Organización de computadoras

Clase 2

Universidad Nacional de Quilmes

Lic. Martínez Federico

Compuertas lógicas:

- Compuertas lógicas:
 - −¿Qué?

- Compuertas lógicas:
 - −¿Qué?
 - Compuerta OR

- Compuertas lógicas:
 - −¿Qué?
 - Compuerta OR
 - Compuerta AND

- Compuertas lógicas:
 - −¿Qué?
 - Compuerta OR
 - Compuerta AND

- Compuertas lógicas:
 - ¿Qué?
 - Compuerta OR
 - Compuerta AND
 - Compuerta NOT

- Compuertas lógicas:
 - ¿Qué?
 - Compuerta OR
 - Compuerta AND
 - Compuerta NOT
 - Otras compuertas

- Compuertas lógicas:
 - ¿Qué?
 - Compuerta OR
 - Compuerta AND
 - Compuerta NOT
 - Otras compuertas
- Circuitos

- Compuertas lógicas:
 - −¿Qué?
 - Compuerta OR
 - Compuerta AND
 - Compuerta NOT
 - Otras compuertas
- Circuitos
 - Formulas y tablas de verdad

- Compuertas lógicas:
 - ¿Qué?
 - Compuerta OR
 - Compuerta AND
 - Compuerta NOT
 - Otras compuertas
- Circuitos
 - Formulas y tablas de verdad
 - Producto de sumas y suma de productos

- Compuertas lógicas:
 - ¿Qué?
 - Compuerta OR
 - Compuerta AND
 - Compuerta NOT
 - Otras compuertas
- Circuitos
 - Formulas y tablas de verdad
 - Producto de sumas y suma de productos
 - Circuitos comunes

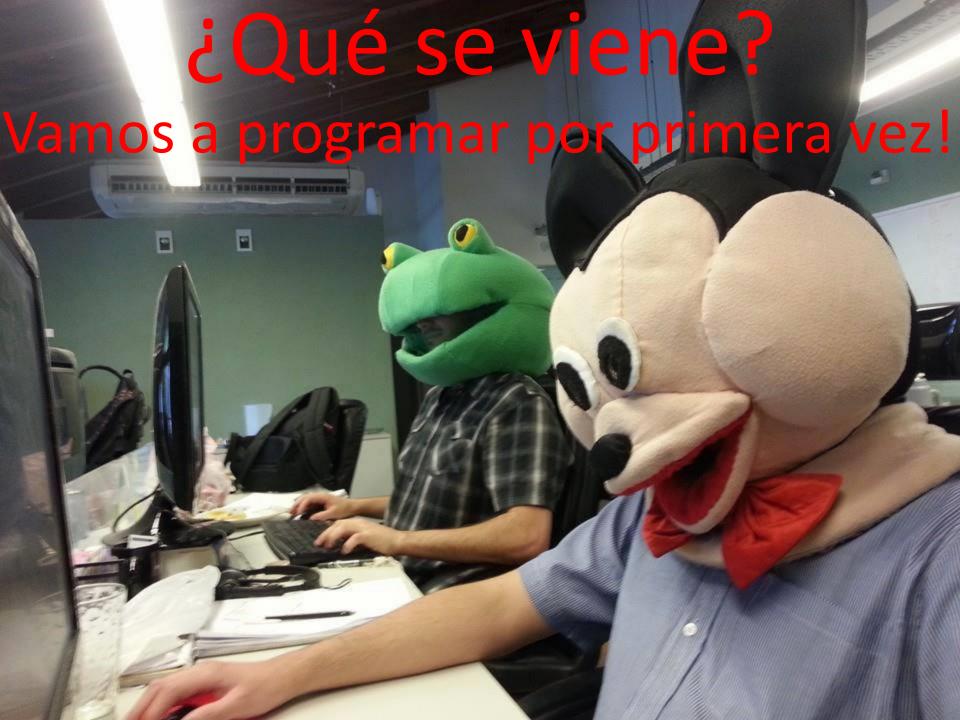
- Compuertas lógicas:
 - ¿Qué?
 - Compuerta OR
 - Compuerta AND
 - Compuerta NOT
 - Otras compuertas
- Circuitos
 - Formulas y tablas de verdad
 - Producto de sumas y suma de productos
 - Circuitos comunes
 - Circuitos aritméticos

Ensamblar y desensamblar

- Ensamblar y desensamblar
- Ciclo de ejecución de instrucción

- Ensamblar y desensamblar
- Ciclo de ejecución de instrucción
- Formato de las instrucciones

- Ensamblar y desensamblar
- Ciclo de ejecución de instrucción
- Formato de las instrucciones
- Primera máquina de uso general Q1



Modelo de Von Neumann

¿Cómo se almacenan las instrucciones?



Ejemplo











Formato de instrucción 000 001

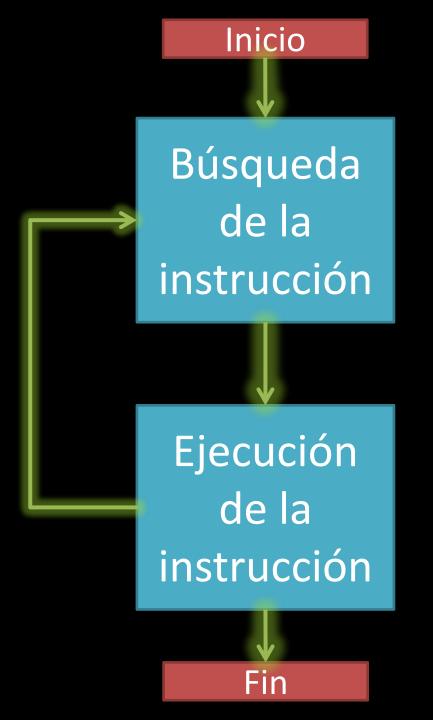


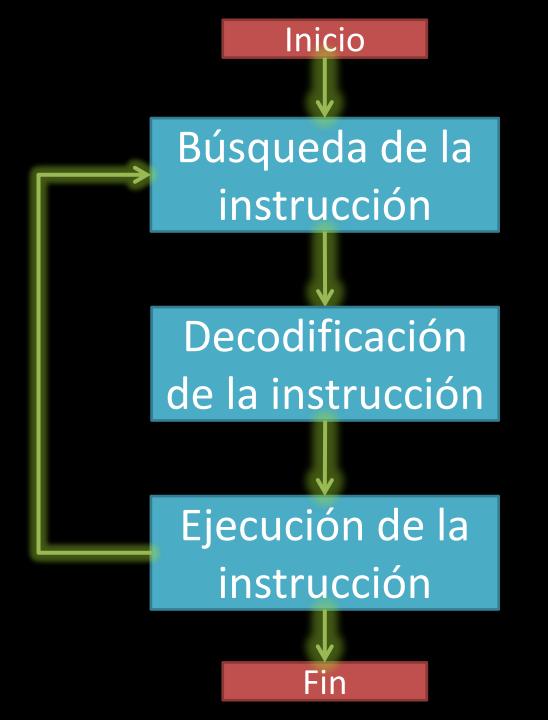












Códigos de operación

• ¿Qué pasaría si quisiéramos codificar las acciones de esta manera?





000



010



110

Códigos de operación

 ¿Qué pasaría si quisiéramos codificar las acciones de esta manera?





1



010



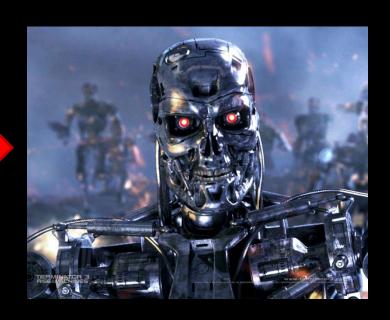
110





Ensamblar





Desensamblar





Ejercicio



- Desensamblar:
- 000 010 001 110 110 001
- 000 111 001





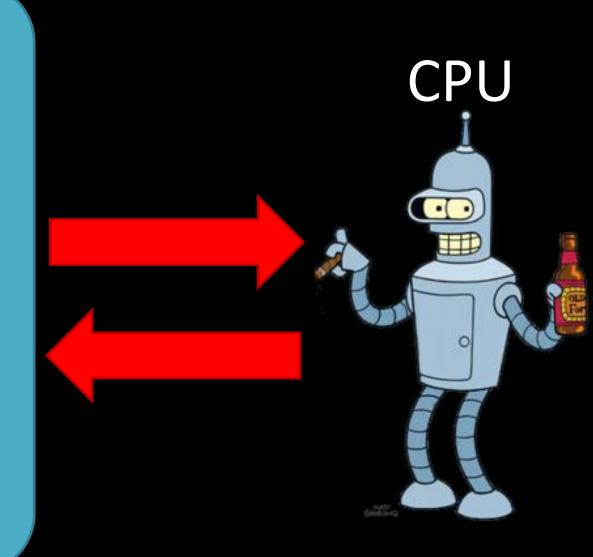
Ensamblado

Ensamblador



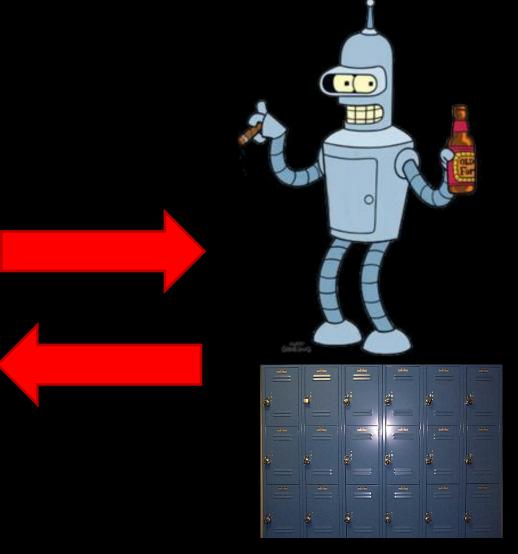
La CPU lo ejecuta

Memoria



Registros

Memoria



Registros del CPU

CPU

Arquitectura





Registros de uso general

R_0	R_1	R ₂	R ₃
R ₄	R ₅	R ₆	R ₇



Operandos:

- Un registro
- Una constante



Instrucciones

Operación	Código	Efecto
MUL	0000	Dest ←Dest * Origen
MOV	0001	Dest← Origen
ADD	0010	Dest ← Dest + Origen
SUB	0011	Dest← Dest - Origen
DIV	0111	Dest← Dest% Origen

• MOV RO, R1

• MUL RO, 7

• ADD R5, 12



• Inmediato: El operando está en la instrucción

Inmediato: El operando está en la instrucción

Registro: El operando es un registro

Inmediato: El operando está en la instrucción

Registro: El operando es un registro

MOV RO, R1
MUL RO, 7

¿Tiene sentido esta instrucción?

MOV 25, R1

• ¿Tiene sentido esta instrucción?

MOV 25, R1

NO!!!



Modo	Código
Inmediato	000000
Registro	100RRR



Formato de instrucción

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

Ensamblemos: MOV R1, 3

Ensamblemos: MOV R1, 3

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

Operación	Cod Op	Efecto
MOV	0001	Dest← Origen

Modo	Código
Inmediato	000000
Registro	100RRR

Ensamblemos: ADD R1, R6

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

Operación	Cod Op	Efecto
ADD	0010	Dest ← Dest + Origen

Modo	Código
Inmediato	000000
Registro	100RRR

 Hacer un programa que multiplique por 12 el valor de RO

 Hacer un programa que sume R0 con R1 y guarde el resultado en R2

Hacer un programa que a R5 le reste
 2 veces el valor que tiene R6

 Hacer un programa que a R4 le sume los valores de R1, R2 y R3; y le reste los valores de R5,R6 y R7

- Ensamblar el siguiente programa:
 - -SUB RO, R1
 - -ADD R2, R0
 - -DIV R2, 7
 - -MUL R5, 14

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

Operación	Cod Op	Efecto
ADD	0010	Dest ← Dest + Origen
SUB	0011	Dest← Dest - Origen
DIV	0111	Dest← Dest% Origen

Modo	Código
Inmediato	000000
Registro	100RRR

• Desensamblar:

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

Operación	Cod Op	Efecto
ADD	0010	Dest ← Dest + Origen
MUL	0000	Dest ←Dest * Origen
MOV	0001	Dest← Origen

Modo	Código
Inmediato	000000
Registro	100RRR

Código maquina

- Código maquina
- Formato de instrucción

- Código maquina
- Formato de instrucción
- Modos de direccionamiento

- Código maquina
- Formato de instrucción
- Modos de direccionamiento
- Arquitectura Q₁



