

Organización de computadoras

Clase 4

Universidad Nacional de Quilmes

Lic. Martínez Federico

¿Qué
PASÓ el martes?



¿Qué pasó?

- Memoria:
 - Organización
 - Lectura
 - Escritura
 - Direcciones

¿Qué pasó?

- Memoria:
 - Organización
 - Lectura
 - Escritura
 - Direcciones
- Buses:
 - ¿Qué son?
 - Tipos
 - Efecto en el tamaño de la memoria

¿Qué pasó?

- Memoria:
 - Organización
 - Lectura
 - Escritura
 - Direcciones
- Buses:
 - ¿Qué son?
 - Tipos
 - Efecto en el tamaño de la memoria
- Q2:
 - Modo de direccionamiento directo

¿Qué pasó?

- Memoria:
 - Organización
 - Lectura
 - Escritura
 - Direcciones
- Buses:
 - ¿Qué son?
 - Tipos
 - Efecto en el tamaño de la memoria
- Q2:
 - Modo de direccionamiento directo
- Fetch de operandos

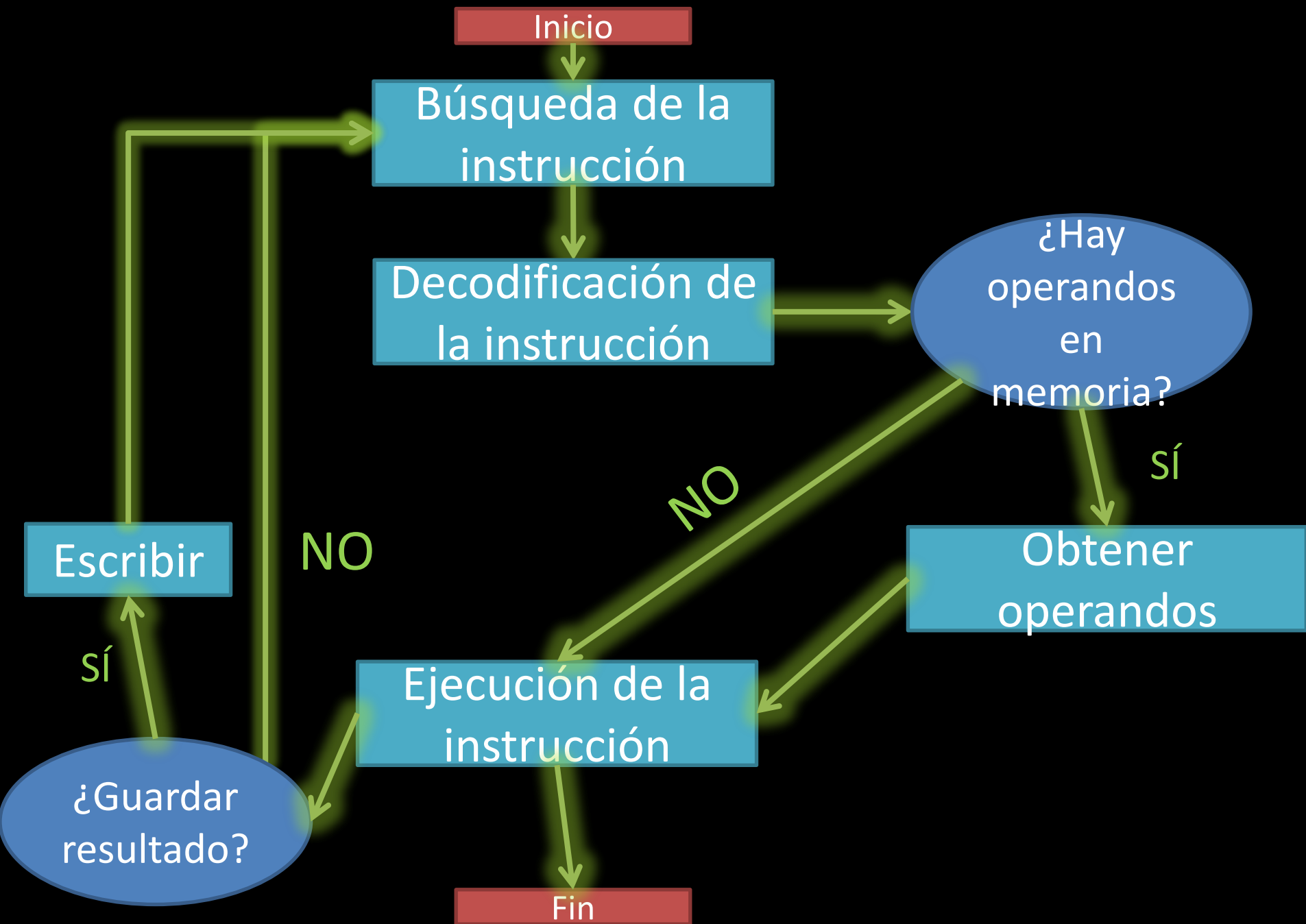
¿Qué pasó?

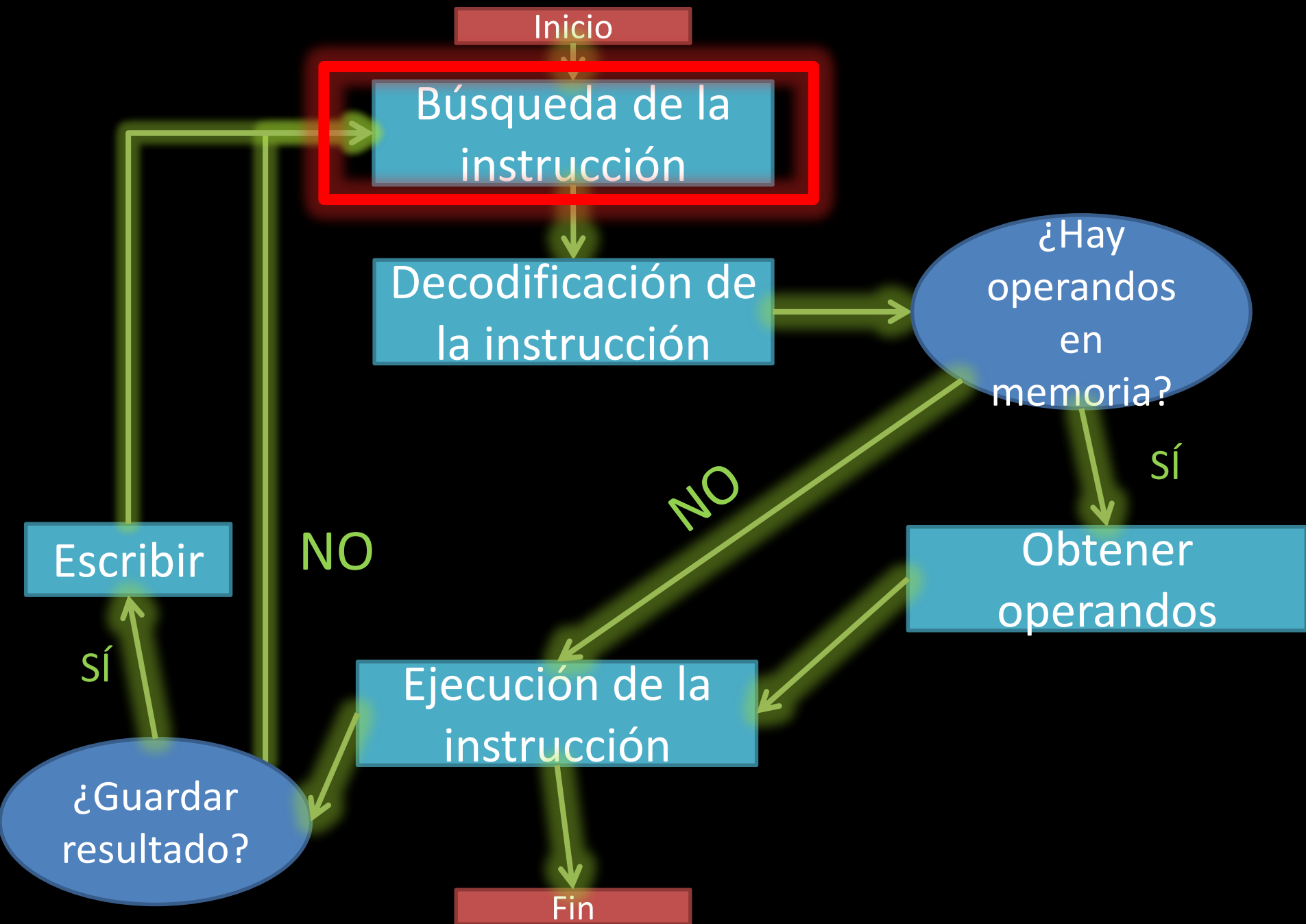
- Memoria:
 - Organización
 - Lectura
 - Escritura
 - Direcciones
- Buses:
 - ¿Qué son?
 - Tipos
 - Efecto en el tamaño de la memoria
- Q2:
 - Modo de direccionamiento directo
- Fetch de operandos
- Accesos a memoria



¿Y ahora?

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





¿Cómo sabe la CPU cual es la próxima instrucción?

¿Cómo sabe la CPU cual es la próxima instrucción?



¿Cómo sabe la CPU cual es la próxima instrucción?



R	O
O	U
G	N
R	T
A	E
M	R

¿Cómo sabe la CPU cual es la próxima instrucción?



R O
ó S
X A
I
M
A

Valor del
PC

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

Instrucción que se
está ejecutando

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

Ejemplo

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

PC = 0x0000

Busqueda de la 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

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

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 0x0008

Mov [??], Inm

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 0x0008

Mov [0x0008], Inm

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 0x0008 0x1111

Mov [0x0008], Inm

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 0x0008 0x1111

Mov [0x0008], 0x1111

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 0x0008 0x1111

Mov [0x0008], 0x1111

Rutina

- Programa que queremos utilizar mas de una vez con otros de nuestros programas

Rutina

- Programa que queremos utilizar mas de una vez con otros de nuestros programas
- Implementa una cierta lógica acotada

Rutina

- Programa que queremos utilizar mas de una vez con otros de nuestros programas
- Implementa una cierta lógica acotada
- Nos permite modularizar y reusar código

Ejemplo

- Hacer un programa que calcule n^5 para los números que están en las celdas 0x1000, ..., 0x1007

Ejemplo

- Hacer un programa que grabe un 1 o un 0 en 0x0000, ..., 0x0005 según si los números de 0xF000, ..., 0xF0005 son impares o no

Modularizar

Modularizar

- Partir un problema en varios mas chicos



Problemón

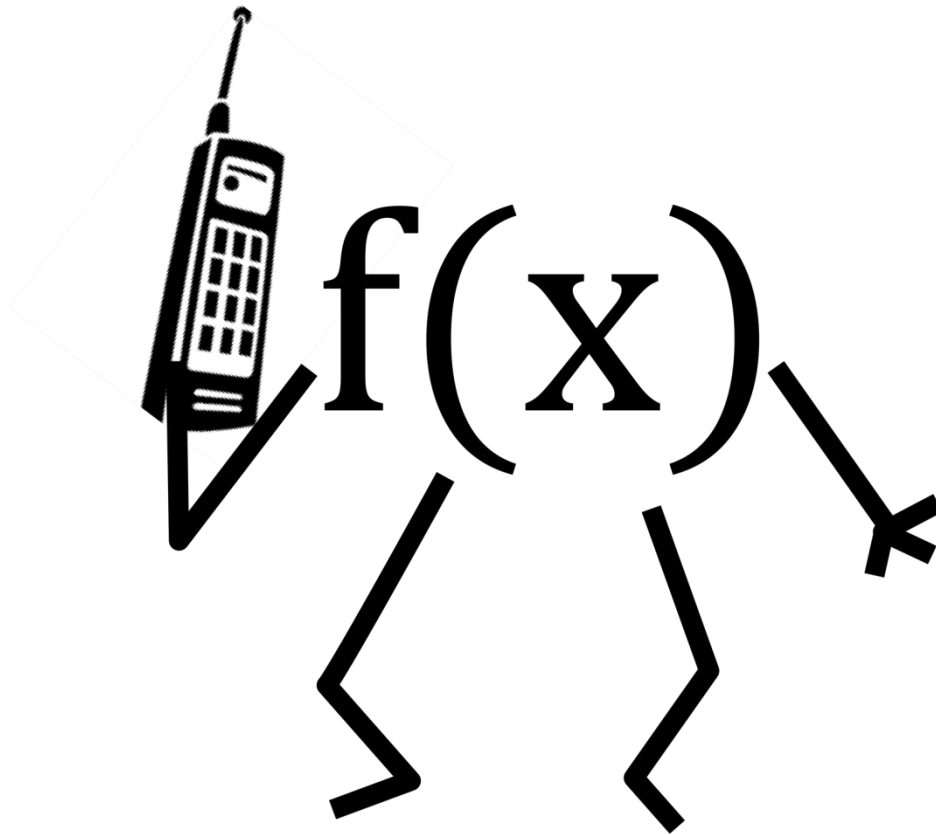
Problemita 1

Problemita 1

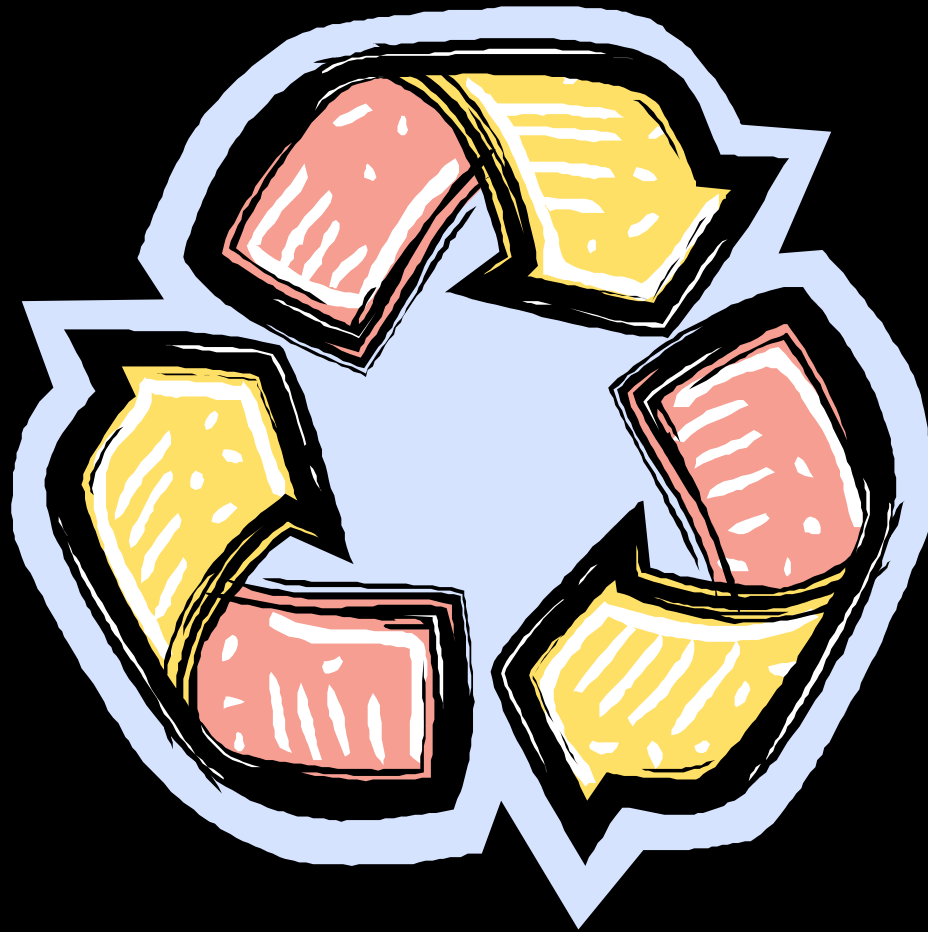
Problemita 1

Modularizar

- Usar llamadas a función



Reuso



Reuso

- 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

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!

Especificación por contratos



¿Que Es Un Contrato?

El diccionario lo define como
un acuerdo que no se puede romper...
...QUE NO SE PUEDE ROMPER...

Especificación por contratos

- Modelamos con:
 - Requiere
 - Retorna
 - Modifica

Especificación por contratos

- Pedimos lo que necesitamos: **Requiere**
- Notificamos qué vamos a devolver: **Retorna**
- Informamos que registros, celdas de memoria, flags, etc; vamos a modificar: **Modifica**

Requiere

- Dónde están los operandos?
- Ej: 0x0F0A es la dirección de uno de los números y en R1 el otro.

Requiere

- Dónde están los operandos?
- Ej: 0x0F0A es la dirección de uno de los números y en R1 el otro.
- Qué características deben tener?
- Ej: Ninguno de los número es 0, la suma de los dos operandos no se va de rango, etc

Retorna

- Cuál es el valor (o valores) que retorna y dónde.
- Ej: En R0 devuelve la suma de los valores. En R1 devuelve 1 si y solo si el número que estaba en R2 era impar.

Modifica

- ¿Cambia el valor de algún registro?
- ¿Cambian celdas de memoria?

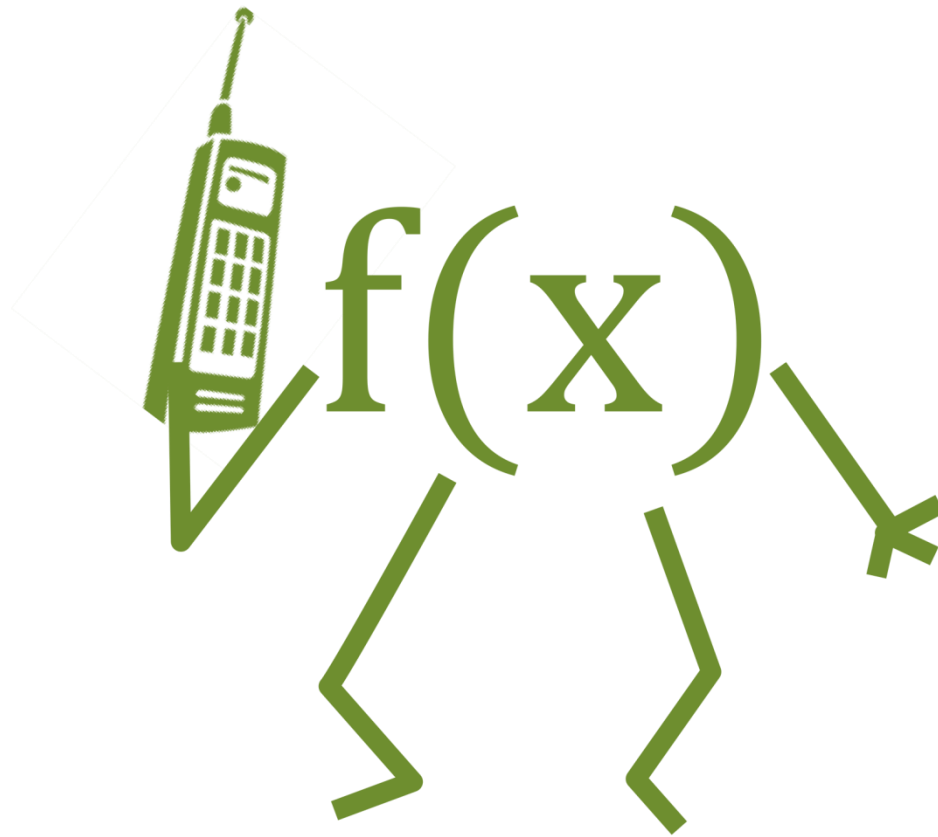
Ejemplo

- Hacer un programa que divida dos números, el primero está en R0 y lo divide por el que está en R1, guardando el resultado en R7

Ejemplo

- Hacer un programa que divide dos números, el primero está en R0 y lo divide por el que está en R1, guardando el resultado en R7
- Hacer un programa que sume los números que está en R1, R2 y R3 guardando el resultado en R3

Llamadas a subrutinas



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?

Llamadas a subrutinas

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

Llamadas a subrutinas

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



Llamadas a subrutinas

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



Llamadas a subrutinas

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



Llamadas a subrutinas

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

Movemos el PC



CALL y RET

- Necesitamos nuevas instrucciones para poder ir a la rutina y volver a donde estábamos

CALL y RET

- Necesitamos nuevas instrucciones para poder ir a la rutina y volver a donde estábamos
- Se cambia el valor de PC

CALL y RET

- Necesitamos nuevas instrucciones para poder ir a la rutina y volver a donde estabamos
- Se cambia el valor de PC
- Se recupera el valor para volver

CALL y RET

- Necesitamos nuevas instrucciones para poder ir a la rutina y volver a donde estabamos
- Se cambia el valor de PC
- Se recupera el valor para volver
- Se usa la pila (Mas de esto el viernes!)

CALL y RET

- CALL: salta, pero guarda la dirección a donde tiene que volver (Mas adelante veremos cómo)
- RET: Salta al valor que guardo el CALL

CALL y RET

¿Qué pasa si la rutina no hace ret?

Sigue corriendo!!



CALL y RET

¿Qué pasa si la rutina no hace ret?





- Modos de direccionamiento

Modo	Código
Inmediato	000000
Registro	100RRR
Directo	001000



- Formato de instrucción:
 - Instrucciones 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	$\text{Dest} \leftarrow \text{Dest} \% \text{Origen}$



- Formato de instrucción:
 - Instrucciones tipo 2:

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



- Operaciones tipo 2:

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



- Formato de instrucción:
 - Instrucciones tipo 3:

Cod Op (4 bits)	Relleno (00000000000000)
----------------------------------	---



- Operaciones tipo 4:

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

Etiquetas



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

Ejemplo

- Hacer un que calcule n^5 para los números que están en las celdas 0x1000, ..., 0x1007

Ejemplo

- Hacer un programa que grabe un 1 o un 0 en 0x0000, ..., 0x0005 según si los números de 0xF000, ..., 0xF005 son pares o no

Qué deberían llevarse en la cabeza



Qué deberían llevarse en la cabeza

- Ciclo de instrucción:
 - PC e IR
 - MAR, MBR (ligeramente)

Qué deberían llevarse en la cabeza

- Ciclo de instrucción:
 - PC e IR
 - MAR, MBR (ligeramente)
- Pila
 - SP
 - Push y Pop

Qué deberían llevarse en la cabeza

- Ciclo de instrucción:
 - PC e IR
 - MAR, MBR (ligeramente)
- Pila
 - SP
 - Push y Pop
- Subrutinas
 - Herramienta para modularizar y reusar

Qué deberían llevarse en la cabeza

- Ciclo de instrucción:
 - PC e IR
 - MAR, MBR (ligeramente)
- Pila
 - SP
 - Push y Pop
- Subrutinas
 - Herramienta para modularizar y reusar
- Contratos
 - Documentación de subrutinas
 - Requiere, Asegura y Modifica

Qué deberían llevarse en la cabeza

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

Qué deberían llevarse en la cabeza

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



¿Preguntas?



Bibliografía

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

