

# Arch-Computer

سوال اول

الف) کدها را متناسب با (op) به فورم مناسب در می آوریم و معادل هر کد را در زبان Assembly پیدا می کنیم و می نویسیم.

E3A0100C = 1110 0011 1010 0000 0001 0000 0000 1100

به فورم Data processing می نویسیم:

$$\begin{array}{ccccccc} \text{1110} & \text{00} & \text{111010} & \text{0000} & \text{0001} & \text{0000} & \text{0000} & \text{1100} \\ \text{cond} & \text{op} & \text{i cmd s} & \text{Rn} & \text{Rd} & \text{rot} & \text{imm8} & \text{src2} \end{array}$$

معادل اسمبلی که بالا را براساس بایه

Mov R19, #12

همین طور برای مابقی کدها عمل می کنیم.

E3A0200F = 1110 0011 1010 0000 0010 0000 0000 1111

$$\begin{array}{ccccccc} \text{1110} & \text{0011} & \text{1010} & \text{0000} & \text{0010} & \text{0000} & \text{0000} & \text{1111} \\ \text{cond} & \text{op i} & \text{cmd s} & \text{Rn} & \text{Rd} & \text{shamt5} & \text{sh} & \text{Rm} \end{array}$$

Mov R2, #15

E1520001 = 1110 0001 0101 0010 0000 0000 0000 0001

$$\begin{array}{ccccccc} \text{1110} & \text{0001} & \text{0101} & \text{0010} & \text{0000} & \text{0000} & \text{0000} & \text{0001} \\ \text{cond} & \text{op i} & \text{cmd s} & \text{Rn} & \text{Rd} & \text{shamt5} & \text{sh} & \text{Rm} \end{array}$$

CMP5 R0, R2, R1

$0422001 = \underbrace{1100}_{cond} \underbrace{0000 \ 0100}_{opi \ cmds} \underbrace{0010}_{Rn} \underbrace{0010}_{Rd} \underbrace{\overset{shamt5}{0000} \overset{sh}{0000} \overset{Rm}{0001}}_{src2}$

SUBGT  $R_2, R_2, R_1$

$B0411002 = \underbrace{1011}_{cond} \underbrace{0000 \ 0100}_{opi \ cmds} \underbrace{0001}_{Rd} \underbrace{0001}_{Rn} \underbrace{\overset{shamt5}{0000} \overset{sh}{0000} \overset{Rm}{0010}}_{src2}$

SUBLT  $R_1, R_1, R_2$

$1AFFFFFFB = \underbrace{0001}_{cond} \underbrace{1010}_{op \ func} \underbrace{1111 \ 1111 \ 1111 \ 1111 \ 1111 \ 1011}_{imm24}$

BNE Flag

ماشین کسی بهای Branch not Equal به آدرس (imm24) است.

Flag در آدرس  $(PC+8)+imm24$  قرار دارد چون  $imm24 = -5$  است پس

از  $PC+8$  باید پنج instruction به عقب برویم.



$$E3A0000A = \underbrace{1110}_{\text{cond}} \underbrace{0011}_{\text{opi}} \underbrace{1010}_{\text{cmd s}} \underbrace{0000}_{R_N} \underbrace{0000}_{R_D} \underbrace{0000 \ 0000 \ 1010}_{\text{src2} \quad \text{imm8}}$$

(2✓)

MOV R0, #10

$$E3A01003 = \underbrace{1110}_{\text{cond}} \underbrace{0011}_{\text{opi}} \underbrace{1010}_{\text{cmd s}} \underbrace{0000}_{R_N} \underbrace{0001}_{R_D} \underbrace{0000 \ 0000 \ 0011}_{\text{src2} \quad \text{rot} \quad \text{imm8}}$$

MOV R1, #3

$$E1500001 = \underbrace{1110}_{\text{cond}} \underbrace{0001}_{\text{opi}} \underbrace{0101}_{\text{cmd s}} \underbrace{0000}_{R_N} \underbrace{0000}_{R_D} \underbrace{0000 \ 0000 \ 0001}_{\text{src2} \quad \text{shamt5} \quad \text{sh} \quad R_m}$$

CMPS R0, R1

$$0A000005 = \underbrace{0000}_{\text{cond}} \underbrace{1010}_{\text{op func}} \underbrace{0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0101}_{\text{imm24}}$$

BEQ FLAG1

$$CA000000 = \underbrace{1100}_{\text{cond}} \underbrace{1010}_{\text{op func}} \underbrace{0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000}_{\text{imm24}}$$

BGT FLAG2

$$EA000001 = \underbrace{1110}_{\text{cond}} \underbrace{1010}_{\text{op func}} \underbrace{0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0001}_{\text{imm24}}$$

B FLAG3

$$E0400001 = \underbrace{1110}_{\text{cond}} \underbrace{0000}_{\text{opi}} \underbrace{0100}_{\text{cmd s}} \underbrace{0000}_{R_N} \underbrace{0000}_{R_d} \underbrace{\overbrace{0000}^{\text{shamt 5}} \overbrace{0000}^{\text{sh}} \overbrace{0001}^{R_m}}_{\text{src 2}}$$

SUB  $R_0, R_0, R_7$

$$EFFFFFF9 = \underbrace{1110}_{\text{cond}} \underbrace{1010}_{\text{op func}} \underbrace{1111 \ 1111 \ 1111 \ 1111 \ 1111 \ 1111}_{\text{imm 24}} \underbrace{1001}_{R_m}$$

B Flag 4

$$E0411000 = \underbrace{1110}_{\text{cond}} \underbrace{0000}_{\text{opi}} \underbrace{0100}_{\text{cmd s}} \underbrace{0001}_{R_N} \underbrace{0001}_{R_d} \underbrace{\overbrace{0000}^{\text{shamt 5}} \overbrace{0000}^{\text{sh}} \overbrace{0000}^{R_m}}_{\text{src 2}}$$

SUB  $R_7, R_7, R_0$

$$EFFFFFF2 = \underbrace{1110}_{\text{cond}} \underbrace{1010}_{B \text{ func}} \underbrace{1111 \ 1111 \ 1111 \ 1111 \ 1111 \ 1111}_{\text{imm 24}} \underbrace{0111}_{R_m}$$

B Flag 4

$$E1A00000 = \underbrace{1110}_{\text{cond}} \underbrace{0001}_{\text{opi}} \underbrace{1010}_{\text{cmd s}} \underbrace{0000}_{R_N} \underbrace{0000}_{R_d} \underbrace{\overbrace{0000}^{\text{shamt 5}} \overbrace{0000}^{\text{sh}} \overbrace{0000}^{R_m}}_{\text{src 2}}$$

MOV  $R_0, R_0, \text{LSL} \#0$



(کد اول)

mov R<sub>1</sub>, #12

mov R<sub>2</sub>, #5

Flag:

cmps R<sub>2</sub>, R<sub>1</sub>

subgt R<sub>0</sub>, R<sub>2</sub>, R<sub>1</sub>

sublt R<sub>1</sub>, R<sub>1</sub>, R<sub>2</sub>

bne Flag

(کد دوم)

mov R<sub>0</sub>, #10

mov R<sub>1</sub>, #3

Flag4:

cmps R<sub>0</sub>, R<sub>1</sub>

beq Flag1

bgt Flag2

b Flag3

Flag2:

sub R<sub>0</sub>, R<sub>0</sub>, R<sub>1</sub>

b Flag4

Flag3:

sub R<sub>1</sub>, R<sub>1</sub>, R<sub>0</sub>

b Flag4

Flag1:

mov R<sub>0</sub>, R<sub>0</sub>, lsl #0

(ب)

در هر دو کد اینتر در مقایسه مقدار دهی می شود سپس آن در مقدار مقایسه

می شوند و Flag آن ثبت می شود بعد در کد اول اگر  $(R_2 > R_1)$  آن گاه

$R_2 = R_2 - R_1$  حساب می شود و اگر  $(R_2 < R_1)$  آن گاه  $R_1 = R_1 - R_2$  حساب می شود

در صورت عدم تساوی  $R_1, R_2$  به Flag با آدرس  $(PC+8) + imm24$  می رویم

در کد دوم بعد از مقایسه  $R_0, R_1$  اگر برابر باشند به Flag1 و اگر بزرگتر باشد به Flag2 و در غیر

این صورت به Flag3 در Flag2  $R_0 = R_0 - R_1$  را حساب کرده و به Flag4 می گردیم

تا دوباره  $R_0$  را با  $R_1$  مقایسه کنیم، در Flag 3  
می کنیم و ب Flag 4 پس گردیم تا  $R_0$  و  $R_1$  را با هم مقایسه کنیم و ادامه کار...  
در Flag 1  $R_0$  را مقدار دهی می کنیم.