

## پاسخ تمرین شماره ۱ درس معماری کامپیوتر

امیر حسین عاصم یوسفی  
۹۶۱۱۰۳۲۳

۱۴ اسفند ۱۳۹۷

### سوال ۱:

با توجه به فرمول محاسبه  $CPU TIME$  که برابر است با  $\frac{CPU\ CLOCK\ CYCLE}{CLOCK\ RATE} = \frac{(Instructioncount * CPI)}{Clock\ Rate}$  برای میتوان تعداد دستورات از هر دو نوع گفته شده را به دست آورد که به شرح زیر می باشد:

$$\begin{aligned} A \text{ in processor } p1 : \frac{3}{200} &= \frac{IC(A) * CPI}{200} \Rightarrow IC(A) = \frac{3}{CPI} \\ A \text{ in processor } p2 : \frac{5}{300} &= \frac{IC(A) * CPI}{300} \Rightarrow IC(A) = \frac{5}{CPI} \\ B \text{ in processor } p1 : \frac{4}{200} &= \frac{IC(B) * CPI}{200} \Rightarrow IC(B) = \frac{4}{CPI} \\ B \text{ in processor } p2 : \frac{3}{300} &= \frac{IC(B) * CPI}{300} \Rightarrow IC(B) = \frac{3}{CPI} \end{aligned}$$

حال با توجه به بالا می توانیم مجموع دستورات از هر نوع را به دست آوریم:

$$Sum\ A\ (p1 + p2) = A \text{ in processor } p1 + A \text{ in processor } p2 = \frac{8}{CPI}$$

$$Sum\ B\ (p1 + p2) = B \text{ in processor } p1 + B \text{ in processor } p2 = \frac{7}{CPI}$$

و چون طبق صورت سوال زمان اجرای یک برنامه خاص در هر دو پردازنده یکسان می باشد بنابراین هر دو مقدار  $CPI$  با یک دیگر برابرند، پس داریم:

$$\frac{IC(A)}{IC(B)} = \frac{8}{7}$$

### سوال ۲:

برای انجام این دستور به ۱۵ کلاک نیاز داریم که دستورات هر کلاک به شرح زیر می باشد:

$$\begin{aligned} T_1 : AR &\leftarrow PC \\ T_2 : DR &\leftarrow M[AR] \\ T_3 : Decode\ "DR" , PC &\leftarrow PC + 1 \\ T_4 : AR &\leftarrow PC \\ T_5 : DR &\leftarrow M[AR] , PC \leftarrow PC + 1 \\ T_6 : AR &\leftarrow DR \\ T_7 : AC &\leftarrow M[AR] , PC \leftarrow PC + 1 \\ T_8 : AC &\leftarrow DR , AR \leftarrow PC \\ T_9 : DR &\leftarrow M[AR] \end{aligned}$$

$$\begin{aligned}
&T_{10} : AR \leftarrow DR \\
&T_{11} : DR \leftarrow M[AR], PC \leftarrow PC + 1 \\
&T_{12} : AC \leftarrow DR + AC, AR \leftarrow PC \\
&\quad T_{13} : DR \leftarrow M[AR] \\
&T_{14} : AR \leftarrow DR, DR \leftarrow AC \\
&T_{15} : M[AR] \leftarrow DR
\end{aligned}$$


---

### سوال ۳ :

$MIPS = \text{Million Instruction Per Second} = \frac{\text{Instruction Count}}{(\text{Execution time} \times 10^6)}$  با توجه به فرمول  
و با توجه به فرمول به دست آوردن زمان اجرا داریم :

$$\text{Execution Time} = \text{Instruction count} \times CPI / \text{ClockRate}$$

از طرفی با توجه به اطلاعات مساله برای پردازنده اول داریم : برای پردازنده اول با توجه به اطلاعات مساله داریم :

<i>Frequency of Instruction</i>	<i>Value</i>	<i>Product</i>
$SUM = 10\%$	4	0.4
$MULT = 7\%$	7	0.49
$DIVISION = 5\%$	9	0.45

با توجه به بالا می توان  $CPI$  را محاسبه کرد ، که برابر است با :

$$CPI = 0.4 + 0.49 + 0.45 = 1.34$$

بنابراین زمئن اجرا برای پردازنده اول برابر است با :

$$\text{Execution Time} = CPI \times IC / \text{ClockRate} = 1.34 \times IC / \text{ClockRate}$$

پس مقدار  $MIPS$  آن برابر است با :

$$MIPS_{c1} = \frac{IC}{1.34 \times IC / (\text{ClockRate} = 600)} = \frac{600}{1.34}$$

از طرفی با توجه به اطلاعات گفته شده در مساله برای پردازنده دوم داریم :

<i>Frequency of Instruction</i>	<i>Value</i>	<i>Product</i>
$SUM = 10\%$	24	2.4
$MULT = 7\%$	36	2.52
$DIVISION = 5\%$	57	2.85

با توجه به بالا می توان  $CPI$  را محاسبه کرد ، که برابر است با :

$$CPI = 2.4 + 2.52 + 2.85 = 7.77$$

بنابراین زمین اجرا برای پردازنده دوم برابر است با :

$$Execution\ Time = CPI \times IC / ClockRate = 7.77 \times IC / ClockRate$$

پس مقدار  $MIPS$  آن برابر است با :

$$MIPS_{c_2} = \frac{IC}{7.77 \times IC / (ClockRate=X)} = \frac{X}{7.77}$$

حال اگر دو مقدار بالا را برابر یک دیگر قرار دهیم داریم :

$$MIPS_{c_1} = MIPS_{c_2} \Rightarrow \frac{600}{1.34} = \frac{X}{7.77} \Rightarrow X = 600 \times 7.77 / 1.34 \cong 3447\ MHz = 3.447\ GHz$$

و این انتظار را داشتیم زیرا تعداد سیکل ها لازم برای انجام هر عملیات ممیز شناور برای پردازنده دوم بیشتر از پردازنده اول است بنابراین باید فرکانس کاری بیشتری نسبت به پردازنده اول داشته باشد تا  $MIPS$  آن ها برابر شود .

## سوال ۴ :

با توجه به این که در هر بار دسترسی به حافظه فقط می توانیم یک عدد را بخوانیم بنابراین می توان گفت برای ضرب هر دو عدد دو بار باید به حافظه دسترسی داشته باشیم . که تعداد آن ها برابر است با  $2n^3$  از طرفی بعد از به دست آوردن مقدار یک درایه باید آن را داخل حافظه بنویسیم بنابراین به ازای هر درایه از ماتریس حاصل ضرب باید به حافظه دسترسی یابی و دستور نوشتن آن درایه را بدهیم بنابراین تعداد دستورات خواندن برابر است با  $n^2$  همچنین برای ضرب دو عدد باید یک دستور ضرب را بدهیم بنابراین برای ضرب دو ماتریس مربعی با اندازه دلخواه به تعداد  $n^3$  دستور ضرب بدهیم و همین تعداد نیز برای عمل جمع باید دستور صادر کنیم . بنابراین تعداد کل دستورات برابر است با

$$4n^3 + n^2$$

حال مانند سوال قبل جدول زیر را تشکیل می دهیم :

Frequency of Instruction	Value	Product
$Read\ or\ Write = \frac{2n^3+n^2}{4n^3+n^2}$	15	$15 \times \frac{2n^3+n^2}{4n^3+n^2}$
$MULT = \frac{n^3}{4n^3+n^2}$	5	$5 \times \frac{n^3}{4n^3+n^2}$
$SUM = \frac{n^3}{4n^3+n^2}$	2	$2 \times \frac{n^3}{4n^3+n^2}$

با توجه به بالا می توان  $CPI$  را محاسبه کرد ، که برابر است با :

$$CPI = 15 \times \frac{2n^3+n^2}{4n^3+n^2} + 5 \times \frac{n^3}{4n^3+n^2} + 2 \times \frac{n^3}{4n^3+n^2} = \frac{1}{4n^3+n^2} (37n^3 + 15n^2)$$

و حال با توجه به فرمول زمان اجرا داریم :

$$Execution\ Time = CPI \times IC / ClockRate = 7.77 \times IC / ClockRate \Rightarrow$$

$$Execution\ Time = \frac{1}{4n^3+n^2} (37n^3 + 15n^2) \times (4n^3 + n^2) \times 0.25 \times 10^{-8}$$

بنابراین زمان اجرا به شکل به صورت زیر می باشد :

$$Execution\ Time = (37n^3 + 15n^2) \times 0.25 \times 10^{-8}$$

## سوال ۵ :

سیگنال *hand shaking* شامل سه سیگنال *END* , *DONE* , *ERROR* بنابراین می توان *RTL* را به صورت زیر نوشت :

```
start  T0 : A ← 0 , sc ← n , b ← divisor , q ← dividend , state ← 1 , done ←
        0 , ERR ← 0
        T1 : SHL(a : q)
        T2 : a ← a - b
        T3 & a : q ← 0 , a ← a + b , ERR ← 1
        T3 & (not)a : set q0 , q0 ← 1
        T4 : sc ← sc - 1
        T4 & sc : q ← q0 , remainder ← a , done ← 1
        T4 & (not)sc : state ← 1
```

توضیح این که این برنامه در صورتی که مقدار *sc* برابر با صفر باشد باید سیگنال *done* را یک کنیم و مقدار دو سیگنال دیگر را برابر با صفر قراردهیم ولی در صورتی که *sc* برابر با یک باشد نگاه باید *state* را برابر با یک قرار دهیم تا به *T<sub>1</sub>* برویم و کار خود را باید ادامه دهیم و جایی که سیگنال *ERR* را یک می کنیم به این معنا میباشد که با تفریق های متوالی عدد مقسوم منفی می شود که یک خطا محسوب میشود .

## سوال ۶ :

( الف )

این قانون بیان می کند که بهبود سرعت قسمتی از یک برنامه محدود به همان قسمت و باعث نمی شود که سرعت کل برنامه به همان نسبت بهبود یابد که به شرح زیر است :

$$Execution\ Time_{new} =$$

$$Execution\ Time_{old} \times [(1 - Frac_{enhanced}) + (Frac_{enhanced} / Speedup_{enhanced})]$$

$$Speedup_{overall} = Exec\ Time_{old} / Exec\ Time_{new}$$

( ب )

اگر فرض کنیم اجرای این برنامه ۱۰۰ ثانیه طول می کشد پس می توان گفت ۶۰ ثانیه از آن را برای ضرب و ۴۰ ثانیه از آن را برای دستورات دیگر استفاده می کند . حال چون گفته شده عملیات ضرب ۴۰ درصد افزایش سرعت داشته بنابراین برنامه عملیات ضرب را در ۳۶ ثانیه انجام میدهم پس زمان اجرا به ۷۶ ثانیه رسیده است حال با توجه به فرمول

$$Speedup_{overall} = \frac{Exec\ Time_{old}}{Exec\ Time_{new}}$$

داریم :

$$Speedup_{overall} = \frac{100s}{76s} = 1.31$$

پ) برای این مورد حال از ۱۰۰ ثانیه اجرای برنامه ضرب هیچ سهمی ندارد بنابراین برنامه در زمان ۴۰ ثانیه اجرا می شود که با توجه به فرمول گفته شده در قسمت قبل داریم :

$$Speedup_{overall} = \frac{100s}{40s} = 2.5$$

ت) برای این مورد می دانیم از ۱۰۰ ثانیه ، ۱۲/۵ ثانیه برای مقایسه و ۶۰ ثانیه برای ضرب و ۵.۲۷ ثانیه برای دستورات دیگر صرف می شود حال برای دو حالت داریم :

حالت ۱) ۳۶ ثانیه عمل ضرب (۴۰ درصد افزایش سرعت عملیات ضرب) و ۱۲/۵ ثانیه برای مقایسه و ۵.۲۷ برای دستورات دیگر بنابراین داریم :

$$Speedup_{overall} = \frac{100s}{36+12.5+27.5s} = \frac{100s}{76} = 1.31$$

حالت ۲) ۶۰ ثانیه برای ضرب و ۶/۲۵ ثانیه برای مقایسه و ۲۷/۵ ثانیه برای دستورات دیگر . بنابراین داریم :

$$Speedup_{overall} = \frac{100s}{60+6.25+27.5s} = \frac{100s}{93.75} = 1.06$$

همان طور که می بینیم با مقایسه مقدار دو حالت در میابیم که حالت ۱ (افزایش ۴۰ درصدی سرعت ضرب) زمان اجرا را کمتر می کند .

## سوال ۷ :

الف) برای این کار ابتدا باید داده را بر روی *BUS* قراردهیم که رجیسترها از روی آن داده را بردارند که کلاک اضافی نیاز نباشد . بنابراین خط ها به صورت زیر در می آید :

$$\begin{aligned} BUS &< -R1, R2 < -BUS, R4 < -BUS \\ BUS &< -R5, R3 < -BUS, R1 < -BUS \\ BUS &< -R6, R5 < -BUS \end{aligned}$$

ب) با توجه به الف می توان از ۳ باس استفاده کرده که هر کدام از خطوط دستور بالا بر روی یک باس انجام شود و در مجموع بتوان کل عملیات را بر روی ۳ باس و در یک کلاک انجام داد و با فرض این که در هر رجیستر بلافاصله از روی باس بردارد و بنابراین این کار قابل انجام است .

## سوال ۸ :

الف) با توجه به فرمول

$$CPU\ Time = CPU\ Clock\ Cycles \times CPU\ Cycle\ Time$$

و با توجه به این که

$$\begin{aligned} CPU\ Clock\ Cycles &= CPI_{ALU} \times IC_{ALU} + CPI_{Load} \times IC_{Load} + CPI_{Store} \times \\ &IC_{Store} + CPI_{Branch\ Jump} \times IC_{Branch\ Jump} \\ CPU\ Clock\ Time &= 0.5 \times 10^{-9} \end{aligned}$$

پس :

$$CPU\ Clock\ Cycles = 6 \times 10^9 + 8 \times 2.1 \times 10^8 + 7 \times 1.9 \times 10^8 + 5 \times 1.2 \times 10^8 = 60 \times 10^8 + 16.8 \times 10^8 + 13.3 \times 10^8 + 6 \times 10^8 = 96.1 \times 10^8$$

بنابراین

$$CPU\ Time = 96.1 \times 10^8 \times 0.5 \times 10^{-9} = 4.805s$$

ب) با توجه به تغییرات انجام شده مقادیر به شکل زیر می باشند :

$$\begin{aligned} IC_{ALU} &= 9 \times 10^8, \quad CPI_{ALU} = 6 \\ IC_{Load} &= 2.1 \times 10^8, \quad CPI_{Load} = 8 \\ IC_{Store} &= 1.9 \times 10^8, \quad CPI_{Store} = 6 \\ IC_{Branch\ Jump} &= 1.2 \times 10^8, \quad CPI_{Branch\ Jump} = 5 \end{aligned}$$

بنابراین :

$$CPU\ Clock\ Cycles = 54 \times 10^8 + 16.8 \times 10^8 + 11.4 \times 10^8 + 6 \times 10^8 = 88.2 \times 10^8$$

پس :

$$CPU\ Time_{new} = 88.2 \times 10^8 \times 0.5 \times 10^{-9} = 4.41s$$

و با توجه به فرمول  $SpeedUp$  داریم :

$$Speedup = \frac{Time_{original}}{Time_{improved}} = \frac{4.805}{4.41} \cong 1.09$$