



# Arquitectura de Computadores

## Control cableado

### Práctica 3

2º Curso – Graduado en Ingeniería Informática



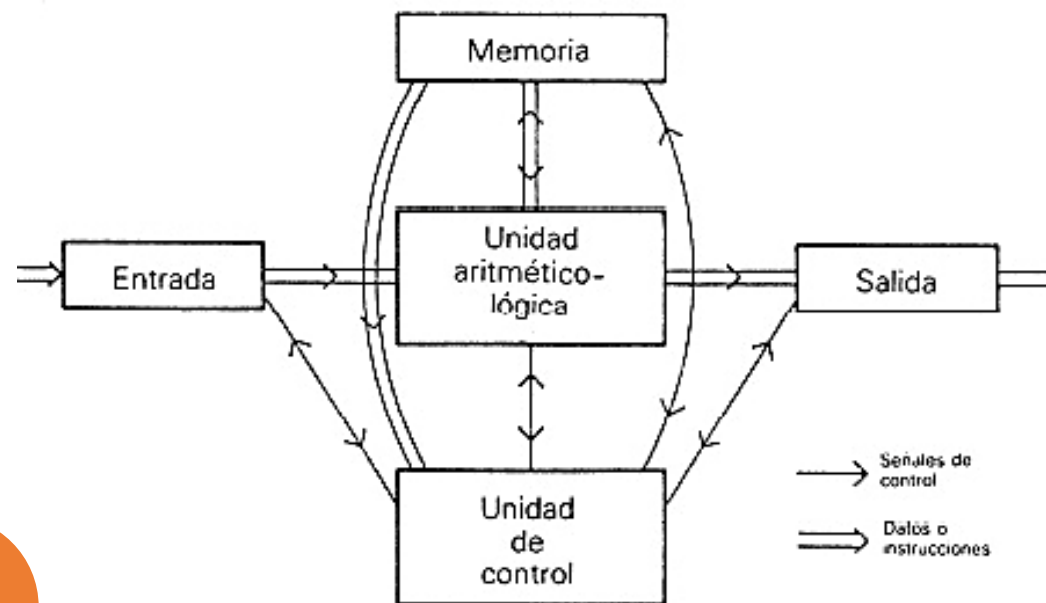
# Camino de datos y unidad de control de una computadora

## • VISIÓN GLOBAL

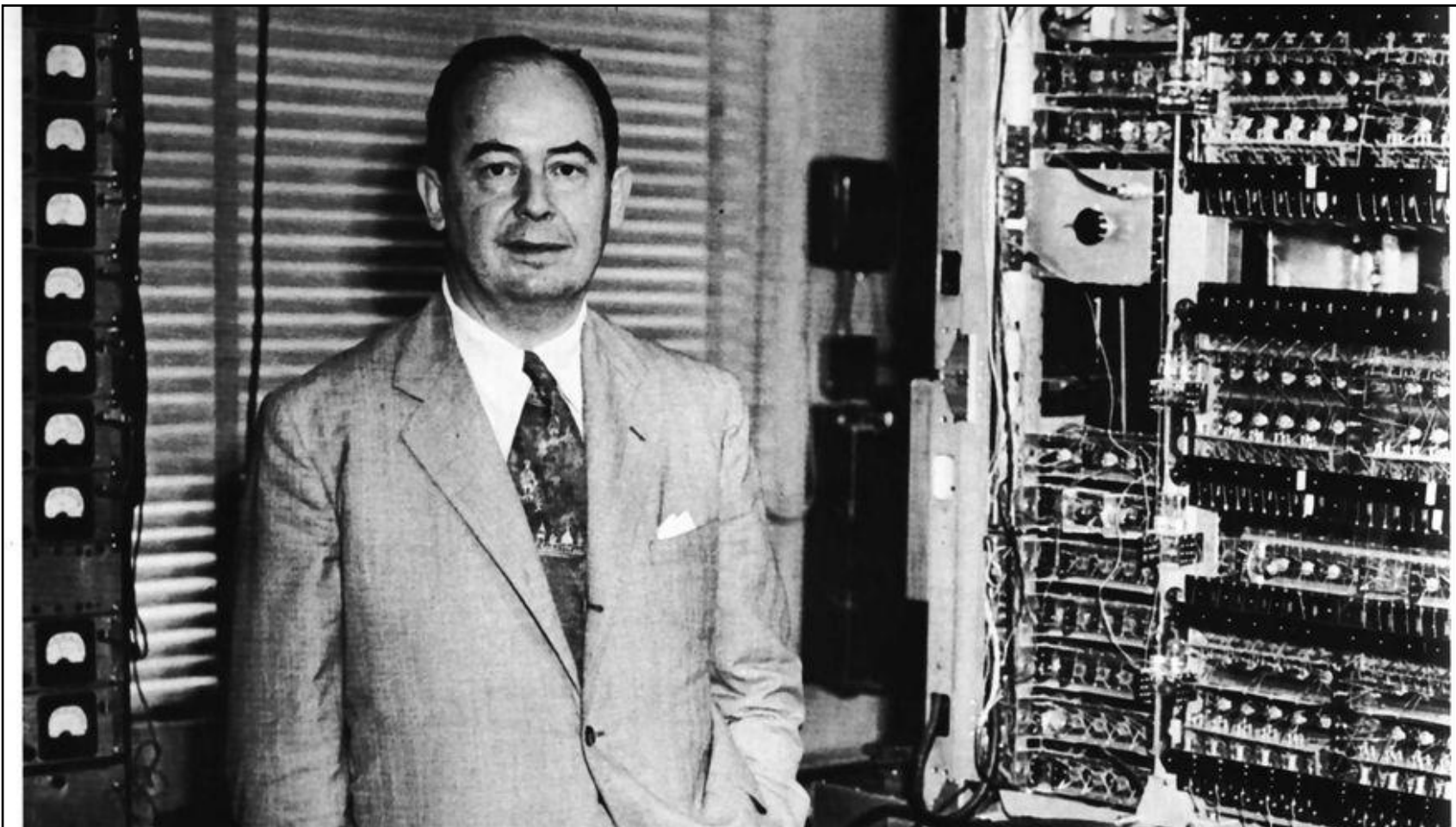
**La unidad de control como bloque de la Estructura Von Neuman**

- Unidad de cálculo
- Unidad de memoria
- Unidad de Entrada/Salida
- Unidad de control

El modelo de Von Neumann también conocido como Arquitectura Von Neumann o arquitectura Princeton es, como su nombre lo indica, una arquitectura de computadoras, es decir un modelo conceptual que muestra cómo funciona una computadora (Máquina electrónica digital programable para el tratamiento automático de la información, capaz de recibirla, operar sobre ella mediante procesos determinados y suministrar los resultados de dichas operaciones) y la forma en la que se interconectan los componentes de hardware.



Esquema de Von Neumann.



John von Neumann, participante en el Proyecto Manhattan y responsable del concepto de "programa almacenado en memoria"

- John Von Neumann fue un matemático de origen húngaro que trabajó en el Proyecto Manhattan, el desarrollo de la bomba atómica de Estados Unidos durante la Segunda Guerra Mundial
- El trabajo necesitaba realizar grandes cálculos y el desarrollo de ordenadores fue indispensable
- Von Neumann contribuyó enormemente al desarrollo de los computadores y posibilitó el desarrollo de computadoras cuyos programas se almacenaban en memoria.

Nuestros ordenadores personales, *smartphones* y *tablets* pueden **ejecutar cualquier programa** que les instalemos. Prácticamente, cualquier dispositivo que tenemos hoy a nuestro alcance es, en el fondo, un **ordenador de propósito general** cuyo ámbito de uso o aplicación lo marca el *software* que está utilizando.



# Función de la Unidad de control

## ● Asegurar la ejecución de los programas ( en Memoria principal)

- 1 Generación de las secuencias de microordenes para la ejecución de todas y cada una de las instrucciones de la computadora.
- 2 Ejecución de la siguiente instrucción del programa

# Técnicas de diseño de la Unidad de control

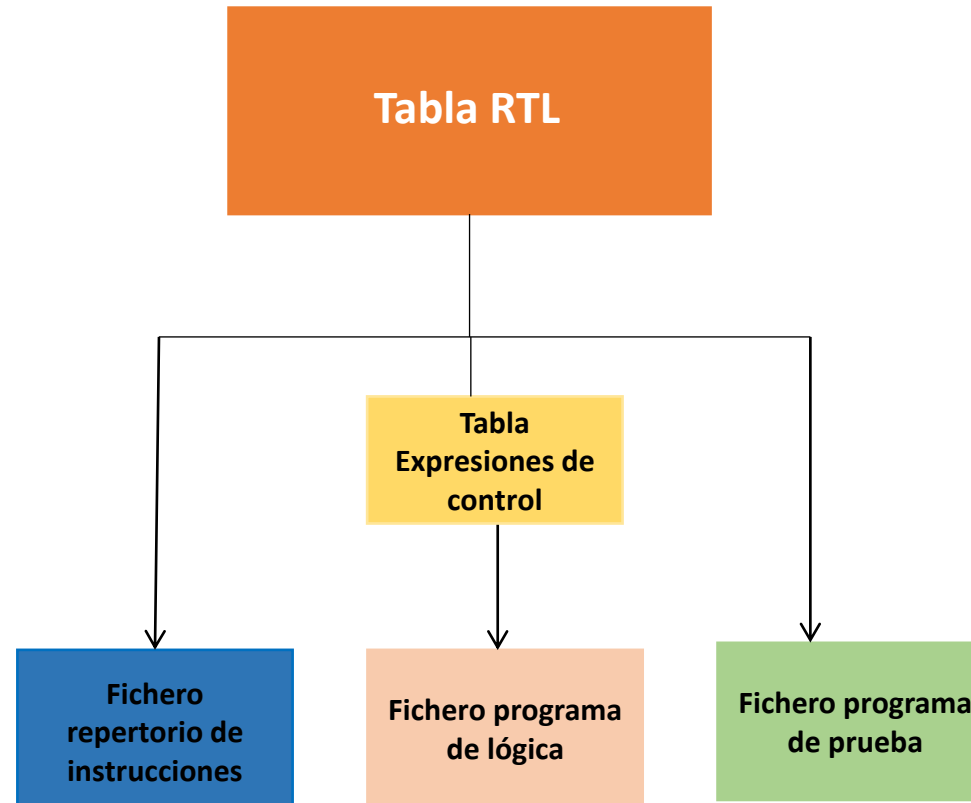
## CABLEADA

- Como Circuito Secuencial Síncrono
  - Tedioso y rígido
  - Difícil de analizar
  - Dificultad de detectar errores
  - Minimiza el número de biestables
  - Es de ejecución rápida
- Mediante Registros de desplazamiento
  - Mas sencillo
  - Flexible
  - Fácil de analizar
  - Minimiza la lógica adicional
  - Bastante rápida
- Mediante Decodificadores de Tiempo e Instrucción
  - Sencilla de diseñar
  - Bastante flexible
  - Fácil de analizar
  - Posee bastante lógica combinacional
  - Es rápida

## MICROPROGRAMADA

- Mediante la utilización de una ROM de Control
  - Muy fácil de diseñar
  - Muy flexible
  - Fácil de analizar
  - Es costosa en circuitería (ROM, Registros y L.C.)
  - Es la mas lenta

## Elementos esenciales para la creación de nuevas instrucciones en control cableado en SiCoMe





# Repertorio de instrucciones

## Tabla RTL

### Instrucciones a implementar

– **FETCH** (ciclo de búsqueda)

– **LDA m (q1)**

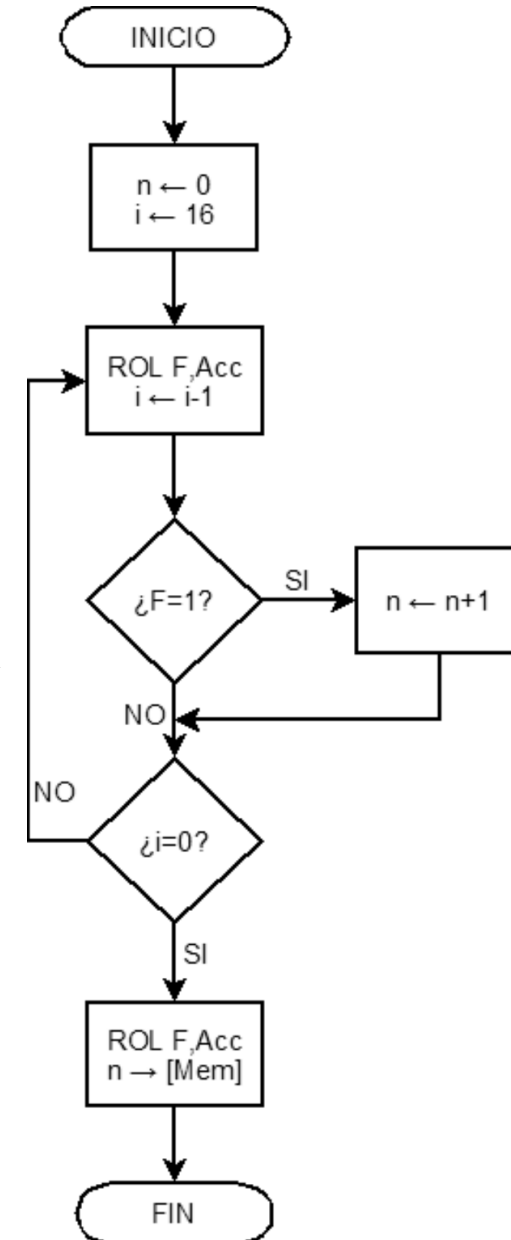
• Carga en el acumulador el contenido de la posición de memoria m

– **ONES m (q2)**

• Cuenta el número de unos en el acumulador y lo almacena en la posición de memoria m

Arquitectura de Computadores

DIAGRAMA DE FLUJO  
INSTRUCCIÓN ONES



# Tabla RTL

## Generación de las microordenes:

- \* Dependiendo de la instrucción (**qi**)
- \* Dependiendo del tiempo (**ti**)
- \* dependiendo de alguna condición

CICLO	MICROOPERACIONES	SIGUIENTE
ADDR(ONES)+0	0→QR; 16→SC	Incremento
ADDR(ONES)+1	QR→GPR	Incremento
ADDR(ONES)+2	ROL F,Acc; SC-1→SC	Incremento
ADDR(ONES)+3	GPR+1→GPR (si F=1)	Incremento
ADDR(ONES)+4		Si $Z_{sc}=0$ bifurca a ADDR(ONES)+2 Si $Z_{sc}=1$ Incrementa
ADDR(ONES)+5	ROL F,Acc; GPR→M	Bifurca a ADDR(FETCH)

CONDICIONES	MICROOPERACIONES	SIGUIENTE
FETCH		
$t_0$	PC→MAR	SR+1→SR
$t_1$	PC+1→PC; M→GPR	SR+1→SR
$t_2$	GPR(OP)→OPR; GPR(AD)→MAR	SR+1→SR
LDA		
$q_1 \cdot t_3$	0→ACC; M→GPR	SR+1→SR
$q_1 \cdot t_4$	GPR+ACC→ACC	0→SR
ONES		
$q_2 \cdot t_3$	0→QR; 16→SC	SR+1→SR
$q_2 \cdot t_4$	QR→GPR	SR+1→SR
$q_2 \cdot t_5$	ROL F,Acc; SC-1→SC	SR+1→SR
$q_2 \cdot t_6 \cdot F$	GPR+1→GPR	
$q_2 \cdot t_6 \cdot Z_{sc}$		SR+1→SR
$q_2 \cdot t_6 \cdot Z_{sc}'$		5→SR
$q_2 \cdot t_7$	ROL F,Acc; GPR→M	0→SR

## Registro de secuencia temporal:

Contador

- Cuenta hacia arriba
- Clear
- Carga paralela



## EXPRESIONES DE CONTROL

0 → SR  
5 → SR

MICROOPERACIÓN	EXPRESIÓN DE CONTROL
OPERACIONES GENÉRICAS	
PC → MAR	$t_0$
PC+1 → PC	$t_1$
M → GPR	$t_1 + q_1 \cdot t_3$
GPR(OP) → OPR	$t_2$
GPR(AD) → MAR	$t_2$
0 → ACC	$q_1 \cdot t_3$
GPR+ACC → ACC	$q_1 \cdot t_4$
0 → QR	$q_2 \cdot t_3$
QR → GPR	$q_2 \cdot t_4$
ROL F, Acc	$q_2 \cdot t_5 + q_2 \cdot t_7$
GPR+1 → GPR	$q_2 \cdot t_6 \cdot F$
GPR → M	$q_2 \cdot t_7$
OPERACIONES SOBRE EL SR	
SR+1 → SR	$t_0 + t_1 + t_2 + q_1 \cdot t_3 + q_2 \cdot t_3 + q_2 \cdot t_4 + q_2 \cdot t_5 + q_2 \cdot t_6 \cdot Z_{SC}$
LOAD SR	$q_1 \cdot t_4 \cdot (0) + q_2 \cdot t_6 \cdot Z_{SC}' \cdot (5) + q_2 \cdot t_7 \cdot (0)$
OPERACIONES SOBRE EL SC	
SC-1 → SC	$q_2 \cdot t_5$
LOAD SC	$q_2 \cdot t_5 \cdot (16)$

# EXPRESIONES DE CONTROL – FICHERO DE LÓGICA

\$

PC→MAR:t0

PC+1→PC:t1

M→GPR:t1 + t3 · q1

GPR(OP)→OPR:t2

GPR(AD)→MAR:t2

0→ACC:t3 · q1

GPR+ACC→ACC:t4 · q1

0→QR:t3 · q2

QR→GPR:t4 · q2

ROL FA:t5 · q2 + t7 · q2

GPR+1→GPR:t6 · q2 · F

GPR→M:t7 · q2

\$

SR+1→SR:t0 + t1 + t2 + t3 · q1 + t3 · q2 +

t4 · q2 + t5 · q2 + t6 · q2 · Zsc

LOAD SR:t4 · q1-0 + t6 · q2 · Zsc'-5 + t7 · q2-0

SC-1→SC:t5 · q2

LOAD SC:t3 · q2-16

\$

Operaciones genéricas

Operaciones sobre el SR

Operaciones sobre el SC

Se puede realizar  
el fichero de lógica  
a través de:

- Fichero de texto plano
- Directamente en SICOME

# CONTROL CABLEADO - LÓGICA DE ACTIVACIÓN DE MICROOPERACIONES

SiCoMe 2.0 - Simulador de la Computadora Mejorada

Archivo Simulador Servidor Ayuda

☐ Control Microprogramado  
☒ Control Cableado

Reiniciar computadora  
Resetear computadora  
Limpiar memoria

Memoria	Datos	Operación
0001	0000	NOP
0002	0000	NOP
0003	0000	NOP
0004	0000	NOP
0005	0000	NOP
0006	0000	NOP
0007	0000	NOP
0008	0000	NOP
0009	0000	NOP

Ir a 0000

Status

Estado de la ejecución: detenida

Repertorio actual:

Programa actual:

Ejecutando: ciclo - instrucción

Computadora

PC: 000  
MAR: 000  
SP: 000  
OPR: 00  
CONTROLADOR  
REGISTRO: 000000  
BITS: Zs Zc Zsc Zov N  
STATUS: 000000  
Qn Qnms Qs Bs X

ALU

Desconectado

Control Cableado - Lógica de Activación de Microoperaciones

Principal

☒ Microoperacion PC+1->PC

☐ Cont SR / SC SR+1->SR 0

Añadir Expresión Lógica Añadir

Eliminar Expresión Lógica Eliminar

Cargar Función Lógica Cargar

Función Lógica

# CONTROL CABLEADO - REPERTORIO Y PROGRAMA

## Repertorio

```
HALT false 0 q0  
LDA true 2 q1  
ONES true 5 q2
```

## Programa de prueba

```
0 1C75  
@  
10  
@  
LDA 0  
ONES 1  
HALT
```