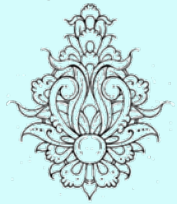
معماری کامپیوتر (۱۳۹۱-۱۱-۱۳)

جلسه‌ی هفدهم

دانشگاه شهید بهشتی
دانشکده‌ی مهندسی برق و کامپیوتر
بهار ۱۳۹۱
احمد محمودی ازناوه

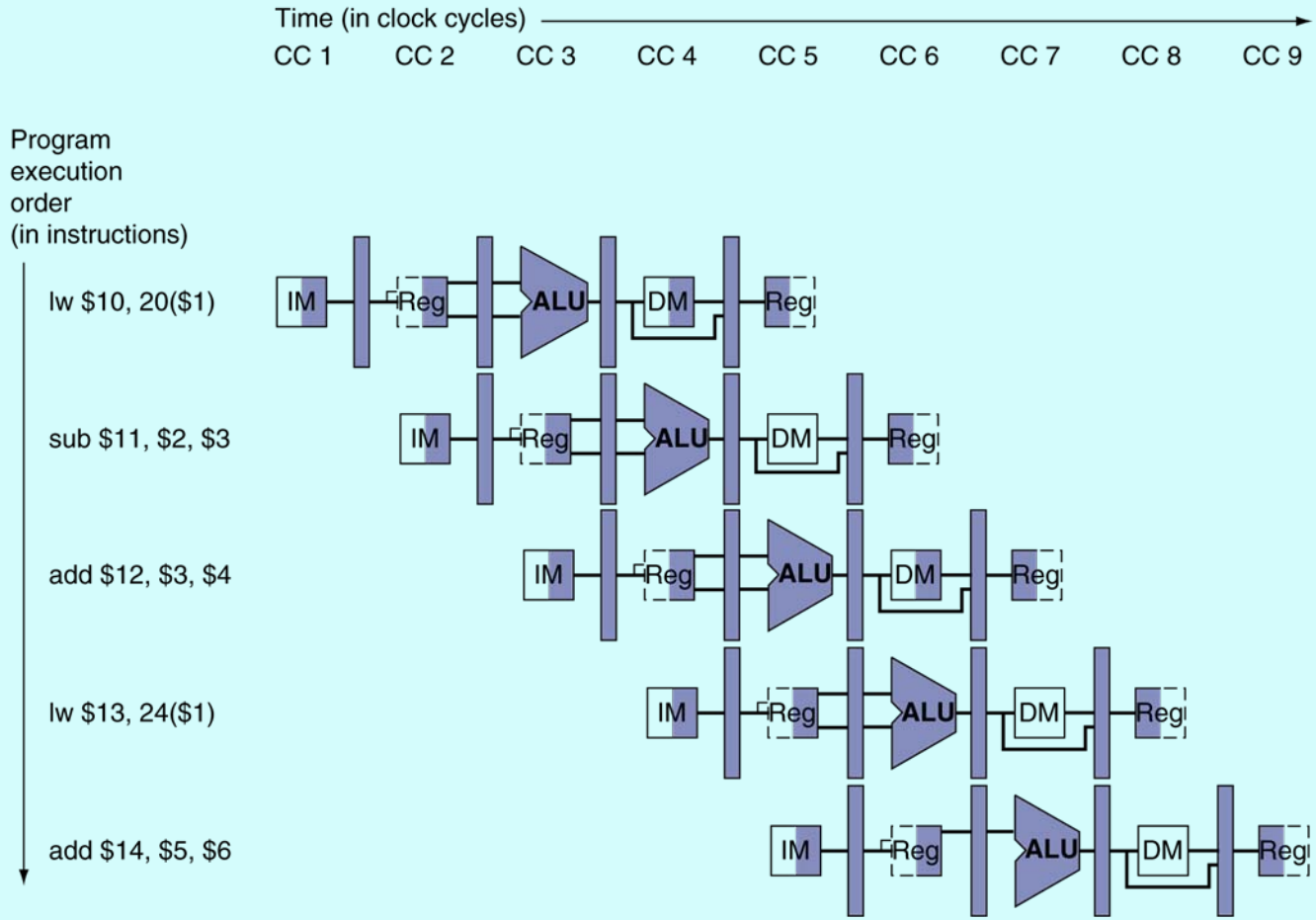
فهرست مطالب

- واحد کنترل در خط لوله
  - پیش‌فرستادن (هدایت رو به جلو)
  - واحد تشخیص مخاطره



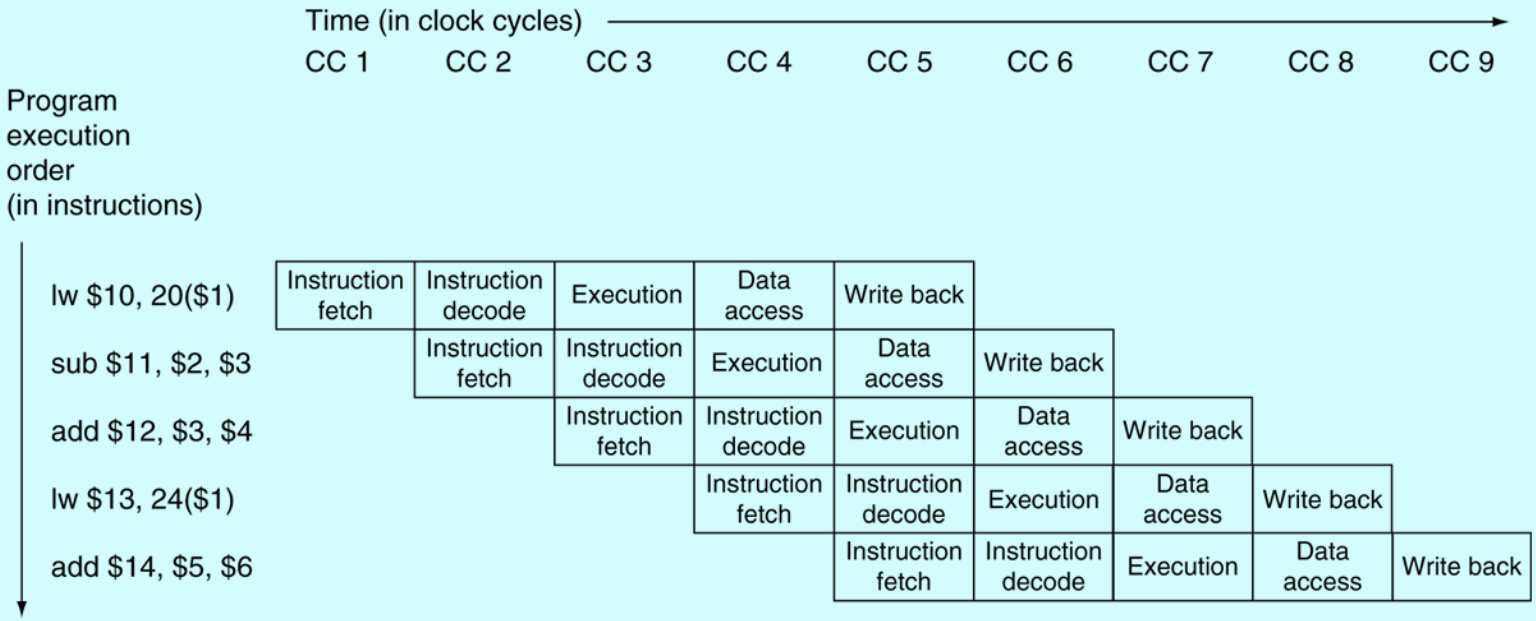
نمودار خط لوله به صورت چندسیکلی

- در این شیوه به کارگیری منابع نشان داده شده است



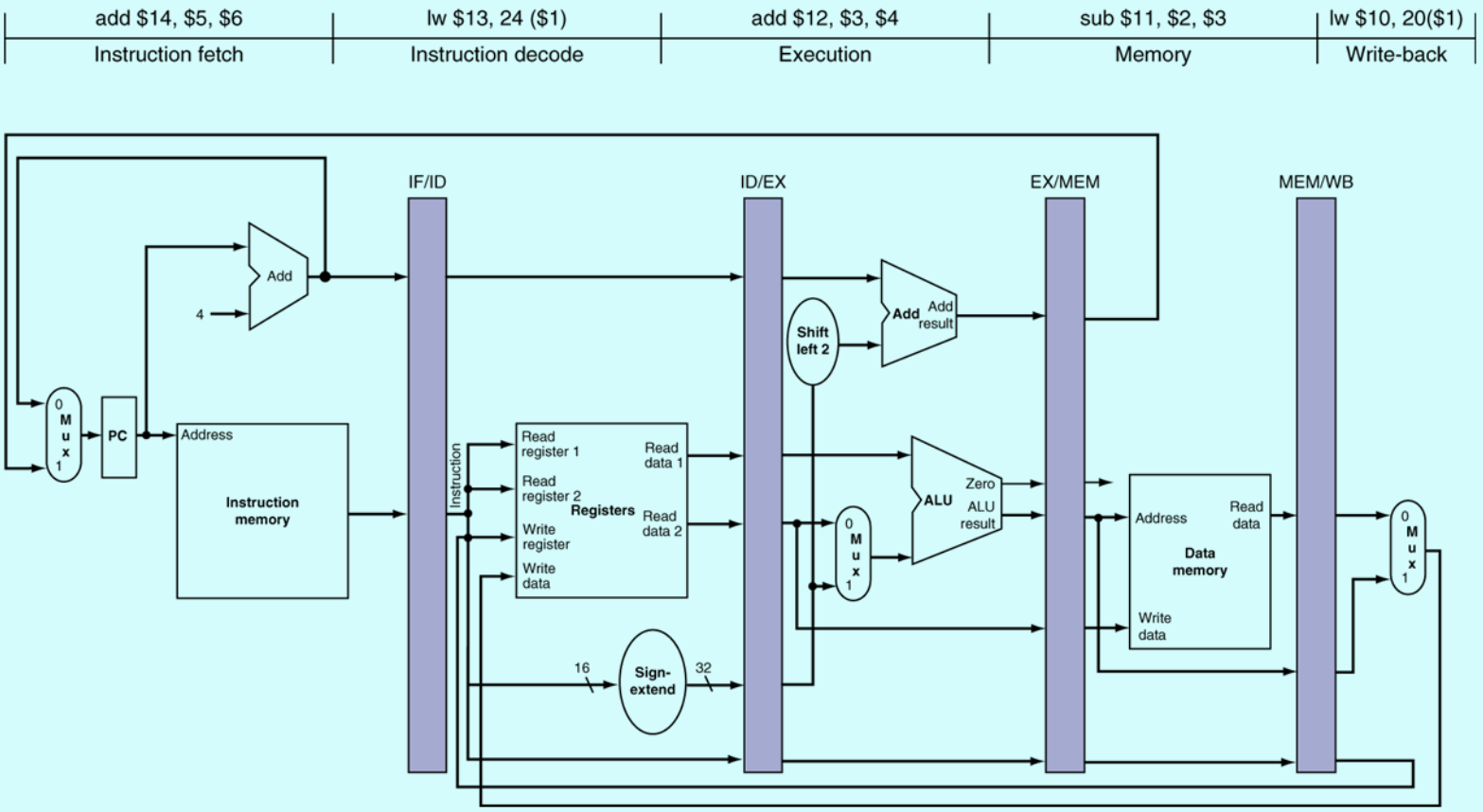
نمودار خط لوله به صورت چندسیکلی (ادامه...)

- **شیوه‌ی متعارف**

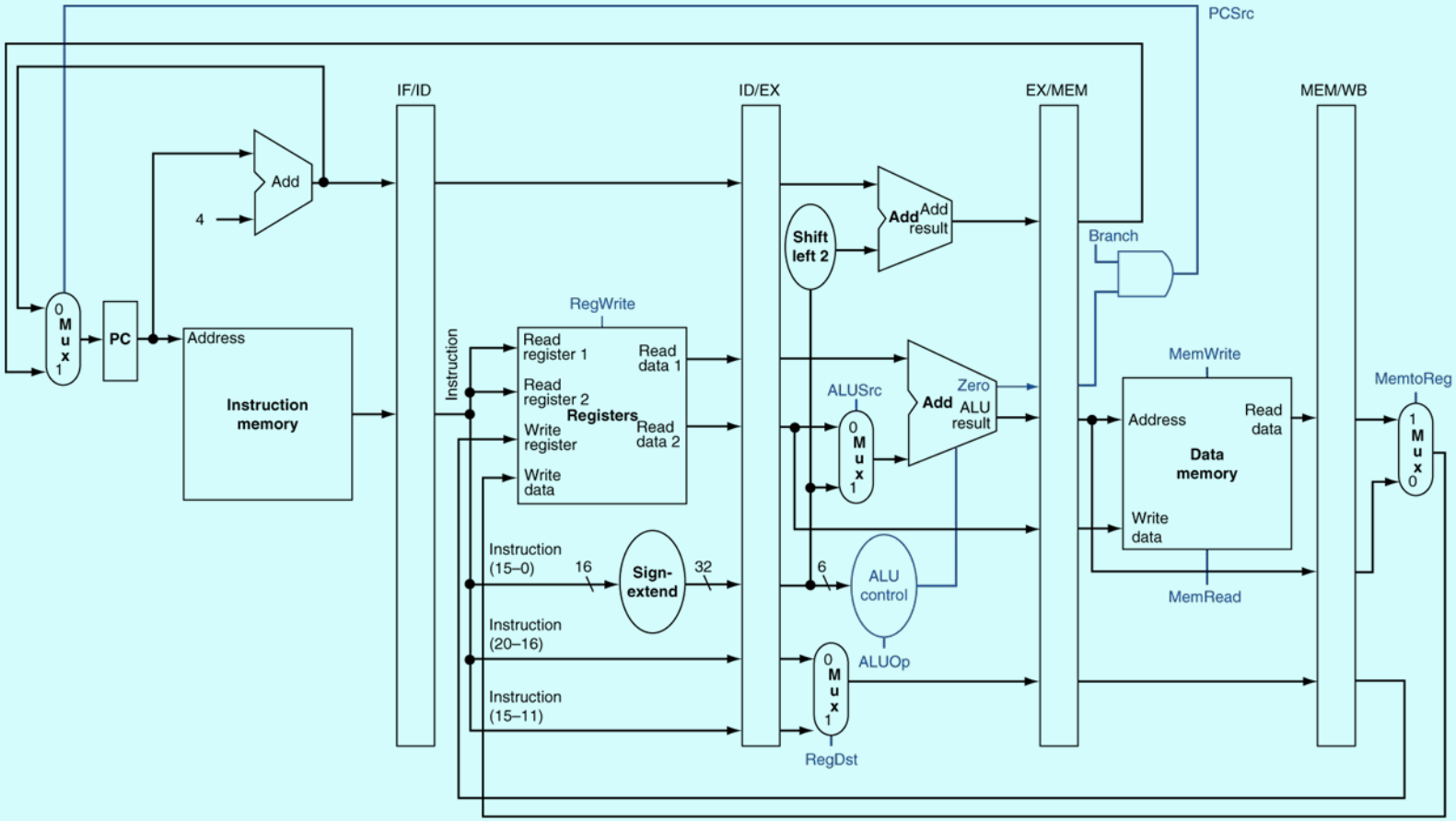


نمودار خط لوله به صورت تک‌سیکلی

- **حالت خط لوله در یک سیکل**

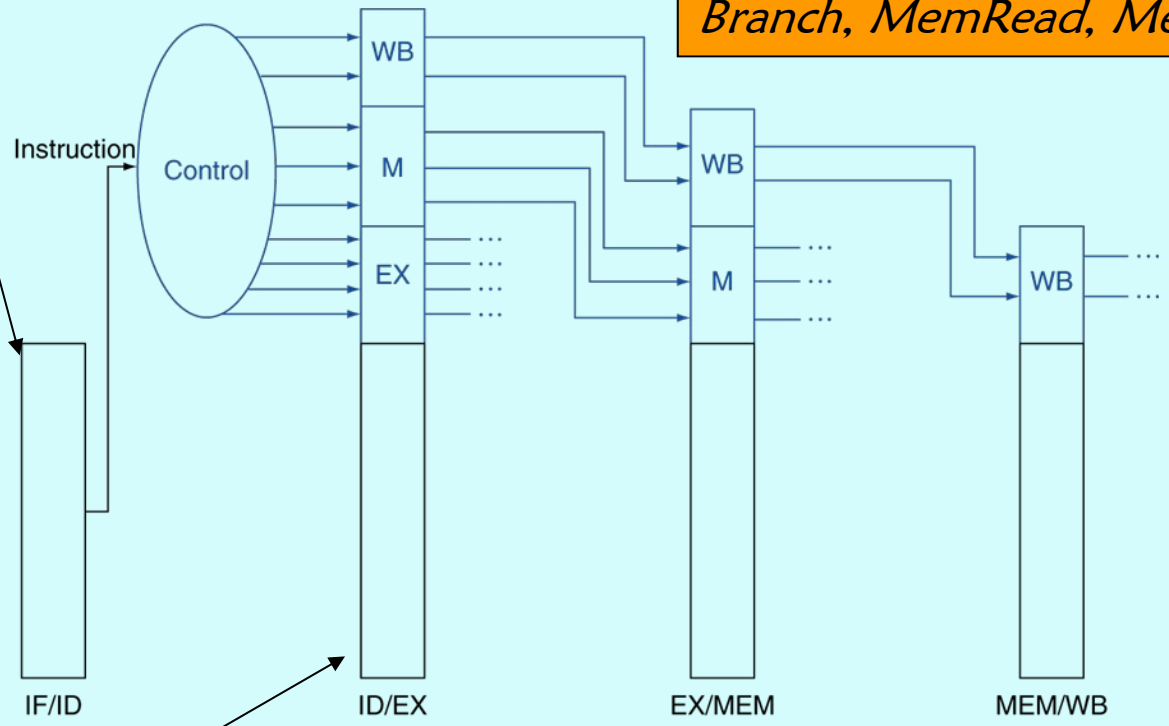


واحد کنترل خط لوله



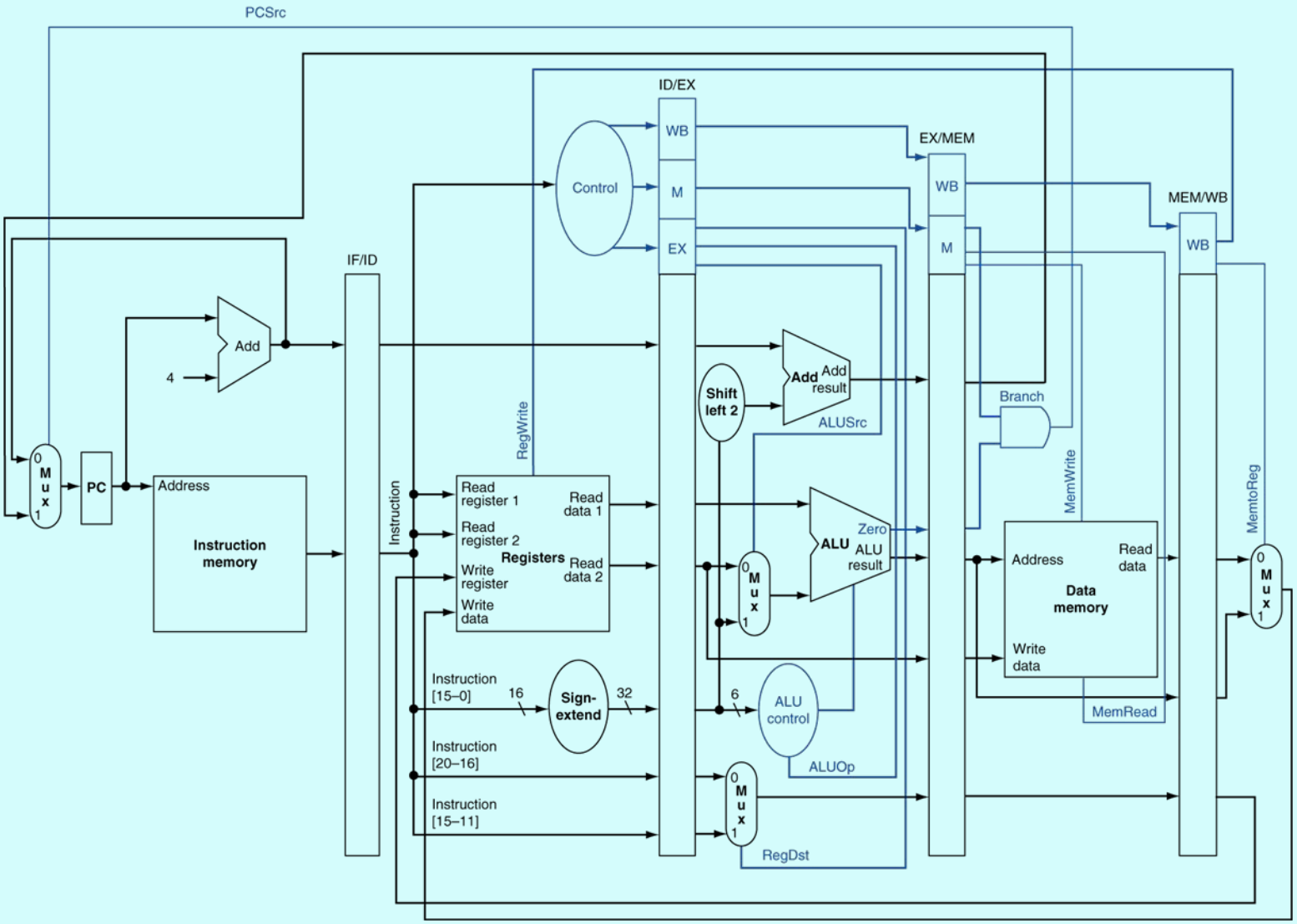
واحد کنترل خط لوله (ادامه...)

در این بخش می‌باید دستورالعمل بعدی واکشی شود، کاری که برای همه‌ی دستورها به یک شیوه خواهد بود



همانند گام پیشین، نیاز به سیگنال کنترلی خاصی نیست

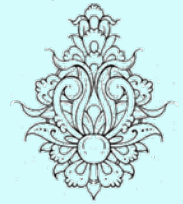
واحد کنترل خط لوله (ادامه...)



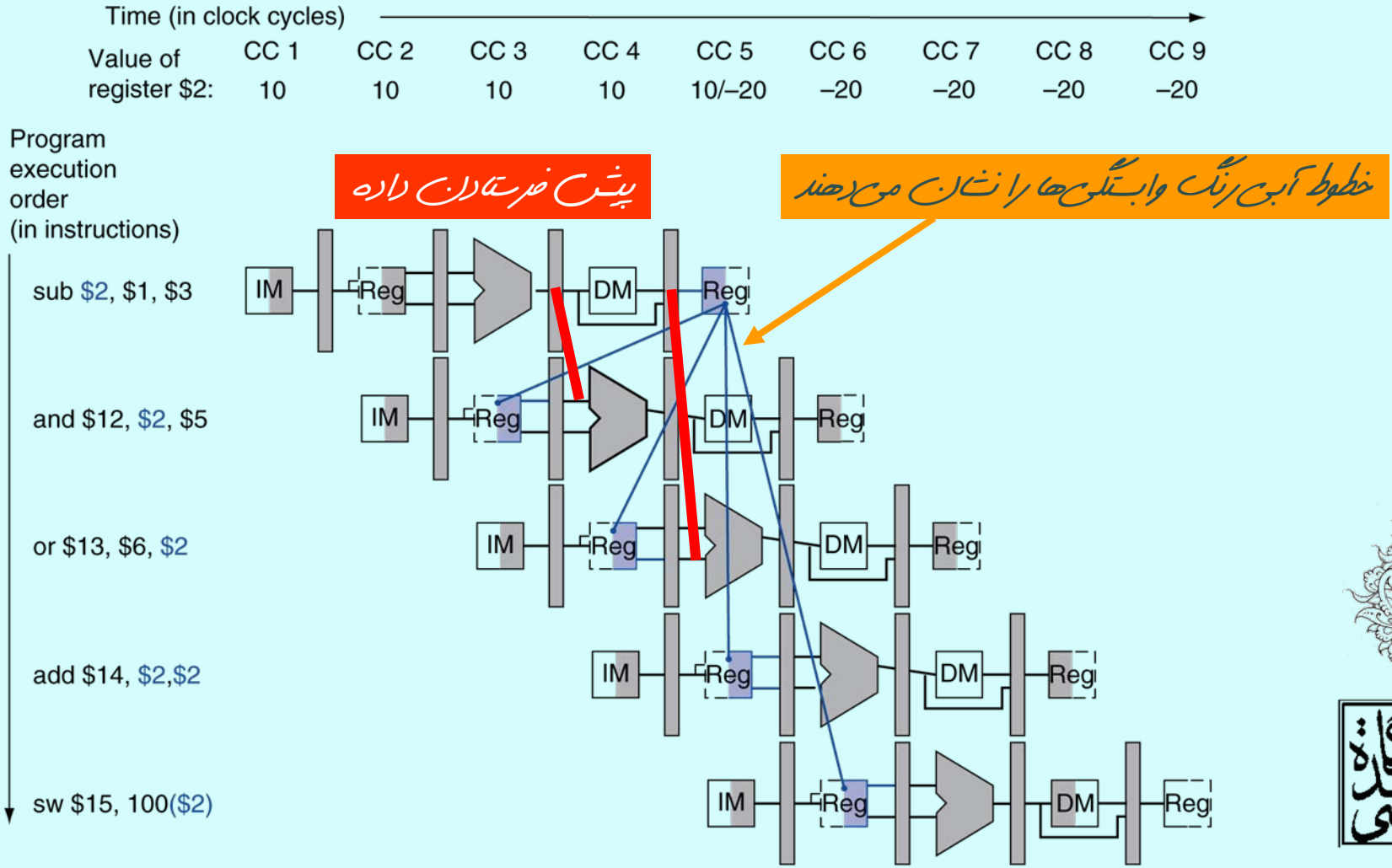
مخاطرات داده‌ای- مثال

```
sub $2, $1,$3
and $12,$2,$5
or  $13,$6,$2
add $14,$2,$2
sw  $15,100($2)
```

- در این قطعه برنامه، چهار دستورالعمل آخر به مقدار $2 وابسته هستند.
- چگونه می‌توان با پیش‌فرستادن مشکل وابستگی را حل کرد؟



مخاطرات داده‌ای (ادامه...)



پیش فرستادن

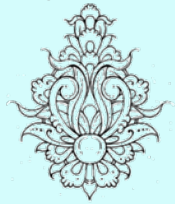
- چنان‌چه دیده شد، برای رهایی از مخاطرات داده، پیش‌فرستادن داده، راهکاری متداول است.
  - در ادامه خواهیم دید پیش‌فرستادن چگونه انجام می‌شود. برای سادگی تنها حالتی را بررسی خواهیم کرد، که داده در مرحله‌ی EX تولید می‌شود.

ID/EX.RegisterRs

شماره‌ی ثباتی را نشان می‌دهد که مقدار آن در ثبات ID/EX خط لوله قرار دارد.

- عملوندهای ALU در کدام ثبات قرار دارند؟

ID/EX.RegisterRs, ID/EX.RegisterRt



پیش فرستادن (ادامه...)

- **مخاطره‌ی داده در موارد زیر روی می‌دهد:**

1a. EX/MEM.RegisterRd = ID/EX.RegisterRs
1b. EX/MEM.RegisterRd = ID/EX.RegisterRt
2a. MEM/WB.RegisterRd = ID/EX.RegisterRs
2b. MEM/WB.RegisterRd = ID/EX.RegisterRt

Fwd from EX/MEM pipeline reg

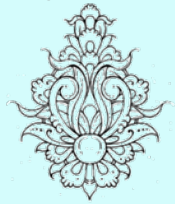
Fwd from MEM/WB pipeline reg

نوع یک

نوع دو

بدون مخاطره

```
sub $2,  $1,$3
and $12, $2, $5
or  $13, $6, $2
add $14, $2, $2
sw  $15, 100($2)
```



تشخیص نیاز به پیش فرستادن

- بنابراین، می‌توان با مقایسه‌ی محتوای ثبات‌ها، مداری برای کنترل پیش‌فرستادن داده طراحی کرد.

```
sub $2, $1,$3
and $12,$2,$5
```

EX/MEM.RegisterRd = ID/EX.RegisterRs=$2

ولی

در همه‌ی دستورالعمل‌های مقدار خروجی ALU، در ثبات نوشته نمی‌شود بدین ترتیب این راهکار در همه‌ی موارد درست نخواهد بود.

برای پیش‌گیری از این مسأله می‌توان از سیگنال‌های کنترلی مربوط به WB ذخیره شده در ثبات‌های خط لوله استفاده نمود.

EX/MEM.RegWrite, MEM/WB.RegWrite

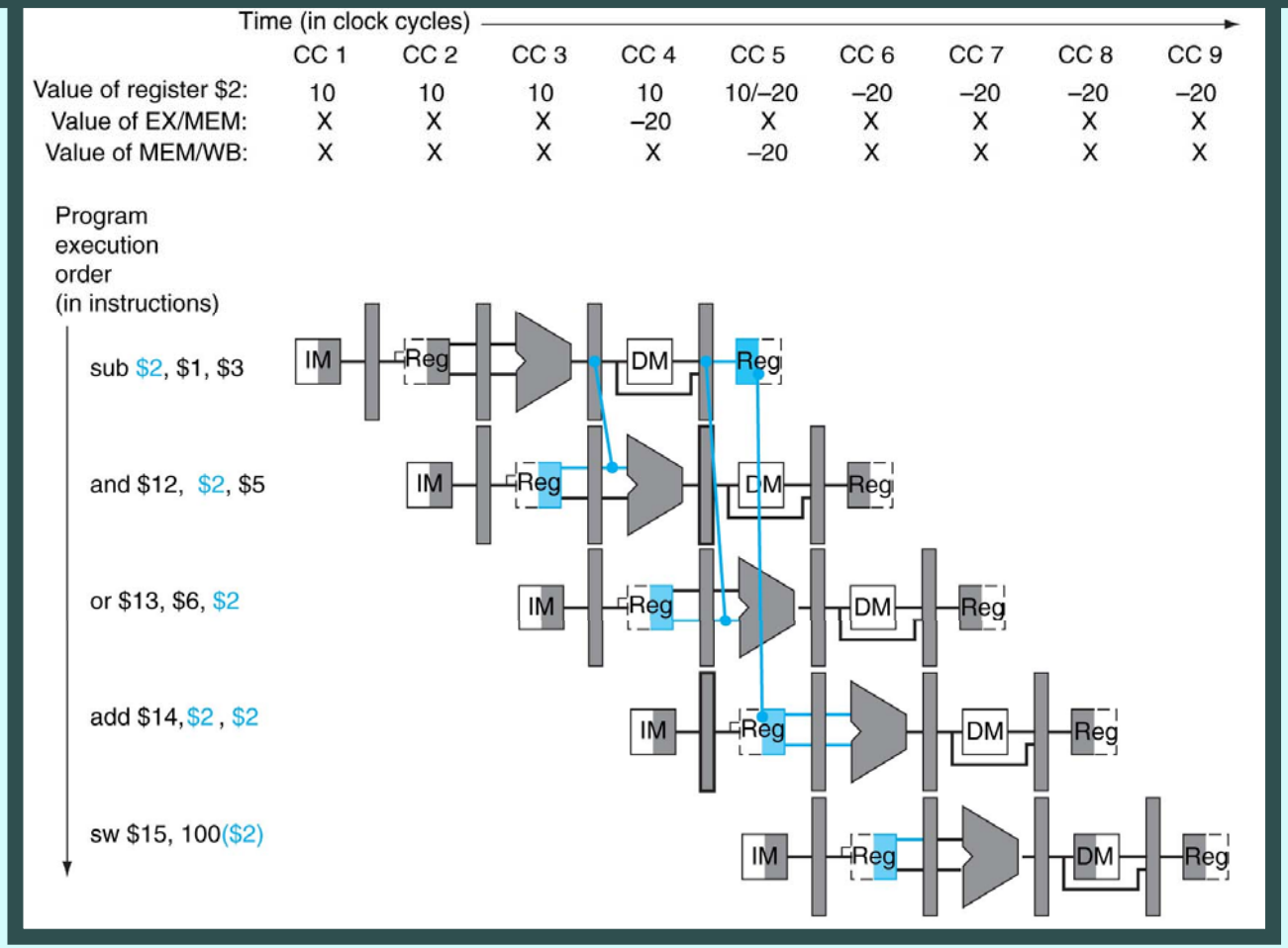
همچنین در صورتی که ثبات شماره‌ی صفر به عنوان مقصد یک دستور استفاده شده باشد، باید از پیش فرستادن جلوگیری کرد.

EX/MEM.RegisterRd ≠ 0,
MEM/WB.RegisterRd ≠ 0

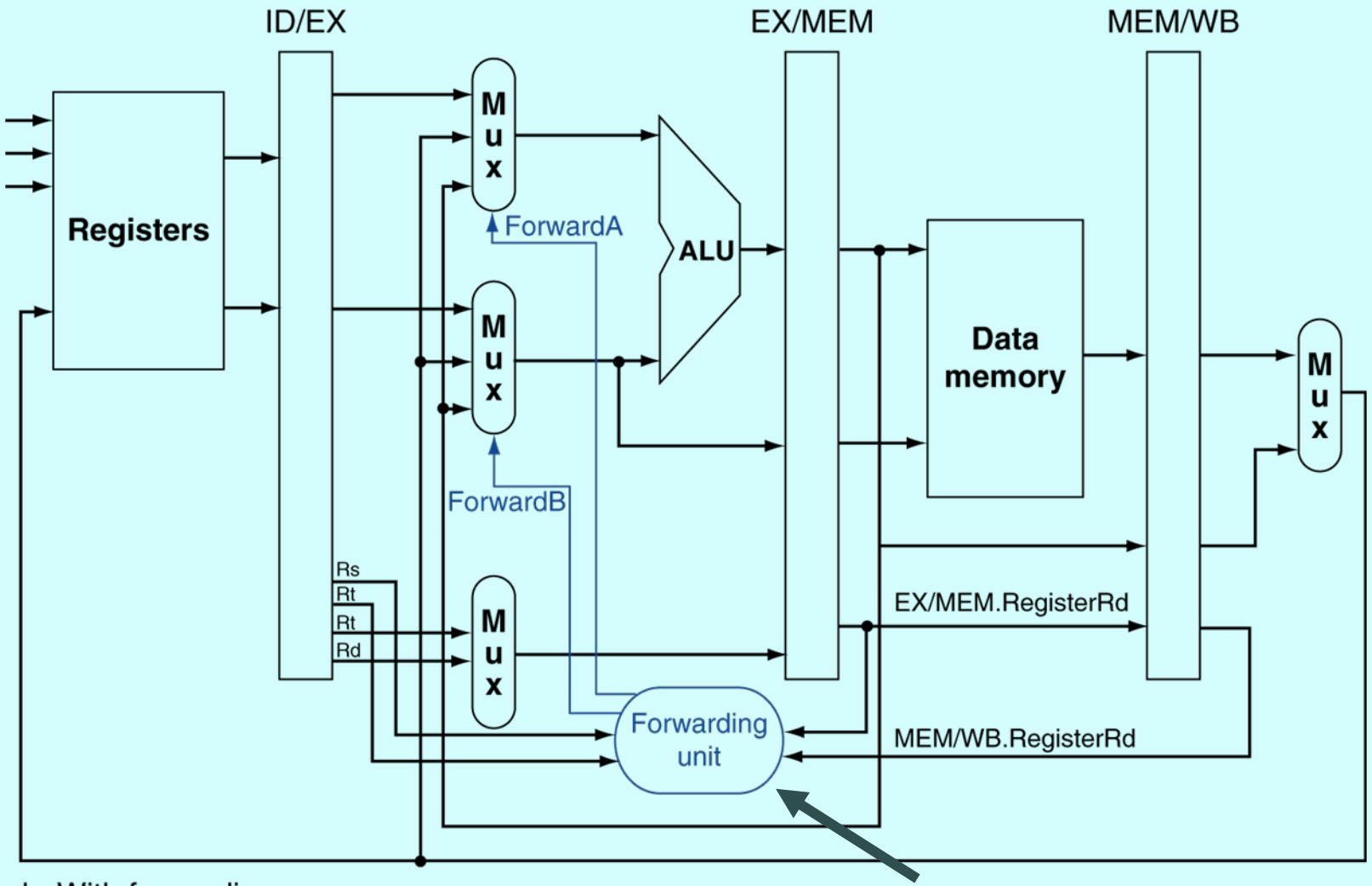
مسیر پیش فرستادن داده

با تشخیص زمان‌هایی که پیش‌فرستادن لازم است، نیمی از مشکلات حل شد،
هنوز نیمه‌ی دیگر باقی مانده است

:D

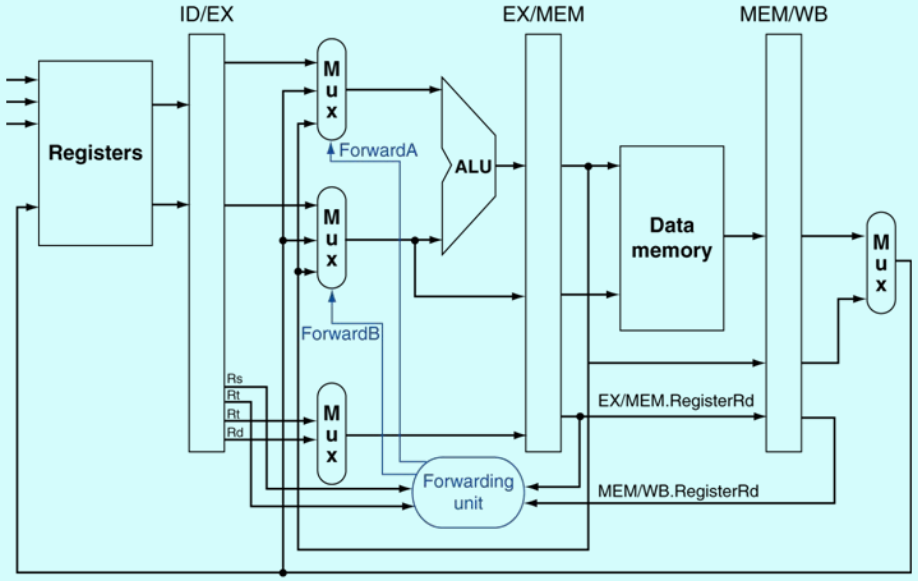


مسیر پیش فرستادن داده (ادامه...)



کنترل پیش فرستادن در مرحله‌ی EX قرار دارد، چرا که
خانه‌ی بافر پیش فرستادن در این مرحله قرار دارد.

مسیر پیش فرستادن داده (ادامه...)



b. With forwarding

Mux control	Source	Explanation
ForwardA = 00	ID/EX	The first ALU operand comes from the register file.
ForwardA = 10	EX/MEM	The first ALU operand is forwarded from the prior ALU result.
ForwardA = 01	MEM/WB	The first ALU operand is forwarded from data memory or an earlier ALU result.
ForwardB = 00	ID/EX	The second ALU operand comes from the register file.
ForwardB = 10	EX/MEM	The second ALU operand is forwarded from the prior ALU result.
ForwardB = 01	MEM/WB	The second ALU operand is forwarded from data memory or an earlier ALU result.

شرایط پیش‌فرستادن

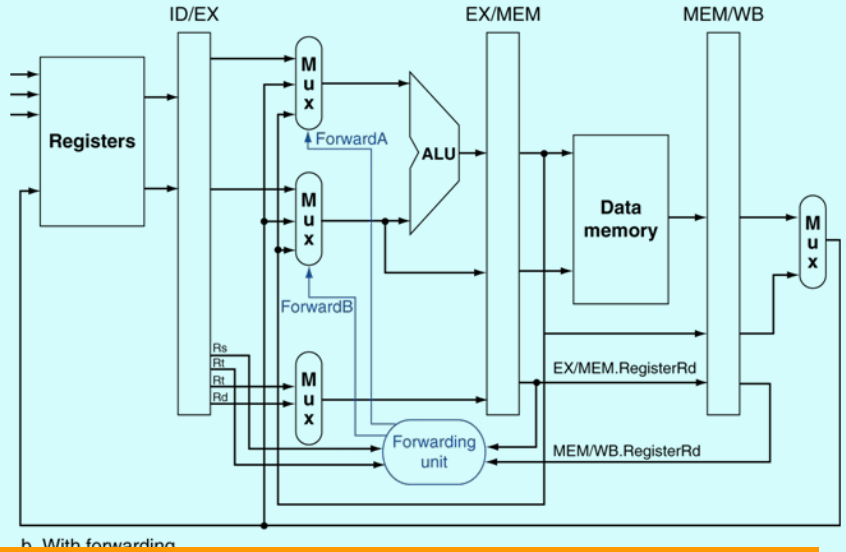
- EX hazard
  - if (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
    and (EX/MEM.RegisterRd = ID/EX.RegisterRs))
    ForwardA = 10
  - if (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
    and (EX/MEM.RegisterRd = ID/EX.RegisterRt))
    ForwardB = 10
- MEM hazard
  - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
    and (MEM/WB.RegisterRd = ID/EX.RegisterRs))
    ForwardA = 01
  - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
    and (MEM/WB.RegisterRd = ID/EX.RegisterRt))
    ForwardB = 01

اگر هر دوی این شرایط با هم رخ داد، چه اشکالی ایجاد می‌شود؟

شرایط پیش‌فرستادن (ادامه...)

- در این قطعه برنامه هر دو نوع مخاطره رخ می‌دهد.

```
add $1,$1,$2
add $1,$1,$3
add $1,$1,$4
```

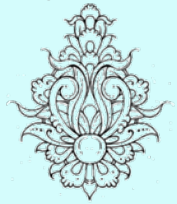


در این حالت آخرین نتیجه باید فرستاده شود،
در نتیجه داده‌ی موجود در مرحله‌ی MEM فرستاده می‌شود.

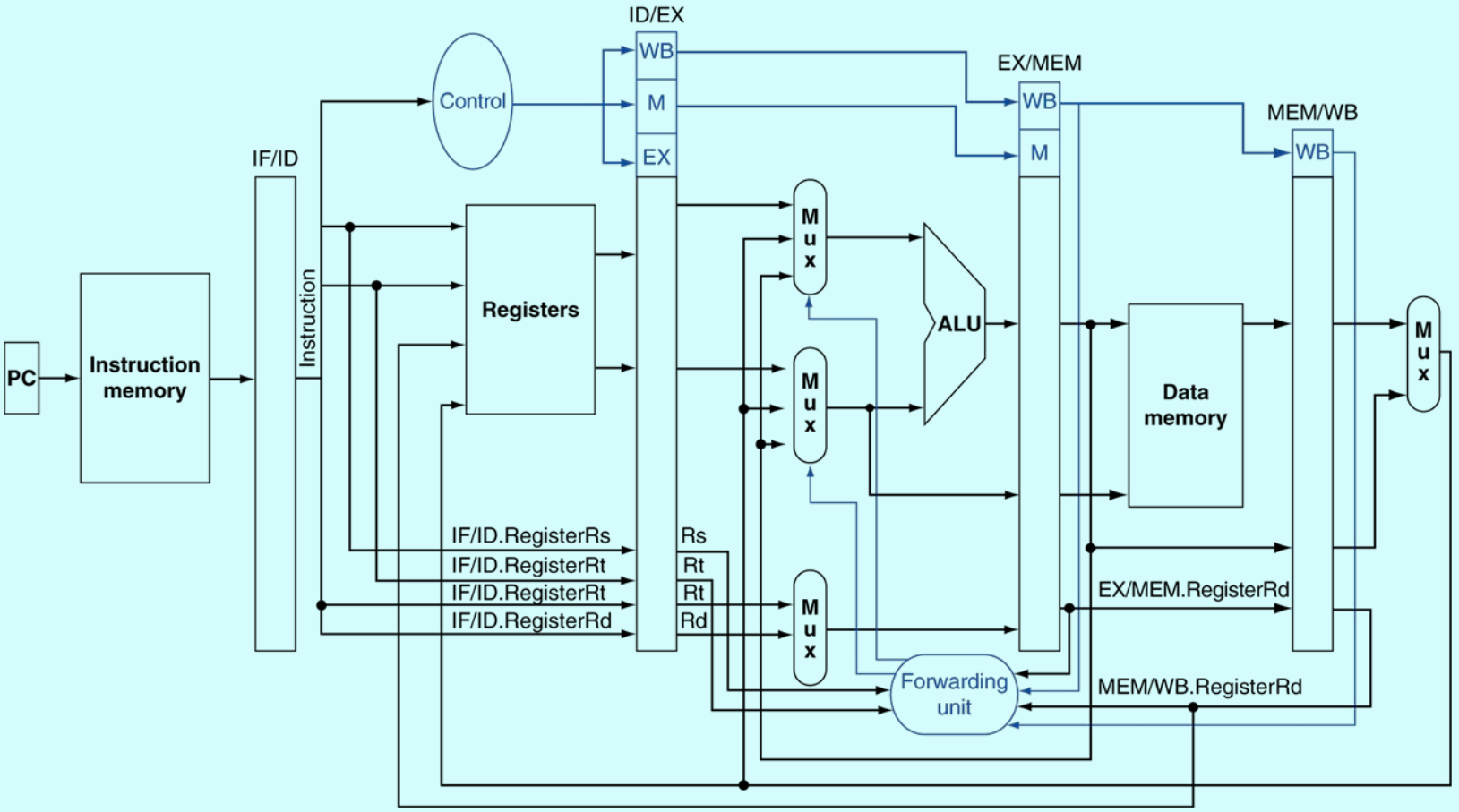
- بنابراین باید تغییراتی در مخاطره‌ی MEM بدهیم

شرایط پیش‌فرستادن (ادامه...)

- MEM hazard
  - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
    and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
    and (EX/MEM.RegisterRd = ID/EX.RegisterRs))
    and (MEM/WB.RegisterRd = ID/EX.RegisterRs))
    ForwardA = 01
  - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
    and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
    and (EX/MEM.RegisterRd = ID/EX.RegisterRt))
    and (MEM/WB.RegisterRd = ID/EX.RegisterRt))
    ForwardB = 01

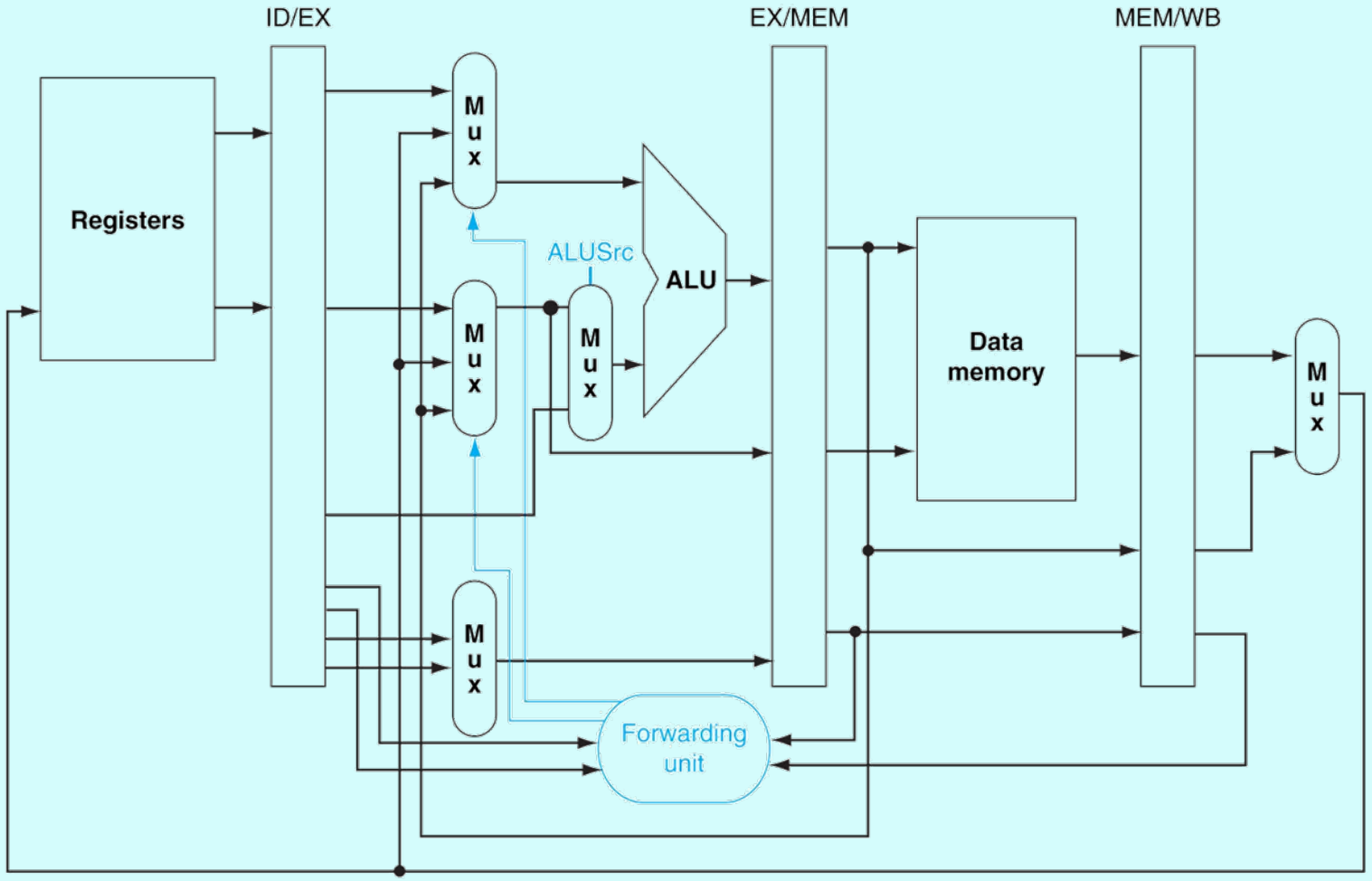


شرایط پیش‌فرستادن (ادامه...)

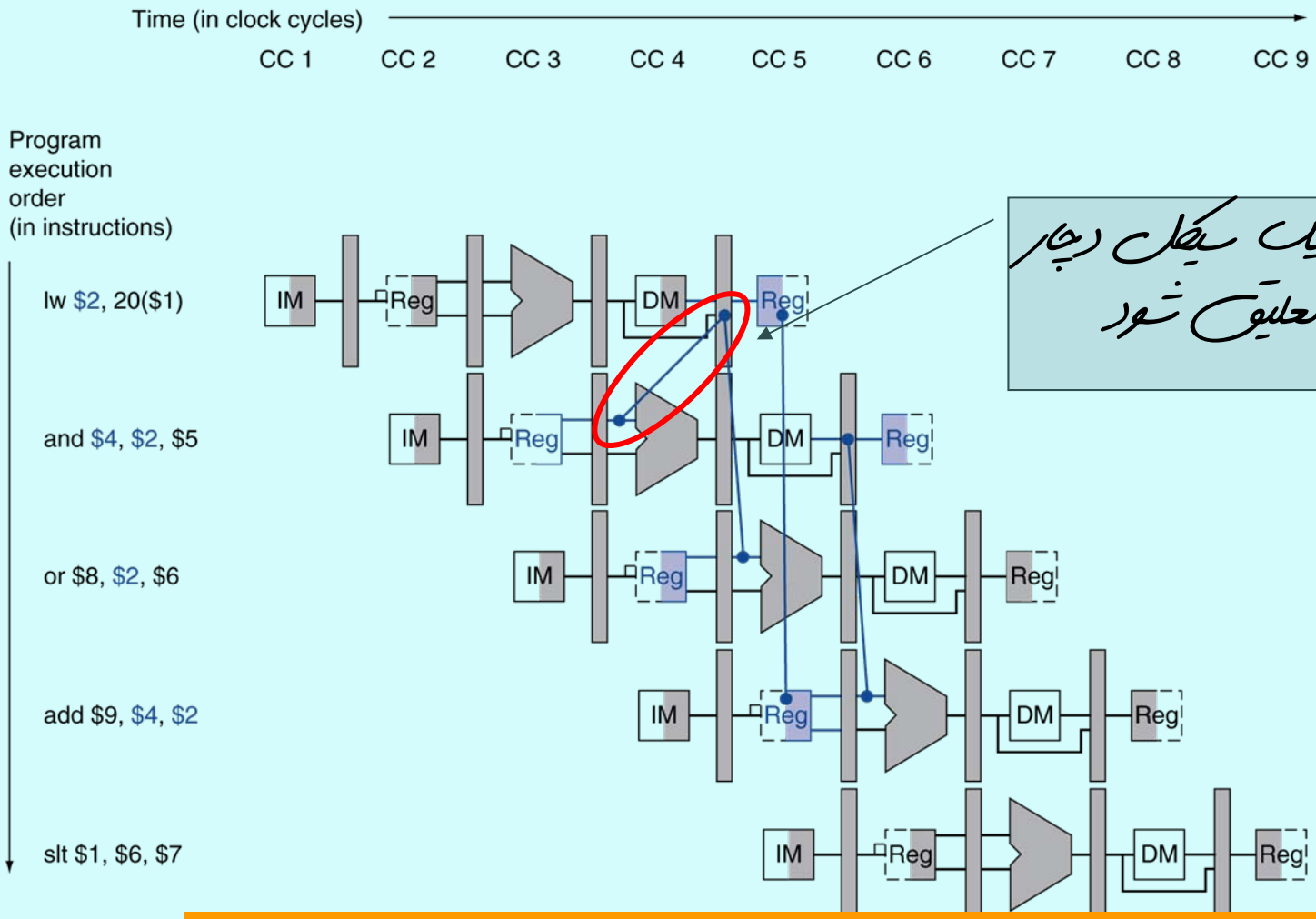


در این شکل بخشی که برای ارسال داده‌ی ثابت به ALU بود، حذف شده است

شرایط پیش‌فرستادن (ادامه...)



پیش فرستادن و تعلیق



بنابراین افزون بر واحد پیش‌فرستادن به واحد تشخیص مخاطره نیز نیاز داریم

واحد تشخیص مخاطره

به نظر شما این واحد در کدام مرحله قرار دارد؟

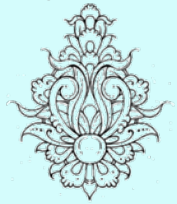
- در ID هنگامی که دستورالعمل کدگشایی می‌شود، وقوع مخاطره بررسی می‌شود.
- به عنوان مثال در استفاده از داده‌ی در حال بارگذاری

– IF/ID.RegisterRs, IF/ID.RegisterRt حاوی شماره‌ی ثبات‌های عملوند ALU می‌باشد.

اگر

ID/EX.MemRead and
((ID/EX.RegisterRt = IF/ID.RegisterRs) or
(ID/EX.RegisterRt = IF/ID.RegisterRt))

تعلیق: یک حباب وارد کن



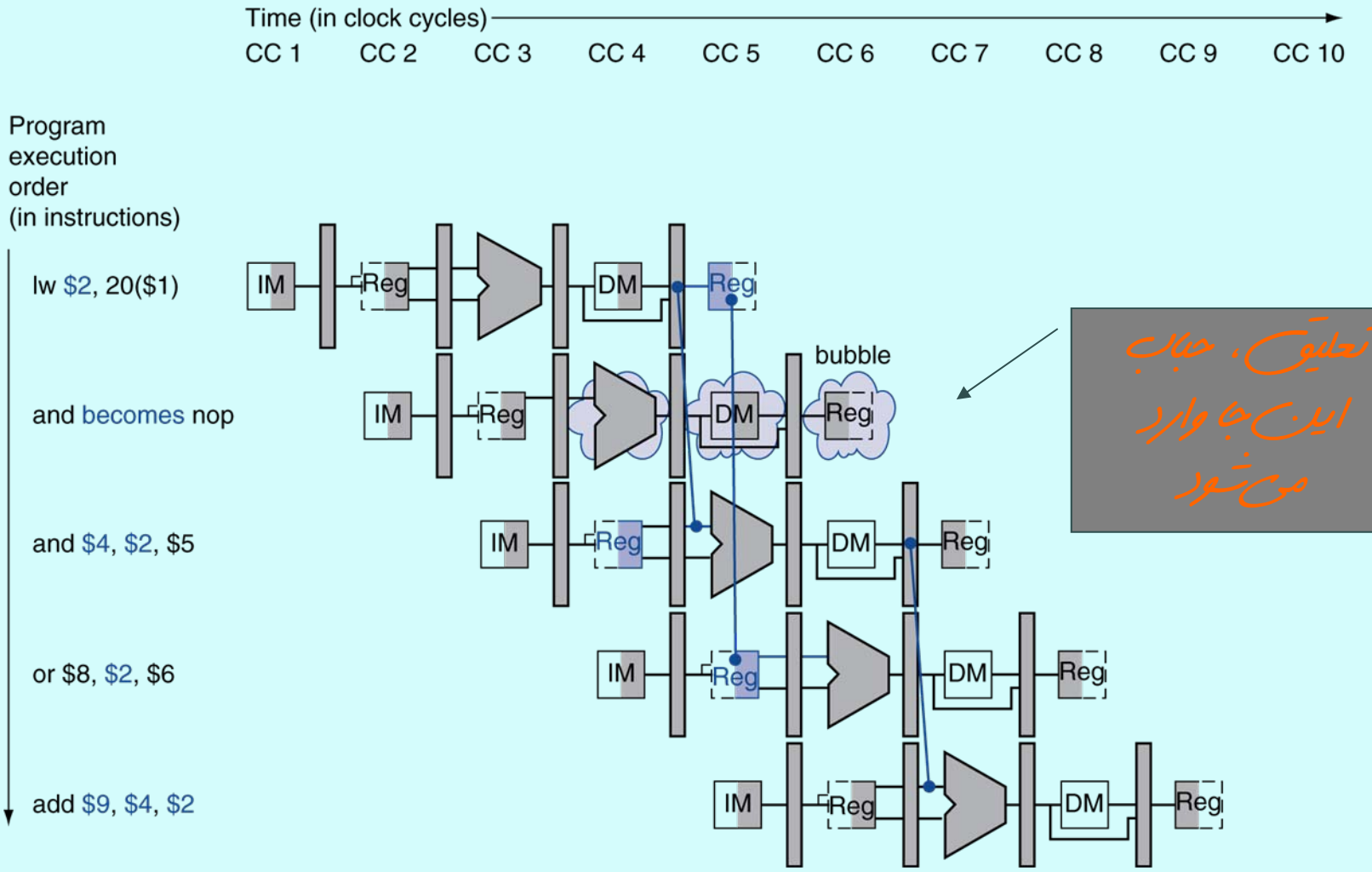
وارد کردن حباب

- در این حال دستور تهی (nop) وارد خط لوله می‌شود.
  - تمام خطوط کنترلی غیر فعال (برابر با ’0‘) می‌شود.
    - به جز سیگنال‌های نوشتن در حافظه، مقدار باقی سیگنال‌ها اهمیتی ندارد.
  - مقدار PC افزایش نمی‌یابد.
  - دستورالعمل دوباره واکشی می‌شود.
  - و دوباره کدگشایی می‌شود.

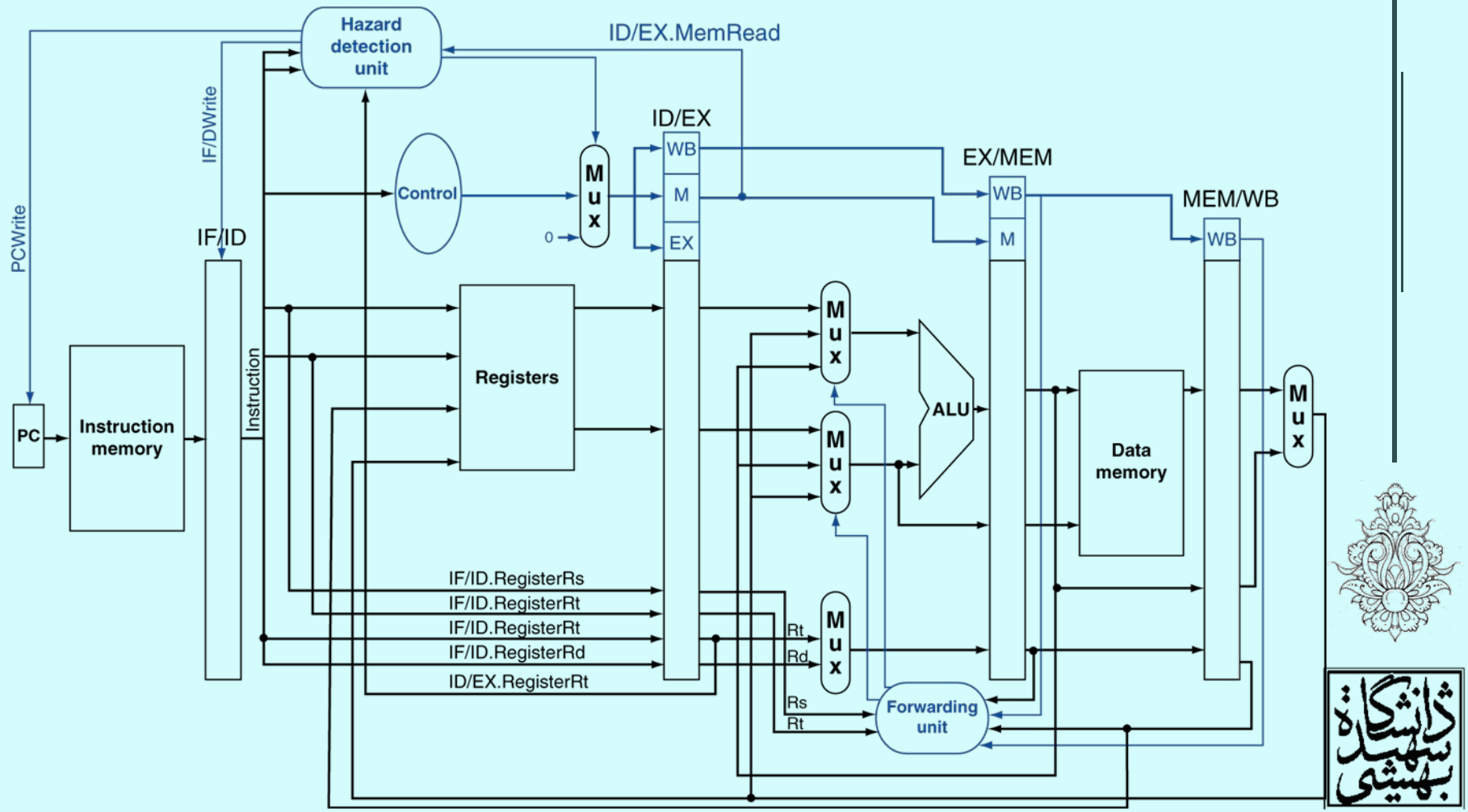
The BIG Picture

هر چند در تشخیص مخاطرات، سخت‌افزار نقش اصلی را ایفا می‌کند لازم است کامپایلر بر نحوه‌ی کار خط لوله مسلط باشد

وارد کردن حباب (ادامه...)

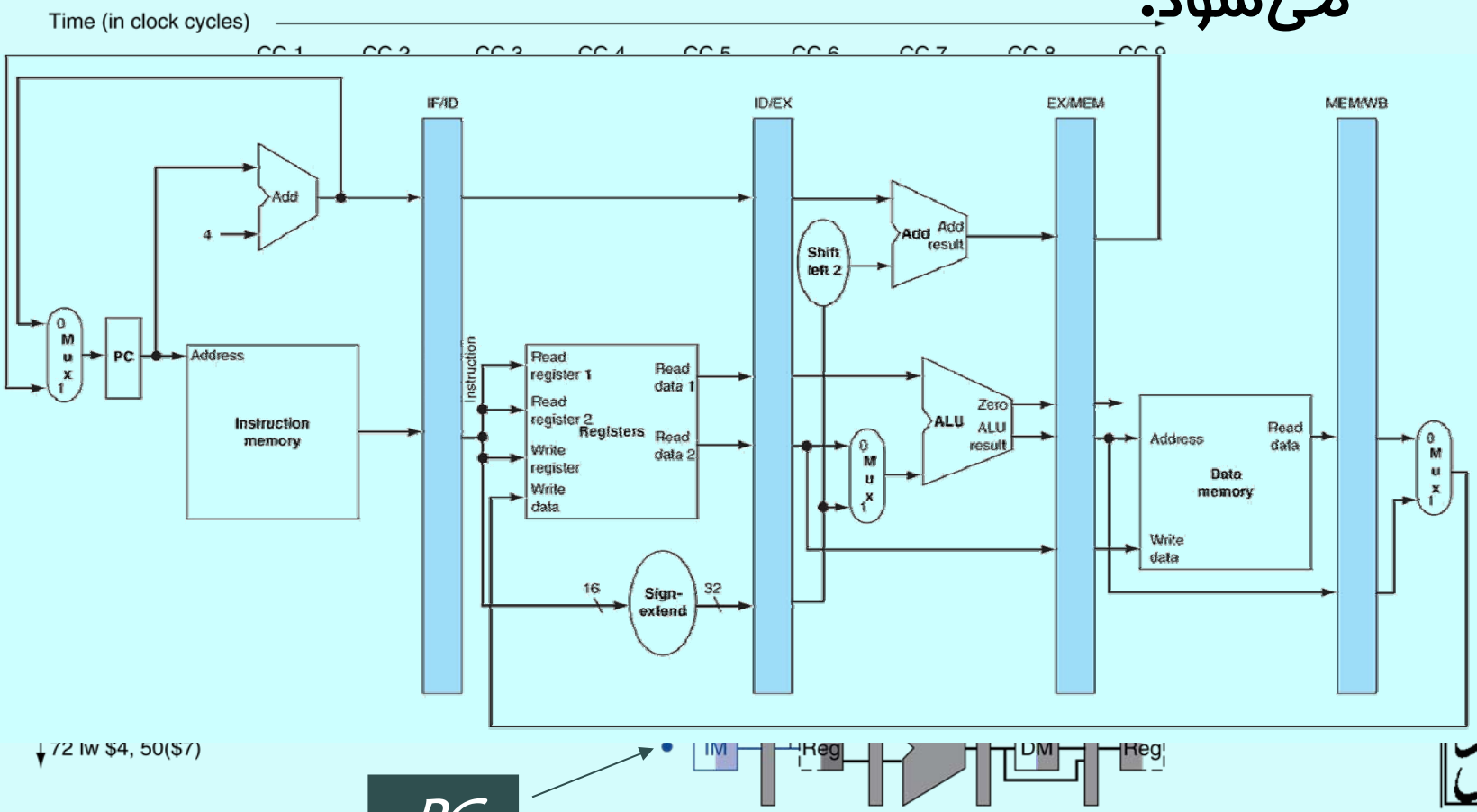


داده‌گذر همراه با مدار تشخیص مخاطره



مخاطرات کنترل

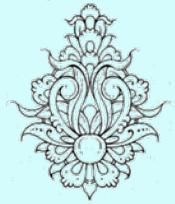
- نتیجه‌ی دستور پرش در مرحله‌ی MEM مشخص می‌شود.



مخاطرات کنترلی (ادامه...)

- ایجاد تعلیق، موجب کندی می‌شود.
- یک راه حل، این است که فرض کنیم هیچ پرشی انجام نمی‌شود.
  - در صورت تحقق، اجرای دستورات واکشی شده، ملغی می‌گردد.
  - برای این کار سیگنال‌های کنترلی غیرفعال می‌شوند.
  - دستورالعمل‌ها از ثبات خط لوله پاک می‌شوند.

flush



مخاطرات کنترلی (ادامه...)

- راه دیگر، کوتاه‌کردن مسیر انجام دستورالعمل‌های پرش شرطیست، (در مرحله‌ی ID) که شامل دو کار است.

Target address adder

  - محاسبه‌ی سریع آدرس محل پرش

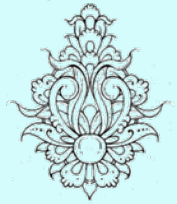
Register comparator

  - محاسبه‌ی سریع شرط

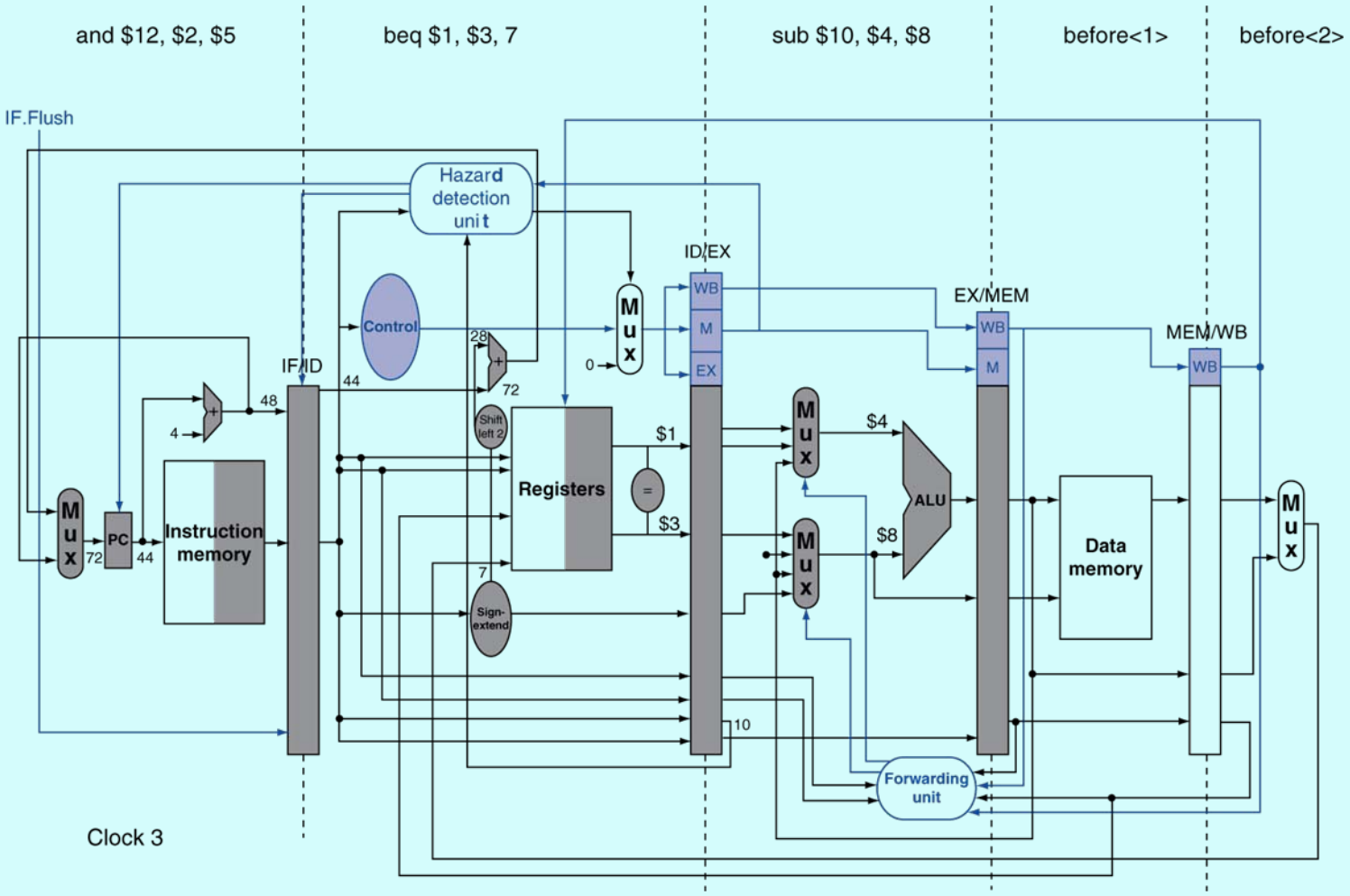
- در صورت انجام چنین کاری می‌باید تغییراتی در مدار تشخیص مخاطره و مدار ایجاد حباب به وجود آورد.

مثال

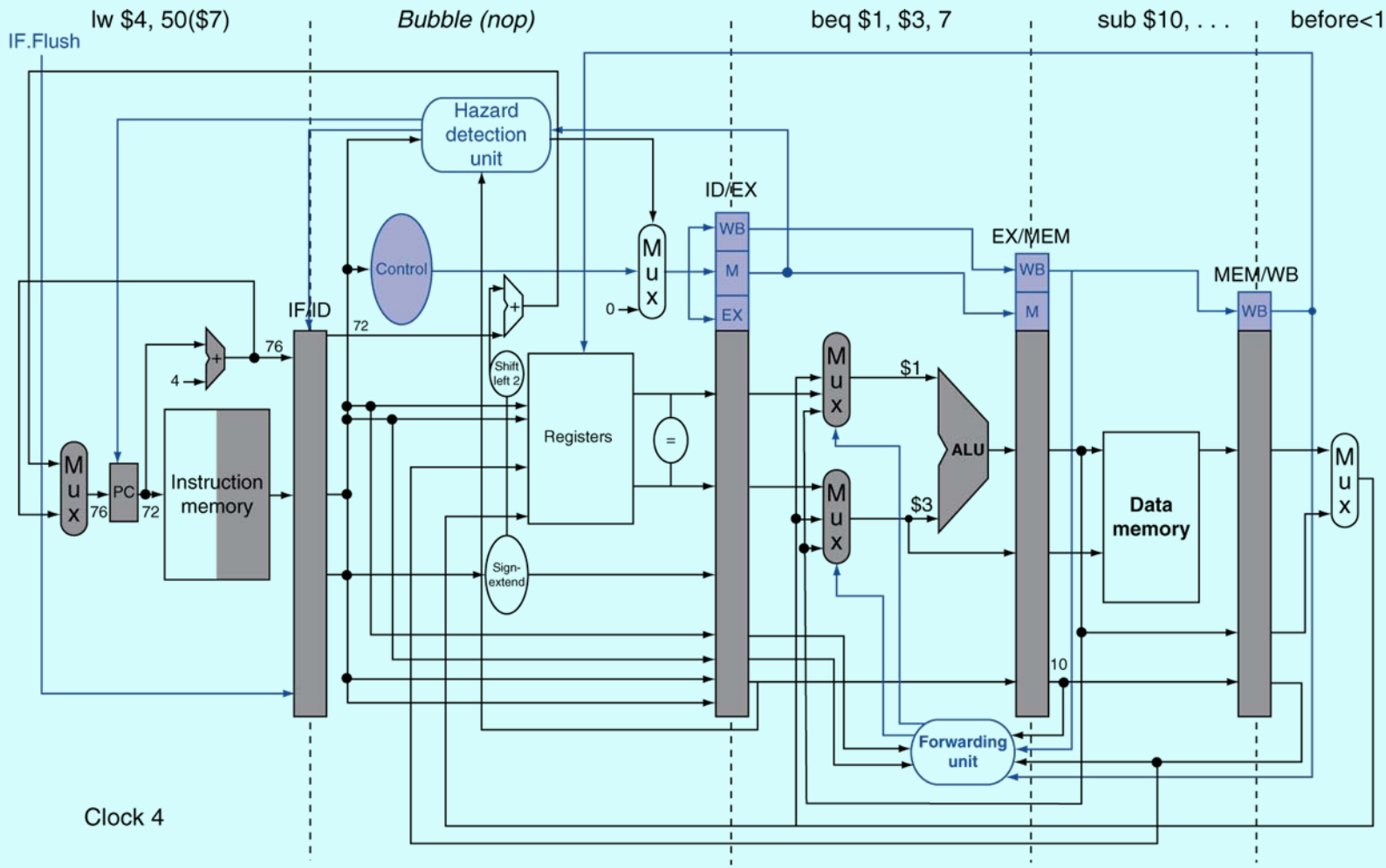
```
36:   sub   $10, $4, $8
40:   beq   $1,  $3, 7
44:   and   $12, $2, $5
48:   or    $13, $2, $6
52:   add   $14, $4, $2
56:   slt   $15, $6, $7
      ...
72:   lw    $4, 50($7)
```



در صورت تحقق شرط

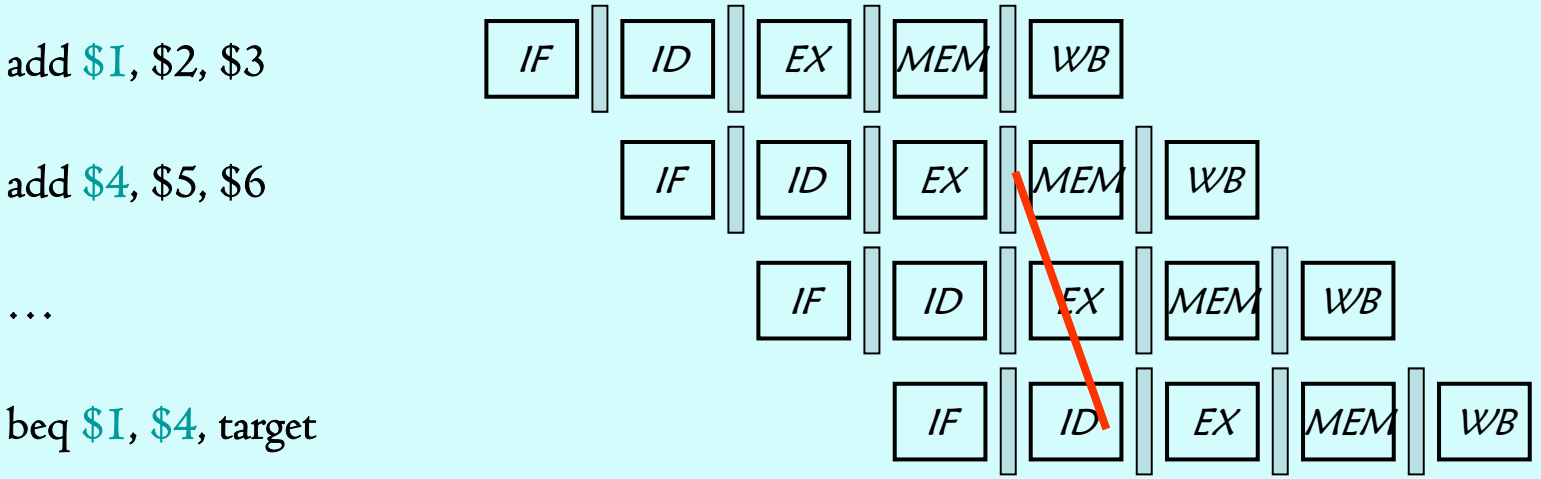


در صورت عدم تحقق شرط (ادامه...)



مخاطره‌ی داده در پرش شرطی

- در صورتی‌که ثبات مقایسه به داده‌ای احتیاج داشته باشد، که هنوز تکمیل نشده، مخاطره‌ی داده رخ می‌دهد.



با پیش‌گرفتن داده قابل حل می‌باشد.