معماری کامپیوتر (۱۳۹۱-۱۱-۱۳)

جلسه‌ی پانزدهم

دانشگاه شهید بهشتی

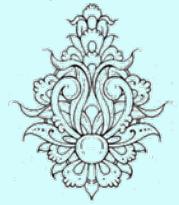
دانشکده‌ی مهندسی برق و کامپیوتر

بهار ۱۳۹۱

احمد محمودی ازناوه

فهرست مطالب

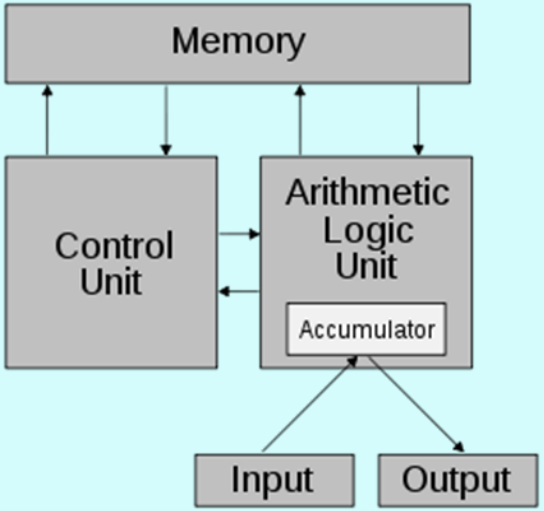
- مروری بر جلسه‌ی پیش
  - مراحل اجرای دستورالعمل‌ها
  - اجرای دستور در چند سیکل
- خط لوله
- مخاطرات خط لوله
  - انواع مخاطرات



معماری Von Neumann

- در این معماری دستورالعمل‌ها و داده‌ها در یک حافظه ذخیره می‌شوند.
- بدین‌ترتیب امکان خواندن همزمان داده و دستورالعمل از حافظه وجود ندارد، چنین مسأله‌ای به تنگنای معماری Von Neumann شهرت دارد.

Von Neumann bottleneck



معماری Harvard

- در معماری Harvard، برای داده‌ها و دستوالعمل‌ها، حافظه و گذرگاه‌داده‌ی جداگانه‌ای در نظر گرفته شده است. این نام در پی ساخت کامپیوتر Harvard Mark I که از حافظه‌های جداگانه برای داده‌ها و دستورالعمل‌ها استفاده می‌کرد، به این معماری اطلاق شده است.
- در اغلب کامپیوترهای امروزه از معماری تعدیل شده‌ی هاروارد (Modified Harvard Architecture) استفاده می‌شود.
- در این شیوه حافظه‌ی نهان مربوط به دستوالعمل و داده‌ها جداگانه هستند، به نوعی می‌توان این شیوه را ترکیبی از دو نوع معماری فوق دانست، چنین شیوه‌ای در پردازنده‌های x86، ARM، PowePc و MIPS مورد استفاده قرار می‌گیرد.

اجرای دستورات به صورت سری

Cycles \ Stages	S1	S2	S3	S4	S5	S6
1	I-1					
2		I-1				
3			I-1			
4				I-1		
5					I-1	
6						I-1
7	I-2					
8		I-2				
9			I-2			
10				I-2		
11					I-2	
12						I-2

- تا پیش از پردازنده‌های ۸۰۴۸۶، اجرای دستورات به صورت ترتیبی در شش مرحله صورت می‌پذیرفت. در پردازنده‌های ۸۰۴۸۶ این شش مرحله به صورت خط لوله در آمد.

خط لوله (pipeline)

- «خط لوله یک شگرد پیاده‌سازی است که در آن چندین دستورالعمل به طور هم‌پوشان (overlapped) به اجرا در می‌آید. در پردازنده‌های خانواده‌ی X86 نخستین بار در ۸۰۴۸۶ از خط لوله استفاده شد.

Stages

Cycles	S1	S2	S3	S4	S5	S6
1	I-1					
2	I-2	I-1				
3		I-2	I-1			
4			I-2	I-1		
5				I-2	I-1	
6					I-2	I-1
7						I-2

در یک خط لوله‌ی k مرحله‌ای، برای انجام n دستور چند سیکل ساعت زمان لازم است؟

k + (n – 1)

خط لوله (ادامه...)

- اگر زمانی که گام‌های مختلف خط لوله نیاز دارند یکسان نباشد، «ظرفیت گذردهی» چه تفاوتی خواهد کرد؟ فرض کنید گام چهارم به دو سیکل نیاز داشته باشد؟

Throughput

Stages

exe

Cycles

	S1	S2	S3	S4	S5	S6
1	I-1					
2	I-2	I-1				
3	I-3	I-2	I-1			
4		I-3	I-2	I-1		
5			I-3	I-1		
6				I-2	I-1	
7				I-2		I-1
8				I-3	I-2	
9				I-3		I-2
10					I-3	
11						I-3

در چنین حالی، اجرای n دستور چند سیکل زمان لازم است؟

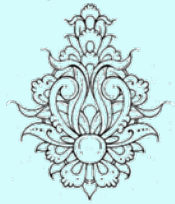
$k + (2n - 1)$

Slides prepared by Kip R. Irvine

طراحی ISA برای خط لوله

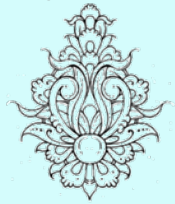
•تمام دستورات تمام دستورات MIPS طول یکسانی دارند. مرحله‌ی واکشی به سادگی انجام می‌شود.

•قالب دستورها مشابه هستند. بدین‌ترتیب کدگشایی ساده‌تر خواهد بود. کدگشایی دستور و خواندن محتوای ثبات در یک گام صورت می‌پذیرد. این که ثبات منبع جای مشخصی دارد، پیش از تمام شدن کدگشایی دستور محتوای ثبات خوانده می‌شود. وگرنه، گام‌های خطلوله به شش گام می‌رسید.



طراحی ISA برای خط لوله (ادامه...)

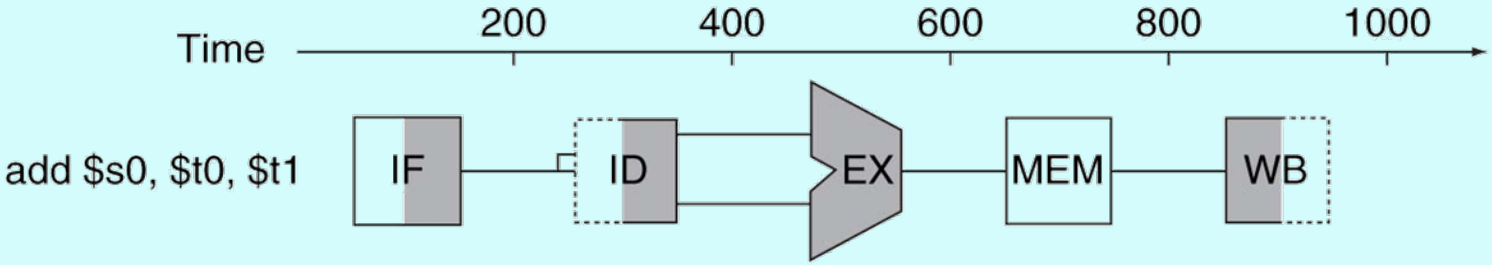
- در مجموعه دستورات MIPS، دسترسی به حافظه فقط در دستورهای lw/sw امکان‌پذیر می‌باشد. در صورتی که در دستورهای محاسباتی استفاده از عملوند حافظه مجاز بود، گام سوم و چهارم به گام‌های یافتن آدرس، خواندن حافظه و اجرای دستور گسترش می‌یافت.



مخاطره / مانع

به نقل از ترجمه‌ی آقای دکتر ملکیان

- گاهی اوقات در مسیر خط لوله شرایطی پیش می‌آید که دستورالعمل بعدی را نمی‌توان در سیکل ساعت بعدی اجرا کرد. به چنین رخدادهایی اصطلاحاً «خطرات ناخواسته» یا به عبارت بهتر «مخاطرات و موانع سدّ راه در خط لوله» گفته می‌شود و با سه نوع مختلف از آن مواجه خواهیم بود.



این شکل مراحل مختلف یک خط لوله، برای دستور add را نشان می‌دهد، شکل‌هایی که سایه نخورده‌اند، در این دستور مورد استفاده قرار نمی‌گیرند، بخش‌های که در سمت راست (چپ) سایه خورده‌اند، بیانگر خواندن (نوشتن) داده از (در) حافظه می‌باشد

مخاطرات خط لوله!!

Structural Hazard

– مخاطرات ساختاری

- یکی از منابع مورد نیاز مشغول است

– مخاطرات داده‌ای

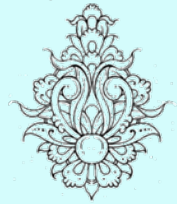
- به داده‌ای نیاز است که توسط دستور قبلی آماده نشده است

Data Hazard

– مخاطرات کنترلی

- چنان‌چه دستورات کنترلی بخواهند بر اساس اجرای دستور قبلی تصمیم‌گیری کنند.

Control Hazard



مخاطرات ساختاری

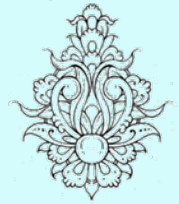
- اولین مانع در خط لوله به **مخاطرات ساختاری** شهرت دارد و بدین معناست که سخت‌افزار فی‌نفسه قادر به پشتیبانی ترکیبی خاص از دستورالعمل‌هایی که می‌خواهیم در یک سیکل واحد آنها را اجرا کنیم نیست.
- تعارض در هنگام استفاده از منابع مشترک
  - در MIPS با یک حافظه برای دستورالعمل‌ها و داده‌ها
    - دستورات خواندن و نوشتن به حافظه احتیاج دارند

stall

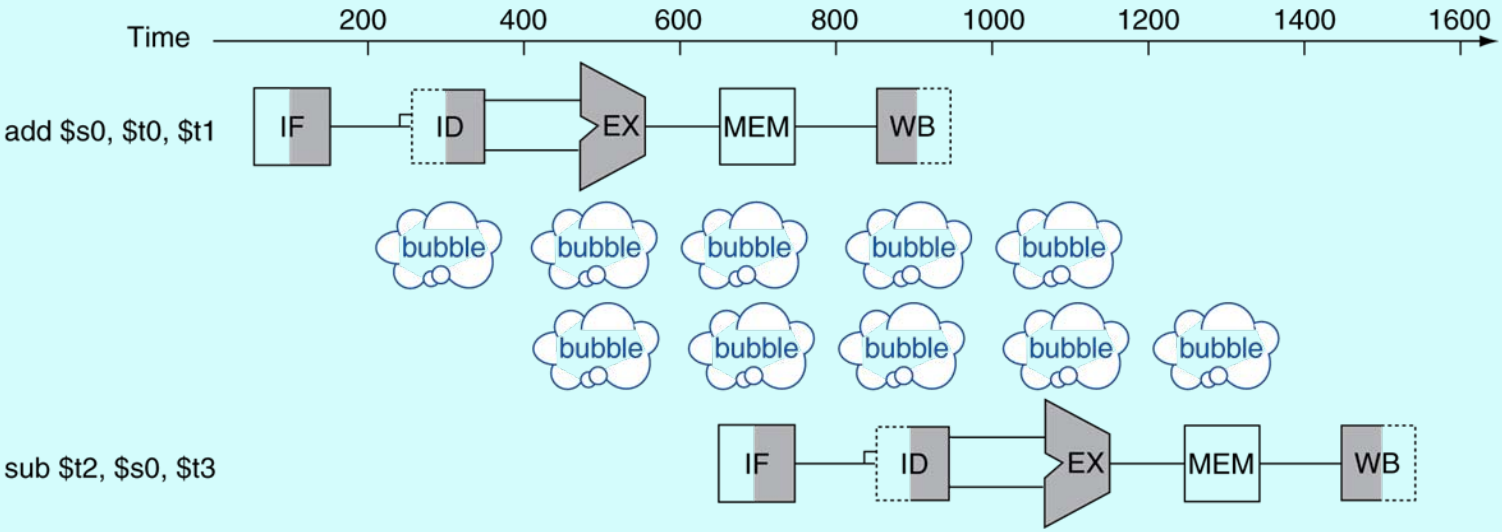
- در این صورت برای واکشی دستورات بعدی، تعلیق می‌شود.
  - در چنین حالاتی به جای دستورات بعدی حباب وارد خط لوله می‌شود.

Bubble

- خط لوله‌ی داده‌گذر (datapath) به دو حافظه‌ی داده و دستور نیاز دارد.



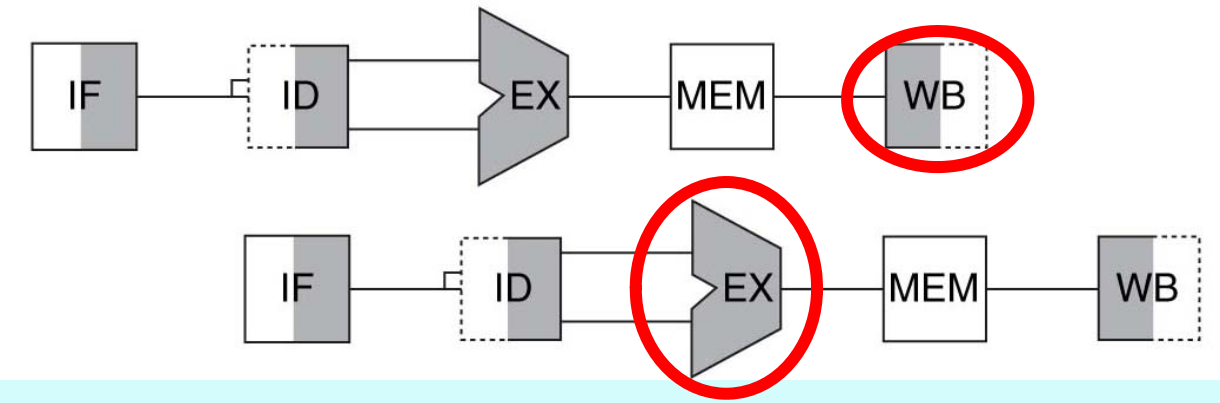
تعلیق

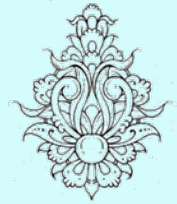


- این شکل یکی از مفاهیم بسیار با اهمیت در ایجاد خط لوله را به تصویر کشیده است که اگرچه عنوان رسمی «تعلیق خط لوله» دارد ولیکن اغلب اوقات از این پدیده با نام ساده‌تر «حباب» یاد می‌شود. در ادامه‌ی بحث به موانع منجر به «تعلیق» در بخش‌های دیگری از خط لوله خواهیم پرداخت.

- مخاطرات و موانع ناشی از **داده** موقعی رخ خواهد داد که خط لوله باید در انتظار تکمیل یکی از مراحل قبلی از حرکت باز داشته شود

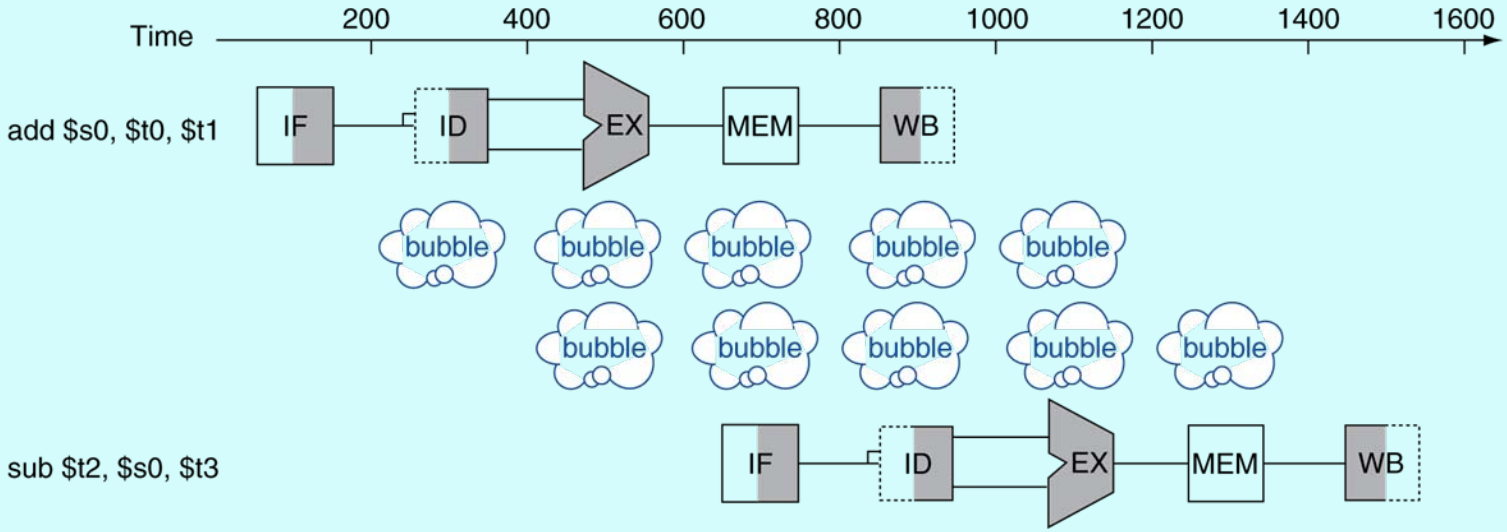
```
add $s0, $t0, $t1
sub $t2, $s0, $t3
```





- یک دستورالعمل به داده‌ای نیاز دارد، که در دستور قبلی مشغول آماده کردن آن است

```
add $s0, $t0, $t1
sub $t2, $s0, $t3
```



پیش فرستادن

- در حالاتی که مخاطره‌ی داده رخ می‌دهد، پس از انجام دستورالعمل و آماده شدن داده، نتیجه به دست آمده پیش از ذخیره در ثبات، در دستور بعدی استفاده می‌شود.
- چنین حالتی، نیاز به اتصالات بیشتری در داده‌گذر (datapath) دارد.