المحاضرة 4 عملي معالجات صغرية – ميكاترونيكس

المثال 1 :

لديك البرنامج التالي:

البرنامج	شرح
.data	انتقل الى ذاكرة المعطيات
.word 3, 10, 7, 4, -9	تخزين كلمات (ممثلة عشرياً) في الذاكرة
.text	انتقل الى ذاكرة التعليمات
lui $s0, 0x1001	تحميل القيمة 0x1001 إلى النصف العلوي من المسجل $s0
ori $a0, $0, 5	تعليمة ori الغرض منها هنا هو وضع قيمة 5 في المسجل $a0
Loop: lw $t0, 0($s0)	تحميل قيمة عنوان الذاكرة الموجود في $s0 بإزاحة صفرية, إلى المسجل t0
addi $t0, $t0, 2	
sw $t0, 20($s0)	تخزين قيمة t0 في الذاكرة في العنوان المحتوى في s0 بإزاحة 20
addi $s0, $s0, 4	
addi $a1, $a1, 1	
beq $a1, $a0, exit	في حال تساوي قيمة المسجلين a1 وa0 اقفز إلى اللافتة :Exit
J loop	اقفز إلى اللافتة Loop
Exit:	

والمطلوب:

1. ما هو تمثيل الذاكرة قبل تنفيذ تعليمات البرنامج.
2. أنشيء جدول لتتبع قيم المسجلات داخل الحلقة أثناء تنفيذ الكود.
3. ما هو تمثيل الذاكرة بعد تنفيذ تعليمات البرنامج.
4. ما هي وظيفة البرنامج؟؟

الحل :

1. تمثيل الذاكرة قبل تنفيذ تعليمات البرنامج:

ملاحظة: يفضل تحويل القيم العشرية إلى ست عشرية قبل وضعها في تمثيل الذاكرة.

	+0	+4	+8	+c
0x1001000	0x3	0xa	0x7	0x4
0x1001010	0xfffffff7			

2. قيم المسجلات:

القيم الابتدائية قبل دخول الحلقة:

$s0= 0x10010000 **$a0=0x00000005**

$a1=0x0 **$t0=0x0**

Round	1	2	3	4	5
$t0	0x5	0xc	0x9	0x6	0xfffffff9
$s0	0x10010004	0x10010008	0x1001000c	0x10010010	0x10010014
$a1	0x1	0x2	0x3	0x4	0x5 (**Exit**)

ملاحظة: سجلنا في الجدول قيمة كل مسجل في نهاية كل لفة من لفات الحلقة.

3. تمثيل الذاكرة بعد تنفيذ البرنامج:

	+0	+4	+8	+c
0x1001000	0x3	0xa	0x7	0x4
0x1001010	0xfffffff7	0x5	0xc	0x9
0x1001020	0x6	0xfffffff9		

4. وظيفة البرنامج هو المرور على أول خمس عناصر في الذاكرة($a1=5), وإضافة القيمة 2 إلى قيمة كل منها ثم تخزين الناتج في موقع الذاكرة الذي يبعد 5 كلمات (20/4=5 words) عن موقع العنصر الحالي.

المثال 2:

لديك البرنامج التالي:

```
        .data
        .word  16, 13, 20, 17, 9
        .text
        lui $s6, 0x1001
Loop:   lw $t0, 0($s6)
        beq $t0, $0, mm
        addi $s5, $s5, 1
        add $s0, $s0, $t0
        addi $s6, $s6, 4
        J   loop
mm:
        sw $s0, 0($s6)
        sw $s5, 4($s6)
```

والمطلوب:

1. ما هو تمثيل الذاكرة قبل تنفيذ تعليمات البرنامج.
2. أنشيء جدول لتتبع قيم المسجلات داخل الحلقة أثناء تنفيذ الكود.
3. ما هو تمثيل الذاكرة بعد تنفيذ تعليمات البرنامج.
4. برأيك ما هي وظيفة البرنامج؟؟
5. ماهي التعديلات الواجب إجرائها على البرنامج في حال التعامل مع مصفوفة أنصاف كلمات half-word؟
6. ماهي التعديلات الواجب إجرائها على البرنامج في حال التعامل مع مصفوفة بايتات bytes؟

الحل:

1. تمثيل الذاكرة :

	+0	+4	+8	+c
0x1001000	0x10	0xd	0x14	0x11
0x1001010	0x9	0x0	0x0	0x0

2. القيم الابتدائية للمسجلات :

$s6= 0x10010000 **$s5=0x0**

جدول تتبع قيم المسجلات :

Round	1	2	3	4	5	6
$t0	0x10	0xd	0x14	0x11	0x9	0x0(**mm:**)
$s5	0x1	0x2	0x3	0x4	0x5	-----
$s0	0x10	0x1d	0x31	0x42	0x4b	----
$s6	0x10010004	0x10010008	0x1001000c	0x10010010	**0x10010014**	---

3. تمثيل الذاكرة بعد تنفيذ كامل البرنامج :

	+0	+4	+8	+c
0x1001000	**0x10**	**0xd**	**0x14**	**0x11**
0x1001010	**0x9**	**0x4b**	**0x5**	**0x0**

4. وظيفة البرنامج هو المرور على عناصر المصفوفة (قيم عناصر الذاكرة) وحساب مجموعها وعددها، ثم تخزين الناتجين بعد المصفوفة مباشرة في الذاكرة. وتتوقف الحلقة عند الوصول إلى قيمة صفرية (شرط الإنهاء $t0==0).
5. لكي يتعامل البرنامج مع أنصاف كلمات يجب إجراء التعديلات التالية:

قبل	بعد التعديل
الموجه .word	.half
التعليمة lw $t0, 0($s6)	lh $t0, 0($s6)
التعليمة addi $s6, $s6, **4**	addi $s6, $s6, **2**
التعليمة sw $s0, 0($s6)	sh $s0, 0($s6)
التعليمة sw $s5, **4**($s6)	sh $s5, **2**($s6)

6. لكي يتعامل البرنامج مع بايتات (ربع كلمة) يجب إجراء التعديلات التالية:

قبل	بعد التعديل
الموجه .word	.byte
التعليمة lw $t0, 0($s6)	lb $t0, 0($s6)
التعليمة addi $s6, $s6, **4**	addi $s6, $s6, **1**
التعليمة sw $s0, 0($s6)	sb $s0, 0($s6)
التعليمة sw $s5, **4**($s6)	sb $s5, **1**($s6)

ملاحظة هامة:

في الحالة العامة إذا كان المطلوب هو تحويل البرنامج ليتعامل مع أنصاف كلمات أو بايتات، نستخدم (lh أو lb) بدلاً من lw , كما فعلنا في مثالنا السابق.

أما في حالة خاصة إذا كان المطلوب هو تحويل البرنامج ليتعامل مع أنصاف كلمات أو بايتات **مع اعتبار** أن الأعداد المخزنة في الذاكرة هي غير مؤشرة (unsigned) (أي اعتبارها جميعها موجبة) حينها فقط نستخدم lbu أو lhu بدلاً من lw