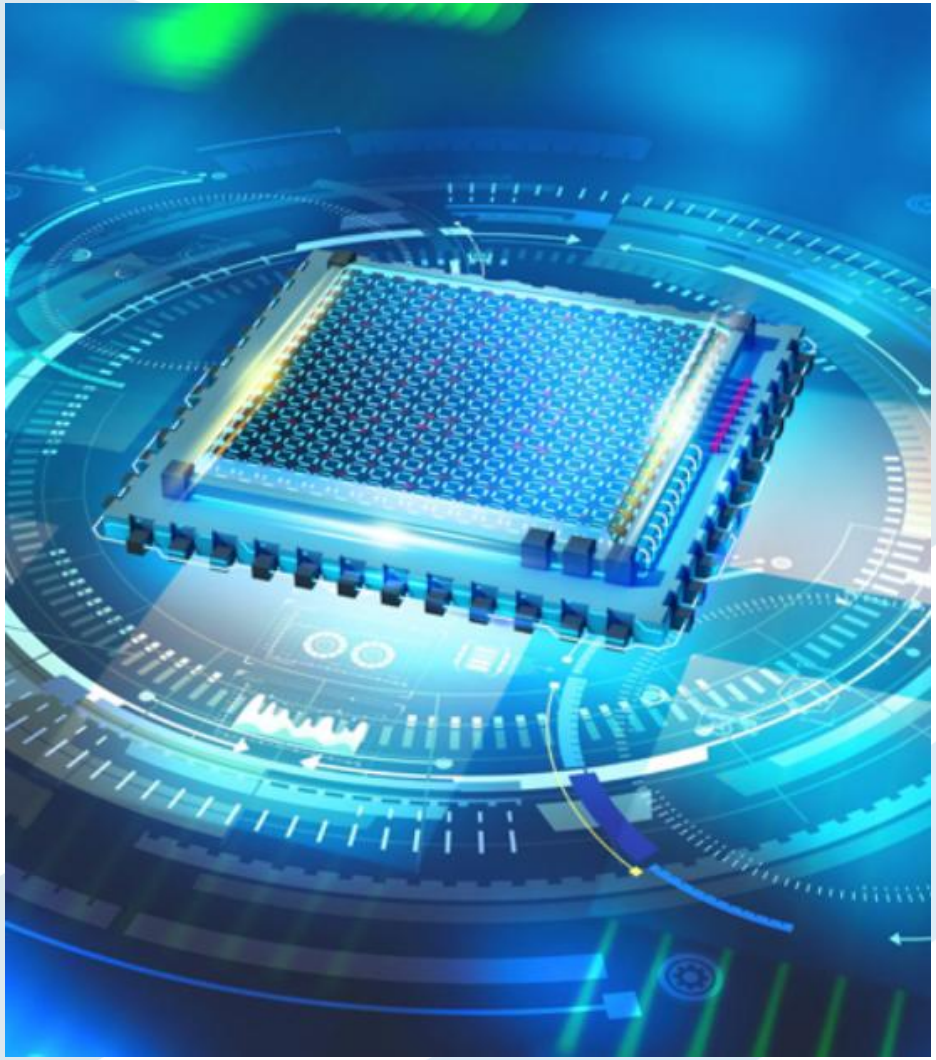
جامعة المنارة الخاصة
كلـــــية الهندســـــة
هندسة ميـــكاترونيك

المعالجات الصغرية ولغة التجميع

المحاضرة السابعة

مدرس المقرر
د. بسام حسن

2025_2026



مفردات من المحاضرة السابعة :

- The Processor: Control Design
- وحدة التحكم بوحدة الحساب والمنطق
- تنفيذ التعليمات من النوع_ R-type
- تنفيذ التعليمات SW , LW
- تنفيذ تعليمات القفز المشروط

- تصميم التحكم
  - الهدف: التحكم بـ Datapath لتنفيذ العملية المطلوبة
  - وحدة التحكم
    - مداخلها: شيفرة العملية Opcode والحقل Function (من التعليمة) + خطوط أخرى
    - مخارجها: خطوط التحكم التي يحتاجها الـ Datapath (بما فيه الـ ALU)
    - يمكن أن تتكون من أكثر من مرحلة: ستكون هنا ثنائية المراحل
      - وحدة التحكم الرئيسية
        - مداخلها: شيفرة العملية المأخوذة من التعليمة Opcode
        - مخارجها: خطوط التحكم التي يحتاجها الـ Datapath + خطان خاصان ALUOp
      - وحدة التحكم بوحدة الحساب والمنطق
        - مداخلها: الخطان ALUOp + الحقل Function المأخوذ من التعليمة
        - مخارجها: أربعة خطوط للتحكم بوحدة الحساب والمنطق

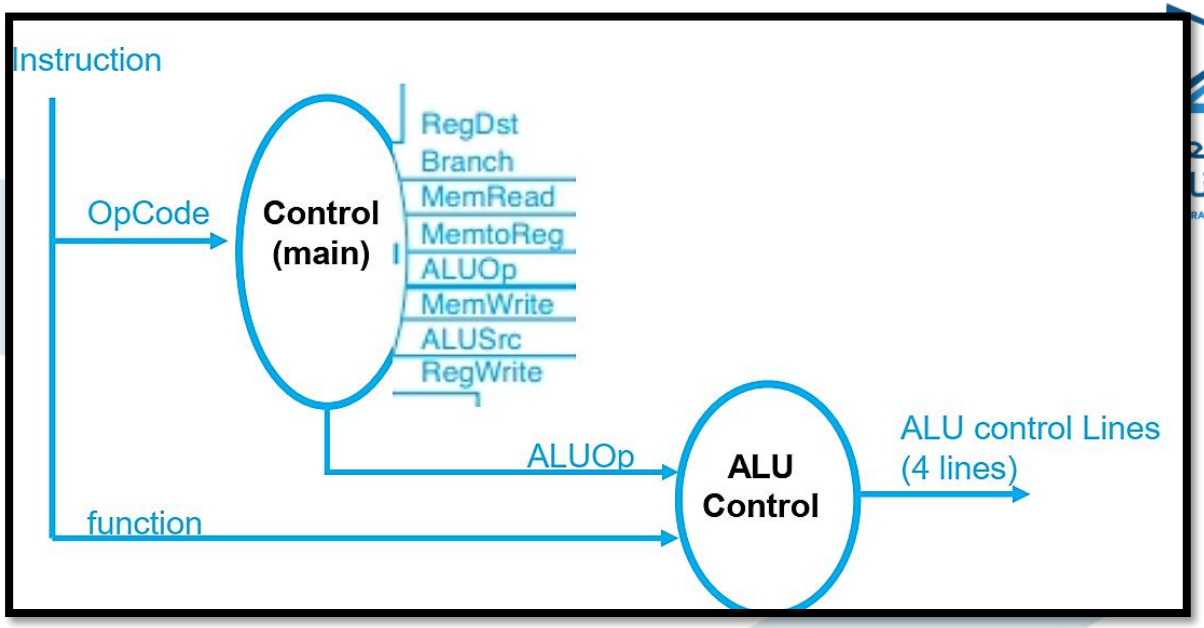
وحدة التحكم بوحدة الحساب والمنطق

- خطوط التحكم بالـ ALU
  - Ainvert
  - Bnegate
  - Operation (خطان للناخب مباشرة)
- يبين الجدول العملية التي تنفذها ال ALU من أجل قيم خطوط التحكم

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

- يجب توليد هذه الخطوط الأربع من
  - الخطين ALUOp
  - الحقل funct في شيفرة التعليمة

وحدة التحكم بوحدة الحساب والمنطق



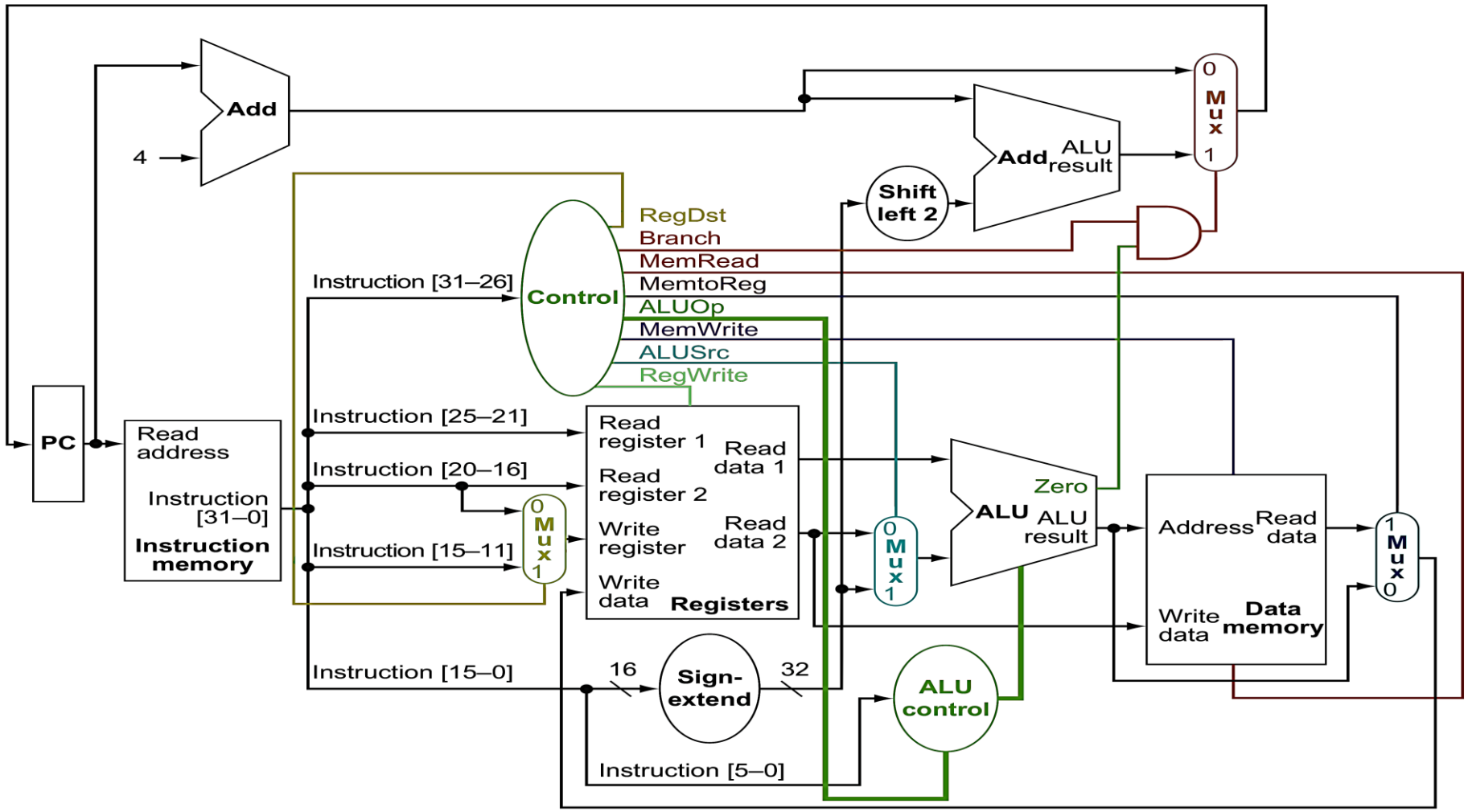
Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	set on less than	101010	set on less than	0111

تنفيذ التعليمات من النوع_ R-type

مثال: add $t1,$t2,$t3 سنتبع أربعة خطوات:

1) جلب التعليمة وزيادة قيمة الـ PC.
2) قراءة قيمة المسجلين $t3 & $t2 من ملف المسجلات وأيضاً وحدة التحكم الرئيسية تقوم بحساب قيم خطوط التحكم خلال هذه الخطوة.
3) الـ ALU تعمل على المعطيات المقروءة من ملف المسجلات مع استخدام شيفرة التابع (bits 5:0) من حقل funct من التعليمة لتحديد العملية التي ستنفذها الـ ALU.
4) نتيجة الحساب (خرج الـ ALU) تكتب في ملف المسجلات مع استخدام البيتات (11:15) من التعليمة لاختيار المسجل الهدف ($t1).

تنفيذ التعليمات من النوع _R-type

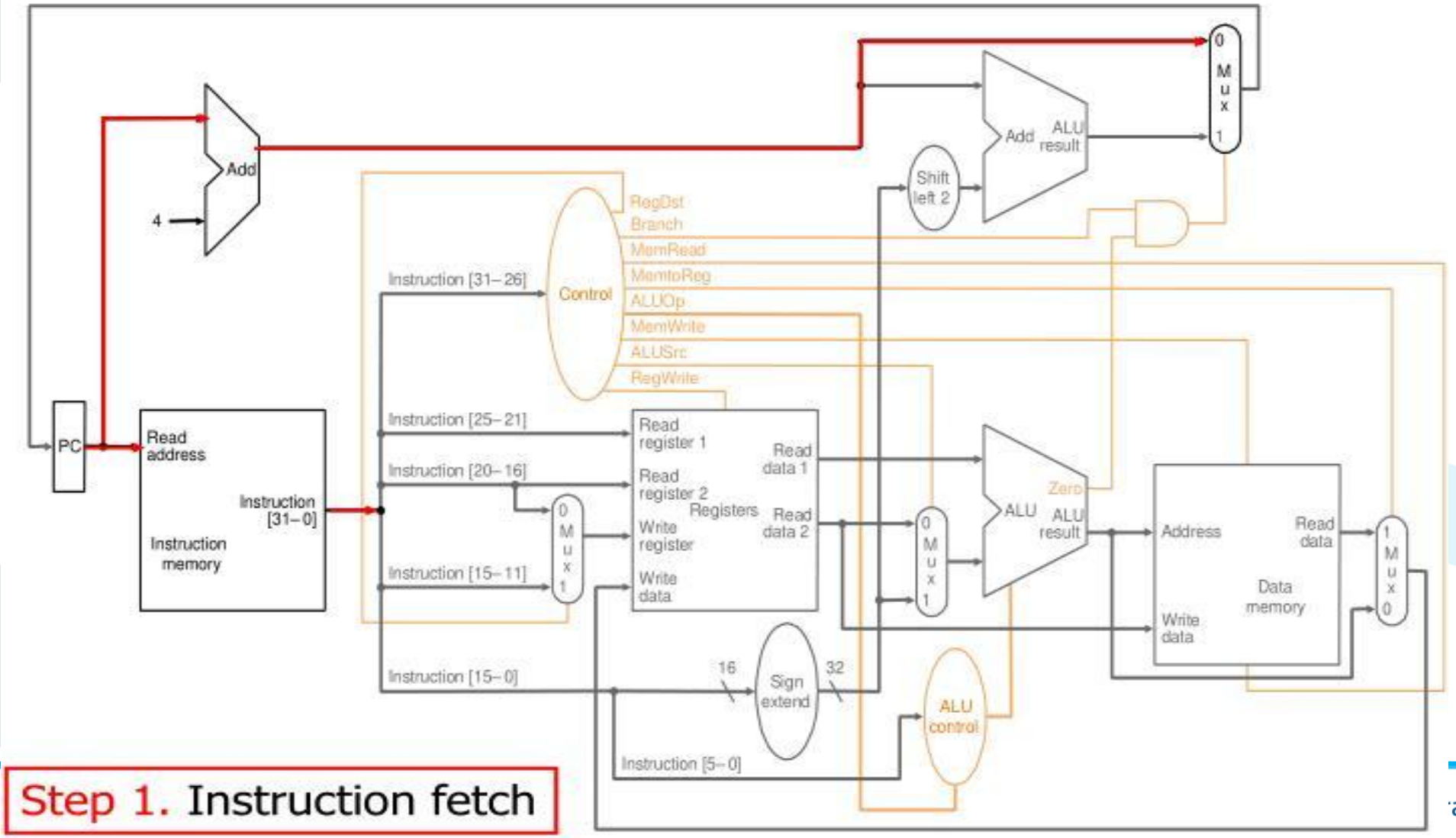


Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0

تنفيذ التعليمات من النوع_ R-type

add $t1,$t2,$t3 => add $9, $10, $11 =>

0	10	11	9	0	32



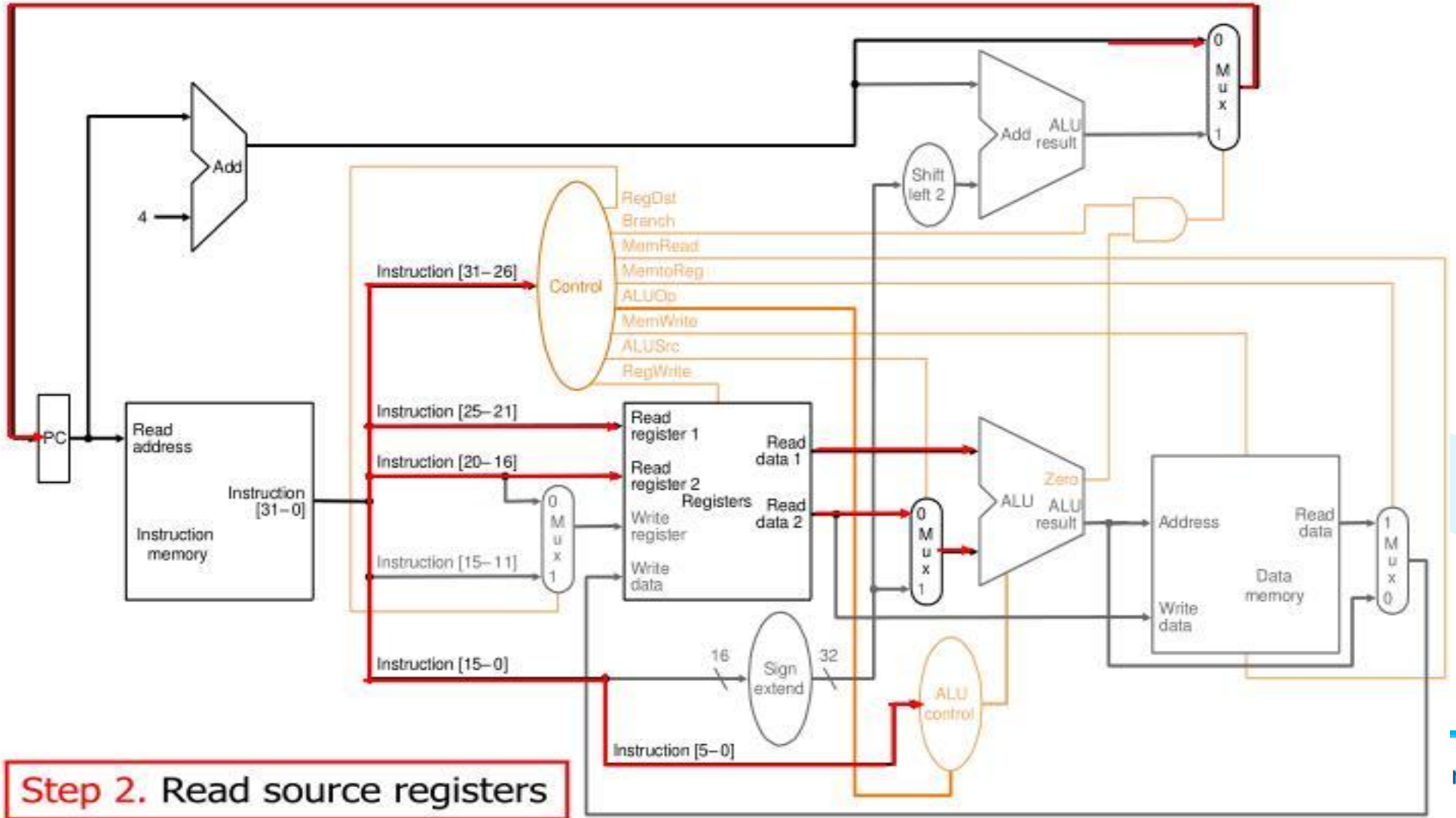
Step 1. Instruction fetch

جَامعة

تنفيذ التعليمات من النوع_ R-type

add $t1,$t2,$t3 =>

0	10	11	9	0	32

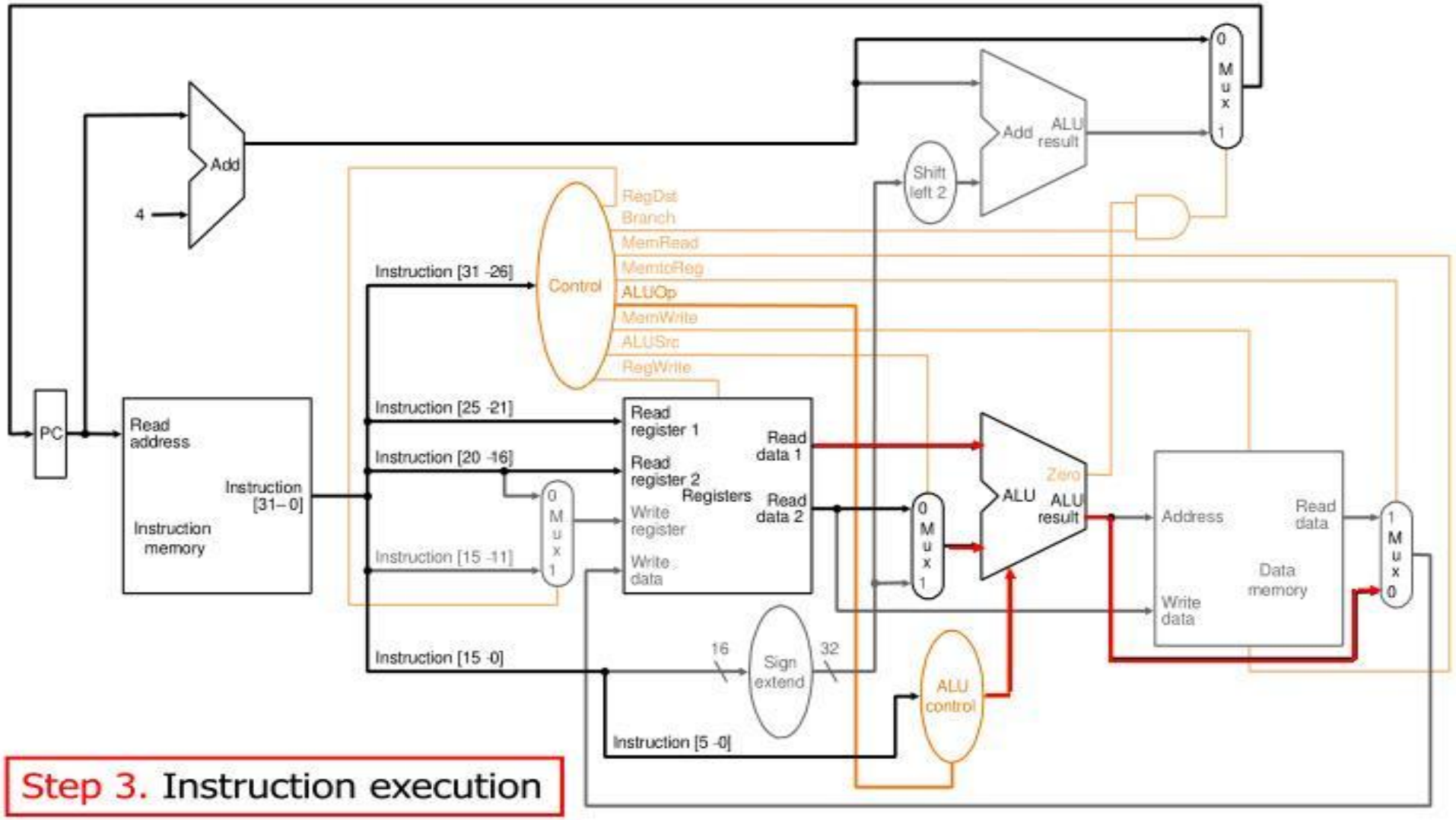


Step 2. Read source registers

تنفيذ التعليمات من النوع_ R-type

add $t1,$t2,$t3 =>

0	10	11	9	0	32

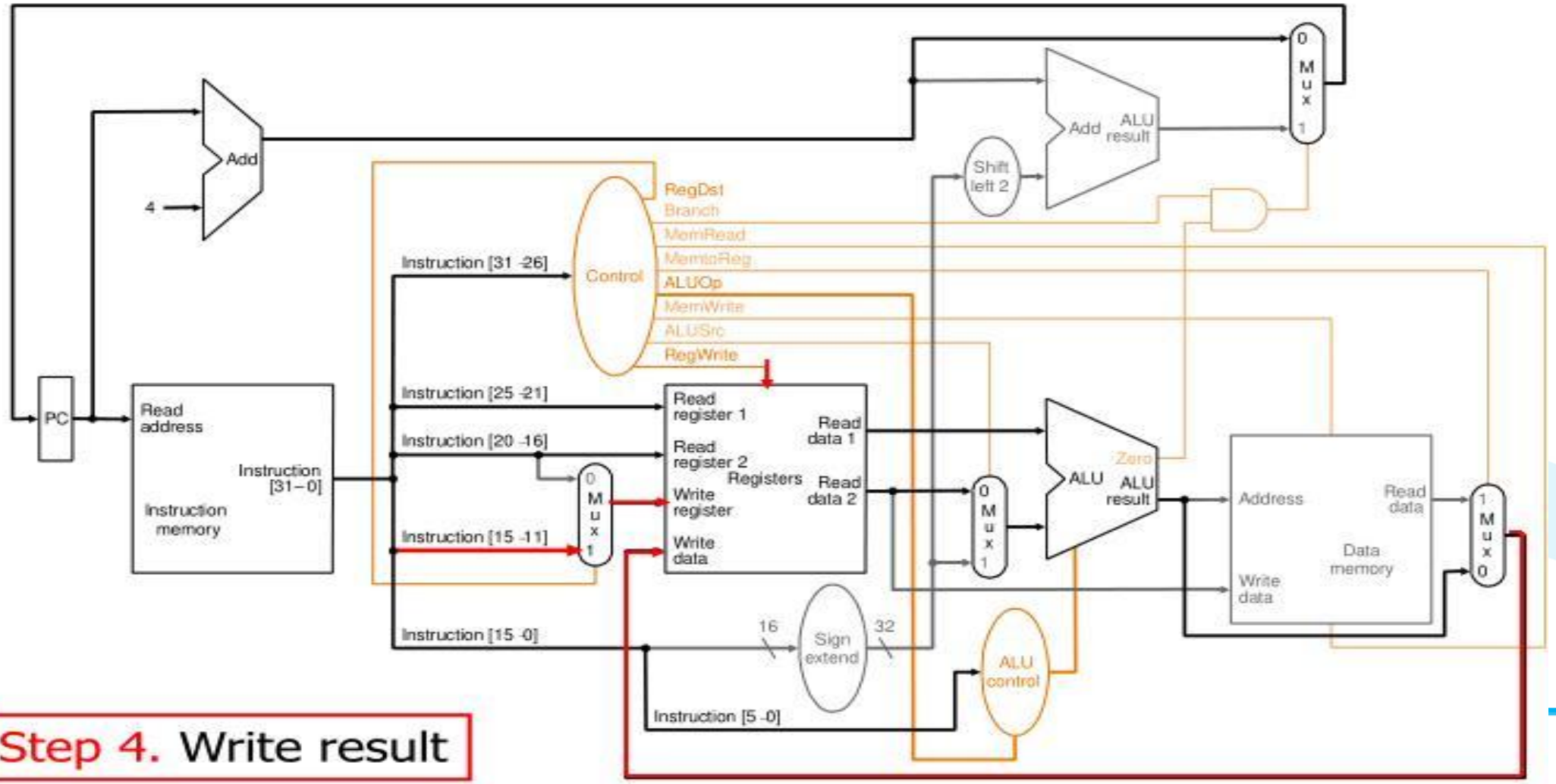


Step 3. Instruction execution

تنفيذ التعليمات من النوع_ R-type

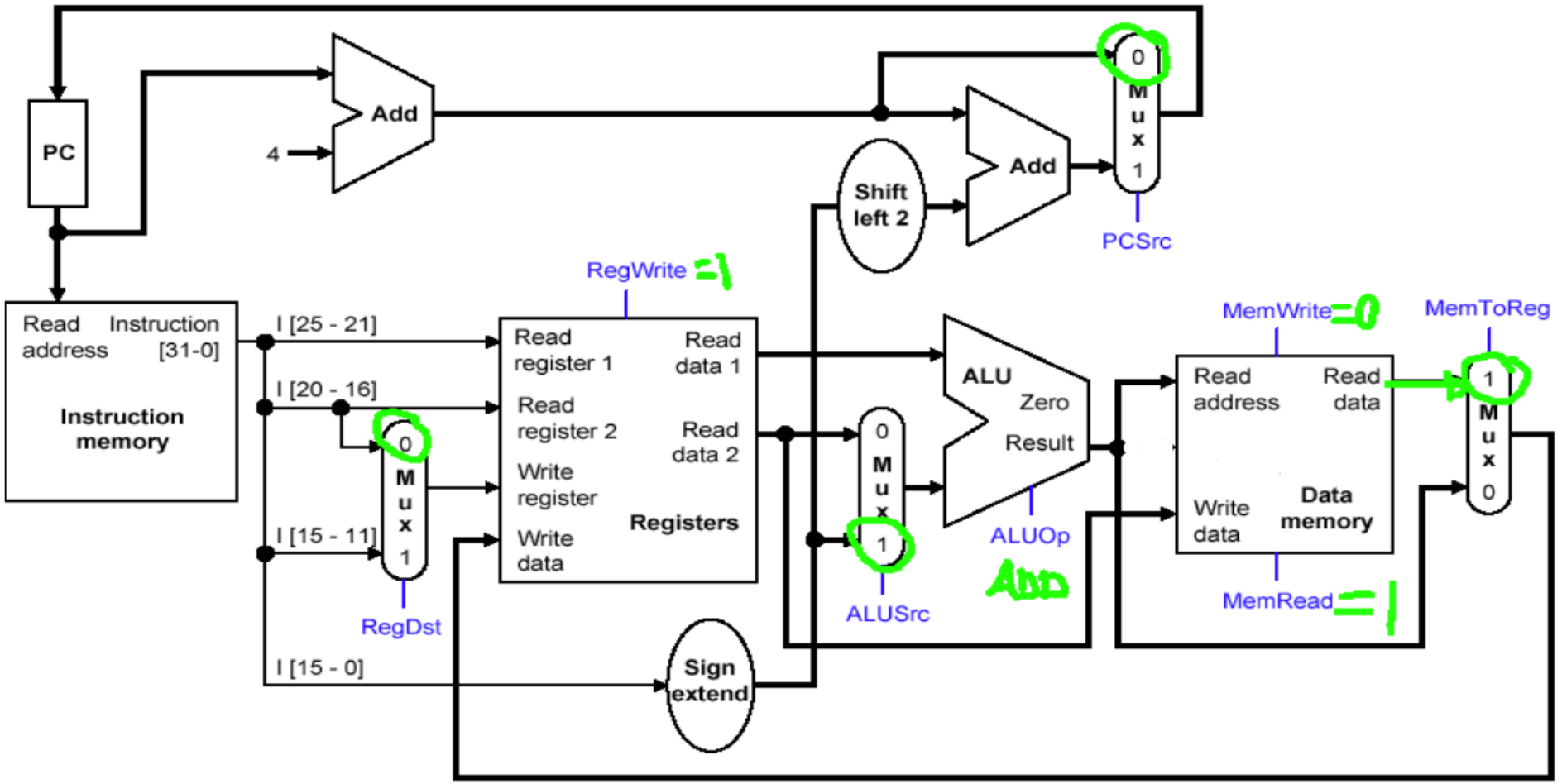
add $t1,$t2,$t3 =>

0	10	11	9	0	32



Step 4. Write result

تنفيذ التعليمتين Load & Store:



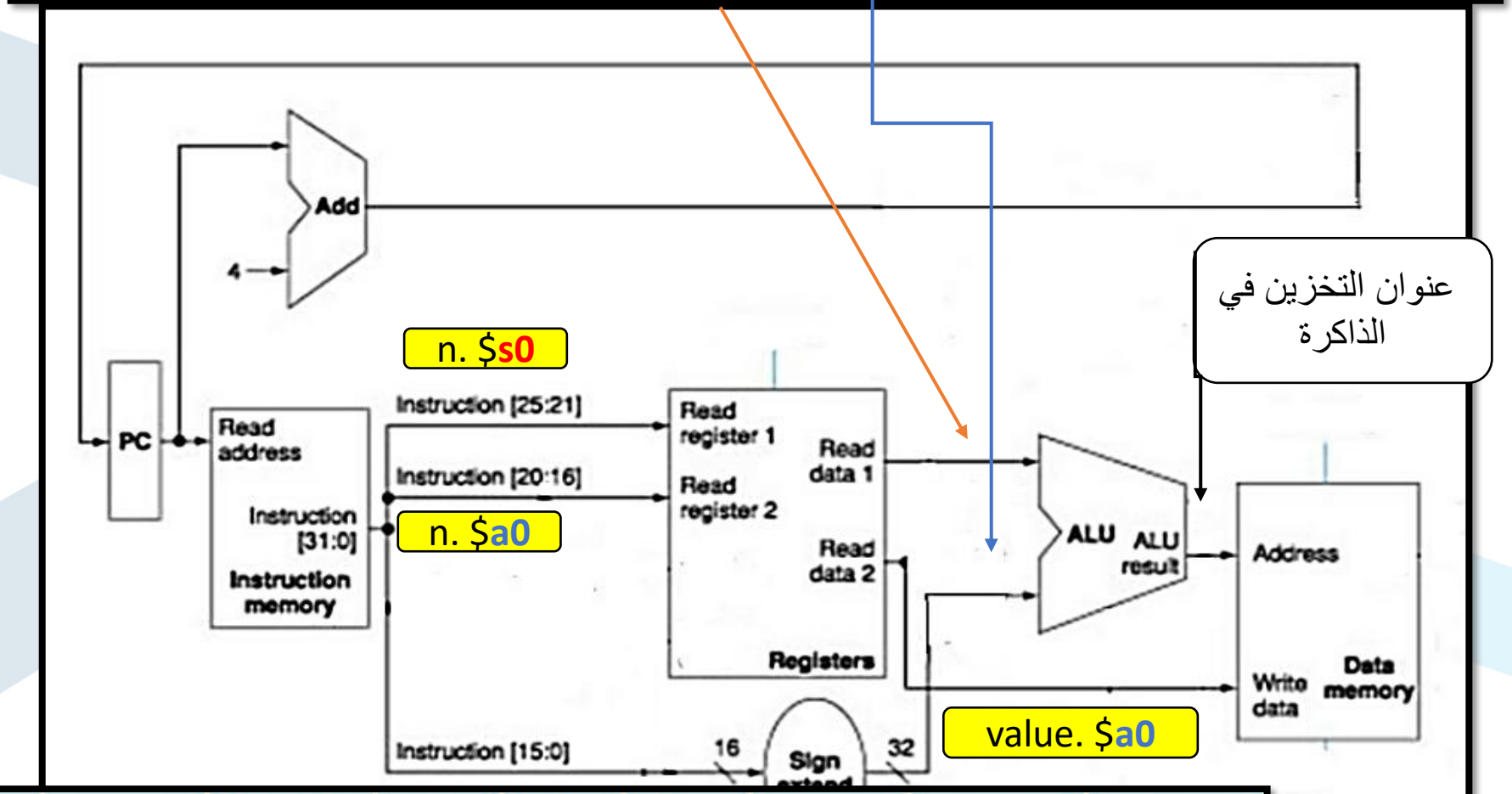
Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
lw	0	1	1	1	1	0	0	0	0

وحدة التحكم بوحدة الحساب والمنطق

تنفيذ التعليمتين Load & Store:

```
sw   $a0, 16($sn)
```

Store Word	sw	I	M[R[rs]+SignExtImm] = R[rt]	(2)	$2b_{hex}$



Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
sw	X	1	X	0	0	1	0	0	0

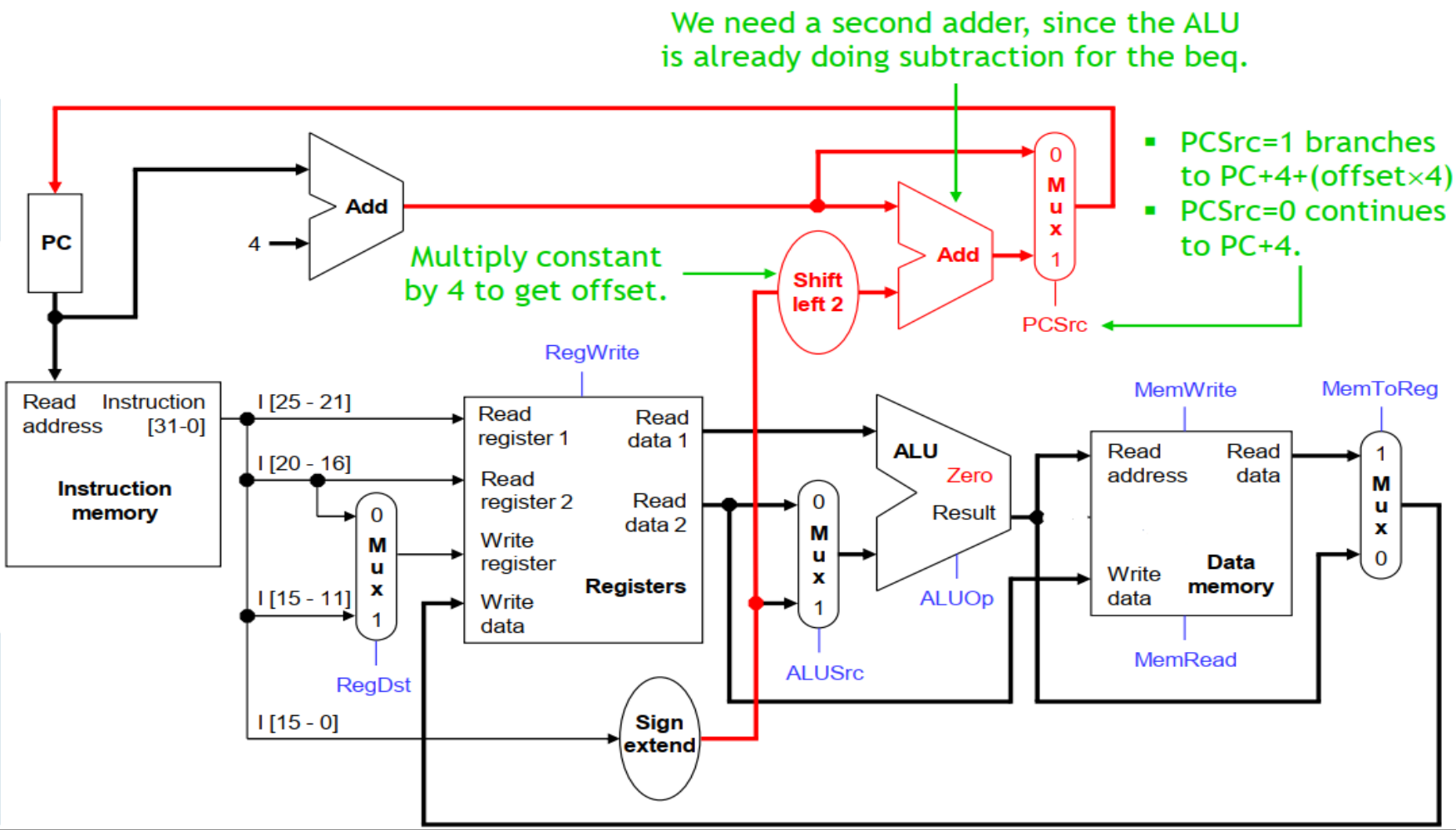
تنفيذ تعليمة القفز المشروط:

مثال: beq $t1,$t2,offset تعمل بشكل مشابه لتعليمات النوع R لكن في هذه الحالة خرج وحدة الحساب يستخدم لتحديد قيمة الـ PC إما PC+4 أو عنوان القفز الهدف وتنفذ التعلية عبر أربعة خطوات:

1) جلب التعليمة من ذاكرة التعليمات وزيادة قيمة الـ PC.
2) قراءة قيمة المسجلين $t3 & $t2 من ملف المسجلات.
3) تقوم الـ ALU بطرح قيمتي المسجلين المقروئين من ملف المسجلات. قيمة PC+4 تضاف إلى 16 bits الدنيا الممددة من حقل الـ offset للتعليمة مزاج بمقدار 2 إلى اليسار والنتيجة ستكون عنوان القفز.
4) الصفر الناتج من الـ ALU يستخدم لتحديد نتيجة أي جامع ستخزن في الـ PC.

وحدة التحكم بوحدة الحساب والمنطق

تنفيذ تعليمة القفز المشروط:

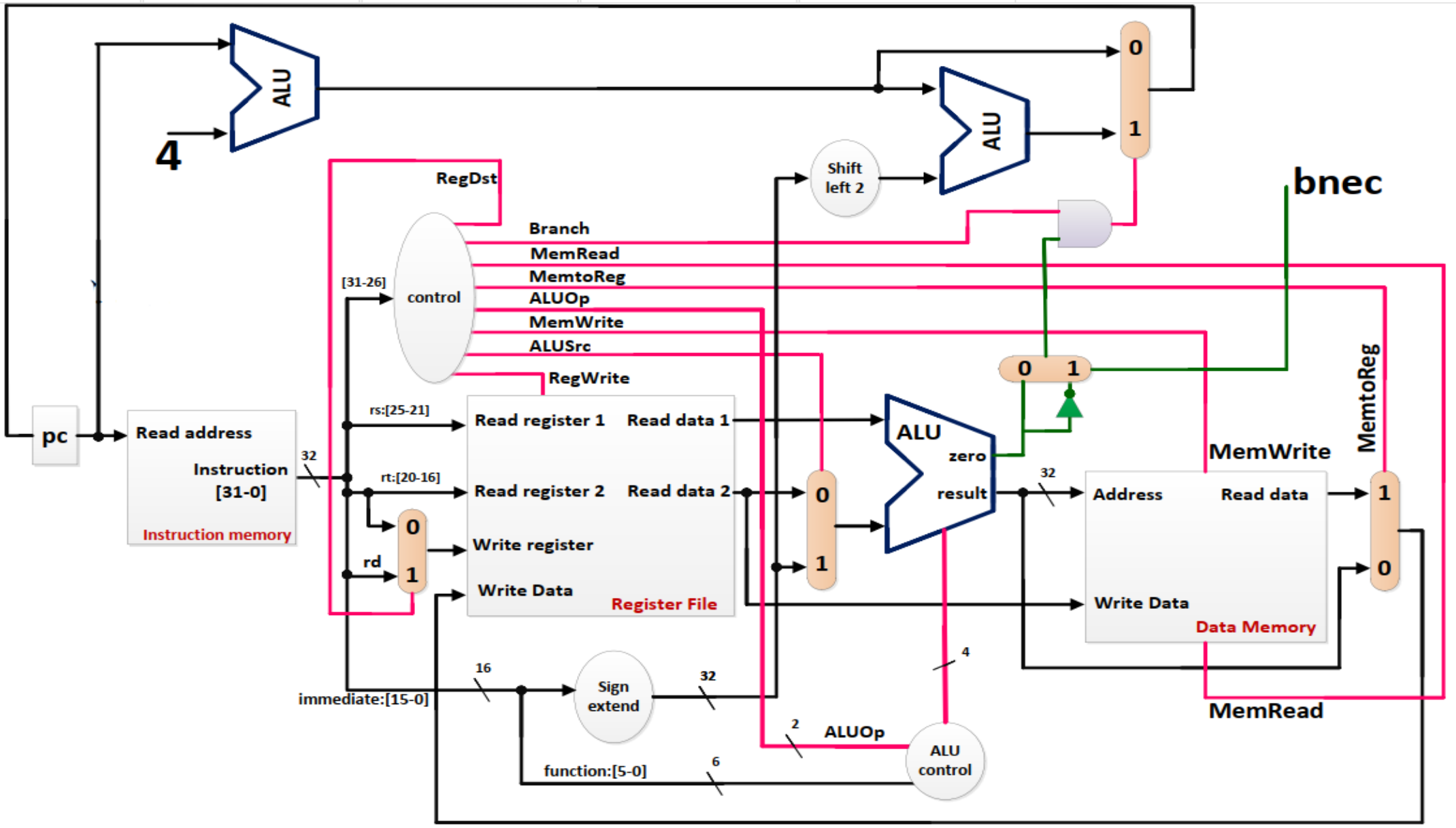


Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
beq	X	0	X	0	0	0	1	0	1

تحقيق تعليمة bne rs, rt, flag

- لا يمكن تحقيق هذه التعليمة باستخدام المخطط السابق
- فهي بعكس التعليمة السابقة
- يجب أن يكون branch=1 للدلالة انها تعليمة قفز
- ويجب ان يكون zero=0 (أي عدم تطابق حتا يتم القفز)
- لا يمكن الحل بوضع بوابة not على الخط zero لان هذا الأمر سيلغي عمل تعليمة beq
- يمكن الحل بإضافة ناخب جديد ذو خط تحكم جديد اسمه bnec (اسم اختياري)
- بحيث يكون عندما bnec=0 فاننا نختار قيمة zero كما هي وبالتالي تفعيل تعليمة beq
- وعندما bnec=1 أي أننا نريد ادخال بوابة not إلى الخط zero من أجل تفعيل تعليمة bne

تحقيق تعليمة bne rs, rt, flag



جَـامعة
المَـنارة
MANARA UNIVERSITY

نـهايـة المحـاضرة السابعة