معماری کامپیوتر (۱۳۹۱-۱۱-۱۳)

جلسه‌ی چهاردهم

دانشگاه شهید بهشتی

دانشکده‌ی مهندسی برق و کامپیوتر

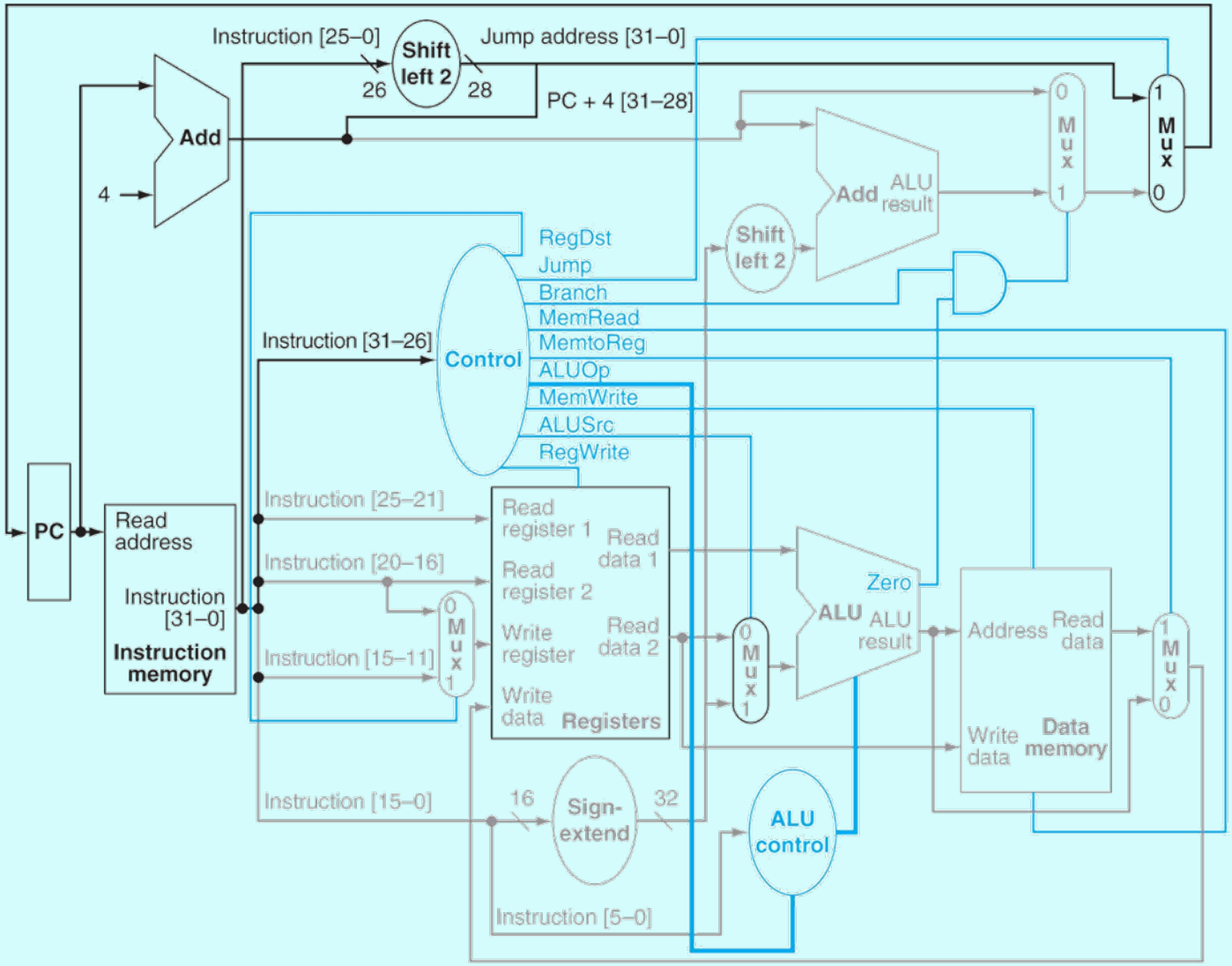
بهار ۱۳۹۱

احمد محمودی ازناوه

فهرست مطالب

- مروری بر جلسه‌ی پیش
- مراحل اجرای دستورالعمل‌ها
- اجرای دستور در چند سیکل
- خط لوله

مسیر گذار داده



مراحل اجرای دستورالعمل در MIPS

- در پردازنده‌های MIPS برای اجرای دستورات به صورت خط لوله، هر دستور در پنج گام انجام می‌شود.
  - واکشی دستورات
  - خواندن محتوای ثبات‌ها و کدگشایی دستورالعمل (قالب منظم MIPS چنین امکانی را مهیا می‌سازد)
  - اجرای دستورالعمل-محاسبه‌ی آدرس
  - دسترسی به حافظه
  - نوشتن پاسخ در ثبات

IF: Instruction fetch

ID: Instruction decode & register read

EX: Execute operation or calculate address

MEM: Access memory operand

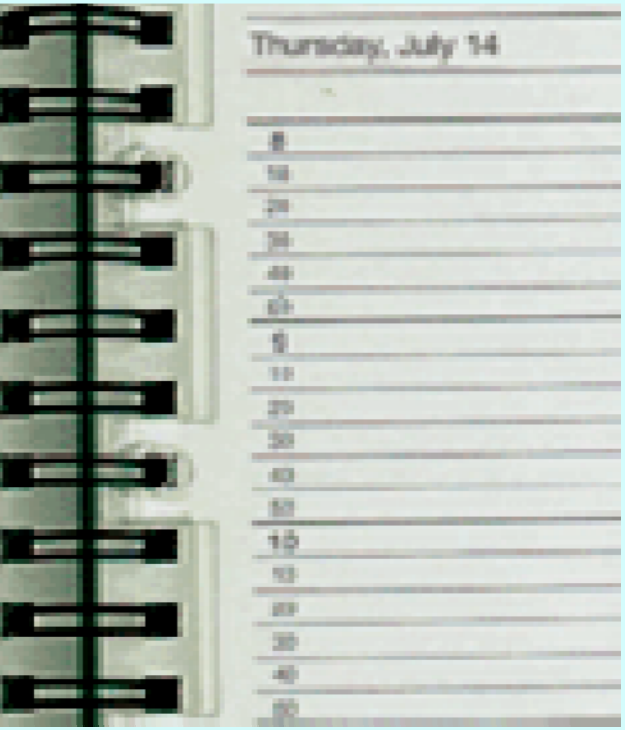
WB: Write result back to register

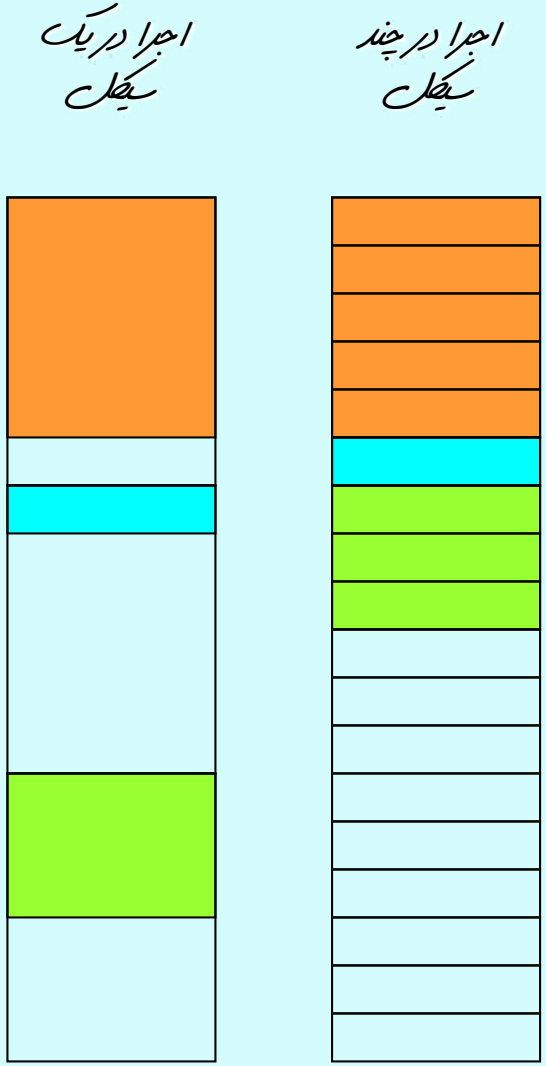
کارایی مسیر گذار داده‌ی تک سیکلی

- فرض می‌کنیم زمان مورد نیاز برای هر کدام از بخش‌های پردازنده به صورت زیر باشد:
  - برای نوشتن در ثبات 100ps
  - دستیابی به حافظه و محاسباتی و ... 200ps
- در معماری یک سیکلی، باید طول پالس را برابر با طول کندترین دستوالعمل در نظر گرفت.

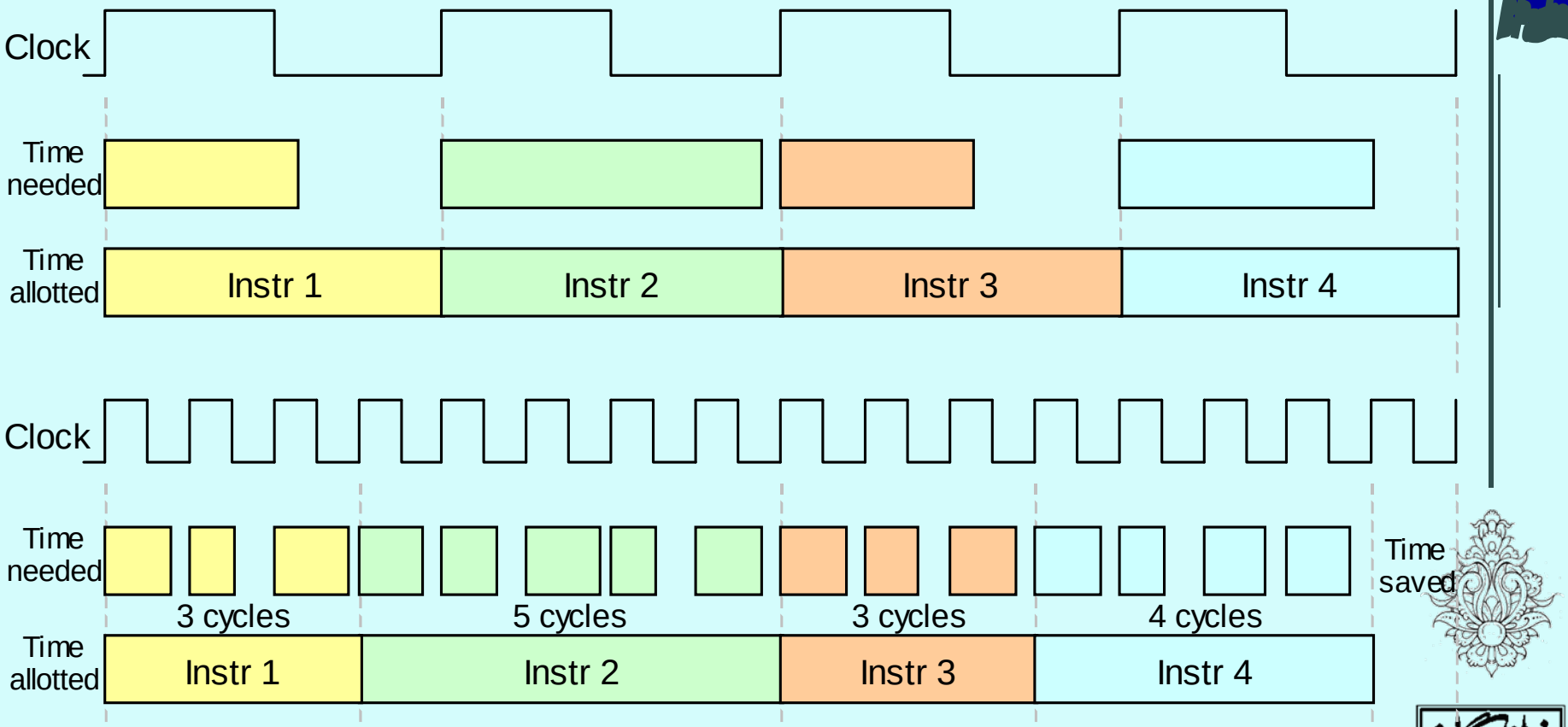
Instr	Instr fetch	Register read	ALU op	Memory access	Register write	Total time
lw	200ps	100 ps	200ps	200ps	100 ps	800ps
sw	200ps	100 ps	200ps	200ps		700ps
R-format	200ps	100 ps	200ps		100 ps	600ps
beq	200ps	100 ps	200ps			500ps

اجرای هر دستورالعمل در چند سیکل(در برابر اجرا در یک سیکل)

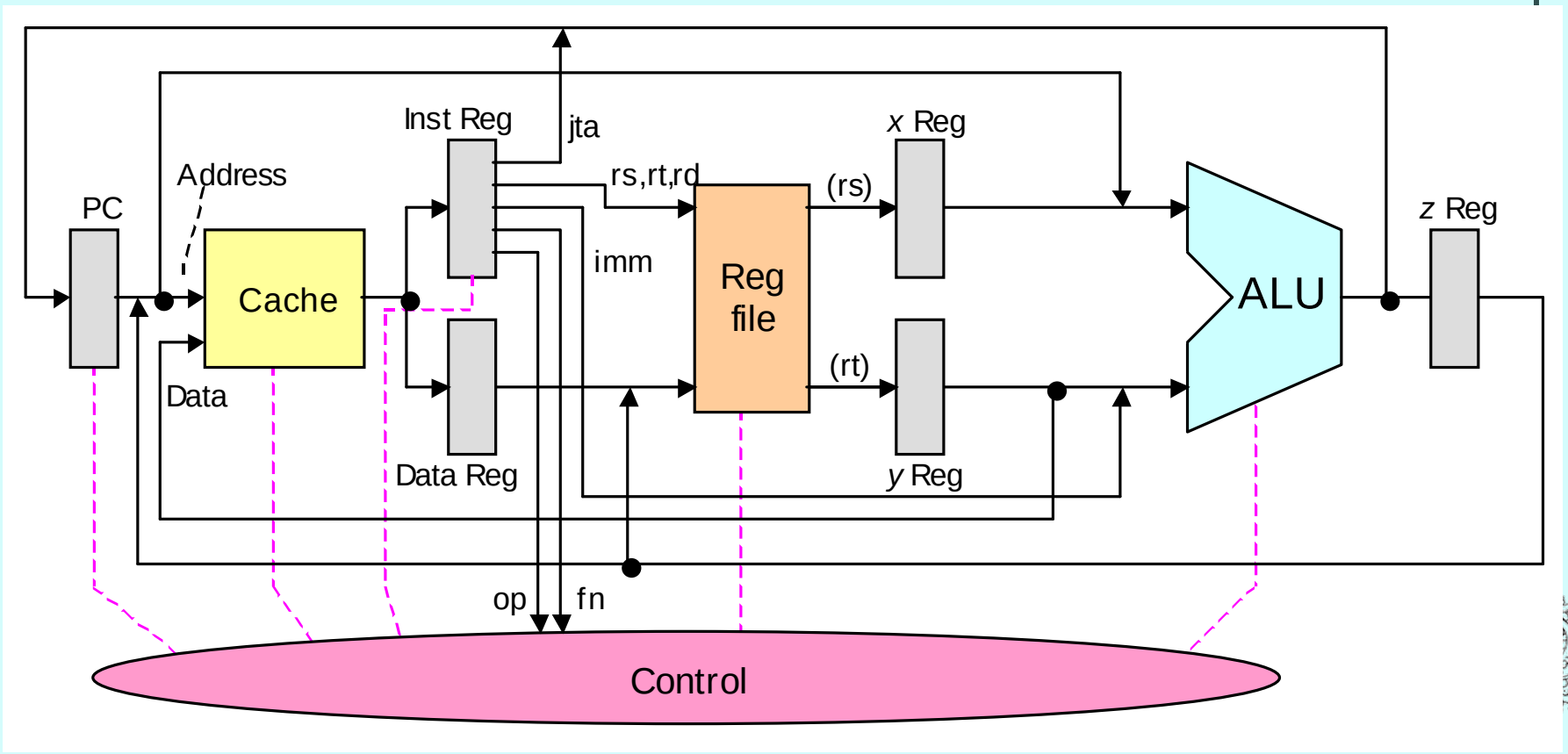




اجرای هر دستورالعمل در چند سیکل(در برابر اجرا در یک سیکل)



مسیر گذار داده‌ی چند سیکلی



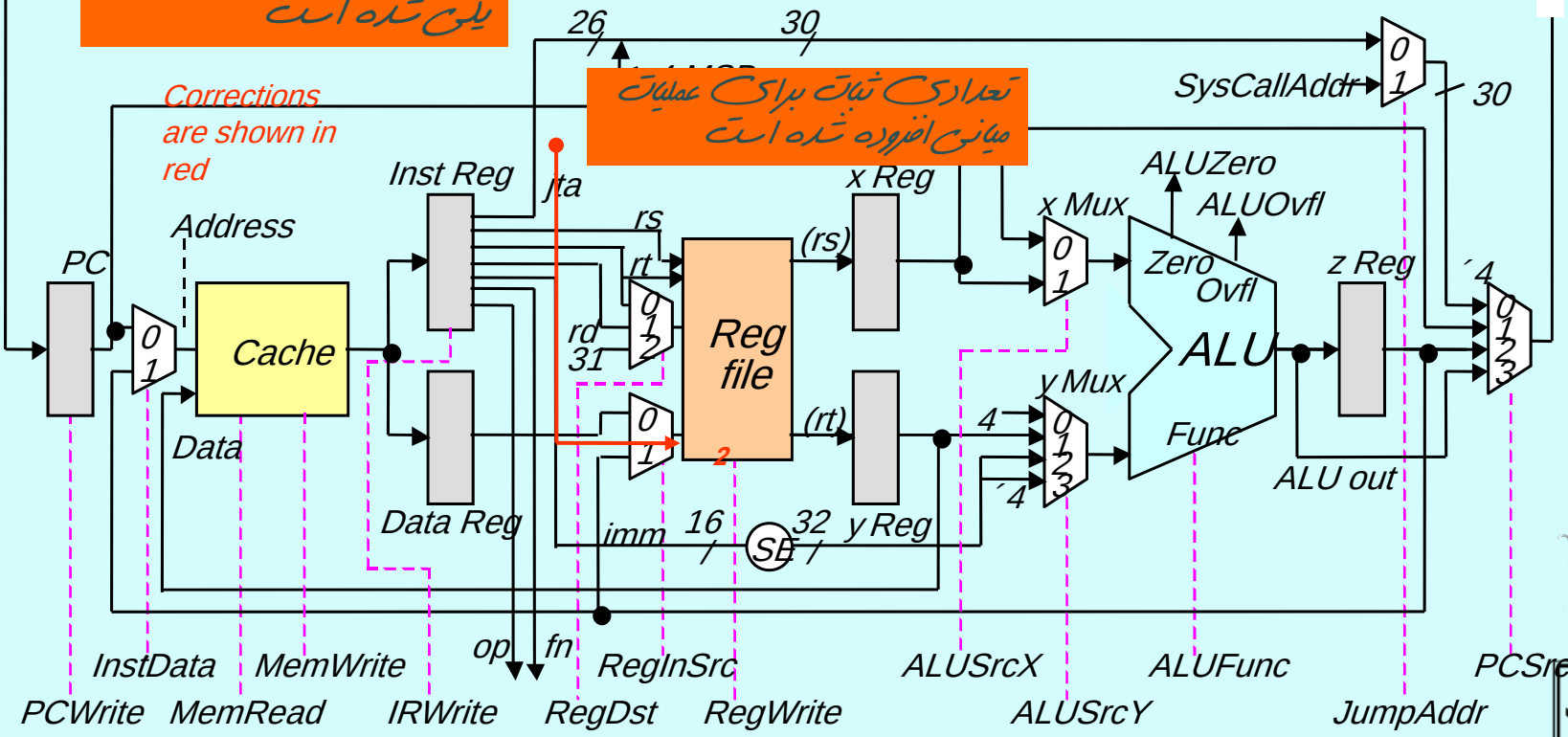
داده‌گذر در حالت چند سیکلی

سه تفاوت عمده با حالت تک سیکلی دارد

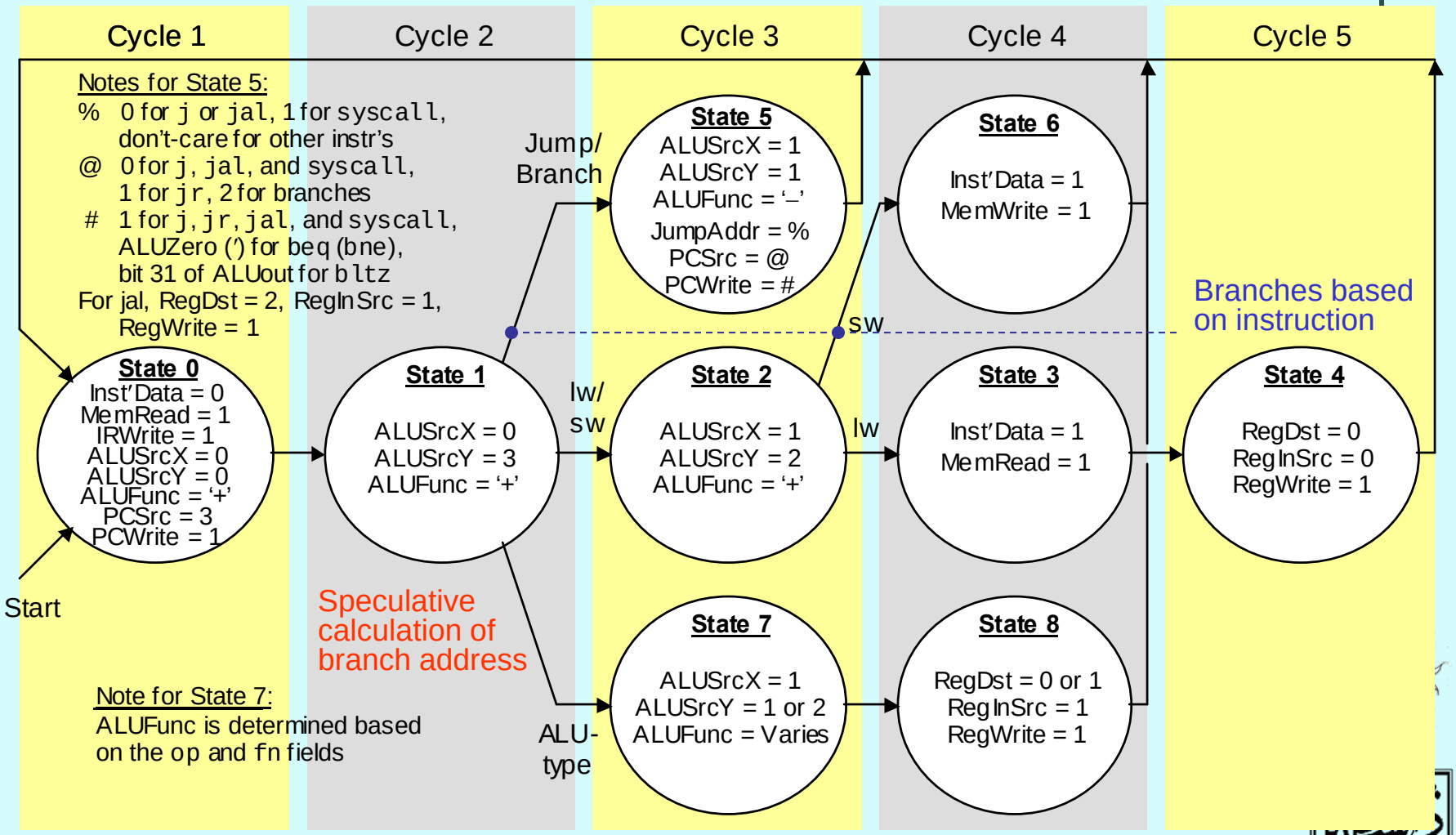
واحد ALU وظیفه‌ی محاسبه‌ی آدرس دستور بعدی را هم دارد

حافظه‌ی برنامه و داده یکی شده است

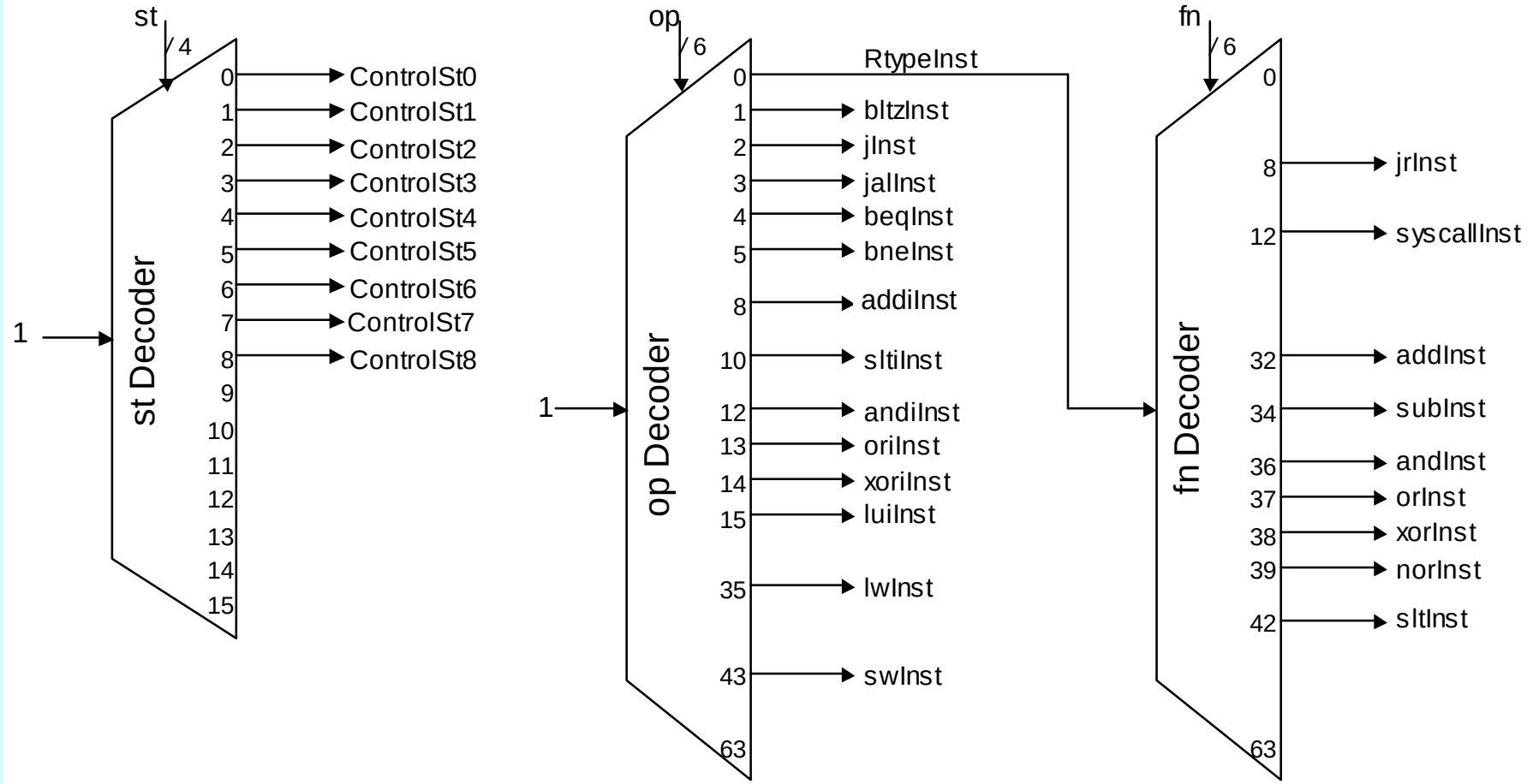
تعدادی ثبات برای عملیات میانی اضافه شده است



ماشین حالت واحد کنترل

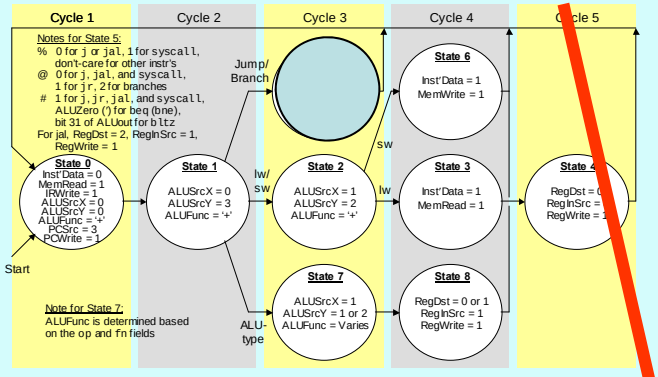


رمزگشایی دستورالعمل‌ها در گام‌های مختلف

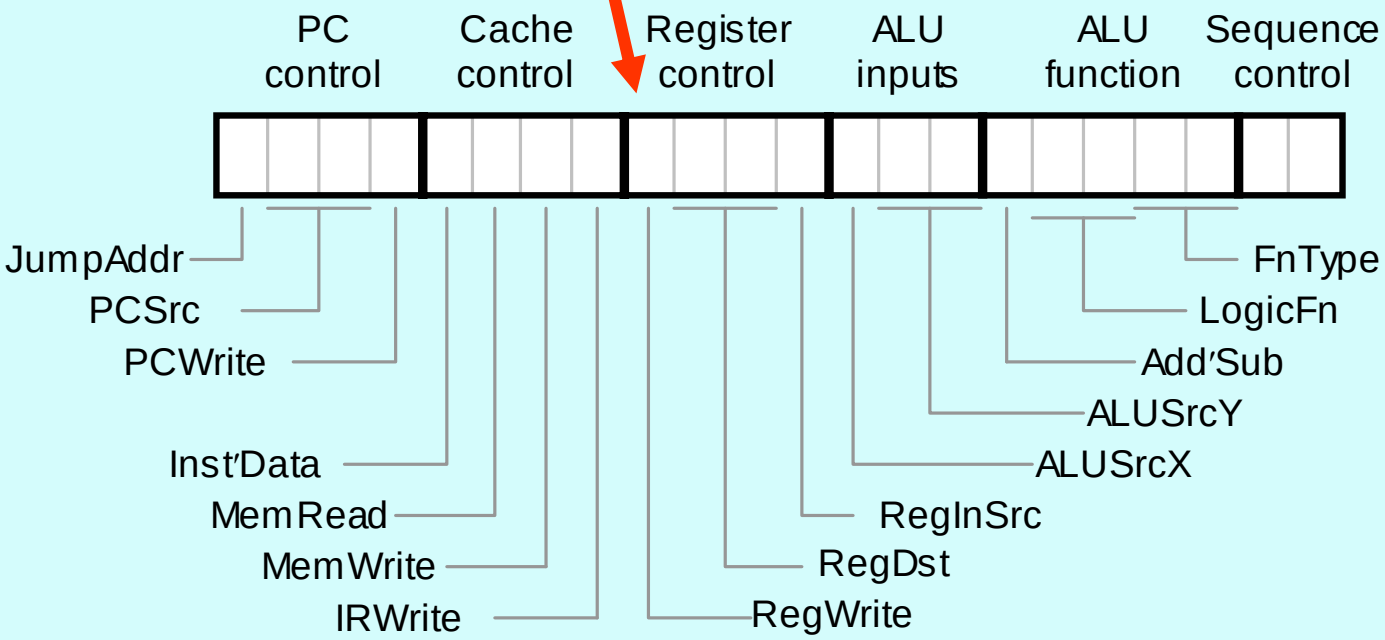


ریزبرنامه

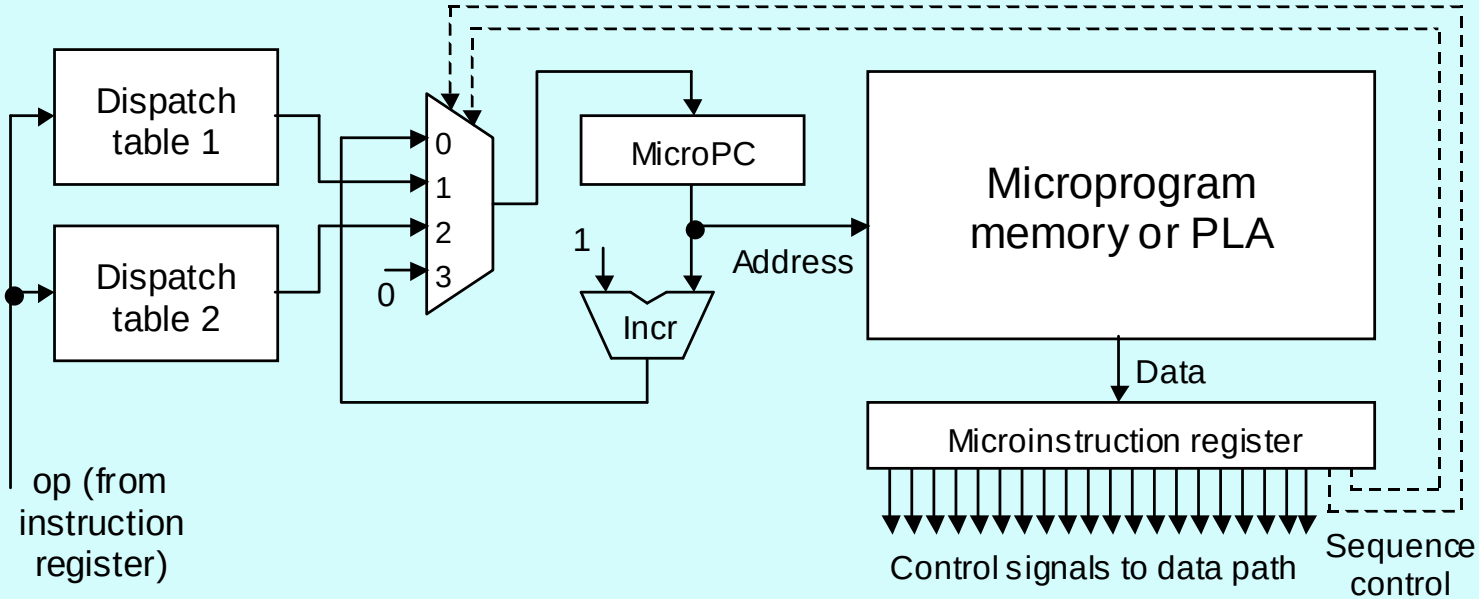
ریزبرنامه



ماشین حالت استفاده شده برای خطوط کنترلی، خود شبیه به یک برنامه است!



ریزبرنامه (ادامه...)



CISC در مقابل RISC

RISC

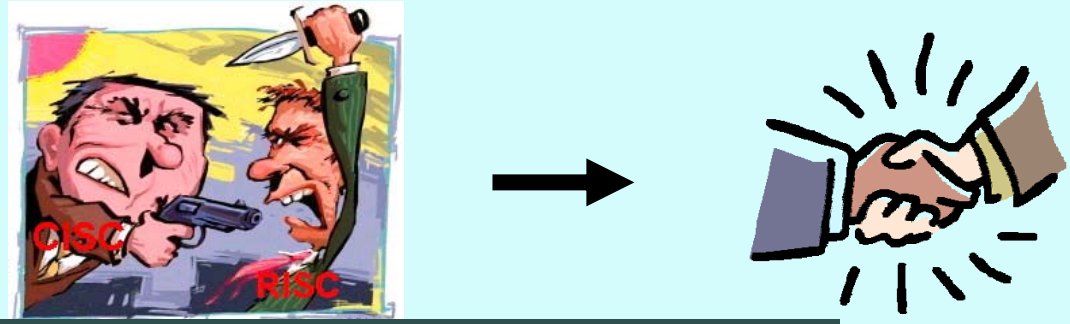
- تعداد دستورات کم
- طول دستورات ثابت
- زمان اجرای ثابت
- هزینه‌ی پایین
- تنها دستورات خواندن و نوشتن به حافظه دسترسی دارند
- همه‌ی عملوندها ثبات‌های پردازنده هستند
- مودهای آدرس محدود
- واحد کنترل به صورت سیم‌بندی

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...)

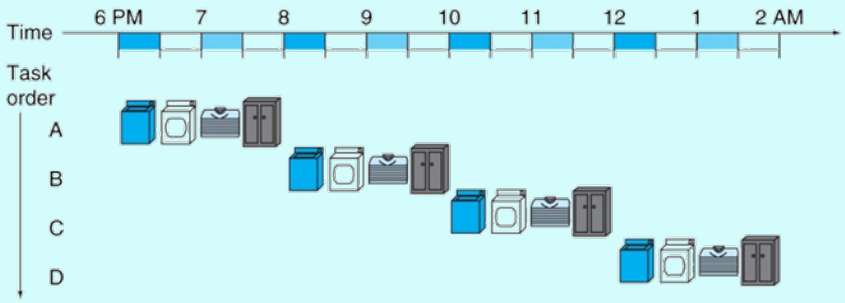
- در عمل مرزهای بین این دو در حال محو شدن هستند.
- پردازنده‌های جدید از خصوصیات هر دو بهره می‌گیرند.
- با این وجود، برای سیستم‌های درون‌کار (توکار) پردازنده‌ی RISC ترجیح داده می‌شوند.



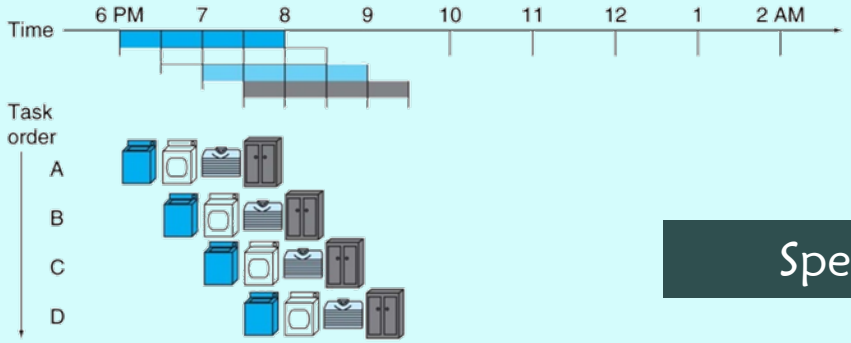
مروری بر خط لوله

- در یک سیستم خط لوله، اجرای چندین دستورالعمل دارای همپوشانی است.
- پایه‌ی خط لوله شبیه خط تولید کارخانه‌هاست.
- تقریبا در تمامی پردازنده‌های موجود از این تکنیک استفاده می‌شود.

مثالی از خط لوله (در رختشویخانه)

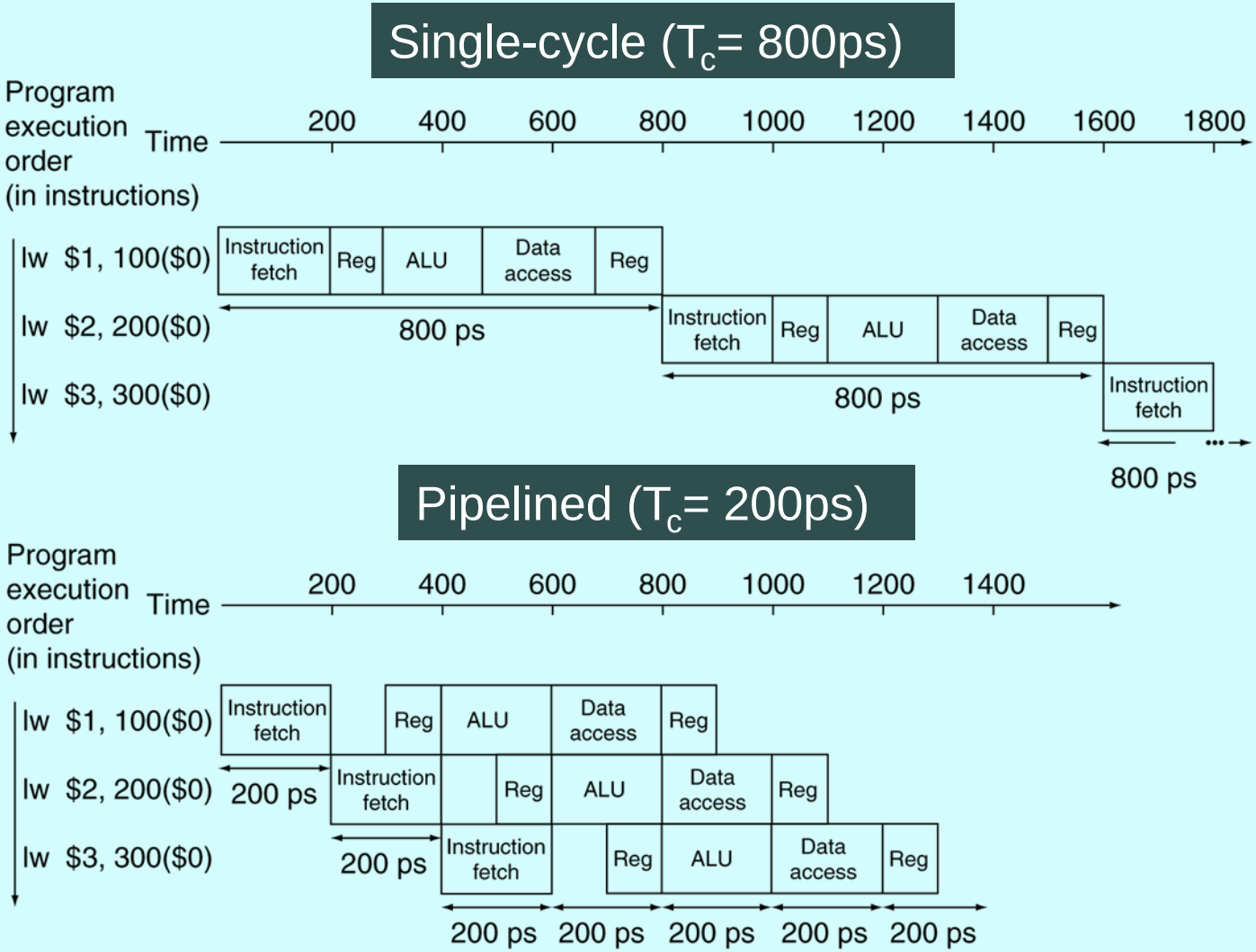


- در صورتی‌که کارها را با همپوشانی انجام دهیم، کارایی افزایش چشمگیری خواهد داشت



Speedup= 8/3.5 = 2.3

کارایی خط لوله (ادامه...)

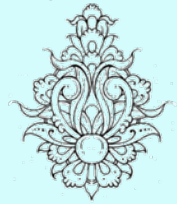


کارایی خط لوله (ادامه...)

- اگر تمامی مراحل متعادل (balanced) باشند؛ تمام مراحل زمان یکسانی صرف کنند

$$\text{Time between instructions}_{\text{pipelined}} = \frac{\text{Time between instructions}_{\text{nonpipelined}}}{\text{Number of stages}}$$

- در صورتی که خط لوله پر باشد، کارایی با تعداد گام‌ها خواهد بود؛ با یک خط لوله‌ی پنج مرحله‌ای سرعت پنج برابر می‌شود



که عشق آسان نمود اول، ولی افتاد مشکل‌ها

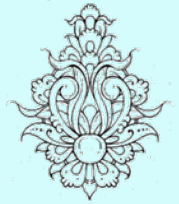
کارایی خط لوله (ادامه...)

- زمانی که هر واحد خط لوله نیاز دارد، یکسان نیست
- افزون بر این، استفاده از خط لوله به سیستم مقداری سربار هم تحمیل خواهد کرد.
- در مثال قبلی، زمان اجرای سه دستور به 1400ps رسید. اگر تعداد دستورات را ۱۰۰۰۰۰۳ در نظر بگیریم، افزایش سرعت تقریبا چهار برابر می‌شود.

1,000,000X800ps+2400

$$\frac{800002400\text{ps}}{200001400\text{ps}} \approx \frac{800\text{ps}}{200\text{ps}} \approx 4.00$$

1,000,000X200ps+1400ps



کارایی خط لوله (ادامه...)

- با استفاده از خط لوله،
  - زمان اجرای یک دستورالعمل افزایش می‌یابد.
  - توان عملیاتی افزایش می‌یابد.

Latency

throughput