معماری کامپیوتر

پایپ‌لاین و پردازش برداری

فصل نهم کتاب موریس مانو

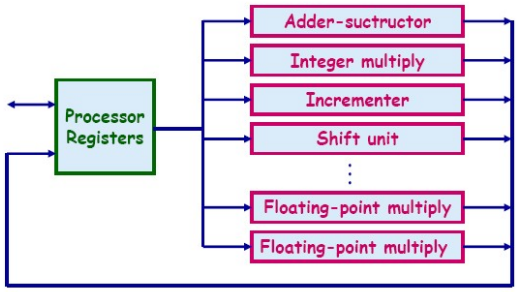
محمدعلی شفیعیان

http://shafieian-education.ir

پردازش موازی

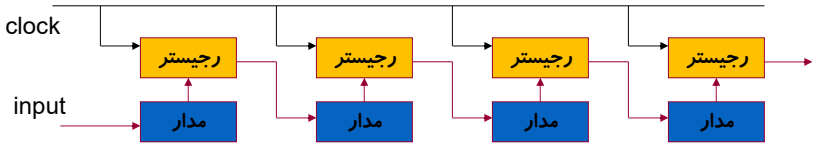
- پردازش چندین کار به صورت همزمان

- هدف از پردازش موازی
  - افزایش سرعت
  - افزایش توان عملیاتی سیستم (تعداد کارها در واحد زمان)



پایپ‌لاین

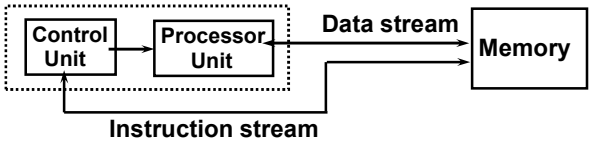
- Pipelining عبارت است از تجزیه یک فرآیند ترتیبی به عملیات ریزتر به‌گونه‌ای که هر یک را بتوان بوسیله یک بخش اختصاصی به‌صورت همزمان انجام داد
- هر قسمت از پایپ‌لاین کار خاصی را روی داده انجام داده و به قسمت بعدی تحویل می‌دهد. وقتی که داده از همه قسمتها رد شد نتیجه در خروجی پایپ‌لاین آماده خواهد بود.
- نام این عمل از خط مونتاژ اتومبیل گرفته شده است.
- هر قسمت از پایپ لاین می‌تواند شامل یک رجیستر و یک مدارترکیبی باشد. رجیستر داده را نگه می‌دارد و مدارترکیبی عمل لازم را روی آن انجام می‌دهد تا در کلاک بعد تحویل رجیستر قسمت بعدی شود.



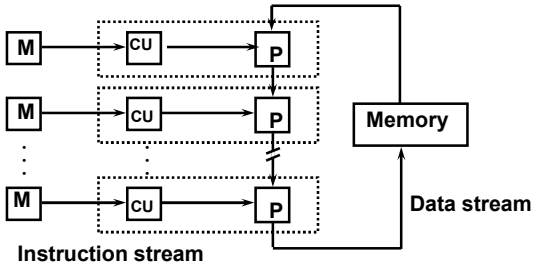
انواع سیستم های پردازش موازی

- SISD
  - یک دستور، یک داده
- SIMD
  - یک دستور، چند داده
  - چندین پردازشگر، که تمام آنها یک دستور را بر روی چندین داده اعمال می کنند.
- MISD
  - چند دستور، یک داده
  - از این نوع ماشینی وجود ندارد
- MIMD
  - چند دستور، چند داده
  - ماشینهای چندپردازنده (مالتی پروسسور)

SISD (Single Instruction Single Data)

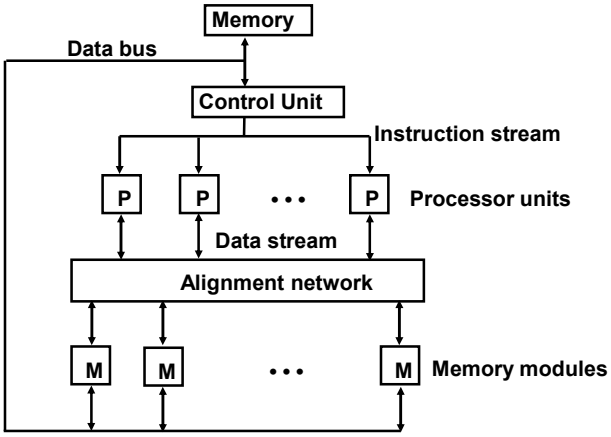


MISD (Multiple Instruction Single Data)



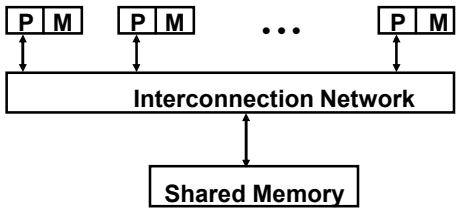
SIMD (Single Instruction Multiple Data)

- تنها یک دستور در هر لحظه اجرا می شود
- این دستور بر روی داده های مختلف



MIMD (Single Instruction Single Data)

- ✓ هر پردازنده یک حافظه مربوط به خود را دارد.
- ✓ در هر لحظه دستورات مختلف بر روی داده های مختلف
- ✓ تمام حافظه ها به یک حافظه مرکزی متصل هستند.
- ✓ مثل، چند پردازنده ها با حافظه مشترک



انواع سیستم های پردازش موازی

نکته ۱: طبقه‌بندی Flynn همه معماری‌ها را پوشش نمی‌دهد.

نکته ۲: فرض کنید مدت زمان اجرای یک برنامه روی یک پردازنده برابر با t است. حال اگر همین برنامه را روی P پردازنده مشابه تقسیم کنیم:

در حالت ایده‌آل ⇐ زمان اجرا $= \frac{t}{P}$

اما در واقعیت چنین نیست، زیرا:

- میزان مبادله اطلاعات بین P پردازنده، زمان اجرا را افزایش می‌دهد.
- ممکن است دستوراتی وجود داشته باشند که قابل موازی شدن نباشند.

انواع سیستم های پردازش موازی

نکته ۳: فرض کنید f درصد از دستورات برنامه $(0 \leq f \leq 1)$ قابل موازی شدن نباشند یعنی به‌هم وابستگی داشته باشند و باید توسط یک پردازنده اجرا شوند که به آن مؤلفه ترتیبی برنامه می‌گویند. اگر از زمان مبادله اطلاعات صرف‌نظر کنیم:

زمان اجرای P پردازنده $= \left(f - \frac{1-f}{P}\right) t$

تسریع P پردازنده نسبت به یک پردازنده: $S = \frac{\text{زمان اجرای یک پردازنده}}{\text{زمان اجرای P پردازنده}} = \frac{t}{\left(f - \frac{1-f}{P}\right) t}$

اگر تعداد پردازنده‌ها را به سمت بی‌نهایت میل دهیم ⇐ $\lim_{P \to \infty} S = S_{max} = \lim_{P \to \infty} \frac{1}{f - \frac{1-f}{P}} = \frac{1}{f}$

انواع سیستم های پردازش موازی

نکته ۴: اگر در برنامه‌ای ۲۰ درصد دستورات موازی نشوند، آن‌گاه تسریع از $\frac{1}{0.2}$ (یعنی ۵) کمتر است و این امر برای پردازنده‌ها یک عامل محدودکننده است.

بنابراین باید به دنبال روشی باشیم که بتوان عملیات ترتیبی را تسریع نمود که برای این منظور می‌توان از پایپ‌لاین (خط لوله) بهره برد.

خط لوله (پایپ‌لاین)

✓ تکنیکی جهت افزایش سرعت

✓ عملیات متوالی (سری) را به چند ریز عملیات موازی تقسیم می‌کنیم.

مثال شستن لباسها

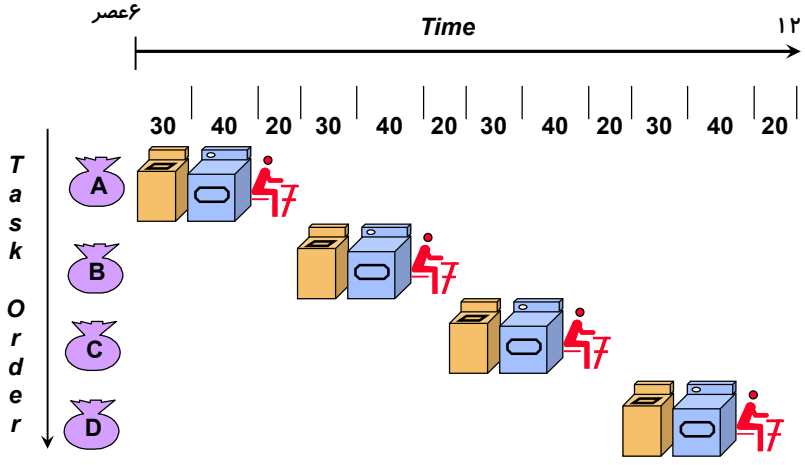
- چهار بسته لباس برای شستن، خشک کردن و تا کردن داریم. برای هر مرحله یک دستگاه جداگانه داریم.

- شستن ۳۰ دقیقه زمان می برد.

- خشک کن ۴۰ دقیقه زمان می برد.

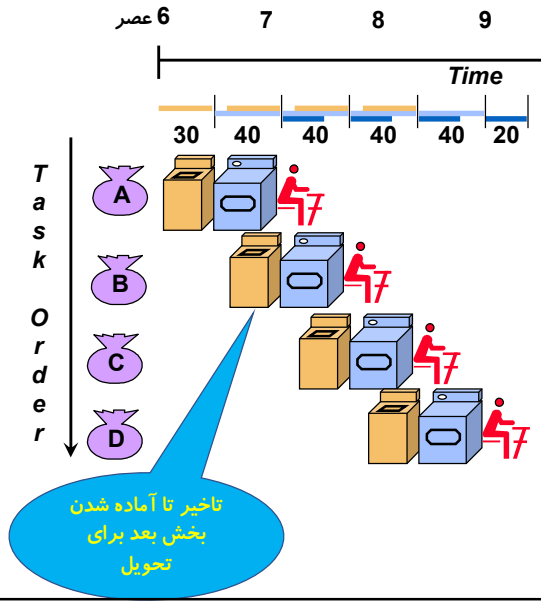
- تا کردن ۲۰ دقیقه زمان نیاز دارد.

شستن ترتیبی



شستن ترتیبی لباس ها شش ساعت طول می کشد.

شستن خط لوله‌ای شده



زمان موردنیاز دراین حالت سه ساعت و نیم است.

ویژگی های خط لوله

- ✓ هر عملیات را که بتوان به چندین عملیات جزئی تقسیم کرد، می‌تواند از تکنیک خط لوله استفاده کند.
- ✓ زمانی مفید است، که بخواهیم آن کار را به تعداد دفعات زیادی تکرار کنیم.
- ✓ خط لوله‌ای زمان اجرای یک عملیات را کاهش نمی‌دهد، بلکه تعداد عملیات انجام شده در واحد زمان را افزایش می دهد (توان عملیاتی)
- ✓ اجرای همزمان چند کار نیاز به منابع متفاوتی دارد.
- ✓ متوقف شدن به خاطر وابستگی ها(STALL)
- ✓ باید سعی شود، بخش هایی با تأخیر یکسان داشت.

مثالی از پایپ لاین

می‌خواهیم عمل جمع وضرب را بر روی دنباله‌ای از داده‌ها انجام دهیم:

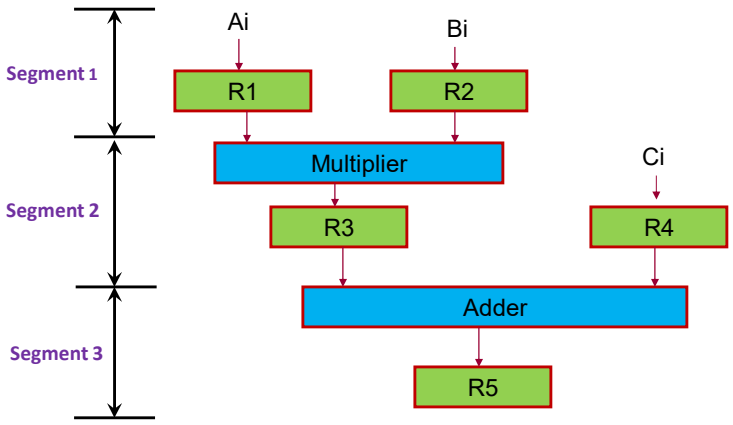
$A_i * B_i + C_i$ for i=1,2,3,..,7

هر عمل را می‌توان در یک قسمت از پایپ‌لاین انجام داد.

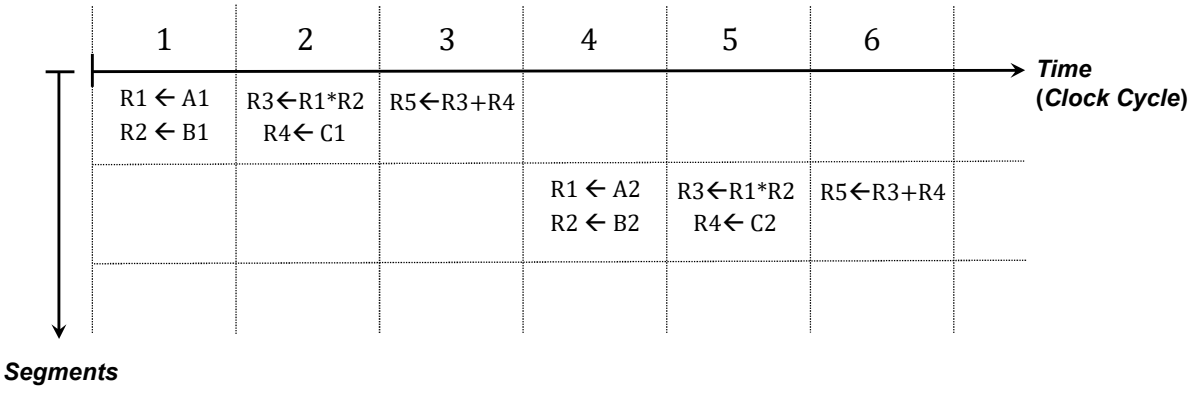
T1: $R1 \leftarrow A_i$, $R2 \leftarrow B_i$

T2: $R3 \leftarrow R1 * R2$, $R4 \leftarrow C_i$

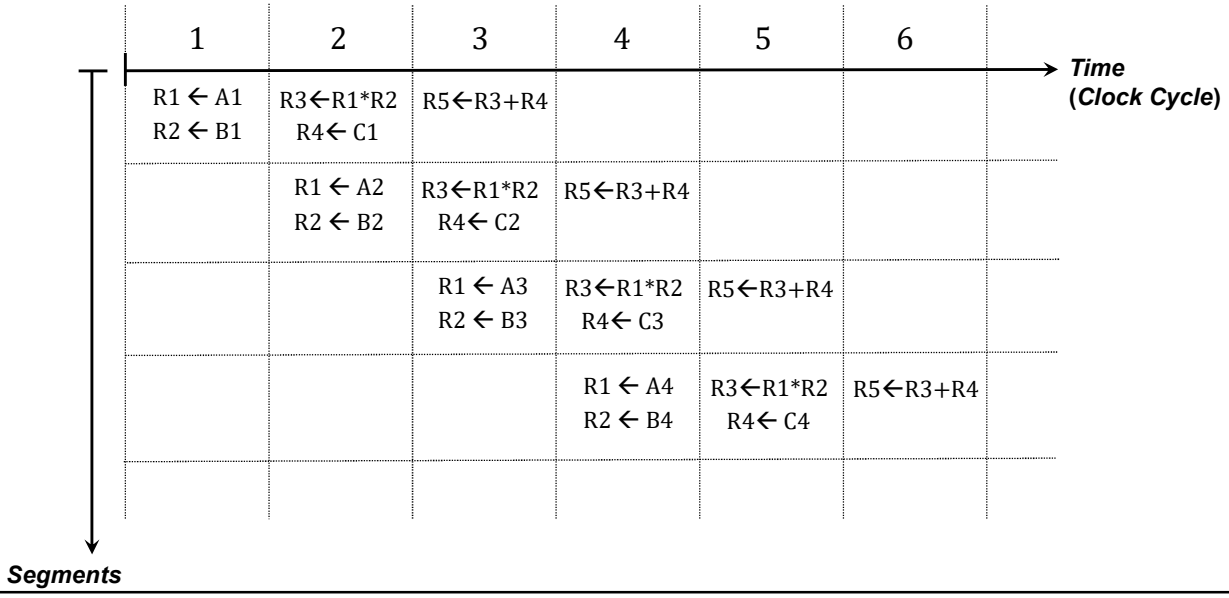
T3: $R5 \leftarrow R3 + R4$



مثالی از پایپ لاین

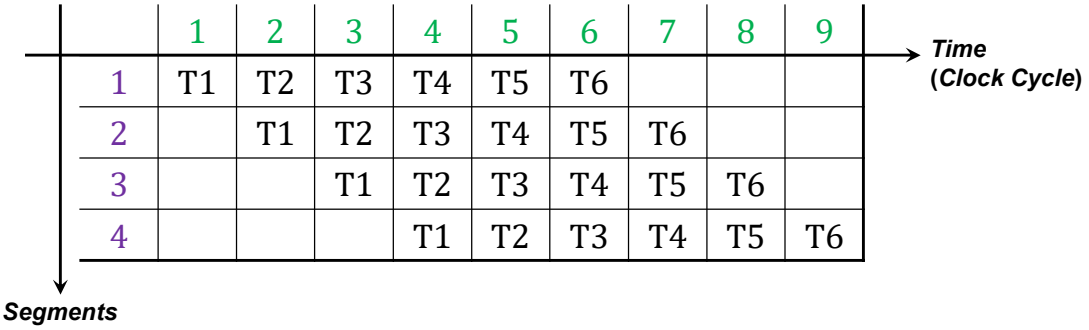


مثالی از پایپ لاین



نمودار Time-Space

- عملکرد پایپ لاین را می‌توان با نمودار Time-Space نشان داد.
- تمام عملیاتی که بر روی اطلاعات به‌ویژه در خط لوله انجام می‌شود یک عمل یا وظیفه (Task) می‌نامیم.
- اگر یک کار (Task) به k قسمت (Segment) تقسیم شده باشد برای انجام n عمل نیاز به k+(n-1) پالس نیاز خواهیم داشت.



ضریب تسریع

$$Speedup = S = \frac{\text{زمان اجرا بدون خط لوله}}{\text{زمان اجرا با خط لوله}} = \frac{\text{تعداد کارها} \times \text{زمان هر کار}}{\text{زمان اجرای یک کار} + \text{تعداد کارها منهای یک}}$$

$$S = \frac{nt_n}{(k+n-1)t_p}$$

- t_p : زمان پالس ساعت در حالت استفاده از پایپ‌لاین
- n : تعداد وظیفه‌ها (Task)
- k : تعداد قسمت‌ها (Segments)
- t_n : زمان اجرای هر وظیفه بدون استفاده از پایپ‌لاین

با زیاد شده n نسبت به k-1 مقدار k+n-1 نزدیک به n می‌شود

$$\lim_{n\to\infty} S = \lim_{n\to\infty} \frac{nt_n}{(k+n-1)t_p} = \frac{t_n}{t_p}$$

ضریب تسریع

✓ در بهترین حالت تمام قسمت‌ها دارای زمان یکسانی هستند.

$$t_n = kt_p = t \quad \rightarrow \quad S = \frac{nkt}{(k+n-1)t} = \frac{nk}{(k+n-1)}$$

✓ وقتی تعداد کارها به سمت بی‌نهایت می‌رود:

$$\lim_{n\to\infty} S = \lim_{n\to\infty} \frac{nk}{(k+n-1)} = k$$

بنابراین حداکثر ضریب تسریع k خواهد بود، ولی در عمل این حالت رخ نمی‌دهد.

- ✓ به دلیل برابر نبودن زمان هر بخش (Segment)
- ✓ تأخیر نگهداری داده
- ✓ وابستگی داده

ضریب تسریع

از لحاظ تئوری مقدار افزایش سرعت k برابر خواهد بود. اما درعمل نمی‌توان همیشه با این سرعت کار کرد:

- ✓ قسمت های مختلف ممکن است زمان متفاوتی داشته باشند. دراین صورت کلاک بر اساس کندترین آنها تنظیم می‌شود.
- ✓ با طراحی یک سیستم بصورت پایپ‌لاین رجیسترهایی به سیستم اضافه می‌شود که در حالت عادی لازم نیستند.

نکته: می‌توان بازدهی یا کارآیی پایپ‌لاین را برابر $E = \frac{S}{k}$ یعنی حاصل تقسیم تسریع به تعداد قسمت‌ها تعریف کرد.

ضریب تسریع

مثال: فرض کنید یک خط لوله ۴ قسمت دارد که تاخیر قسمت‌ها ۶۰، ۵۰، ۸۰ و ۹۰ نانوثانیه است و تأخیر ثبات ۱۰ نانو ثانیه می باشد. حداکثر تسریع این خط لوله نسبت به اجرای غیر خط لوله را محاسبه نمایید.

پریود کلاک : $T = \max(t_i) + t_R = 90 + 10 = 100\ \text{ns}$

زمان در حالت استفاده از پایپ‌لاین : $t_p = 4 \times 100 + (\text{n} - 1) \times 100 = 100\text{n} + 300$

زمان بدون استفاده از پایپ‌لاین : $t_n = n \times (60 + 50 + 90 + 80) = 280n$

تسریع : $S = \frac{t_n}{t_p} = \frac{280n}{100n + 300} \rightarrow S_{max} = \lim_{n\to\infty} S = \frac{280}{100} = 2.8$

ضریب تسریع

مثال: فرض کنید یک خط لوله ۴ قسمت دارد و برای آن $t_p = 20\ ns$ است. ضمناً این خط لوله ۱۰۰ وظیفه را انجام می‌دهد. ضریب تسریع این خط لوله را محاسبه نمایید.

زمان اجرای کل در خط لوله $= (k + n - 1)t_p = (4 + 99) \times 20 = 2060\ \text{ns}$

$$t_n = k\ t_p = 4 \times 20 = 80\ \text{ns}$$

زمان اجرای کل بدون استفاده از پایپ‌لاین $= nk\ t_p = 100 \times 80 = 8000\ \text{ns}$

تسریع : $S = \dfrac{8000}{2060} = 3.88 \quad \rightarrow S_{max} = \lim_{n\to\infty} S = 4$

ضریب تسریع در مثال شستشوی لباس

برای n بسته لباس:

$$S = \frac{n(30 + 40 + 20)}{(30 + 40 + 20) + (n - 1)(40)}$$

30	40	20	30	40	20
شستن	خشک کردن	تا کردن			
			شستن	خشک کردن	تا کردن

30	40	40	40	40	40	20
شستن	خشک کردن	تا کردن				
	شستن	خشک کردن	تا کردن			
		شستن	خشک کردن	تا کردن		
			شستن	خشک کردن	تا کردن	
				شستن	خشک کردن	تا کردن

طبقه بندی خط لوله

۱- خط لوله محاسباتی
عملیات ممیز شناور

۲- خط لوله دستورالعمل

۳- خط لوله پردازنده
- ✓ سری کردن چندین پردازنده با وظایف مختلف

خط لوله دستورالعمل

برای این منظور باید مراحل یک دستورالعمل را به چندین قسمت تقسیم کنیم:

۱- واکشی دستورالعمل
۲- دیکد(رمزگشایی) دستورالعمل و واکشی عملوند
۳- اجرای دستورالعمل
۴- دستیابی به حافظه
۵- کامل نویسی

IF	ID	EX	ME	WB

1	2	3	4	5	6	7	8	9
IF(1)	ID(1)	EX(1)	ME(1)	WB(1)				
	IF(2)	ID(2)	EX(2)	ME(2)	WB(2)			
		IF(3)	ID(3)	EX(3)	ME(3)	WB(3)		
			IF(4)	ID(4)	EX(4)	ME(4)	WB(4)	
				IF(5)	ID(5)	EX(5)	ME(5)	WB(5)

پایپ لاین دستورالعمل

اگر زمان واکشی و اجرا مساوی باشند با این نوع پایپ‌لاین سرعت اجرای دستورات دو برابر می‌شود. اما در عمل چنین اتفاقی نمی افتد. زیرا:

- ✓ زمان اجرا معمولاً بیشتر از زمان واکشی است لذا واحد واکشی مجبور است تا تمام شدن اجرا صبر کند.
- ✓ وجود دستورات انشعاب شرطی باعث می‌شود تا توالی اجرای دستورات به‌هم خورده و با معلوم شدن شرط وآدرس مقصد واحد واکشی مجبور می‌شود تمامی دستورات بافر را دورریخته و مجدداً واکشی را شروع کند. در این حالت واحد اجرا مجبور است تا واکشی دستور صبر کند.
- ✓ رخ دادن وقفه هم تأثیر مشابهی خواهد داشت.

تاثیر طول پایپ لاین

هر چه طول پایپ‌لاین بیشتر شود کارآیی آن بیشتر می‌شود ولی این امر مسایل زیر را بدنبال خواهد آورد:

- ✓ انتقال داده بین قطعات باعث بروز تاخیر و در نتیجه طولانی شدن زمان اجرا میشود.
- ✓ هر چه واحد ها بیشتر می‌شوند واحد کنترل پیچیده‌تر می‌شود.

پایپ لاین

- برای اینکه پایپ‌لاین موثر باشد:
  - همه دستورات باید دارای طول یکسان باشند.
  - قالب دستورات محدود باشد
  - دسترسی به حافظه محدود به دستورات load و store باشد
  - عملوندها در حافظه به ترتیب قرارگیرند

- موارد زیر پایپ‌لاین را مشکل می‌کنند:
  - structural hazards: suppose we had only one memory
  - control hazards: need to worry about branch instructions
  - data hazards: an instruction depends on a previous instruction

مخاطره‌های پایپ‌لاین (Pipeline Hazards)

مشکلات اصلی در خط لوله که باعث کاهش کارآیی می‌شوند:

❖ وابستگی داده ها (Data Dependency)

یک دستور به نتیجه دستور قبلی نیاز دارد، که هنوز آماده نیست.

```
Load  R1, M[376]
Load  R2,M[378]
Multi R1,R2
```

❖ وابستگی مراجعه به حافظه (Resource Conflict)

دو دستور به طور همزمان می خواهند، به حافظه دسترسی داشته باشند، مثلاً دستور یک در بخش چهار است و دسترسی به حافظه می خواهد و همزمان دستور دوم (مثل LD) که در بخش سوم هست، می خواهد به حافظه دسترسی داشته باشد.

❖ مشکلات مربوط به پرش (Branch Difficulties)

ما دستور بعد را وارد کرده ایم ولی وقتی به دستور پرش می رسیم متوجه میشویم باید دستور دیگری را اجرا می کردیم و باید خط لوله خالی شود.

مخاطره‌های ساختاری (Structural Hazard)

وابستگی مراجعه به حافظه (Resource Conflict)

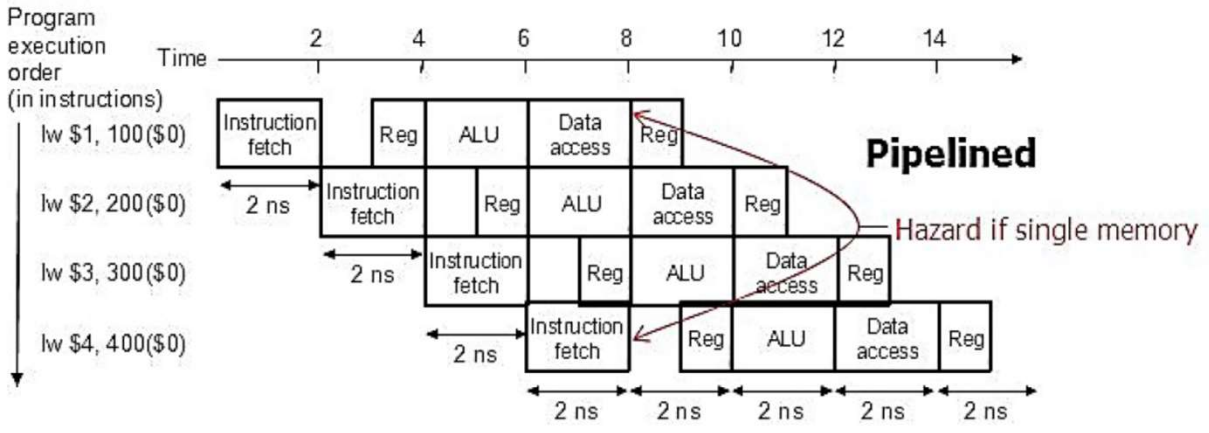
این نوع برخورد وقتی بوجود می آید که دو واحد به‌طور همزمان بخواهند به حافظه دسترسی پیدا کنند.

راه حل آن، اضافه کردن یک حافظه مجزا برای داده در کنار حافظه ای برای دستورات کاری پر هزینه است.

در بعضی مواقع این برخورد با اختصاص دو حافظه جداگانه (دو cache) برای دستور و داده حل می‌شود، در غیر این‌صورت اجرای دستورات تا آزاد شدن منابع به تأخیر خواهد افتاد.این کار به مدار کنترل پیچیده ای نیاز دارد.

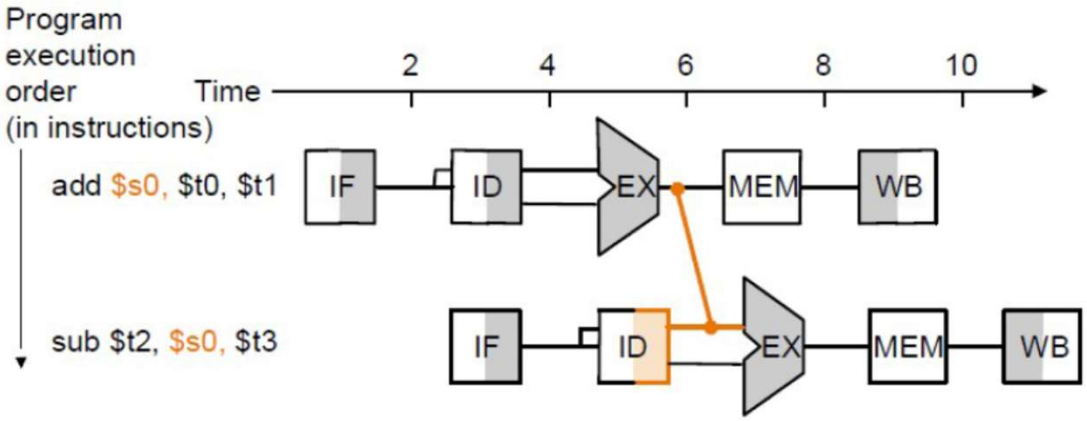
مخاطره‌های ساختاری (Structural Hazard)

وابستگی مراجعه به حافظه (Resource Conflict)



مخاطره‌های داده‌ای (Data Hazard)

وقتی به‌وجود می آید که اجرای یک دستور به نتیجه دستور قبلی که هنوز آماده نشده است بستگی داشته باشد.



مخاطره‌های داده‌ای (Data Hazard)

مثال: تمام دستورات بعد از ADD به نتیجه آن نیاز دارند.

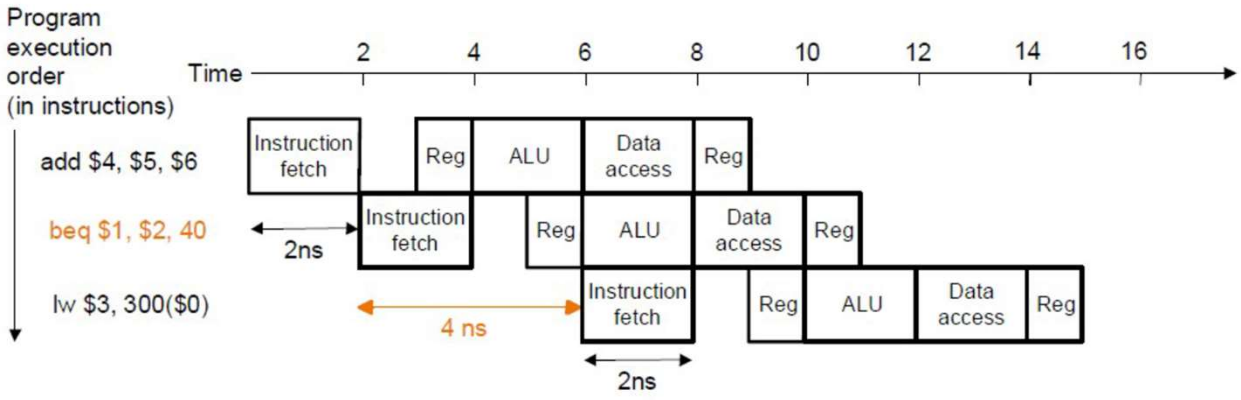
		1	2	3	4	5	6	7	8	9
ADD	**R1** ,R2, R3	IF	ID	EX	MEM	**WB**				
SUB	R4, R5 ,**R1**		IF	**ID_{sub}**	EX	MEM	WB			
AND	R6 ,**R1** ,R7			IF	**ID_{and}**	EX	MEM	WB		
OR	R8 ,**R1** ,R9				IF	**ID_{or}**	EX	MEM	WB	
XOR	R10,**R1**,R11					IF	**ID_{xor}**	EX	MEM	WB

IF: Instruction fetch
ID: Instruction decode/register read
EX: Execute
MEM: Memory access
WB: Write Back results in register

Data Hazard

مخاطره‌های کنترلی (Control Hazard)

مخاطره کنترلی یعنی وجود دستورات انشعاب شرطی در پایپ‌لاین.



مخاطره‌های کنترلی (Control Hazard)

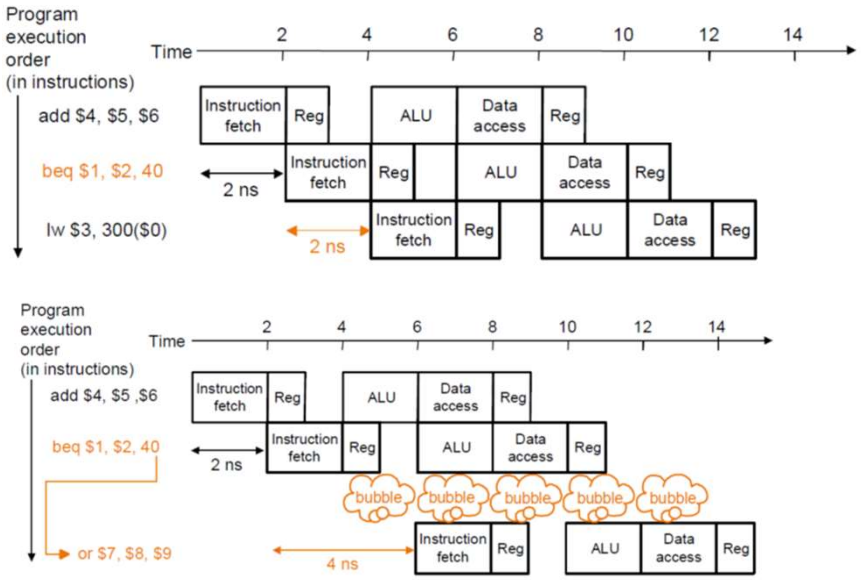
روش های رفع مخاطره کنترلی

۱. واکشی دستورِ بعد از پرش و مقصد پرش تا مشخص شود که بالأخره کدام اجرا شود.

۲. پیش بینی انشعاب:. برای دستورات پرش شرطی می توان یک سابقه از دستورات پرشی که تاکنون اجرا شده اند نگهداری کرد و بر اساس آن پیش بینی کرد.

۳. انشعاب با تأخیر: کامپایلر با جابجایی دستورات، بتواند دستورات مفیدی که حتما باید اجرا شوند را به جای NOPها قرار دهد

مخاطره‌های کنترلی (Control Hazard)



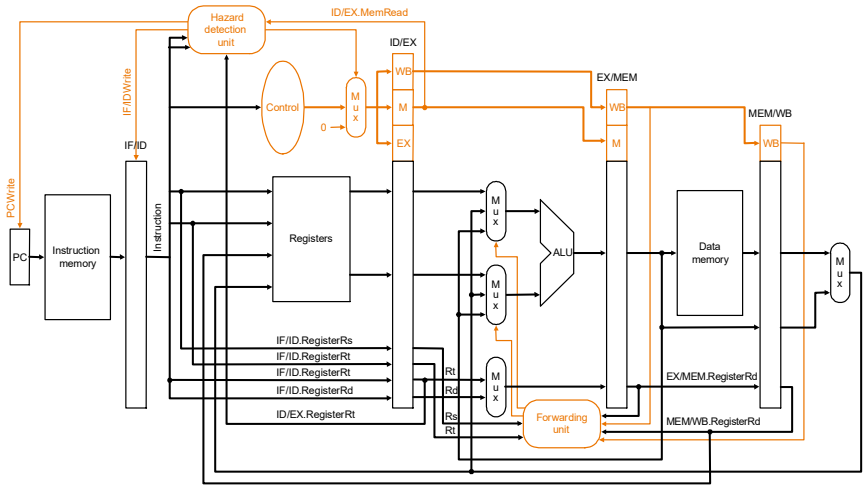
متوقف کردن پایپ‌لاین (Pipeline Stall)

در صورتیکه راه حلی اندیشیده نشده باشد شرایط فوق باعث توقف پایپ لاین خواهند شد.

سخت افزاری به نام Hardware interlock وظیفه تشخیص برخورد را بر عهده داشته و با ایجاد تاخیر لازم در اجرای دستورات بعدی مشکل را حل میکند.

		1	2	3	4	5	6	7	8	9
ADD	R1 ,R2, R3	IF	ID	EX	MEM	WB				
SUB	R4, R5 ,R1		IF	stall	stall	ID_{sub}	EX	MEM	WB	
AND	R6 ,R1 ,R7			stall	stall	IF	ID_{and}	EX	MEM	WB

واحد تشخیص (Hazard Detection Unit)



Stall by letting an instruction that won't write anything go forward

دل گر چه درین بادیه بسیار شتافت

یک موی ندانست و بسی موی شکافت

گرچه ز دلم هزار خورشید بتافت

آخر به کمال ذره‌ای راه نیافت

ابوسعید ابوالخیر

برای کسب اطلاعات بیشتر در مورد این درس می‌توانید به وب سایت
آموزشی در لینک زیر مراجعه نمایید

http://shafieian-education.ir

43