

* 8 بیت داده هست، به خاطر همین به اندازه‌ی پهنای حافظه هست.

میانجی - کد: ۹۸۳۱۰۷۵ - استاد: دکتر زندی

Subject: متمرکز بررسی دوازدهم درس

معماری کامپیوتر

۱. مجموعه دستورالعمل‌های یک پردازنده ۱۴ بیتی در جدول زیر مشخص شده است. حافظه سه مانه یک کیلوریف دجیاتی است (طول هر ریف یا خانه از حافظه = دجیات).

* ISA	MOV <immediate 8 bit>	← مقدار لازم:
	ADD < " >	
	ADD <Memory Address>	
	SUB < register >	
	PUSH < " >	
	POP < " >	

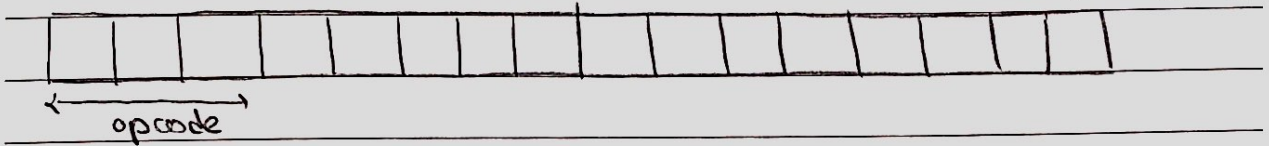
* تعداد بیت‌های خاص منظوره = به مقدار نیاز

* تعداد بیت‌های عام منظوره = ۴

اینجا

بیت‌های پردازنده: AX, BX, CX, DX

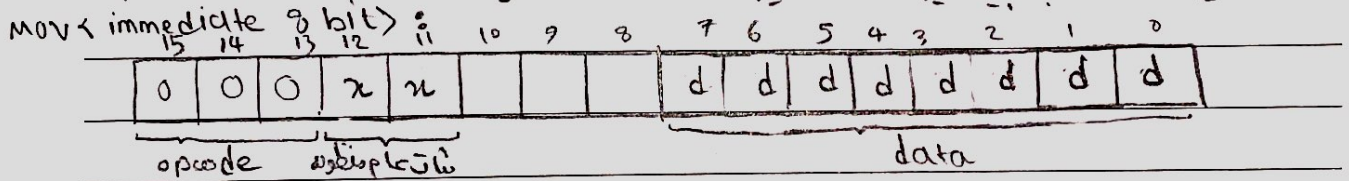
* حافظه: $2^{10} = 1.024$ ریف (۲ بیتی (۱۴ بیتی)
① برای کدب دستورالعمل:



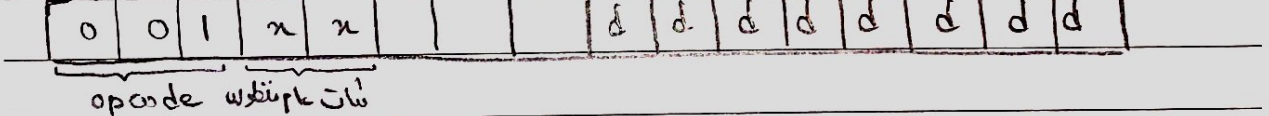
۴ بیت عام منظوره داریم ← نیاز به ۲ بیت داریم.

۴ دستور داریم ← نیاز به ۲ بیت داریم تا مشخص کنیم کدام دستورالعمل خواهم.

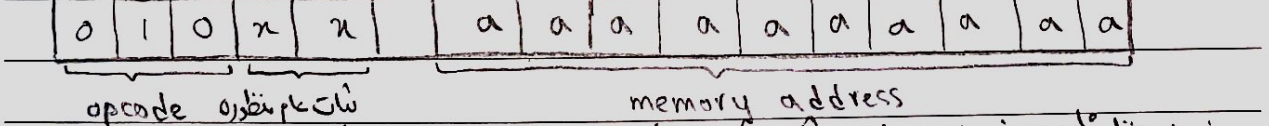
opcode = 3 bit



ADD <immediate 8 bit>:

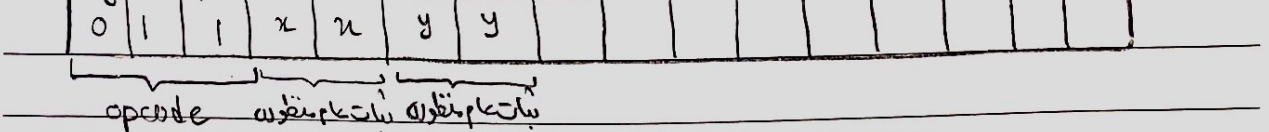


ADD <Memory Address>:



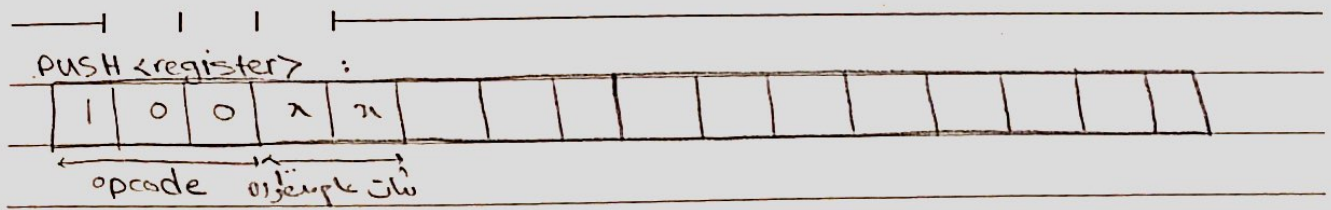
زیرا حافظه 2^{10} ریف دارد برای مشخص کردن کیلوریف نیاز به ۱۰ بیت داریم.

SUB <register>



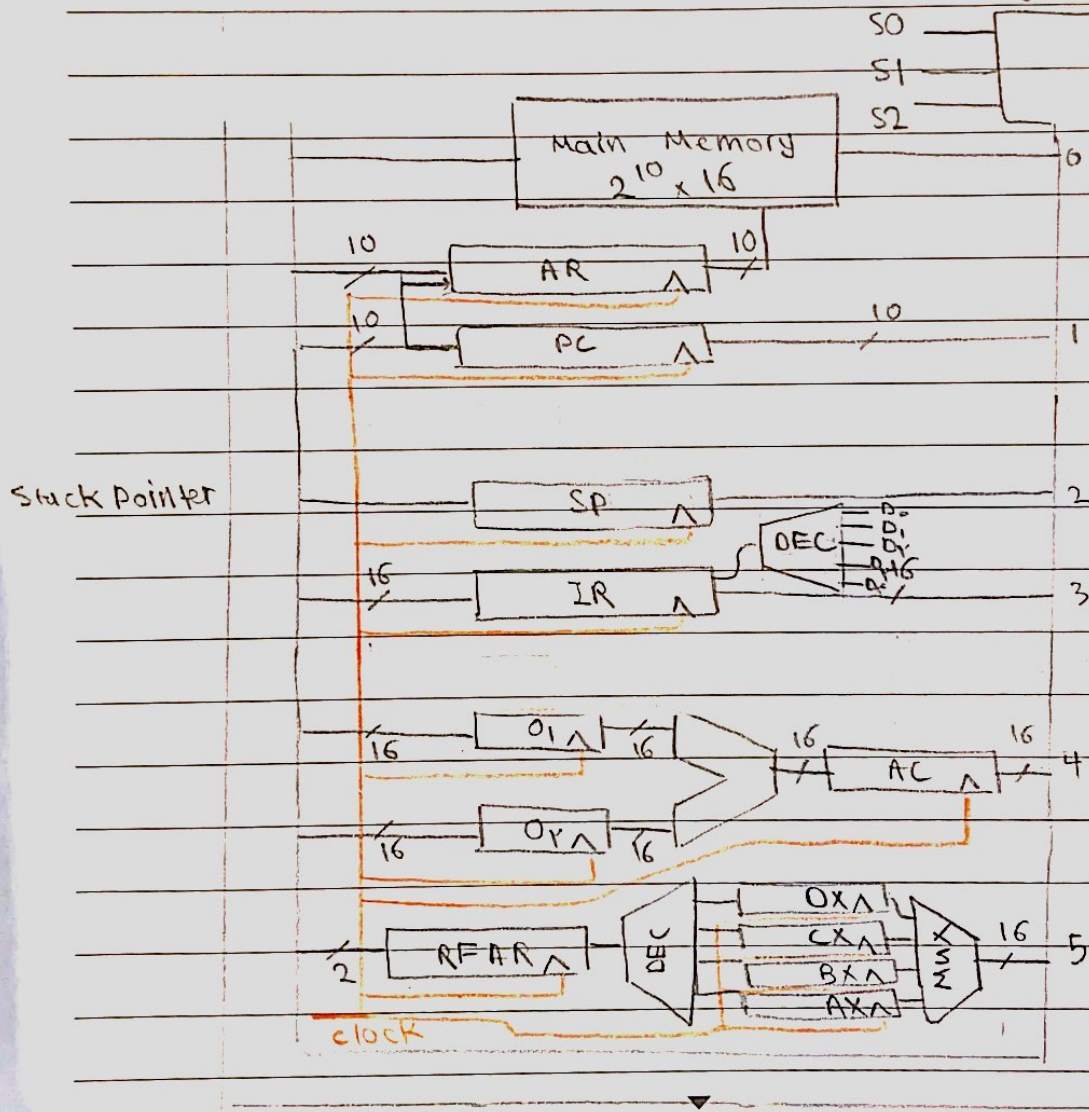
IDEA

Subject:



4- برای هر دستور 14 بیت در نظر می‌گیریم. عبارت‌های دستور POP، PUSH، SUB، برای حالت‌های خاص داریم ولی برای راحتی کار در fetch کردن همه دستورات را 14 بیت می‌گیریم و برای حالت PC هم به همان 14 بیت می‌افزاییم. $PC \leftarrow PC + 1$ می‌شود به همین خاطر برای ما می‌ماند 15 بیت.

: Data Path (2)



IDEA

دستور کارها با بیت‌های خاص منظور، جدولی برای آن می‌نویسیم استفاده کنیم درین temp * register =
 به عنوان مقادیر 16 بیت خاص منظور را می‌نویسیم. RFAR * =

* قبل از شروع کار CPU مقدار SP را مشخص کنیم: $SP \leftarrow (1111111111111111)_2$

Micro Operations (3)

$T_0 : AR \leftarrow PC$

1. Instruction Fetch

$T_1 : IR \leftarrow M[AR], PC \leftarrow PC + 1$

$T_2 : \text{Decode } IR[15:13] \text{ into } D_0 \text{ to } D_4$

2. Instruction Decode

به این ترتیب دستور را میخوانیم

دستورات (4)

MOV <immediate 8 bit>

$T_0.D_0 : RFER \leftarrow IR[12:11]$

ابتدا مشخص می‌کنیم کدام رجیستر

$T_1.D_0 : R[\underbrace{RFER}_{AX}] \leftarrow IR[7:0], SC \leftarrow 0$

مقدار را به آن رجیستر (در اینجا AX) می‌دهیم

ADD <immediate 8 bit>

$T_0.D_1 : RFER \leftarrow IR[12:11]$

ابتدایات را مشخص می‌کنیم که در AX است

$T_1.D_1 : O_1 \leftarrow R[\underbrace{RFER}_{AX}]$

چون جمع توسط ALU انجام می‌شود

$T_2.D_1 : O_2 \leftarrow IR[7:0]$

مقدار را در O_2 می‌نویسیم

$T_3.D_1 : AC \leftarrow O_1 + O_2$

$T_4.D_1 : R[\underbrace{RFER}_{AX}] \leftarrow AC, SC \leftarrow 0$

ADD <Memory Address>

$T_0.D_2 : AR \leftarrow IR[9:0]$

$T_1.D_2 : O_1 \leftarrow M[AR]$

$T_2.D_2 : RFER \leftarrow IR[12:11]$

$T_3.D_2 : O_2 \leftarrow R[\underbrace{RFER}_{AX}]$

$T_4.D_2 : AC \leftarrow O_1 + O_2$

$T_5.D_2 : R[\underbrace{RFER}_{AX}] \leftarrow AC, SC \leftarrow 0$

IDEA

Subject:

SUB < register >

$T_{P.DP} : RFER \leftarrow IR[10:9]$

$T_{F.DP} : OI \leftarrow RF[RFER]$

$T_{A.DP} : RFER \leftarrow IR[12:11]$

$T_{Y.DP} : OY \leftarrow \underbrace{AF[RFER]}_{AX}$

$T_{V.DP} : AC \leftarrow OI - OY$

$T_{A.DP} : \underbrace{RF[RFER]}_{AX} \leftarrow AC, SC \leftarrow 0$

PUSH < register >

$T_{P.DP} : RFER \leftarrow IR[12:11]$

$T_{F.DP} : AR \leftarrow SP$

$T_{A.DP} : M[AR] \leftarrow RF[RFER], sp = sp - 1, SC \leftarrow 0$

(will be decremented by 1 initially stack by 1)

POP < register >

$T_{P.DP} : RFER \leftarrow IR[12:11]$

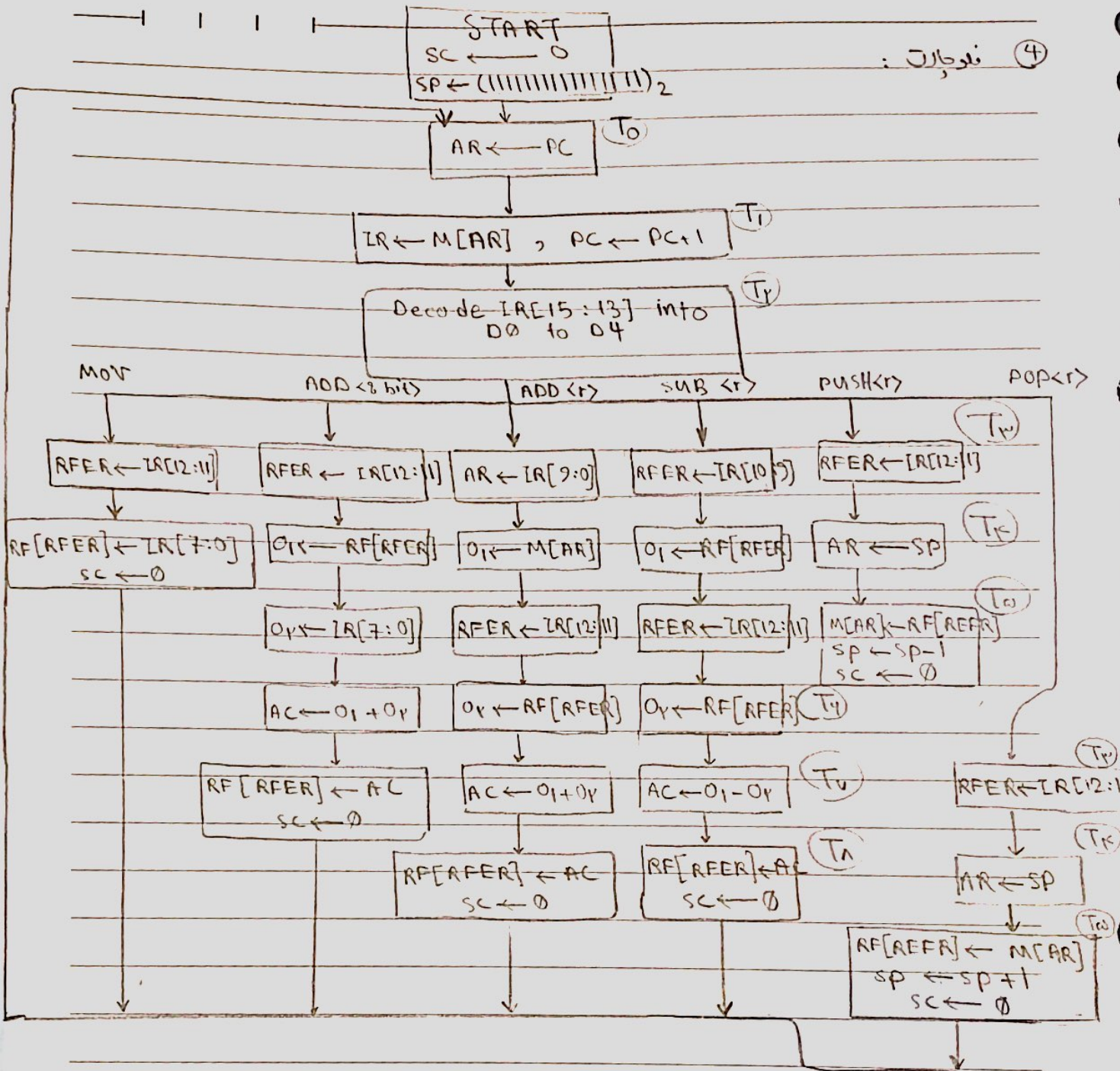
$T_{F.DP} : AR \leftarrow SP$

$T_{A.DP} : RF[RFER] \leftarrow M[AR], sp = sp + 1, SC \leftarrow 0$

IDEA

(K)

Subject:



(5) طولانی ترین مسیر : دستوران ADD <register> , SUB <register> طولانی ترین

مسیر با طول ۸ کلاک رادارند .

کوتاه ترین مسیر : دستور MOV <immediate 8 bit> / کوتاه ترین مسیر با طول

۴ کلاک رادارند

IDEA

(۵)

Subject:

Increment

⑥ مشخص کردن نامها (load, clear, Decrement, V) و طراحی cpu براساس شتابانها.

: m.m*

$$read = T_1 + T_f.D_r + T_w.D_w$$

: خواندن از حافظه

$$write = T_w.D_w$$

: نوشتن در حافظه

: load و clear *

$$L-AR = T_0 + T_p.D_r + T_f.D_f + T_t.D_w$$

: مربوط به آدرسهای رجیستر AR و عملیات

$$L-PC =$$

از آنجا که PC همیشه یک آدرس است و عملیات load براساس آن انجام میشود، مقدار PC در INC دارد.

$$L-IR = T_1$$

$$L-SP =$$

PC در load و مقدار PC در PC

$$L-O_1 = T_f.D_1 + T_f.D_r + T_f.D_p$$

$$L-O_r = T_w.D_1 + T_f.D_r + T_f.D_p$$

$$L-AC = T_f.D_1 + T_v.D_r + T_v.D_p$$

$$L-RFAR = T_p.D_0 + T_r.D_1 + T_w.D_r + T_p.D_p + T_p.D_f + T_p.D_w$$

: ALU *

$$ADD: T_f.D_1 + T_v.D_r$$

0

: در اینجا 0 یعنی ALU به plus

$$SUB: T_v.D_r$$

1

یعنی برای command 0 یعنی add و 1 یعنی sub

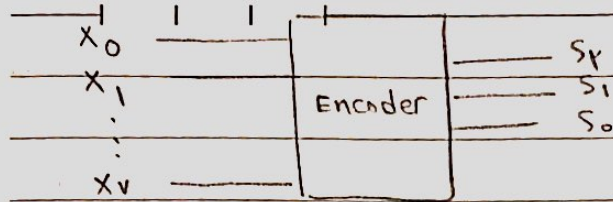
: MUX *

	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	select	mux	mux	mux
memory data out	0	0	0	0	0	0	0	1	\bar{s}_2	\bar{s}_1	s_0	
PC	0	0	0	0	0	0	1	0	0	0	1	
SP	0	0	0	0	0	1	0	0	0	1	0	
IR	0	0	0	0	1	0	0	0	0	1	1	
AC	0	0	0	1	0	0	0	0	1	0	0	
registers	0	0	1	0	0	0	0	0	1	0	1	

IDEA

(4)

Subject:



X_0 : X_0 (مخرج) $M[AR]$ (مخرج) $M[AR]$ (مخرج) $M[AR]$ (مخرج)

m.m $X_0 = T_1 + T_f.D_f + T_w.D_w$

PC $X_1 = T_0 + T_1$

SP $X_f = T_w.D_f + T_w.D_w$

IR $X_r = T_r.D_0 + T_f.D_0 + T_r.D_1 + T_w.D_1 + T_r.D_f + T_w.D_f + T_r.D_r + T_w.D_r + T_r.D_r + T_w.D_r$

AC $X_f = T_v.D_1 + T_A.D_f + T_A.D_r$

registers $X_w = T_f.D_1 + T_y.D_f + T_f.D_r + T_y.D_r + T_w.D_f$ (مخرج) $RF[REFR]$ (مخرج) $RF[REFR]$ (مخرج)

: Clear , Decrement , Increment (مخرج)

PC-Increment = T_1

SP-Increment = $T_w.D_w$

SP-decrement = $T_w.D_f$

sc-Clear = $T_f.D_0 + T_v.D_1 + T_A.D_f + T_A.D_r + T_w.D_f + T_w.D_w$

: register file (مخرج) MUX (مخرج) Decoder (مخرج)

Ren for mux = $T_f.D_1 + T_y.D_f + T_f.D_r + T_y.D_r + T_w.D_f$

مخرج $RF[REFR]$ (مخرج) $RF[REFR]$ (مخرج)

Wen for decoder = $T_f.D_0 + T_A.D_f + T_A.D_r + T_w.D_w$

مخرج $RF[REFR]$ (مخرج) $RF[REFR]$ (مخرج)

IDEA

(v)

Subject:

۲. قطعه برنامه‌ای مستقل از حلقه‌ای است که ۱۰ دستوردار دایره حلقه ۵۰ بار تکرار می‌شود. اگر فرکانس ساعت کامپیوتر ۱ میلیارد هرگز دستوردار باشد برای اجرای هر دستور ۱.۲۵ نایس باشد، آن‌گاه

$$MIPS = \frac{\# \text{ instructions}}{10^4 \times \# \text{ seconds}} = \frac{IPC \times f \times 10^{-4}}{CPI} = \frac{1}{CPI} \times f \times 10^{-4}$$

$$f = 1 \times 10^9$$

$$CPI = \frac{\# \text{ clocks}}{\# \text{ Instructions}} \rightarrow CPI = 1.25$$

$$MIPS = \frac{1}{1.25} \times 10^9 \times 10^{-4} = 80000$$

۳. اگر منظور از میلیون، در زبان فارسی باشد، 10^6 می‌شود و جواب همین است. ولی اگر میلیون در زبان انگلیسی (بر آمریکا) منظور باشد، 10^9 است و جواب 80000 می‌شود.

$$80000 = \frac{1}{1.25} \times 10^9 \times 10^{-4}$$

۳. یک پردازنده دارای ۴ گروه دستوردار است و فرکانس کاری ۴۰ میلیون هرگز است. تعداد نایس‌های لازم برای اجرای هر دستور در جدول زیر داده شده است. میخواهیم کاری این پردازنده را برای یک برنامه که در هر دستورهای هرگز آن در جدول زیر آمده است ارزیابی کنیم. مقدار MIPS این پردازنده چقدر است؟

تعداد دستورهای برنامه	تعداد نایس لازم برای اجرا	گروه دستورها
۴	۱	A
۷	۳	B
۳	۱	C
۱۰	۲	D

$$MIPS = \frac{\# \text{ instructions}}{10^4 \times \# \text{ seconds}} = \frac{IPC \times f \times 10^{-4}}{CPI} = \frac{1}{CPI} \times f \times 10^{-4}$$

$$CPI = \frac{\# \text{ clocks}}{\# \text{ instructions}} \rightarrow \text{تعداد کل نایس لازم} = 0.4 \times 1 \times n + 0.2 \times 3 \times n + 0.3 \times 1 \times n$$

$$= (0.4 + 0.6 + 0.3) n = 1.3 n \rightarrow CPI = \frac{1.3 n}{n} = 1.3$$

$$MIPS = \frac{1}{1.3} \times 40 \times 10^6 \times 10^{-4} = 30769$$

IDEA