدرس دستورالعمل جاری که نوبت fetch شدن آن است
- Address Register (AR): نگهداری آدرسی از حافظه که دستور یا داده در آن است
- Instruction Register (IR): نگهداری دستوری که بتازگی fetch شده است
- Data Register (DR): نگهداری داده‌ها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

- همه ثبات‌ها پایه load و clk دارند

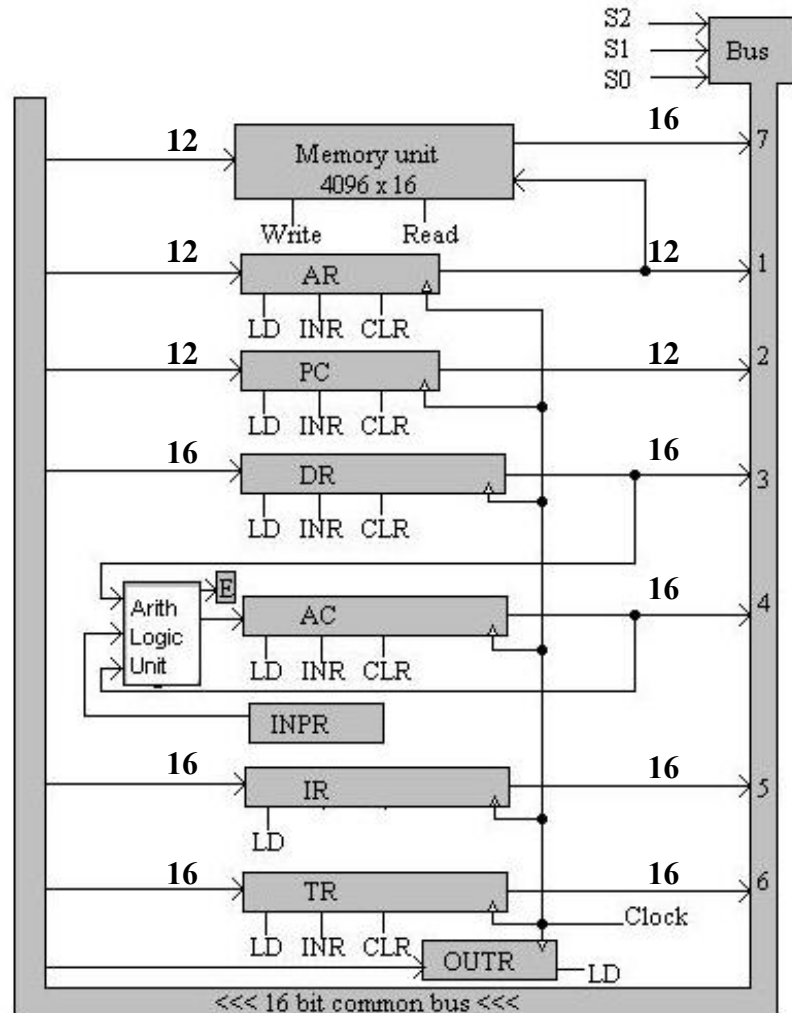
- مقدار روی باس وارد کدام ثبات شود

- مدیریت باس با MUX سه‌تایی

- باس ۱۶ بیتی (common bus)

- اگر ۱۲ بیت روی آن باشد: آدرس

- اگر ۱۶ بیت روی آن باشد: داده



اتصال حافظه و ثبات‌ها



- طراحی باس مشترک در کامپیوتر پایه

- واحد حساب و منطق

- عملیات جمع و تفریق دارد به علاوه اعمال منطقی

- ورودی‌ها: DR، INR و AC

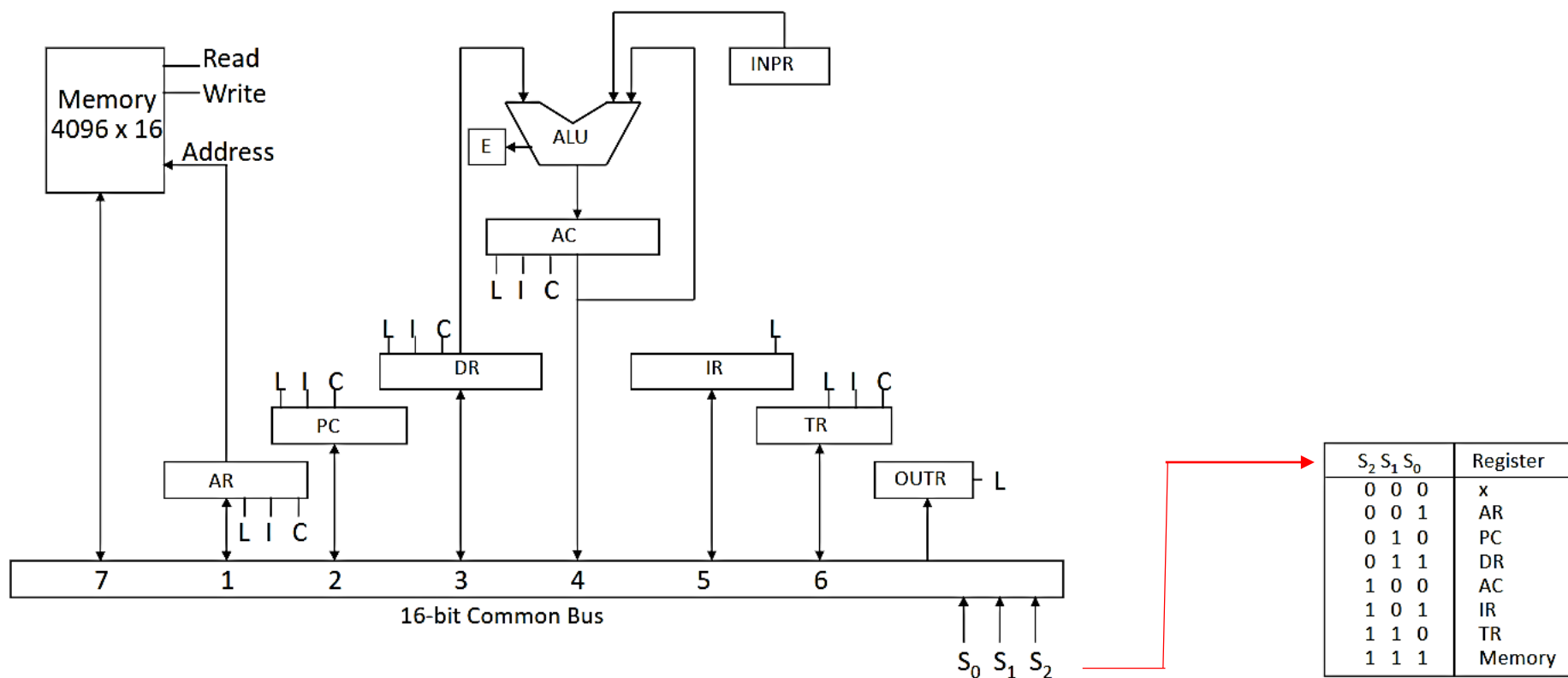
- دسترسی به حافظه

- نوشتن آدرس متناظر در ثبات AR

اتصال حافظه و ثبات‌ها از طریق باس



- باس مشترک آدرس و داده



طراحی واحد کنترل



- یکی از اجزای پردازنده مرکز سیستم‌های کامپیوتری است
- مسئول، مدیریت و هدایت عملیات پردازنده است
- وظیفه واحدهای دیگر (حافظه، ALU و I/O) را در مواجهه با دستورات معین می‌کند
- مانند مدیریت پایه‌های load ثبات‌ها و تنظیم ورودی‌های انتخاب MUX
- ترتیب اجرا، طراحی مدار کنترلی و اجرای الگوریتم فون نیومن
- Decoder, sequence counter, Mux و سخت‌افزار کنترل‌کننده

طراحی توالی گر در واحد کنترل

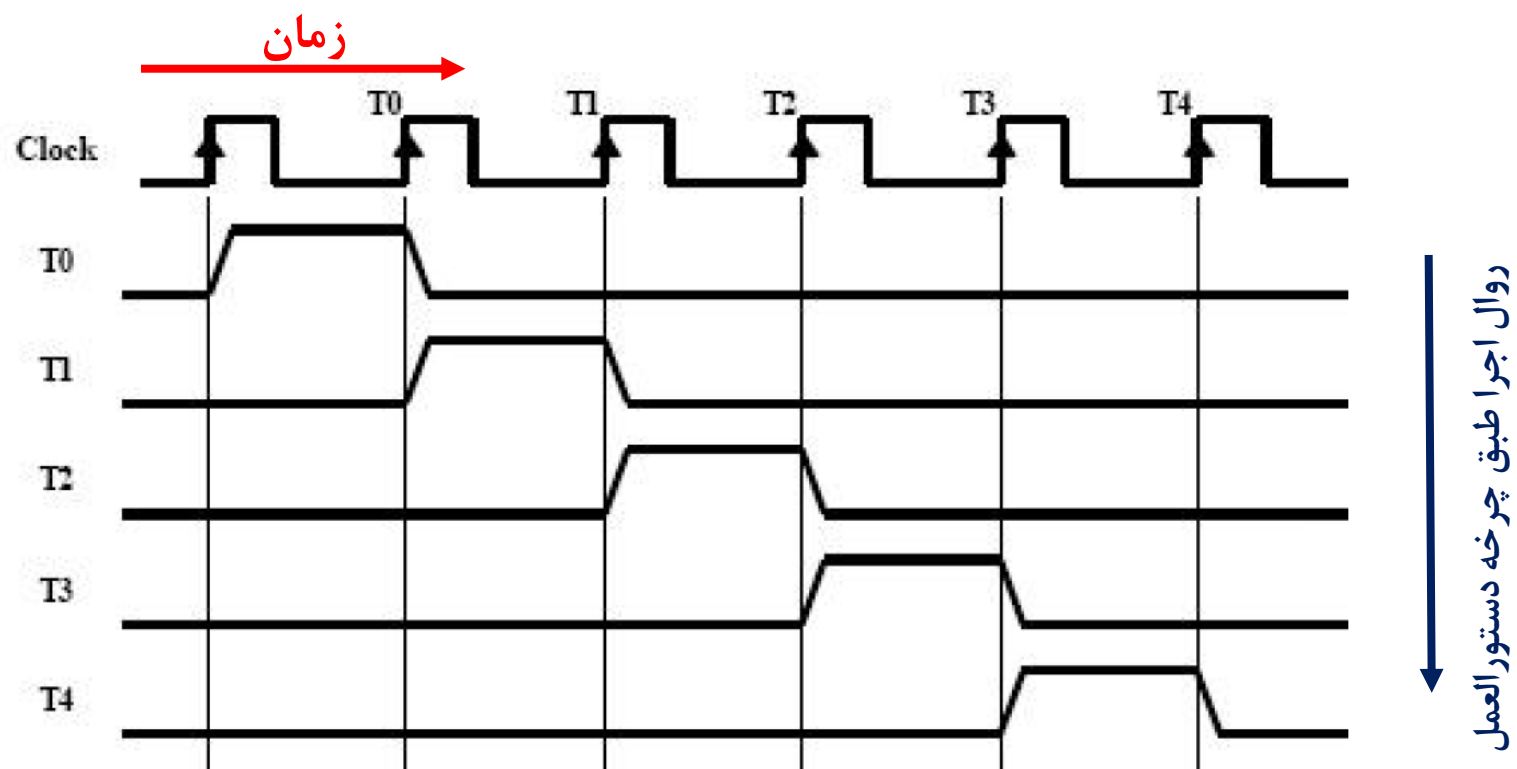


- به منظور اعمال الگوریتم نیومن در اجرای دستورات
- نیاز به اجرای ترتیبی در سخت افزار داریم
- نیاز به طراحی واحد سخت افزاری با هدف ایجاد توالی: توالی گر (sequencer)
- واحد توالی گر، گام های الگوریتم نیومن را برای یک چرخه دستورالعمل ایجاد می کند
- در هر زمان یکی از مراحل فعال و سایرین غیرفعال باشند
- ترتیب فعال شدن فعالیت ها مشخص و طبق روال باشد

طراحی توالی گر در واحد کنترل



- خروجی توالی گر (شمارنده) مورد نیاز:

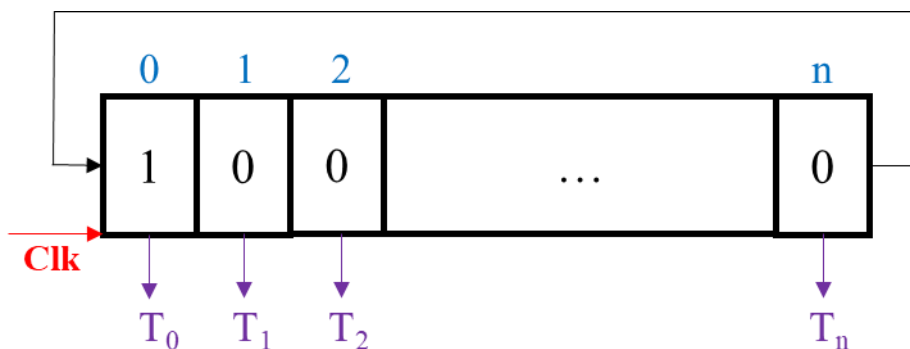


طراحی توالی گر در واحد کنترل



- روش اول: شمارنده بیت لغزان

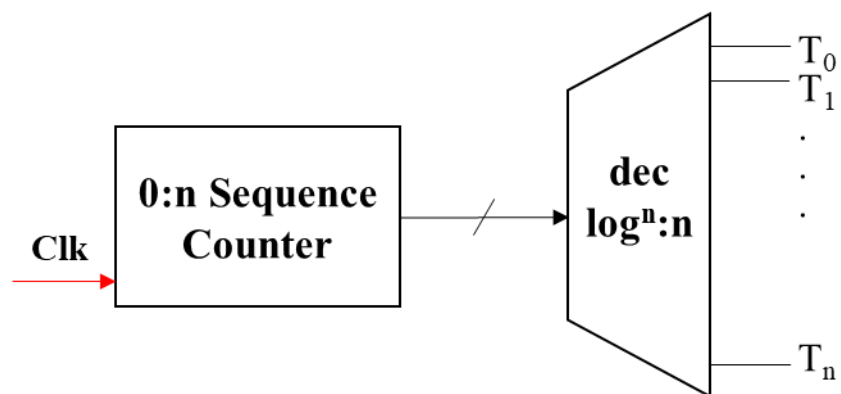
- برای طراحی این بخش می توان از شیفت رجیستر استفاده کرد
- یک بیت ثبات را برابر یک گذاشته و بقیه بیت ها را صفر می کنیم
- با هر کلاک، «یک» وارد شده یک بیت شیفت داده می شود
- در هر لحظه یک FF مقدار یک و باقی مقدار صفر را می گیرند
- مقدار یک با هر کلاک یک واحد شیفت داده می شود



طراحی توالی‌گر در واحد کنترل



- روش دوم: شمارنده صعودی
- اتصال یک دیکدر به یک شمارنده صعودی
- شمارنده‌ی باینری که از 0 تا n را به صورت صعودی می‌شمارد
- خروجی شمارنده به یک دیکدر وصل شده و در خروجی آن، مرحله فعال مشخص می‌شود



طراحی واحد کنترل



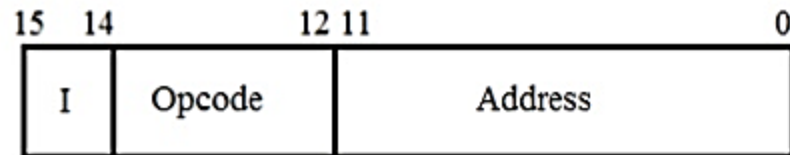
- هدف: مدیریت چرخه اجرای دستورالعمل طبق مدل نیومن و کنترل جریان داده
- **Sequence counter**, Decoder, Mux و سخت افزار کنترل کننده
- می دانیم در کدام مرحله اجرایی هستیم پس باید عملیات آن مرحله را بسازیم
- چه ثبات هایی فعال می شوند،
- نحوه ارتباط ثبات ها با یکدیگر، حافظه و ALU
- شیوه مدیریت باس براساس MUX موجود و فعال شدن load ثبات ها

مجموعه دستورالعمل‌ها در کامپیوتر پایه



- سه نوع دستور تعریف شده است
 - دستورات حافظه‌ای
 - آدرس‌دهی مستقیم ($I=0$) و غیرمستقیم ($I=1$) دارند
 - دستورات ثباتی
 - آدرس‌دهی مستقیم دارند
 - دستورات ورودی/خروجی
 - ارتباط داده‌ای با پورت‌های ورودی و خروجی

مجموعه دستورالعمل‌ها در کامپیوتر پایه



• قالب دستورالعمل

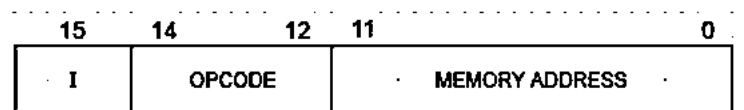
- یک بیت مود آدرس دهی، سه بیت نوع عملگر و دوازده بیت عملوند
- کد عملیاتی ۳ بیتی بوده و هشت حالت قابل تعریف دارد
- هفت کد اول، دستورات حافظه‌ای مانند load, jump, store ... (محدوده 000-110)
- کدهایی که چهاربیت سمت چپ آن‌ها 0111 باشد، دستورات ثباتی
- دوازده دستور مختلف تعریف می‌کنیم که در هر کدام، یک بیت بخش آدرس یک می‌شود
- کدهایی که چهاربیت سمت چپ آن‌ها 1111 باشد، دستورات I/O
- یک بیت برای تمایز دادن بین ورودی و خروجی و باقی برای شماره پورت

مجموعه دستورالعمل‌ها در کامپیوتر پایه



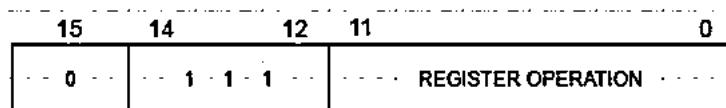
• قالب دستورالعمل

• تعداد دستورات حافظه‌ای



- هفت دستور با آدرس دهی مستقیم و هفت دستور با آدرس دهی غیرمستقیم

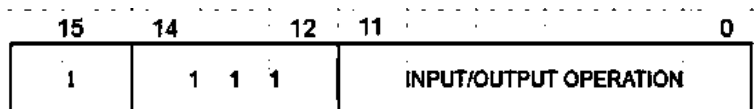
• تعداد دستورات ثباتی



- دوازده دستور به تعداد بیت‌های آدرس (هر بار یکی از بیت‌های آدرس یک باشد)

- چون تعداد ثبات‌ها محدود است، از بخش آدرس به عنوان کد عملیاتی استفاده می‌کنیم

• دستورات ورودی خروجی



- دوازده بیت برای تعیین پورت