

# Organización de Computadoras

---

SEMANA 4

UNIVERSIDAD NACIONAL DE QUILMES

# ¿Qué vimos?

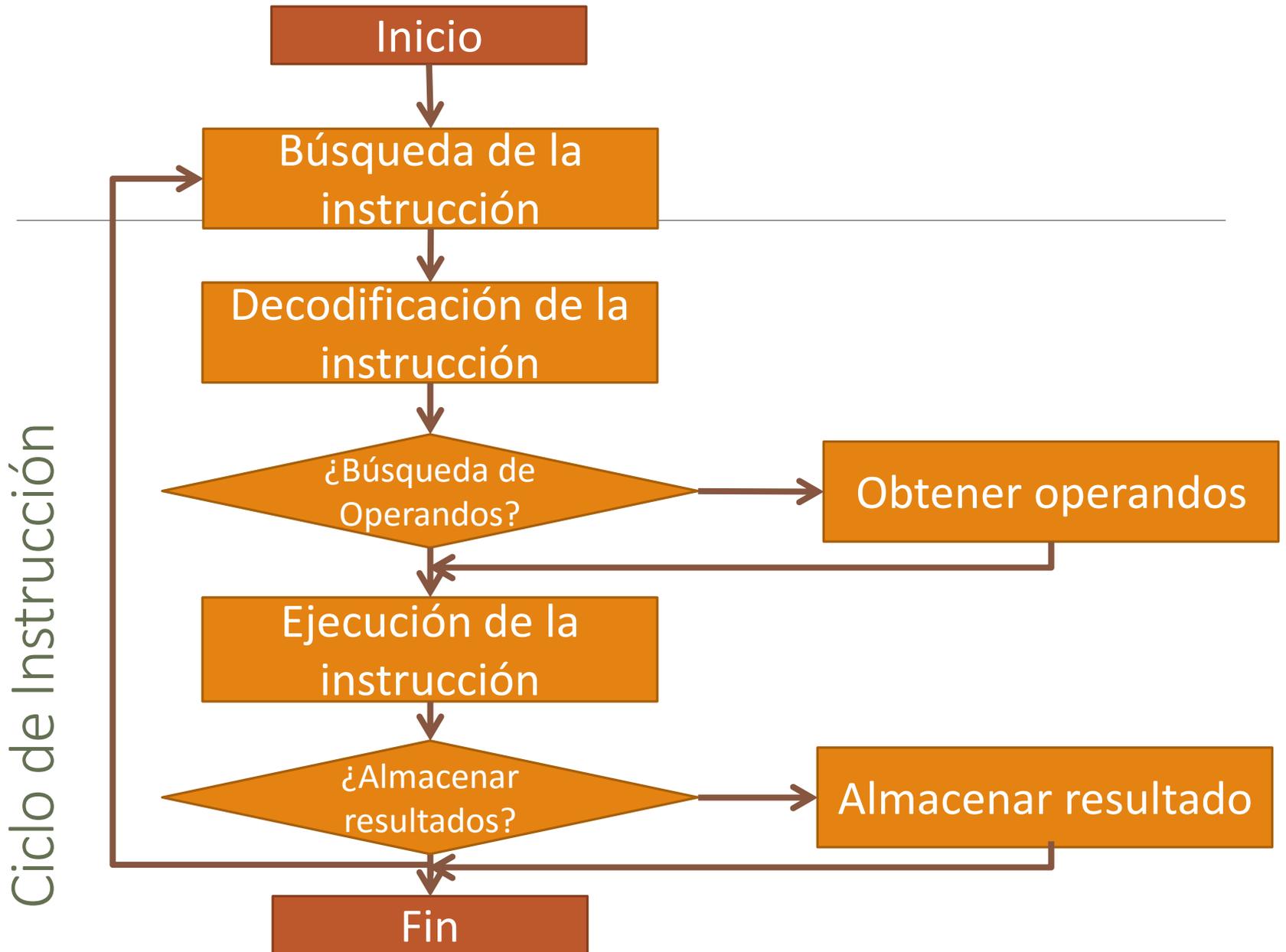
---

- Memoria:
  - Organización
  - Lectura
  - Escritura
- Buses:
  - ¿Qué?
  - Tipos
- Arquitectura Q2
- Accesos a memoria

# Hoy!

---

- Program Counter
- Modularización
- Reuso
- Especificación por contratos
- Llamadas a subrutinas
- Q3



# Ciclo de Instrucción

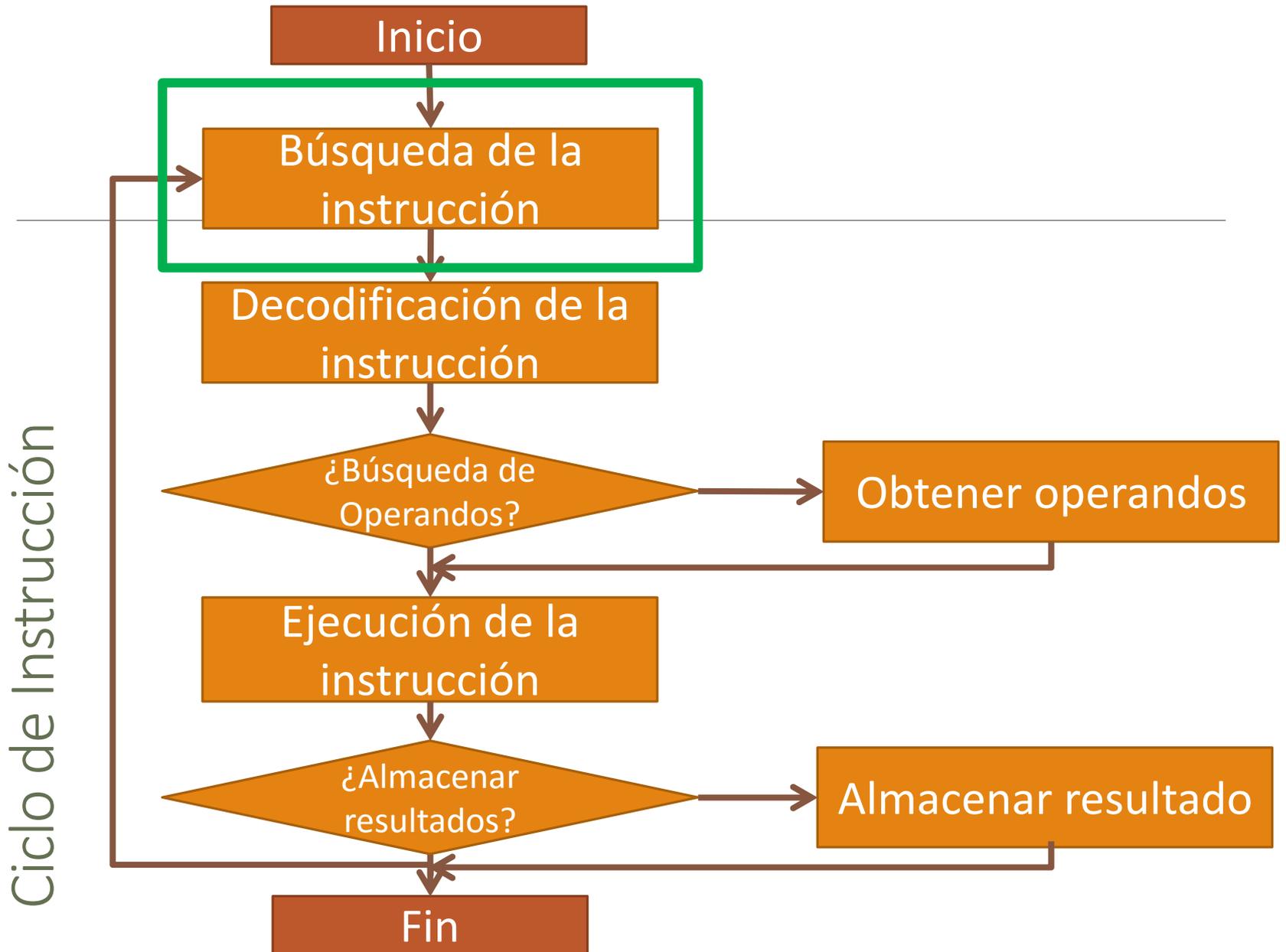
---

¿Cómo sabe la CPU cual es la próxima instrucción?

**PC**

**P**rogram **C**ounter

**P**roxima **C**osa



# Ciclo de Instrucción

---

Dirección	Contenido
0x0001	0xFFFF0
0x0002	0xAA0E
0x0003	0x00E7
0x0004	0x0672
0x0005	0x2059
0x0006	0x5BD3

Instrucción que se está ejecutando

Valor del PC

# Ciclo de Instrucción

## Registros Importantes

---

- **IR** (Instruction Register): contiene la instrucción que se está ejecutando
- **MBR** (Memory Buffer Register): Registro desde donde se lee o escribe a memoria.
- **MAR** (Memory Address Register): Registro para indicar la dirección para escribir o leer

# Ciclo de Instrucción

## Ejemplo

---

Dirección	Contenido
0x0000	0x1200
0x0001	0x0008
0x0002	0x1111
0x0003	0x29C8
0x0004	0xA0A0
0x0005	0x0000
0x0006	0x0000
0x0007	0x0000
0x0008	0x0000

PC = 0x0000

# Ciclo de Instrucción

## Ejemplo – Búsqueda de Instrucción

---

Dirección	Contenido
0x0000	0x1200
0x0001	0x0008
0x0002	0x1111
0x0003	0x29C8
0x0004	0xA0A0
0x0005	0x0000
0x0006	0x0000
0x0007	0x0000
0x0008	0x0000

PC = 0x0001

IR = 0x1200

# Ciclo de Instrucción

## Ejemplo – Interpretación

---

Dirección	Contenido
0x0000	0x1200
0x0001	0x0008
0x0002	0x1111
0x0003	0x29C8
0x0004	0xA0A0
0x0005	0x0000
0x0006	0x0000
0x0007	0x0000
0x0008	0x0000

PC = 0x0001

IR = 0x1200

Mov [ ? ],Inm

# Ciclo de Instrucción

## Ejemplo – Búsqueda de instrucción

---

Dirección	Contenido
0x0000	0x1200
0x0001	0x0008
0x0002	0x1111
0x0003	0x29C8
0x0004	0xA0A0
0x0005	0x0000
0x0006	0x0000
0x0007	0x0000
0x0008	0x0000

PC = 0x0002

IR = 0x1200 0008

Mov [ ? ],Inm

# Ciclo de Instrucción

## Ejemplo – Interpretación

---

Dirección	Contenido
0x0000	0x1200
0x0001	0x0008
0x0002	0x1111
0x0003	0x29C8
0x0004	0xA0A0
0x0005	0x0000
0x0006	0x0000
0x0007	0x0000
0x0008	0x0000

PC = 0x0002

IR = 0x1200 0008

Mov [0008],Inm

# Ciclo de Instrucción

## Ejemplo – Búsqueda de instrucción

---

Dirección	Contenido
0x0000	0x1200
0x0001	0x0008
0x0002	0x1111
0x0003	0x29C8
0x0004	0xA0A0
0x0005	0x0000
0x0006	0x0000
0x0007	0x0000
0x0008	0x0000

PC = 0x0003

IR = 0x1200 0008 1111

Mov [0008],Inm

# Ciclo de Instrucción

## Ejemplo – Interpretación

---

Dirección	Contenido
0x0000	0x1200
0x0001	0x0008
0x0002	0x1111
0x0003	0x29C8
0x0004	0xA0A0
0x0005	0x0000
0x0006	0x0000
0x0007	0x0000
0x0008	0x0000

PC = 0x0003

IR = 0x1200 0008 1111

Mov [0008],0x1111

# Ciclo de Instrucción

## Ejemplo – Ejecución de la instrucción

---

Dirección	Contenido
0x0000	0x1200
0x0001	0x0008
0x0002	0x1111
0x0003	0x29C8
0x0004	0xA0A0
0x0005	0x0000
0x0006	0x0000
0x0007	0x0000
0x0008	0x1111

PC = 0x0003

IR = 0x1200 0008 1111

Mov [0008],0x1111

# DESAFIO!

---

Hacer un programa que calcule  $n^5$  para los números que están en las celdas 0x1000, ..., 0x1003

Dirección	Contenido
0x1000	0x1200
0x1001	0x0008
0x1002	0x1111
0x1003	0x29C8
0x1004	0x40A0
0x1005	0x0000

# Desafío de programación

$n^5$

---

## ¿Que Hay que hacer?

1. Copiar el contenido de 1000 a un registro
2. Multiplicarlo 4 veces por si mismo
3. Copiar el resultado a 1000
4. Copiar el contenido de 1001 a un registro
5. Multiplicarlo 4 veces por si mismo
6. Copiar el resultado a 1001
7. Copiar el contenido de 1002 a un registro
8. Multiplicarlo 4 veces por si mismo
9. Copiar el resultado a 1002
10. Copiar el contenido de 1003 a un registro
11. Multiplicarlo 4 veces por si mismo
12. Copiar el resultado a 1003
13. ...

# Rutinas

## Definición

---

- Programa que resuelve un problema acotado
- Programa que queremos utilizar mas de una vez con otros de nuestros programas
- Nos permite modularizar y reusar código.
- También se la llama subrutina (Es sub porque se la piensa para ser utilizada dentro de otro programa)

# Modularización

## Definición

---

- Partir un problema en varios mas chicos



# Modularización – Reuso

## Definición

---

- Escribir funciones que puedan ser usadas para resolver otros problemas. Y aprovecharlas!



# Modularización – Reuso

## Definición

---

- Escribir funciones que puedan ser usadas para resolver otros problemas. Y aprovecharlas!

f1

f3

f2

f4

Problema 1

f3 f1 f2 f3 f4

Problema 2

f4 f2

# Modularización – Reuso

## Especificación

---

- ¿Por qué?
  - Queremos poder reutilizar subrutinas
  - No podemos andar leyendo cada una para saber donde hay que pasarle los parámetros
  - Ni para saber que hacen

**Lo documentamos!**

# Modularización – Reuso

## Especificación por contratos

---



### contrato

*nombre masculino*

1. Acuerdo, generalmente escrito, por el que dos o más partes se comprometen recíprocamente a respetar y cumplir una serie de condiciones.  
"contrato de trabajo; contrato de alquiler"
2. Documento en que figura este acuerdo, firmado por todas las partes.  
"sometido a las condiciones que figuran en el contrato"

# Modularización – Reuso

## Especificación por contratos

---

- ¿Cómo documentar el código?
  - **Requiere**            Qué necesita la rutina (Parámetros y precondiciones)  
                                 ¿Dónde están los parámetros? (en que variables) ¿Que características deben tener? (distinto de 0, etc)
  - **Retorna**                En que variable (registro o memoria) se retorna el resultado
  - **Modifica**              Que variables auxiliares se utilizan (registros, memoria, flags)

# Modularización – Reuso

## Especificación por contratos – Ejemplo

---

- Documentar la rutina promedio
  - **Requiere** Las notas están en los registros R4 y R5. Los valores están en BSS(16)
  - **Retorna** El promedio en el registro R6, donde el promedio resulta de la división entera:  $(R4+R5)\%2$
  - **Modifica** Nada

# Modularización – Reuso

## Llamadas a subrutinas

---

```
R0_a_5: MOV R1, R0
        MUL R1, R0
        MUL R1, R0
        MUL R1, R0
        MUL R1, R0
```

¿Cómo hago para calcular R7, R6 y R5 a la 5 usando esa función?

# Modularización – Reuso

## Llamadas a subrutinas

---

- Queremos dejar de ejecutar el código actual y pasar a ejecutar el de la subrutina

The letters 'PC' are rendered in a large, bold, orange font with a halftone dot pattern. The letters have a slight 3D effect with a darker orange shadow on the right side.

**Movemos el PC!**

# Modularización – Reuso

## Llamadas a subrutinas

---

- Necesitamos nuevas instrucciones para poder ir a la rutina y volver a donde estábamos
- CALL: salta, pero guarda la dirección a donde tiene que volver (Mas adelante veremos cómo)
- RET: Salta al valor que guardo el CALL



Q1

QII

# Arquitectura

---



La llamada

# Q3

## Modos de direccionamiento

---

- Mismos modo de direccionamiento

Modo	Código
Inmediato	000000
Registro	100RRR
Directo	001000

# Q3

## Formato de instrucción

---

- Formato de Instrucción: Tipo 1

Cod Op (4bits)	Modo Destino (6 bits)	Modo origen (6 bits)	Destino (16 bits)	Origen (16 bits)
-------------------	--------------------------	-------------------------	----------------------	---------------------

Operación	Código	Efecto
MUL	0000	$\text{Dest} \leftarrow \text{Dest} * \text{Origen}$
MOV	0001	$\text{Dest} \leftarrow \text{Origen}$
ADD	0010	$\text{Dest} \leftarrow \text{Dest} + \text{Origen}$
SUB	0011	$\text{Dest} \leftarrow \text{Dest} - \text{Origen}$
DIV	0111	$\text{Dest} \leftarrow \text{Dest} \% \text{Origen}$

# Q3

## Formato de instrucción

---

- Formato de Instrucción: Tipo 2

Cod Op (4bits)	Relleno (000000)	Modo origen (6 bits)	Origen (16 bits)
-------------------	---------------------	-------------------------	---------------------

Operación	Código	Efecto
CALL	1011	$[SP] \leftarrow PC; SP \leftarrow SP-1;$ $PC \leftarrow \text{Origen}$

# Q3

## Formato de instrucción

---

- Formato de Instrucción: Tipo 3 (sin operandos)

Cod Op (4bits)	Relleno (000000000000)
-------------------	---------------------------

Operación	Código	Efecto
RET	1100	$PC \leftarrow [SP+1]; SP \leftarrow SP + 1$

# Q3

## Etiquetas

---

- Le ponen nombre a posiciones de memoria donde queremos ir, para no tener que calcular posiciones de memoria a mano

# Q3

## Ejercicios

---

- Escriba una rutina aLaQuinta, que eleve a la 5ta potencia el contenido del registro R0 . Documente la rutina.
- Utilizando la rutina del ejercicio anterior, escriba un programa que calcule  $n^5$  para los números que están en las celdas 0x1000, ..., 0x1007.

# ¿Qué pasó hoy?

---

- Ciclo de instrucción:
  - PC e IR
  - MAR, MBR (ligeramente)
- Subrutinas
  - Herramienta para modularizar y reusar
- Contratos
  - Documentación de subrutinas
  - Requiere, Asegura y Modifica
- Call y Ret
  - ¿Qué?
  - ¿Cómo?
- Arquitectura Q3

# Necesito algo mas!

---

- Organización y Arquitectura de computadoras, Stallings, Capítulo9.4, pág 333-336.
- Organización y Arquitectura de computadoras, Stallings, Capítulo9.4, pág 312, 317, 353-357