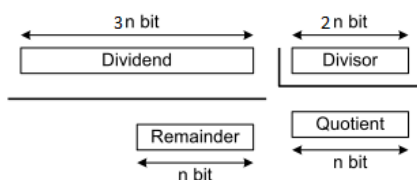


1- ساده

2- ساده

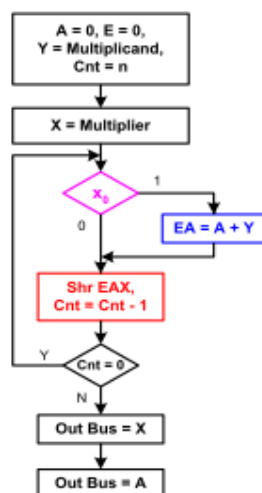
3- در صورتی که فرض کنیم Quotient بیشترین مقدار خود را داشته باشد ($2^n - 1$) آنگاه حداکثر می‌تواند وقتی که در Divisor ضرب می‌شود آن را n بیت شیفت دهد. پس در صورتی که Divisor کمتر از $2n$ بیت پرارزش Dividend باشد ما حتما سرریز داریم.



4- یک نمونه از ضرب non-restoring در سایت قرار گرفته است.

5- با توجه به شکل زیر که الگوریتم‌های دو نوع ضرب را نشان می‌دهد، در صورتی که عدد 01111111 را در Y ضرب کنیم بیشترین تسریع را از ضرب booth می‌گیریم.

عملیات	booth	Shift&add
مکمل‌گیری	1	0
شیفت	7	7
جمع	2	7
تاخیر کل	75 _{ns}	140 _{ns}



shift&add



booth

-6 الف

$$\text{MIN} = 0\ 00000\ 000001 = (-1)^0 \times 2^{(0-16)} \times 1.000001 \approx 2^{-16}$$

$$\text{MAX} = 0\ 11111\ 111111 = (-1)^0 \times 2^{(31-16)} \times 1.111111 \approx 2^{16}$$

ب

$$0\ 10001\ 000000 = (-1)^0 \times 2^{(17-16)} \times 1.000000 = (2)_d$$

$$1\ 01111\ 101010 = (-1)^1 \times 2^{(15-16)} \times 1.101010 = (0.1101010)_b = 2^{-1} + 2^{-2} + 2^{-4} + 2^{-6} = (0.828125)_d$$

ج

$$-0.3432 \approx (-0.34375)_d = -(0.25 + 0.0625 + 0.03125) = -(2^{-2} + 2^{-4} + 2^{-5}) = -(0.0101100)_b = -1.011000 \times 2^{-2}$$

$$= -1.011000 \times 2^{14-16} = 1\ 01110\ 011000$$

-7 الف

$$\text{Max Opcode\#1} = (2^{16} - 120 \times 2^6) / 2^{12} = 14$$

ب

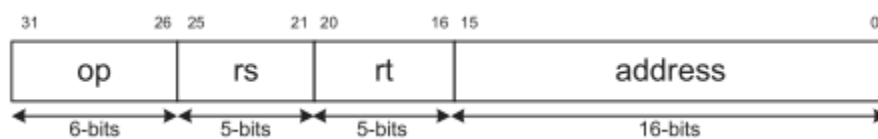
$$\text{Max Opcode\#0} = (2^{16} - 120 \times 2^6 - 14 \times 2^{12}) / 2^0 = 512$$

-8

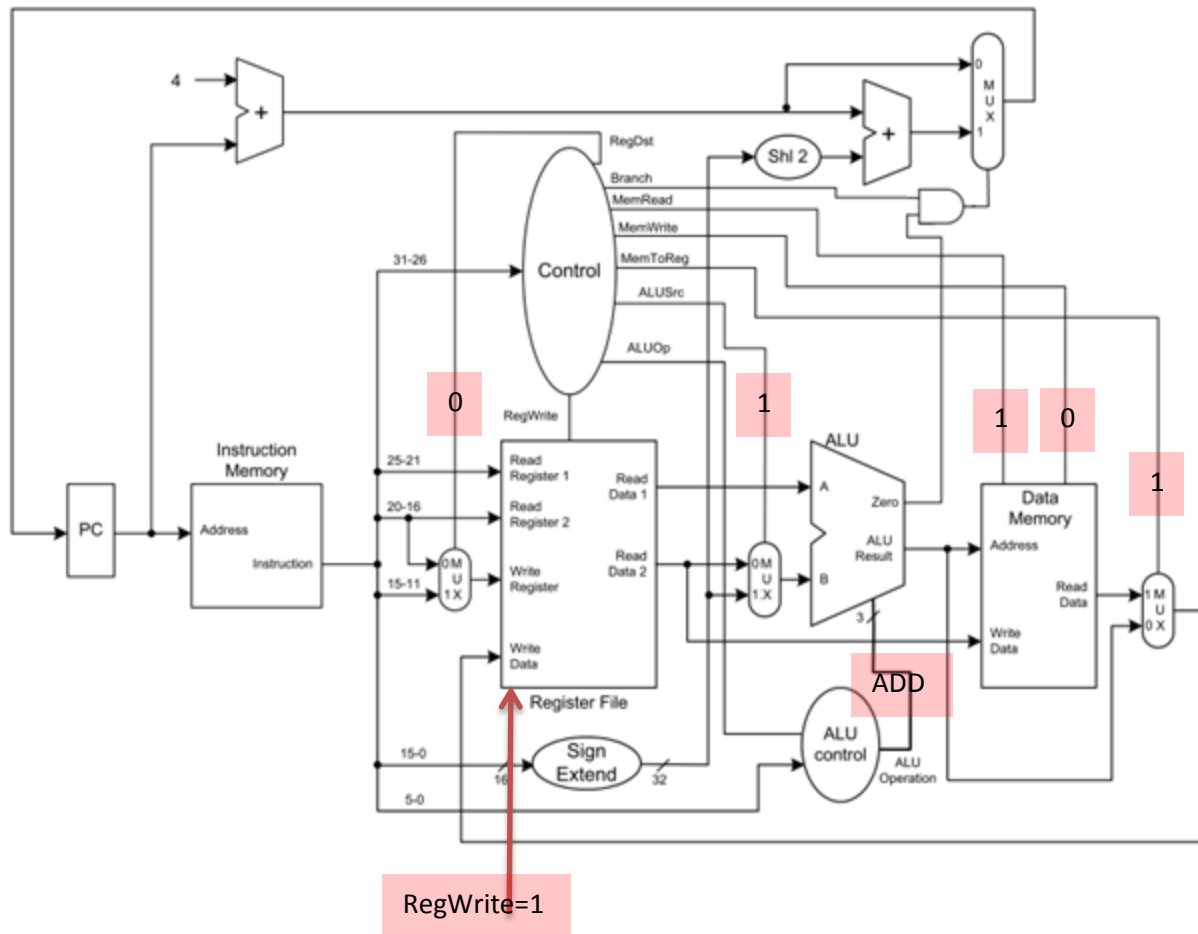
Fib iterative:

```
Lw    $1,$0(0)
ADDi   $2,$0,0
ADDi   $3,$0,1
ADDi   $4,$0,-1
Beq    $1,$0,4
ADD    $3,$2,$3
Sub    $2,$3,$2
ADD    $1,$1,$4
Jump   1
Sw     $2,$0(4)
```

-9 الف

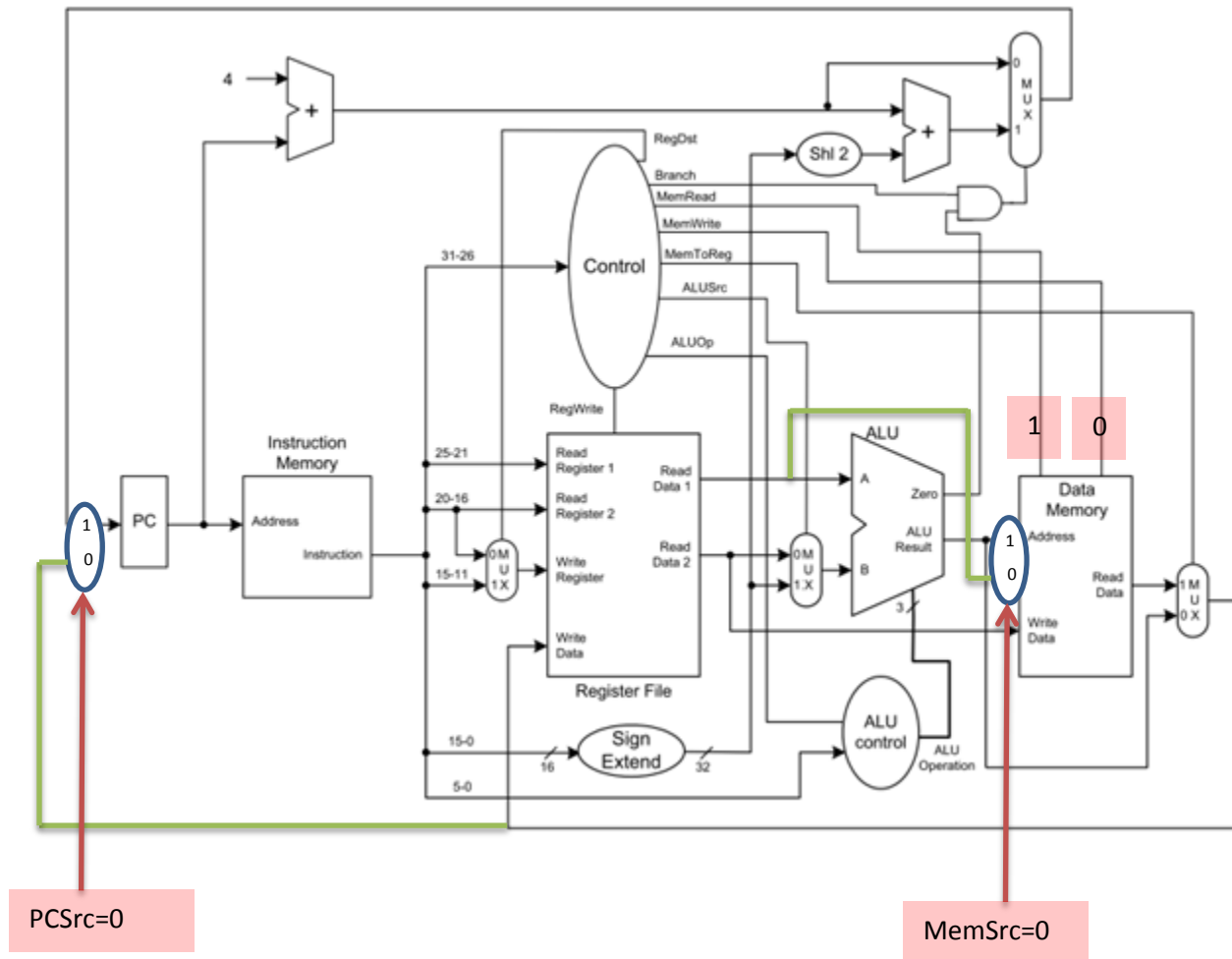


با فرض $rt=\$i$ و $rs=\$j$ و سیگنال های کنترلی نشان داده شده:



(ب) در صورتی که $rs=\$i$ باشد:





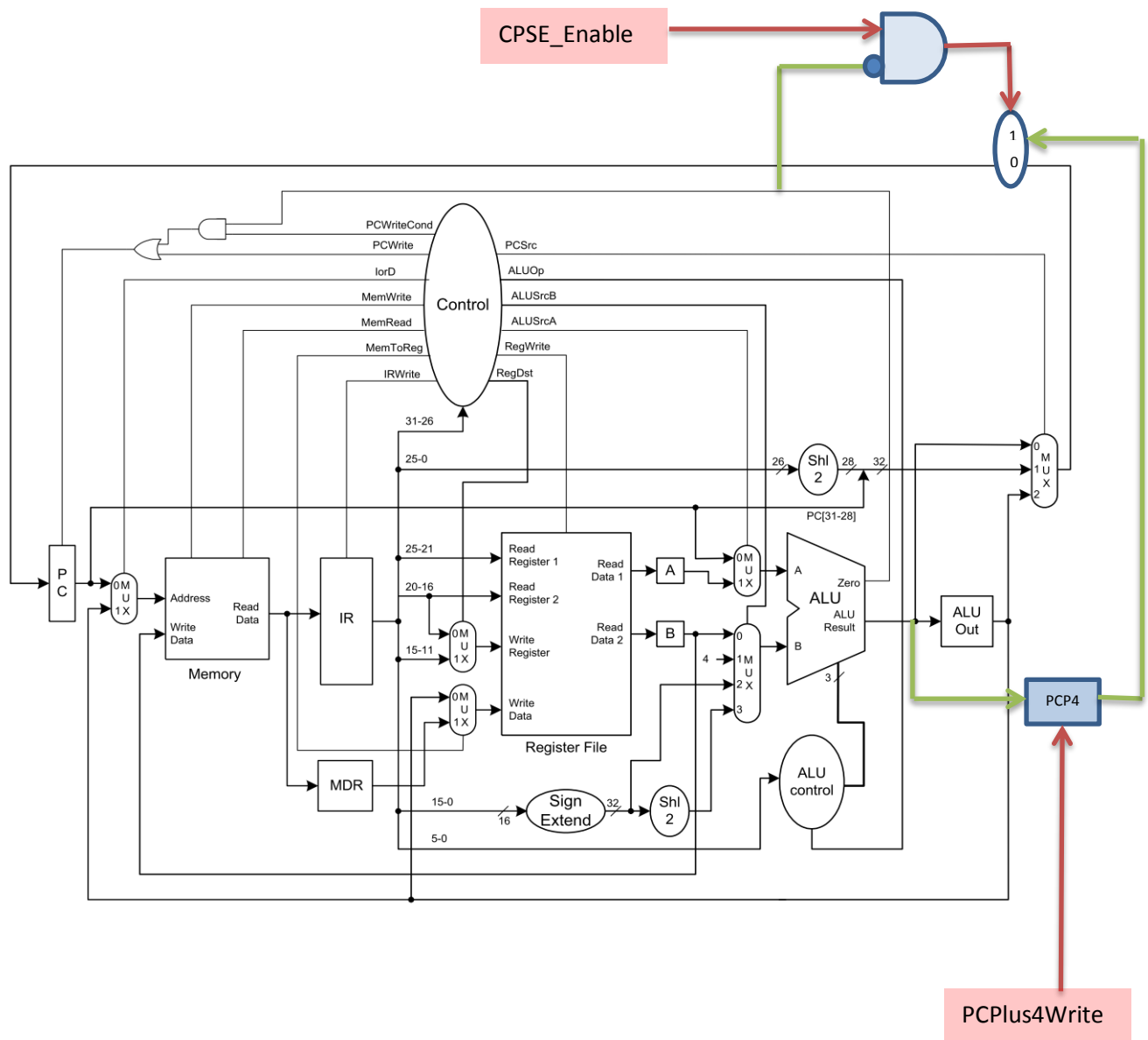
10- الف)

$$-b = \sim b + 1 \quad \Rightarrow \quad \sim b = -1 - b$$

کنترل:

در زیر مراحل اجرای دستور را می بینید:

- 1- IF: مانند قبل
- 2- ID: مانند قبل
- 3- COM: ALUSrcA=2; ALUSrcB=4; ALUOP=SUB;
- 4- COM: RegDst=0; MemToReg=0;



11- فرض می‌کنیم حاصل در \$1, \$2 ذخیره شود. \$2=LSB ، \$1=MSB

کنترلر: در زیر state های کنترلر آورده شده است:

1-IF: مانند قبل

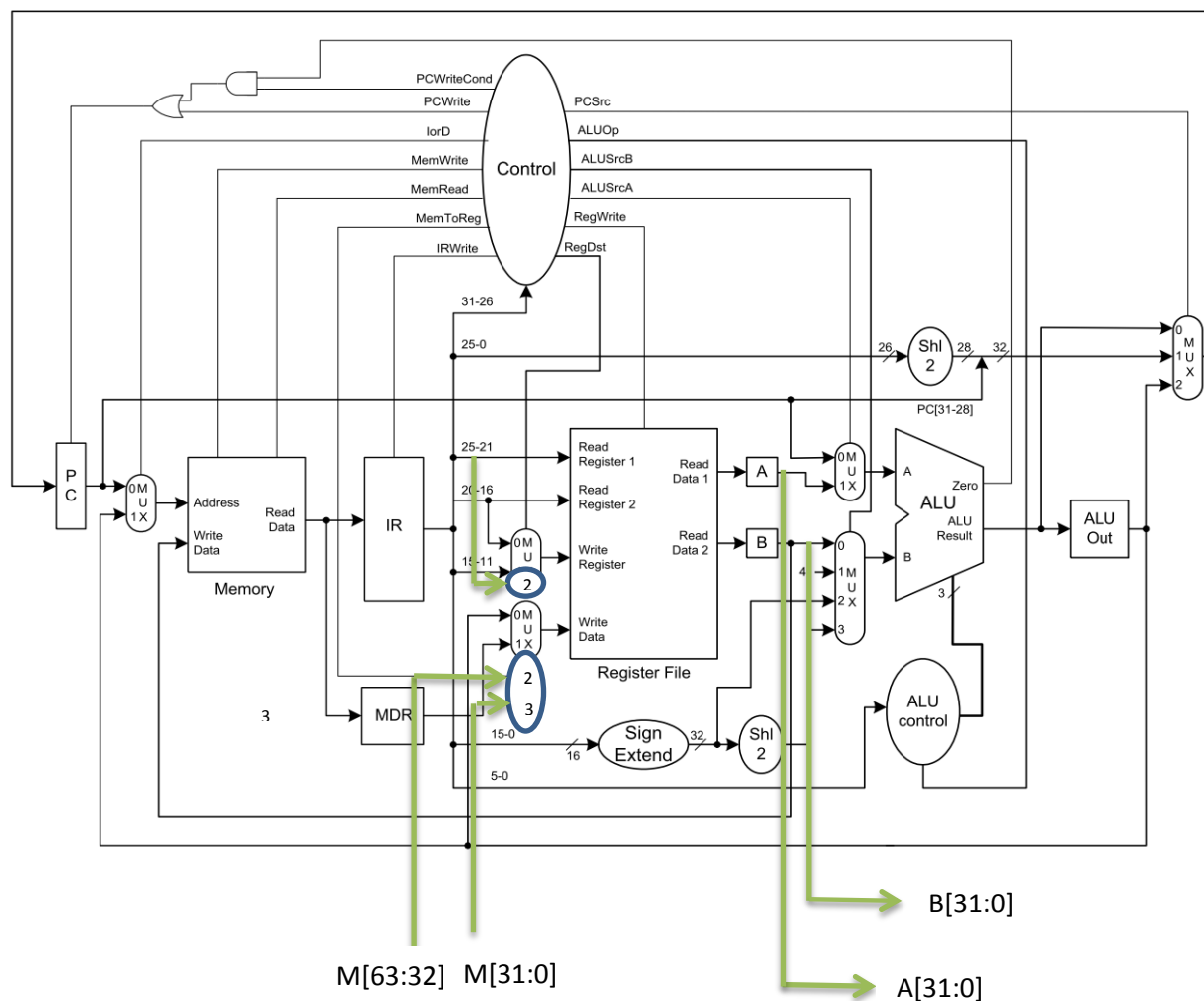
2-ID: مانند قبل

3-MUL: start=1;

4-MUL: multiply=1; if (ready==1) then goto state_5; else goto state_4;

5-MUL: multiply=0; RegDst=0; MemToReg=3;

6-MUL: RegDst=2; MemToReg=2;



12- کارایی:

$$\frac{Pa}{Pb} = \frac{3 * 500}{4.5 * 200}$$

13-

تعداد کلاک هایی که کاهش یافته است: $1 * 1M = 1M$ CC $\Rightarrow 10_5 - 9_5 = 1_5$

بهبود CC $4 - 2 = 2$

تعداد دستورات جایگزین شده $= 1000000 / 2 = 500000$ Inst.

14- الف)

$$\text{CPI} = 0.3 \times 0.1 \times 20 + 0.27 \times 4 + 0.7 \times 2.75 = 3.605$$

ب)

$$\text{CPI}_{\text{new}} = 0.03 \times 10 + 0.27 \times 4 + 0.7 \times 2.75 = 3.305$$

$$\text{Speedup} = \frac{3.605}{3.305} = 1.091$$

9٪ باید افزایش فرکانس داشته باشیم.