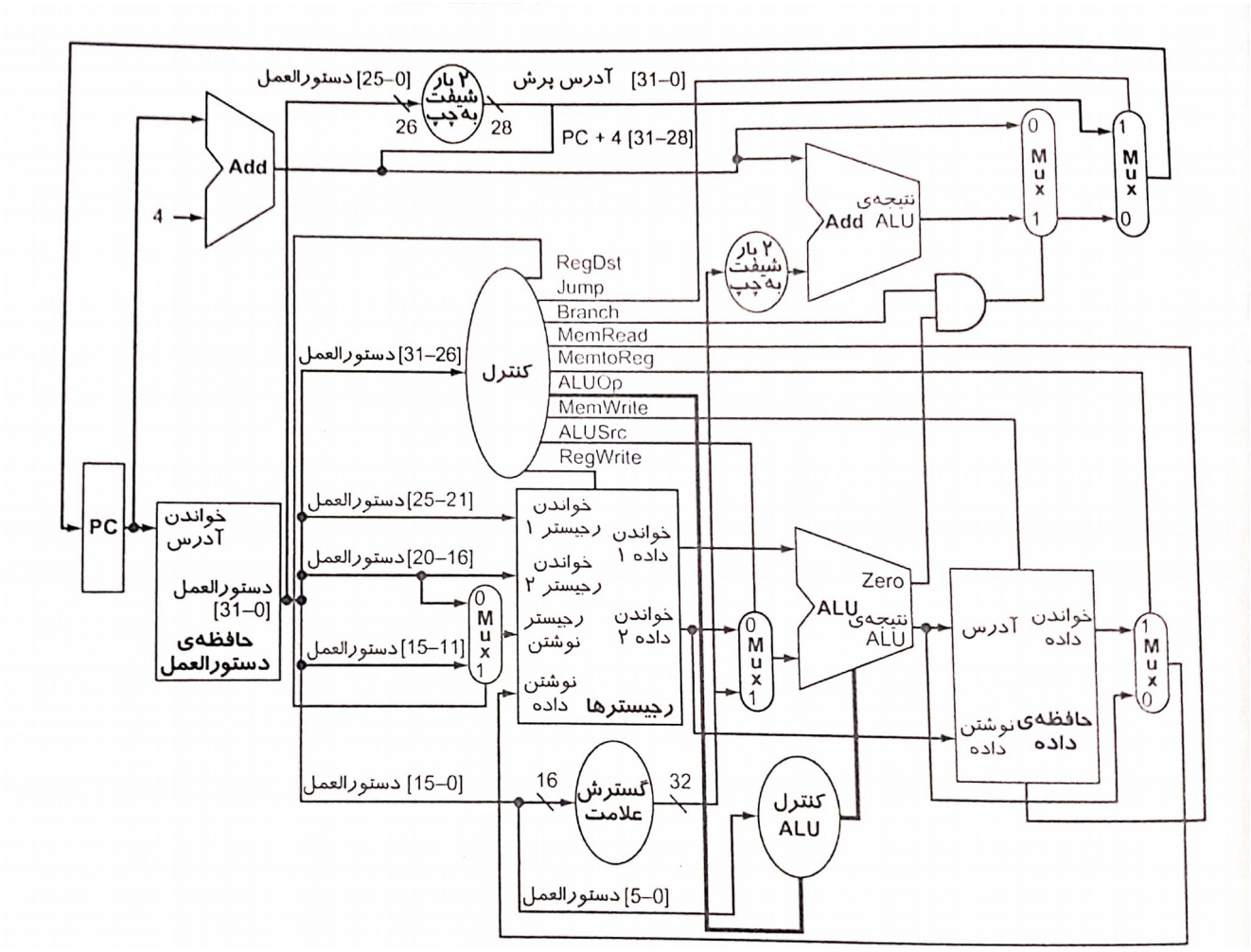
مریم سعیدمهر تکلیف سری سوم معماری 9629373

پاسخ سوال اول :

این مسیرداده خاص نمیتواند دستورات پرش را انجام دهد یعنی «دستور C» قابلیت اجرا ندارد.

حال شکل اصلاح شده ی آن به فرم زیر است :



پاسخ سوال دوم :

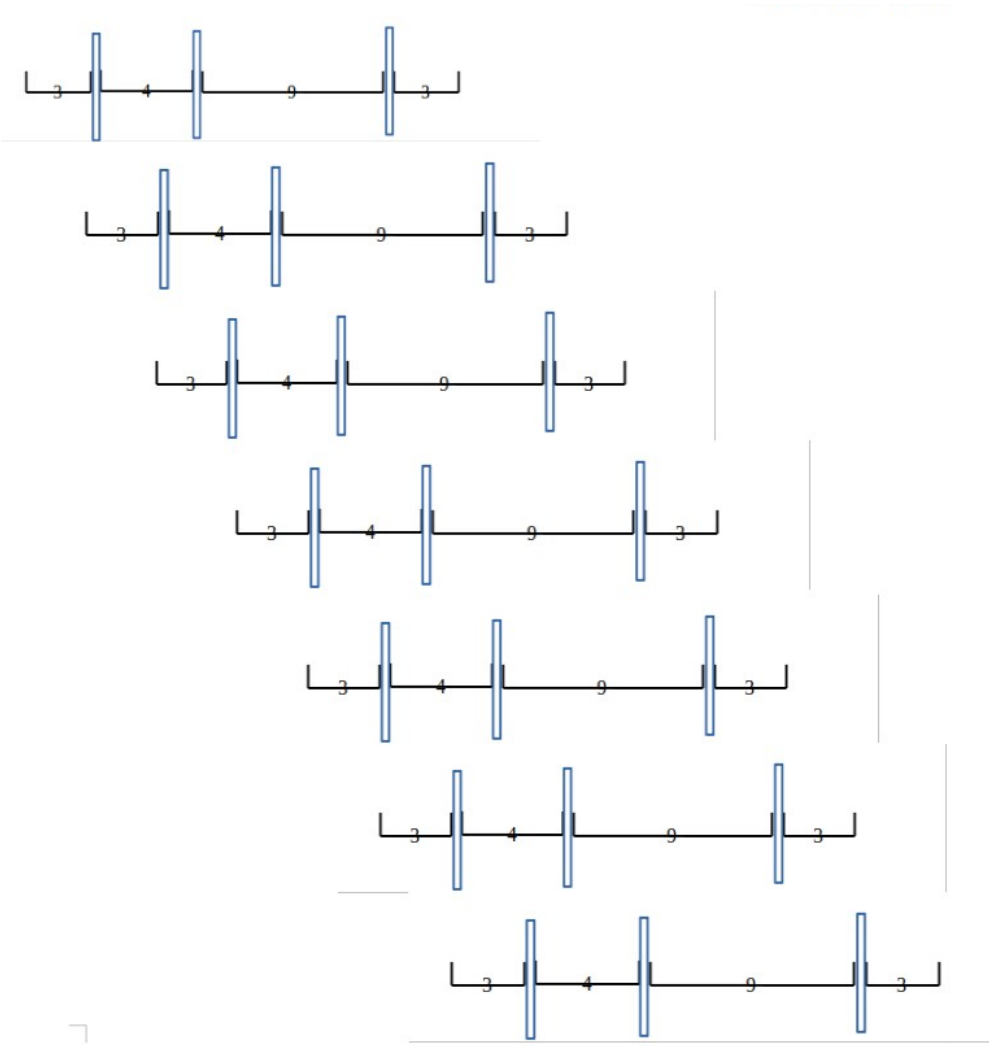
اگر این دو دستور به عنوان دستور اصلی شناخته بشن باید برای هرکدام کد ماشین منحصر به فرد در نظر گرفته شود و همچنین در واحد ALU باید تغییراتی صورت گیرد در حالی که میتوان این دو دستور را با دستورات ساده تری جایگزین کرد بدون اینکه نیاز به تغییراتی در سخت افزار باشد.

پاسخ سوال سوم :

در عدم وجود خط لوله :

```
Totall time = 7 * (3+4+9+3 ) = 7 * 19 = 133 ns
```

در صورت وجود خط لوله :



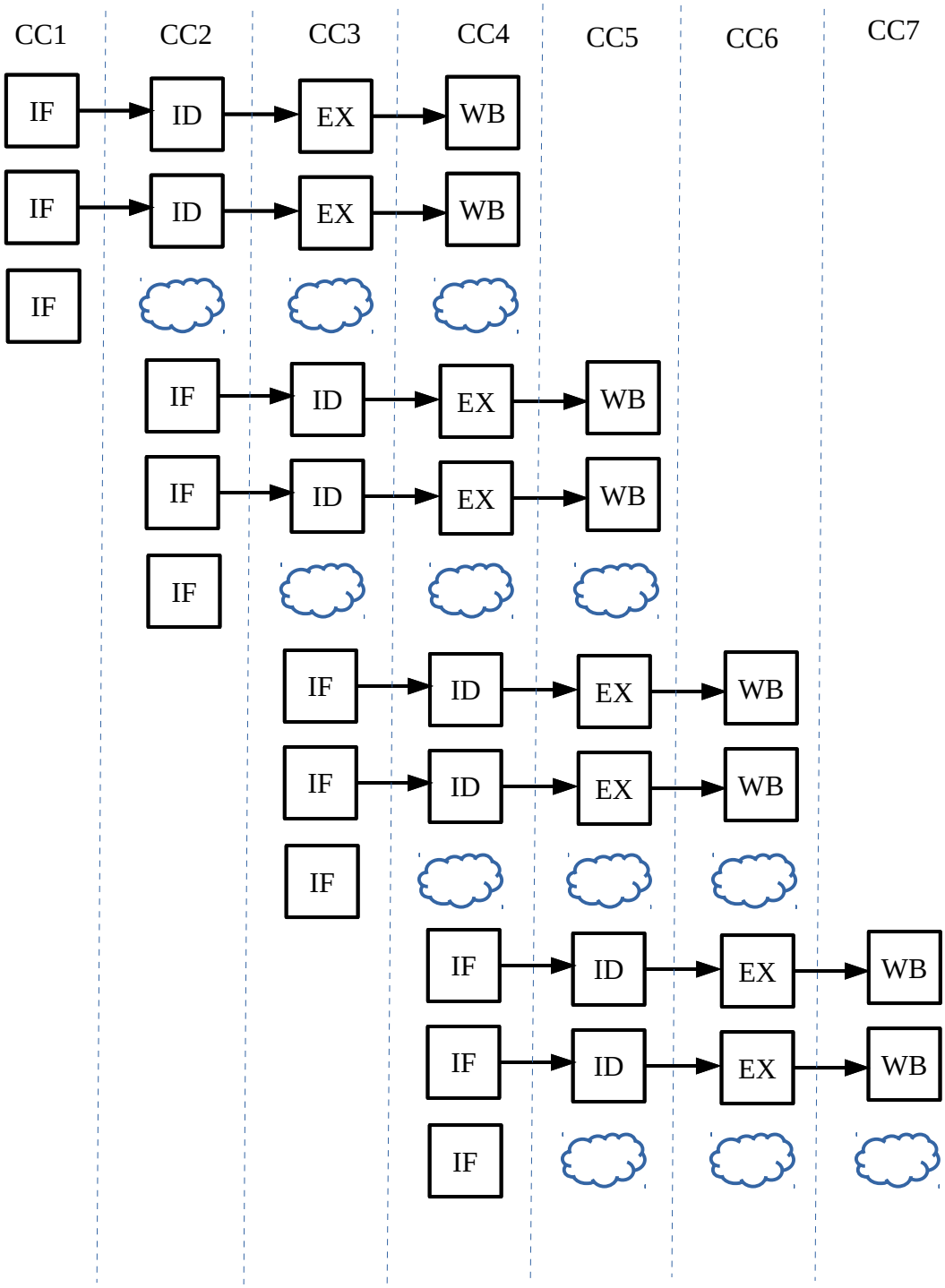

```
Totall time = ( 3+1 )+( 4+1 )+( 9+1 )+( 3+1 )+ [ 6 times ( 3+1 ) ] = 47 ns
```

پس با اضافه کردن خط لوله سرعت پردازنده در اجرای این ۷ دستور متوالی غیر وابسته ~ 2.83 برابر میشود :

پاسخ سوال چهارم :

اگر این شکل همان منظور سوال باشد /: ، در نهایت برای ۳۳ دستورالعمل ، ۲۰ کلاک لازم است.

```
number of clocks needed =  33/2 + (3+1) = 20
```



پاسخ سوال پنجم :

FE	DE	EXE	MEM	WB		
۱۰ns	۷ns	۱۰ns	۱۲ns	۷ns	درصدوقوع	نوع‌دستور
✓	✓	✓	✓	✓	۲۰٪	الف
✓	✓	✓	—	✓	۴۰٪	ب
✓	✓	✓	✓	—	۲۰٪	ج
✓	✓	✓	—	—	۲۰٪	د

(اندازه ی کلاک باید ۱۲ نانوثانیه باشد حداقل)(فرض میکنیم دستورات وابستگی ندارند)(فرض میکنیم دستورات به ترتیب ۲۰ تای اول از نوع الف ، ۴۰ تای بعد از نوع ب ، ۲۰ تای بعدی از نوع ج و ۲۰ تای آخر از نوع د هستند)

در صورت عدم وجود خط لوله و اجرای ۱۰۰ دستور از انواع الف تا د :

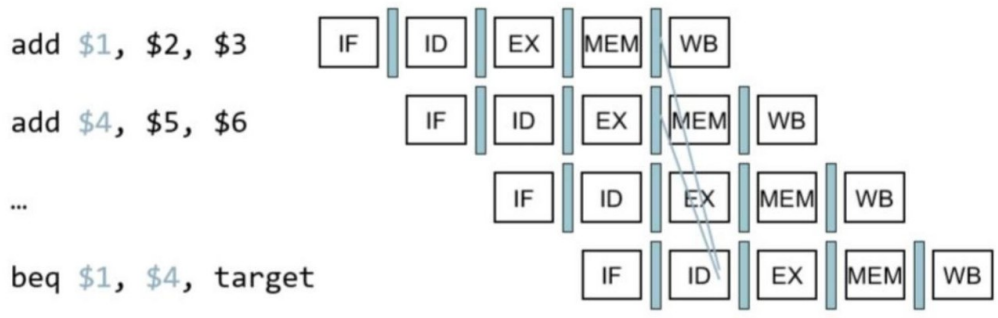
20 * (5 * 12) + 40 * (4 * 12) + 20 * (4 * 12) +20 * (3 * 12) = 4800 ps

در صورت وجود خط لوله و اجرای ۱۰۰ دستور از انواع الف تا د :

((19+5) * 12)+(39 * 12)+(20 * 12)+(19 * 12) = 1224 ps

پس با وجود خط لوله ، پردازنده ~ 3.92 بار سریعتر کار میکند!

پاسخ سوال ششم :



if ((MEM/WB.RegWrite and (MEM/WB.RegisterRd %notequal 0) and MEM/WB.RegisterRd = ID/EXE.RegisterRs) and (EX/MEM.RegWrite and (EX/MEM.RegisterRd %notequal 0) and (EX/MEM.RegisterRd = ID/EXE.RegisterRt))

ForwardA = 01 , ForwardB = 10

if ((MEM/WB.RegWrite and (MEM/WB.RegisterRd %notequal 0) and MEM/WB.RegisterRd = ID/EXE.RegisterRt) and (EX/MEM.RegWrite and (EX/MEM.RegisterRd %notequal 0) and (EX/MEM.RegisterRd = ID/EXE.RegisterRs))

ForwardB = 01 , ForwardA = 10

پاسخ سوال هفتم :

الف)

کافیست تمام nop ها را حذف و همچنین جای دو دستور - -i و sum += I را عوض کنیم.

در این صورت با یک عمل forwarding مشکل حل میشود.

نتیجه :

```
L : add $t1,$t1,$t0
    addi $t0,$t0,-1
    bne $t0,$0,L
    j  Exit
```

ب)

اگر معنای « تسریعی که حاصل میشود برحسب n » رو درست فهمیده باشم :

در کدی که در صورت سوال آمده در حقیقت جمع اعداد از n-1 تا 1 رو حساب میکنه ولی در کدی که من در قسمت الف نوشتم ، جمع اعداد از n تا 1 حساب میشود.

پاسخ سوال هشتم :

الف)

```
sub $2, $1,$3
and $12,$2,$5
or  $13,$6,$2
nor $14,$2,$2
sw  $15,100($2)
lw  $15,80($13)
xor $8,$8,$15
add $7,$15,$3
```

در مورد $2 در دستور دوم با data hazard مواجه میشیم که البته با forwarding برطرف میشه.

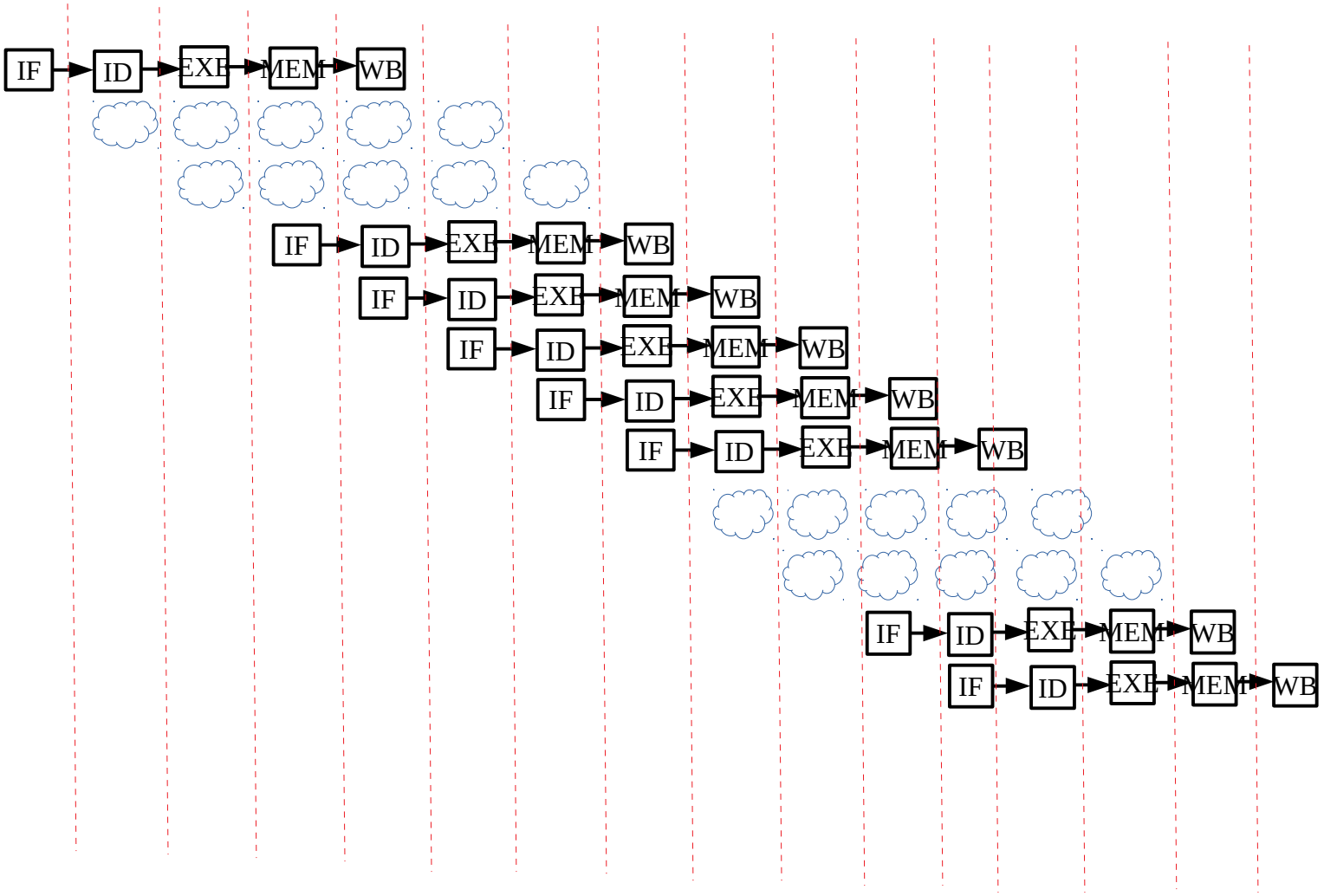
مجدداً در مورد $2 در دستور سوم با data hazard مواجه میشیم که البته با forwarding برطرف میشه.

در مورد $15 در دستور هفتم با data hazard مواجه میشیم که نیاز به stall دارد و صرفاً با forwarding حل نمیشود.

اگه stall که در قسمت قبل گفتم اعمال شود دیگر با هیچ مدل hazard در دستورات بعدی مواجه نمیشویم.

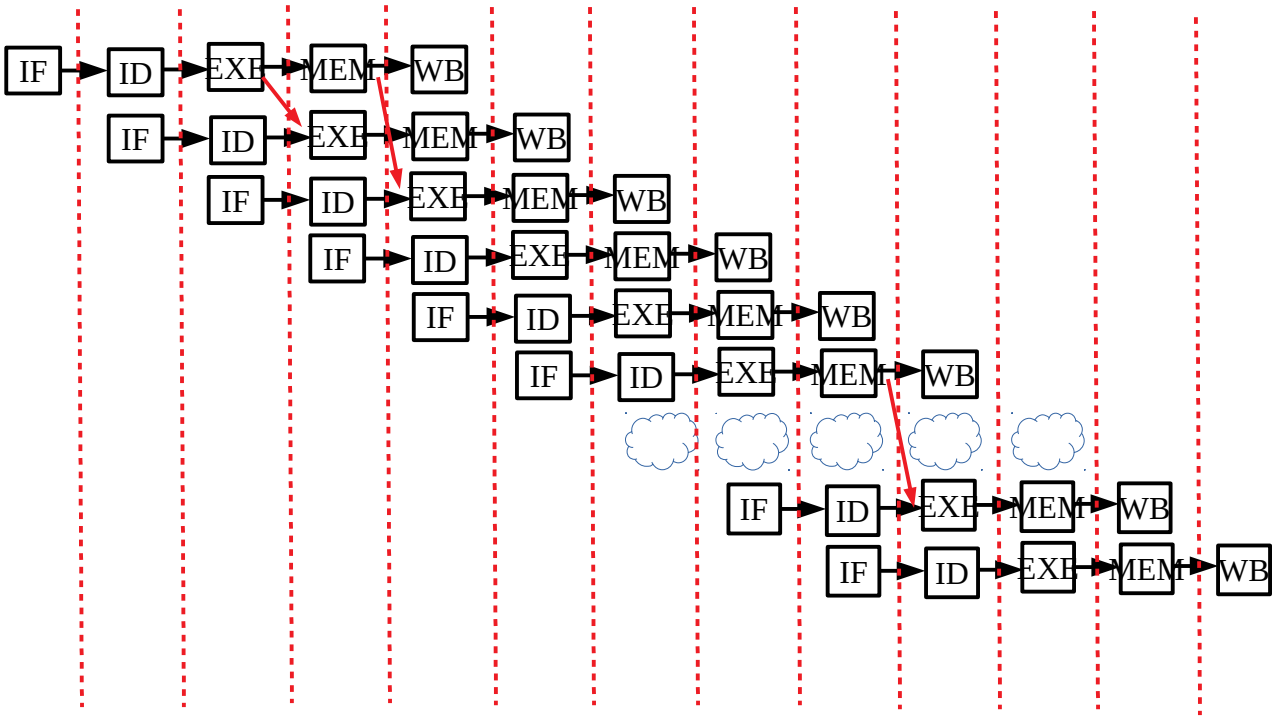
همچنین control hazard نداریم چون دستورات پرشی نداریم.

ب)



به ۱۶ کلاک نیاز دارد .

ج)



به ۱۳ کلاک نیاز دارد .