

if (a > b) { x = 5; y = c + d; } else x = c - d;

CMP R0, R1;
 $\downarrow \quad \downarrow$
 a b
 \downarrow
 if (R0 > R1)

(a-b) $\begin{cases} \text{if } a < b & \rightarrow \text{N Flag} = 1 \\ \text{if } a = b & \rightarrow \text{Z Flag} = 1 \end{cases}$

Loops: C;

for (i = 0, P = 0; i < N; i++)
 P = P + C[i] * x[i];

Assembler:

LDR R3, =C \rightarrow R3 carries the address of C[0] (base element)
 LDR R5, =x \rightarrow R5 // // // x[0] // //
 MOV R0, #0 \rightarrow move the constant (0) to register R0 (R0 = 0)

To set:

x |= 0x03; $\xrightarrow{\text{بعد ذلك}} (x \text{ OR } 0000\ 0003) \rightarrow$ النتيجة في الـ x

LDR R0, =x
 LDR R1, [R0]
 ORR R1, R1, #0x03 $\rightarrow x = x \text{ OR } 0x03$
 STR R1, [R0] \rightarrow stores the result in x

This instruction sets
 the bits 0 and 1
 to 1

To Toggle (ااكس ال bit)

$x \wedge 0x80;$ \rightarrow toggles bit no. 7 in x
 \downarrow
XOR

EOR R1, R1, #0x80 \rightarrow toggle

Shift operations

(LSR) Logical shift Right ينزق ال bits لليمين عدد معين من المرات
وهبط 0

(ASR) Arithmetic shift Right sign bit وفي الآخر اسبب ال
في ما هي (يحتفظ بإشارة الرقم)

(LSL) Logical shift left ينزق ال bits للشمال ونظ صفر على اليمين
(نظ الإشارة في bit لوصها ونحتفظ بيها)

(ROR) Rotate Shift Right يعمل rotation من غير ال carry (بشيل آخر bits
على اليمين واحطها في الشمال)

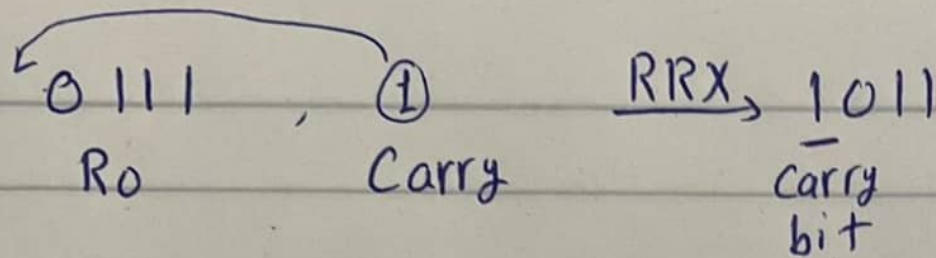
(RRX) Rotate Right Extended يرجع ال carry من العملية الى قبل كدة
bit

في ال LSR ، الرقم الى بيتشال في الآخر على الشمال بيتظ في ال carry وبيطع برة لكن في ال ROR
ننا عايز ارجع ال bit لي واحطها على اليمين الشمال.

exi-

0111 $\xrightarrow{\text{LSR}}$ 0011

0111 $\xrightarrow{\text{ROR}}$ 1011



لوسى register فيه عدد من البتات
وسى carry bit يبقى @ دى ال
RRX ال

The stack:

BX LR \rightarrow return to a certain instruction address (Link Register)