

به نام خدا

سرکار جناب آقای دکتر اسدی

سارا آذرنوش 98170668

تمرین 4

درس معماری کامپیوتر

بهار 1400

بنابر گفته استاد چندسیکلی ها حذف شده اند.

(1)

$$\text{If} + \text{id} + \text{ex} + \text{we} = 30 + 20 + 25 + 20 = 95$$

$$\text{If} + \text{id} + \text{ex} + \text{mem} + \text{we} = 30 + 20 + 25 + 30 + 20 = 125$$

$$\text{If} + \text{id} + \text{ex} + \text{mem} + \text{we} = 30 + 20 + 25 + 30 + 20 = 125$$

$$\text{If} + \text{id} + \text{ex} + \text{we} = 30 + 20 + 25 + 20 = 95$$

$$\text{If} + \text{id} + \text{ex} + \text{mem} + \text{we} = 30 + 20 + 25 + 30 = 105$$

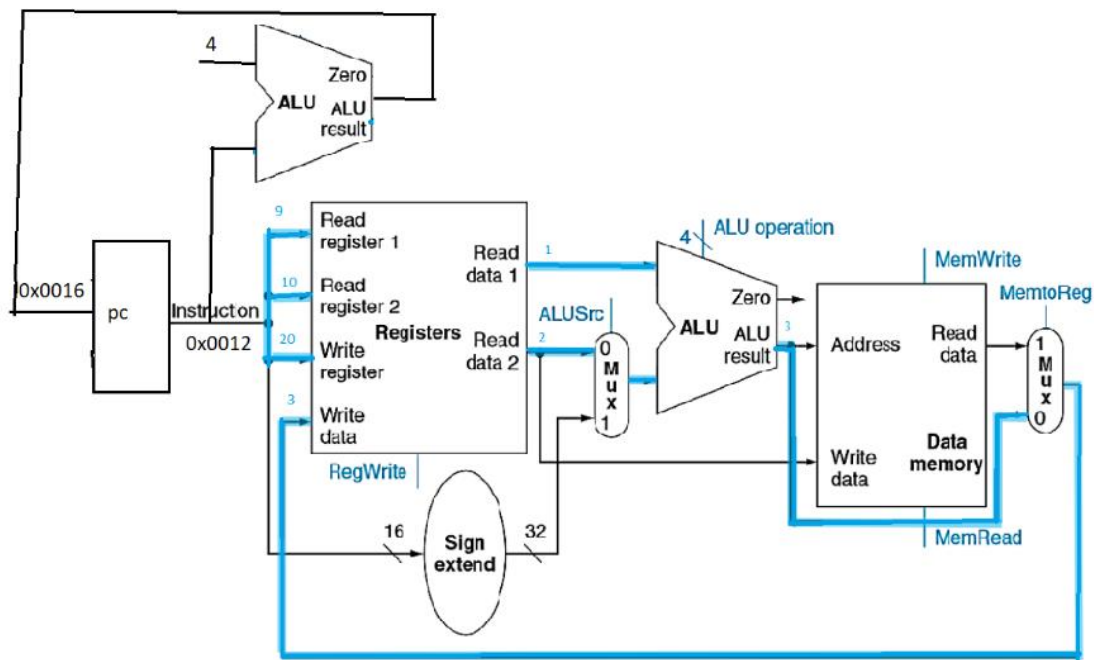
$$\text{If} + \text{id} + \text{ex} + \text{we} = 30 + 20 + 25 + 20 = 95$$

$$\text{If} + \text{id} + \text{ex} + \text{we} = 30 + 20 + 25 + 20 = 95$$

$$\text{If} + \text{id} + \text{ex} = 30 + 20 + 25 = 75$$

$$125 * 2 + 4 * 95 + 75 + 105 = 810$$

(2)



(3)

$$Lw\ 10\% \Rightarrow 25 + 15 + 20 = 60$$

$$Sw\ 10\% \Rightarrow 25 + 15 + 20 = 60$$

$$Branch\ 20\% \Rightarrow 20 + 15 = 35$$

$$Jump\ 20\% \Rightarrow 15$$

$$r\text{-}type\ 40\% \Rightarrow 20 + 15 = 35$$

میانگین هر دستور

$$(10 \times 60 + 10 \times 60 + 20 \times 35 + 20 \times 15 + 40 \times 35) / 100 = 36$$

$$36 / (2 \times 60 + 2 \times 35 + 15) = 0.175$$

اگر کلاک را برابر با بیشترین زمان بگیریم

$$CPU\ TIME = 36 \times 60 = 2160$$

(4)

به بیشتری از یک سیکل نیازمند است.

(5)

(الف)

$$I\text{-}type + R\text{-}type + store + load = 24 + 25 + 10 = 59$$

(ب)

$$I\text{-}type + R\text{-}type + store + load + branch + jump = 24 + 28 + 25 + 2 + 10 + 11 = 100$$

(پ)

$$I\text{-}type + store + load + branch = 28 + 10 + 25 + 11 = 74$$

(ت)

16 بیت را به 32 بیت تبدیل میکند

مواردی که 16 بیت داریم اما برای مثال در *alu* عدد دوم 32 بیت است عدد را به 32 بیت تبدیل میکنیم موارد استفاده نیز قسمت پ هستند.

(6)

(الف)

$$R_type: Alu + adder + rf + id/meme + control + mux = 200 + 150 + 150 + 250 + 50 + 25 = 825$$

مقدار محاسبه شده دوباره به rf بازگشته و در آن ذخیره میشود اگر rf را دوبار محاسبه کنیم:

$$825 + 150 = 975$$

(ب)

$$Load: Alu + adder + rf + id/meme + sign extend + rr + 4 * mux + control + and = 200 + 150 + 150 + 250 + 50 + 30 + 5 + 50 + 4 * 25 = 985$$

مقدار محاسبه شده دوباره به rf باز میگردد اگر rf را دوبار محاسبه کنیم:

$$985 + 150 = 1135$$

(پ)

$$Store: Alu + adder + rf + id/meme + sign extend + rs + 4 * mux + control = 200 + 150 + 150 + 250 + 50 + 4 * 25 + 25 + 50 = 975$$

(ت)

$$Beq: Alu + 2 * adder + rf + id/meme + sign extend + control + 3 * mux + shifleft = 200 + 2 * 150 + 150 + 250 + 50 + 50 + 3 * 25 + 0 = 1075$$

(ث)

$$i_type: alu + sign extend + rf + rr + rs + id/meme + 3 * mux + control + adder = 200 + 50 + 150 + 250 + 3 * 25 + 50 + 150 + 30 + 25 = 980$$

(ج)

کمینه:

$$1/d mem + rf + alu + control + adder + 2 * mux + rs =$$

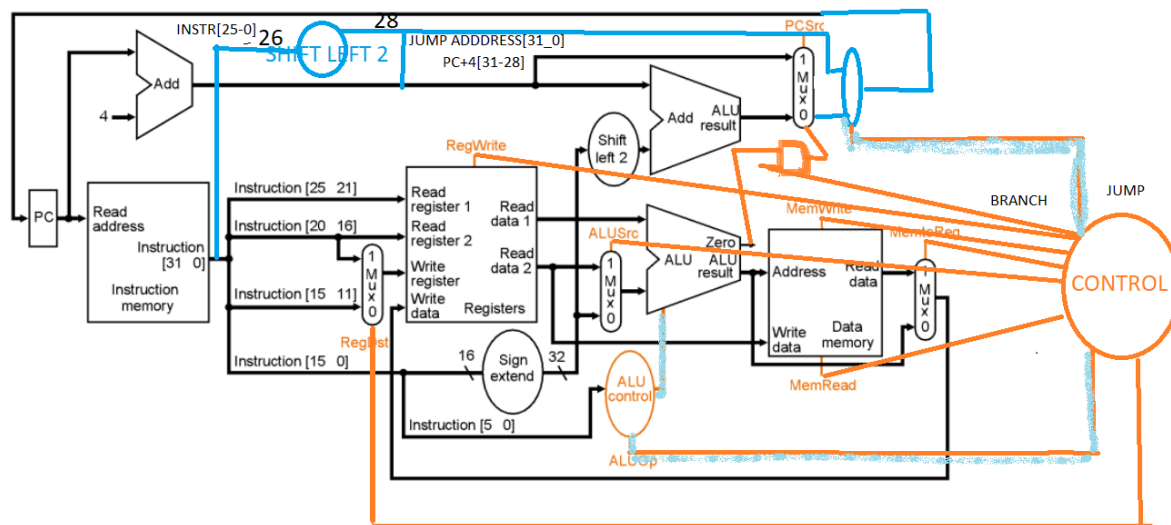
$$250 + 150 + 200 + 50 + 150 + 50 + 25 = 875$$

(برای کلاک، کلاک باید به اندازه آهسته ترین دستور باشد.)

(7)

با افزودن بخش آبی میتوانیم آدرس را تولید و بدست آورده دستورات *jr* را انجام دهیم.

نارنجی مربوط به کنترل است.



(8)

101011 00010 00010 00000000000010100 *op rt rs imm*

Sw 2 2 20

دستور 101011 میباشد که یعنی *sw* است و 2 بیتی بعدی برای ادرس هستند و 16 بیت آخر برای ایمیدیت است.

شکل زیر *sw datapath* است.

در شکل مقادیر نمایش داده شده اند.

(الف)

signextend = 000000000000000000000000000010100

shift left 2 = استفاده نمیشود

(00000000000000000000000000001010000)

(ب)

ماکس مقدار *sign extend* را خروجی میدهد بقیه ماکس ها استفاده نمیشوند.

(خروجی $alu = 22$ - خروجی $pc = 4$ - $adder$ - $rt = 2$)

(پ)

در ابتدا 0 بوده است سپس 4 میشود.

(ت)

Alu متصل به *rf* به دو عدد 32 بیتی 2 هستند.

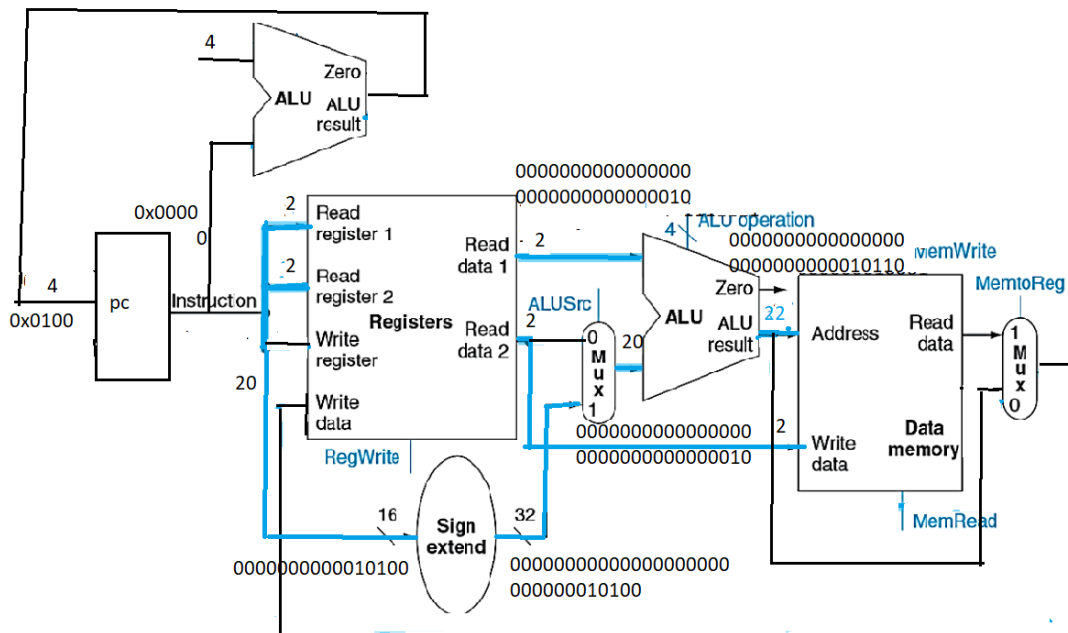
Alu متصل به *pc* به 0 و 4 هستند.

دیگری استفاده نمیشود.

(0000000000000000000000001010000 و 4 هستند.)

(ث)

آدرس رجیستر ها که دو 2 هستند است.



عملی:

ورودی ها را مطابق فرمت گفته شده میگیریم و با توجه به مقدار *write register* و دیکودر 5 به 32 آدرس ثبات مورد نظر را پیدا و اگر *regWrite* فعال باشد مقدار *write data* در آن مینویسیم. (اگر کلیر نیز فعال بود کلیر میکنیم برای اینکار *regWrite* و *clr* را نند میکنیم تا اگر هر دو 1 باشد *regWrite* 0 شود و *clr* انجام شود.

2 باس (*read register 1/2*) که برای آدرس ثبات ها میباشد را هر یک به *32 mux* به 5 متصل میکنیم تا بنابر آدرس داده شده مقدار آن دو ثبات را خروجی دهد.

(متاسفانه برنامه *quartus* بنده علی رغم تلاش فراوان قسمت *vaweform* را کامپایل نمیکند)