جامعة المنارة الخاصة
كلـــــية الهندســـــة
هندسة ميـــكاترونيك

المعالجات الصغرية ولغة التجميع

المحاضرة الثالثة

مدرس المقرر
د. بسام حسن

2025_2026

مفردات من المحاضرة الثالثة :

- الذاكرة في معالجات MIPS
- التعامل مع ذاكرة المعطيات وذاكرة البرامج
- البرامج المكتوبة بلغة الأسمبلي

الذاكرة في معالجات MIPS

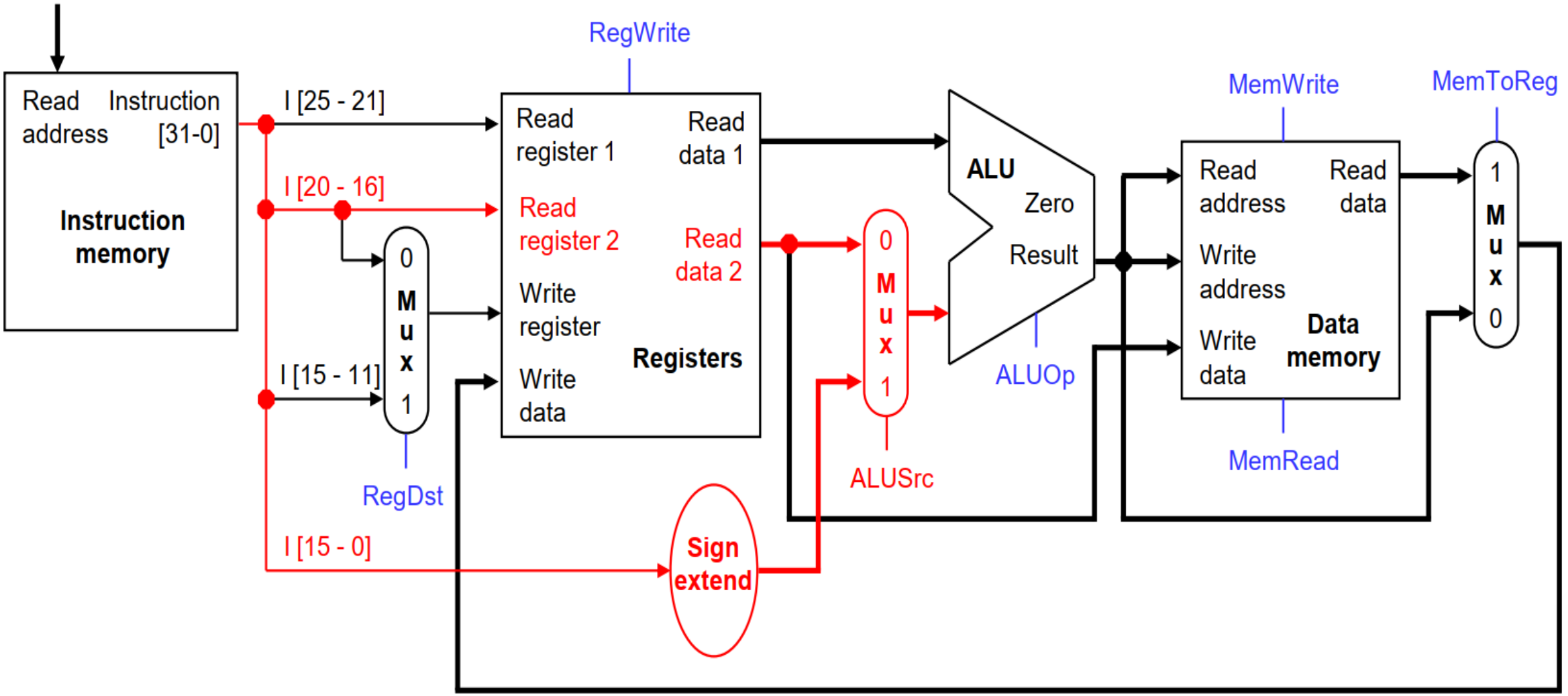
تقسم الذاكرة التي تحويها معالجات MIPS إلى قسمين رئيسيين:

ذاكرة المعطيات Data-memory : والتي يتم فيها تخزين قيم عددية ومعطيات معينة تلزمنا من أجل الكود.

وذاكرة التعليمات Instruction-memory: التي يتم فيها تخزين تعليمات الكود المكتوب

Instruction memory	Data memory

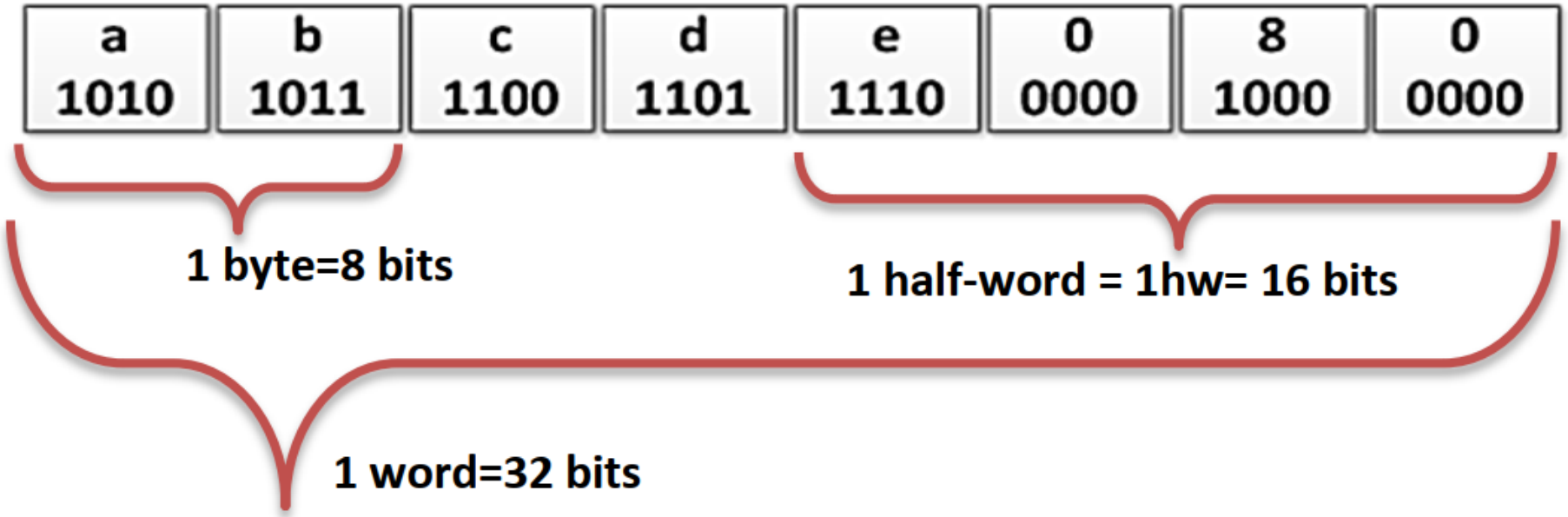
مخطط المعالج



الذاكرة في معالجات MIPS

اضافة القيم الجديدة إلى data memory

- يتم ذلك من خلال استخدام التعليمة **word.** والتي تعني إضافة كلمة جديدة إلى الذاكرة (إضافة قيمة جديدة)
- هذه الكلمة يجب ان تكون مؤلفة من 32 bit وبالتالي يجب ان تكون مؤلفة من 32/4=8 ثمان خانات ست عشرية (كون كل خانة ست عشرية تمثل بأربع خانات ثنائية) **وبالتالي حجم كل كلمة هو 4byte ... علل ذلك!!!**
- لاحظ اننا أضفنا الكلمة الجديدة abcde080 وهي كما تلاحظ مؤلفة من ثمان خانات ست عشرية ولذا نستخدم الدلالة 0x لندل انها قيمة من النظام الست عشري فتصبح التعليمة word 0xabcde080.



الذاكرة في معالجات MIPS

اضافة القيم الجديدة إلى data memory

- **يتم حفظ القيم في ذاكرة المعطيات بدأ من العنوان 0x10010000** (لاحظ ان العنوان قيمة ست عشرية وليست ثنائية... في حال أردت تحويله إلى قيمة ثنائية سيكون مؤلف من 32bits حاول ذلك بنفسك)
- لكل خانة من خانات الذاكرة عنوان محدد تعرف به ويمكننا الوصول إليها باستخدام هذا العنوان ... بذلك تكون القيمة 0xabcde080 قد تم تخزينها في خانة من الذاكرة ذات العنوان 0x10010000
- ماذا لو أردنا إضافة قيمة جديدة إلى ذاكرة البيانات؟؟؟؟ كيف سيتم ذلك؟؟؟ وفي أي موقع سيتم تخزين القيمة الجديدة؟؟؟
- في المثال الأخير يمكنك التعرف على كيفية إضافة قيم أكثر إلى ذاكرة البيانات.
- في حال أضفنا قيمة جديدة إلى الذاكرة ستخزن في الخانة ذات العنوان 0x10010004 علل ذلك!!! سبب ذلك هو كون كل كلمة حجمها 4byte بالتالي 0x10010000+4=0x10010004
- في حال أضفنا قيمة ثالثة ستخزن في العنوان 0x10010008 **علل ذلك!!**
- والتي تليها ستخزن في الخانة ذات العنوان 0x1001000C **علل ذلك!!**

كود الأسمبلي في MIPS

التعامل مع ذاكرة المعطيات وذاكرة البرامج

جَامعة المَنارة

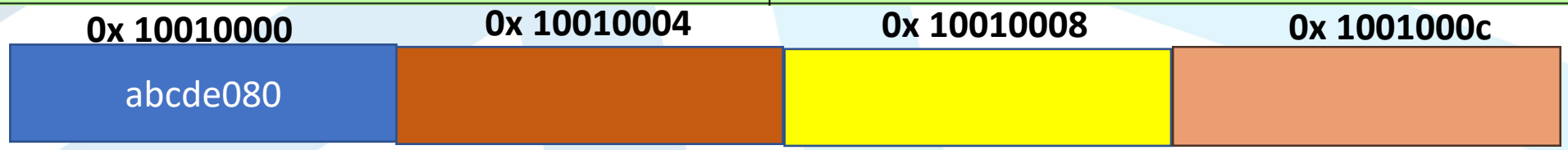
هنا سنتعامل مع ذاكرة المعطيات

هنا سنتعامل مع ذاكرة التعليمات

تعليمات البرنامج

. data	موجه يدل على بداية القسم الخاص بذاكرة المعطيات
.word 0xabcde080	موجه لتخزين الكلمة 0xabcde080 في أول عنوان في ذاكرة المعطيات أي بدءاً من العنوان 0x10010000
.text	موجه يدل على بداية القسم الخاص بذاكرة التعليمات التي تبدأ بالعنوان 0x00400000
lui $a0,0x1001	وضع القيمة 0x1001 في النصف الأعلى من المسجل $a0
lw $t0,0($a0)	تحميل **كلمة كاملة** من عنوان الذاكرة الموجود في المسجل $a0 تماماً (بإزاحة صفرية) ووضع قيمة الكلمة في المسجل $t0

تمثيل الذاكرة



ناتج التنفيذ

$a0= 0x 10010000

$t0= 0x abcde080

جَـامعة المَـنارة
MANARA UNIVERSITY

لدينا الكود التالي المكتوب بلغة الأسمبلي

```
.data
.word 10,10,10,9,0,2
.text
 lui $s6,0x1001
 lui $s5,0x0000
 ori $s5,$s5,10
 loop: sll $t1,$s3,2
 add $t1,$t1,$s6
 lw $t0,0($t1)
 bne $t0,$s5,exit
 addi $s3,$s3,1
 j loop
exit:
```

المطلوب :

١. أوجد تمثيل الذاكرة الناتج عن تنفيذ الكود السابق.
٢. أنشيء جدول لتتبع قيم المسجلات داخل الحلقة أثناء تنفيذ الكود.
٣. ما هي وظيفة الكود؟

```
.data
.word 10,10,10,9,0,2
.text
 lui $s6,0x1001
 lui $s5,0x0000
 ori $s5,$s5,10
 loop: sll $t1,$s3,2
 add $t1,$t1,$s6
 lw $t0,0($t1)
 bne $t0,$s5,exit
 addi $s3,$s3,1
 j loop
exit:
```

لدينا الكود التالي المكتوب بلغة الأسمبلي

الحل:

1. تمثيل الذاكرة :

	+0	+4	+8	+c
0x1001000	0xa	0xa	0xa	0x9
0x1001010	0x0	0x2		

جَـامعة المَـنارة
MANARA UNIVERSITY

```
.data
.word 10,10,10,9,0,2
.text
  lui $s6,0x1001
  lui $s5,0x0000
  ori $s5,$s5,10
  loop: sll $t1,$s3,2
  add $t1,$t1,$s6
  lw $t0,0($t1)
  bne $t0,$s5,exit
  addi $s3,$s3,1
  j loop
exit:
```

الحل:

1. تمثيل الذاكرة :

	+0	+4	+8	+c
0x1001000	0xa	0xa	0xa	0x9
0x1001010	0x0	0x2		

2. القيم الابتدائية للمسجلات خارج الحلقة:

$s6=0x10010000 **$s5=0x0000000a**

جدول تتبع قيم المسجلات من داخل الحلقة حتى نهاية البرنامج:

Round:	1	2	3	4
$t1	0x1001000	0x10010004	0x10010008	0x1001000c
$t0	0xa	0xa	0xa	0x9 **(exit)**
$s3	0x1	0x2	0x3	---------

الحل:

```
.data
.word 10,10,10,9,0,2
.text
 lui $s6,0x1001
 lui $s5,0x0000
 ori $s5,$s5,10
 loop: sll $t1,$s3,2
 add $t1,$t1,$s6
 lw $t0,0($t1)
 bne $t0,$s5,exit
 addi $s3,$s3,1
 j loop
exit:
```

تم انهاء البرنامج (القفز إلى اللافتة exit) في الدورة الرابعة للحلقة عندما أصبحت قيمة t1 لا تساوي 10 أي عندما $\$t1 \neq \$s5$.

3. وظيفة الكود هي المرور على عناصر المصفوفة عنصر عنصر بالترتيب والانهاء عندما تكون قيمة العنصر لا تساوي 10.

جَـامعة
المَـنارة
MANARA UNIVERSITY

نـهايـة المحـاضرة الثـالثـة