

# Arquitectura de computadores II

## Unidad de control

Mayo de 2022

1 Microoperaciones

2 Control del procesador

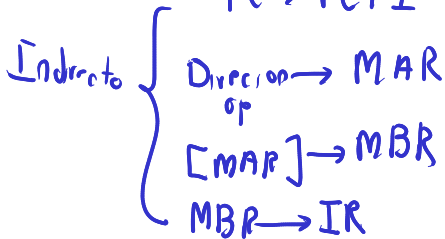
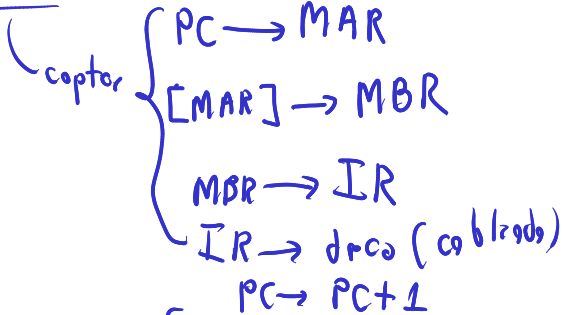
3 Implementación cableada

# Contenido

- 1 Microoperaciones
- 2 Control del procesador
- 3 Implementación cableada



LOAD AX



# Microoperaciones

## Ciclo de captación

Se examina el ciclo de captación, que tiene lugar al principio de cada instrucción, el cual involucra los siguientes registros del procesador

1. **Registro de dirección de memoria (MAR)** Está conectado a las líneas de dirección del bus
2. **Registro intermedio de memoria (MBR)** Está conectado a las líneas de datos del bus
3. **Contador de programa (PC)** Contiene la dirección a la siguiente instrucción a captar
4. **Registro de instrucción (IR)** Contiene la última instrucción captada

# Microoperaciones

## Ciclo de captación

El proceso puede ser estudiado en las siguientes unidades de tiempo

1. **Primera unidad de tiempo:** Transferir PC a MAR
2. **Segunda unidad de tiempo:** Direccionar en la memoria (MAR) y almacenar en MBR e incrementar PC
3. **Tercer unidad de tiempo:** Transferir de MBR a IR (Instrucción a ejecutar)

# Microoperaciones

## Ciclo de captación

t1:  $MAR \leftarrow (PC)$   
t2:  $MBR \leftarrow Memory$   
     $PC \leftarrow (PC) + I$   
t3:  $IR \leftarrow (MBR)$



# Microoperaciones

## Ciclo de captación

|      |                  |
|------|------------------|
| tMAR |                  |
| MBR  |                  |
| PC   | 0000000001100100 |
| IR   |                  |
| AC   |                  |

(a) Beginning (before  $t_1$ )

|     |                  |
|-----|------------------|
| MAR | 0000000001100100 |
| MBR |                  |
| PC  | 0000000001100100 |
| IR  |                  |
| AC  |                  |

(b) After first step

|     |                  |
|-----|------------------|
| MAR | 0000000001100100 |
| MBR | 0001000000100000 |
| PC  | 0000000001100101 |
| IR  |                  |
| AC  |                  |

(c) After second step

|     |                  |
|-----|------------------|
| MAR | 0000000001100100 |
| MBR | 0001000000100000 |
| PC  | 0000000001100101 |
| IR  | 0001000000100000 |
| AC  |                  |

(d) After third step

# Microoperaciones

## Ciclo indirecto

Una vez se capta una instrucción, el siguiente paso es captar los operandos fuente. Las microoperaciones requeridas son.

1. **Primera unidad de tiempo:** Transferir a MAR la dirección del operando
2. **Segunda unidad de tiempo:** Direccionar en la memoria (MAR) y almacenar en MBR e incrementar PC
3. **Tercer unidad de tiempo:** Transferir de MBR a IR (Instrucción a ejecutar)

# Microoperaciones

## Ciclo indirecto

```
t1: MAR  $\leftarrow$  (IR (Address))  
t2: MBR  $\leftarrow$  Memory  
t3: IR (Address)  $\leftarrow$  (MBR (Address))
```

## Ciclo de interrupción

Este se presenta cuando hemos tenido una señal de interrupción durante la ejecución de la instrucción.

1. **Primera unidad de tiempo:** Transferir a MBR el valor de PC
2. **Segunda unidad de tiempo:** Transferir a MAR la dirección donde quedamos al momento de llegar la interrupción. PC ahora tiene la dirección de la instrucción de la interrupción
3. **Tercer unidad de tiempo:** Transferir a memoria el valor de MBR (Guardar el contexto)

# Microoperaciones

## Ciclo de interrupción

t1: MBR  $\leftarrow$  (PC)  
t2: MAR  $\leftarrow$  Save\_Address  
    PC  $\leftarrow$  Routine\_Address  
t3: Memory  $\leftarrow$  (MBR)

*Guarda contexto*

# Microoperaciones

## Ciclo de ejecución

Es más complejo ya que depende de las operaciones se requieran hacer en el procesador. Las micro-operaciones dependen de:

1. Número de operadores se requieren en la instrucción
2. Si se requiere almacenamiento temporal durante la ejecución
3. Donde se almacena el resultado de la operación

# Microoperaciones

## Ciclo de ejecución

Por ejemplo **ADD** R1,X.

t1:  $MAR \leftarrow (IR(address))$  } 30  
t2:  $MBR \leftarrow Memory$   
t3:  $R1 \leftarrow (R1) + (MBR)$

Ejutando

# Microoperaciones

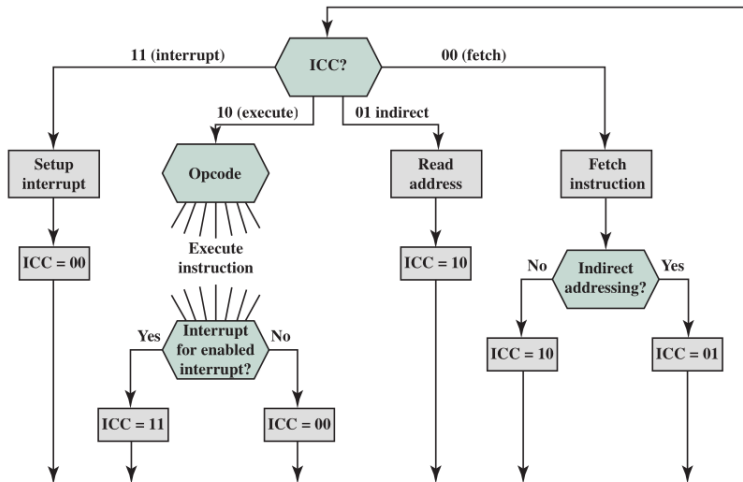
## Ciclo de Instrucción

Puede ser descompuesto en varias micro-operaciones, en el siguiente ejemplo, se ha designado el estado del procesador en términos del registro ICC(Instruction Cycle Code):

- 00 Captación
- 01 Indirecto
- 10 Ejecución
- 11 Interrupción



# Microoperaciones



# Contenido

- 1 Microoperaciones
- 2 Control del procesador
- 3 Implementación cableada

# Control del procesador

## Requisitos funcionales

De acuerdo a lo visto anteriormente, se han descompuesto las instrucciones en microoperaciones o operaciones elementales. Por ello se deben definir los requisitos funcionales de la unidad de control del procesador y por ello, se debe realizar la caracterización del procesador:

*R<sub>g</sub>, ALU, FAULT*

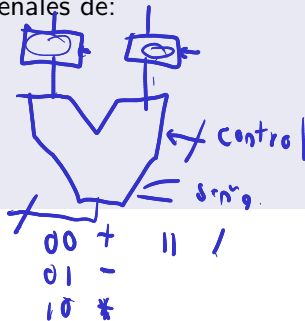
1. Definir elementos básicos del procesador
2. Describir las microoperaciones del procesador
3. ¿Que debe hacer la unidad de control para que el procesador haga las microoperaciones?

# Control del procesador

## Requisitos funcionales

La unidad de control debe regular las señales de:

1. ALU
2. Registros
3. Rutas de datos internos
4. Rutas de datos externos



# Control del procesador

## Requisitos funcionales

Las microoperaciones de un procesador se pueden clasificar de la siguiente forma:

1. Transferir datos en un registro a otro
2. Transferir datos de un registro a un E/S
3. Transferir datos de E/S a un registro
4. Realizar alguna operación (aritmética o lógica)

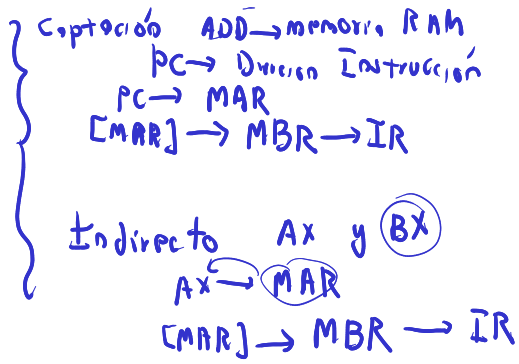
# Control del procesador

## Requisitos funcionales

La unidad de control realiza dos tareas básicas:

1. **Secuenciamiento:** Hace que el procesador avance a través de una serie de microoperaciones para realizar alguna tarea
2. **Ejecución:** Hace que ejecute cada micropoperación

ADD AX, BX



DIV BL, CL

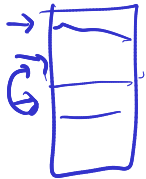
AH AL  
~ ~  
coef residuo



Ejecución {  
Valores AX, BX  
y la ALU  
Enviamos lo single  
Almacenamos en el registro  
resultado

# Interrupcion

int 21h



PC  $\rightarrow$  MBR (guardar el contexto por actual)

[ / ]  $\rightarrow$  MAR (Direccion pila)

MBR  $\rightarrow$  [MAR] (código)

MAR  $\rightarrow$  Dir. intr.

Instrucción  $\rightarrow$  MBR  
int

MBR  $\rightarrow$  PC



# Control del procesador



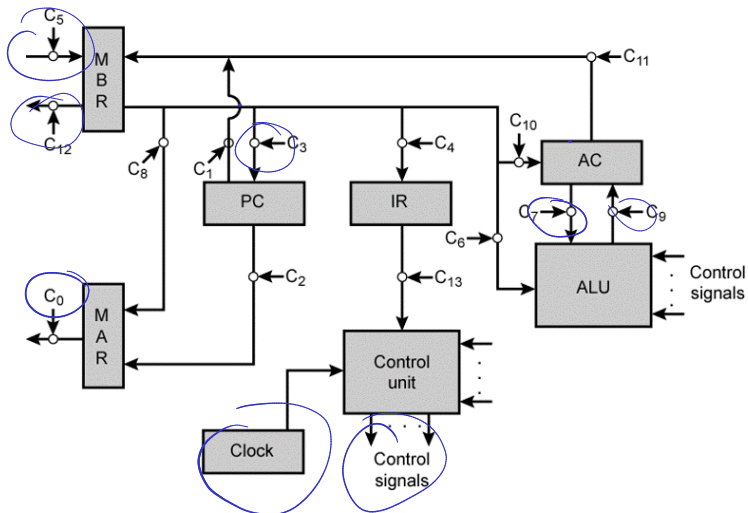
## Señales de control

Son aquellas señales para controlar los elementos en el procesador

1. **Reloj:** Para sincronizar los elementos del procesador
2. **Registro de instrucción:** Código de la operación a realizar
3. **Indicadores:** Banderas de estado del procesador
4. **Señales del bus de control:** Manejar las señales de interrupción
5. **Señales de control internas del procesador:** Registros y ALU
6. **Señales de control hacia el bus de control:** Memoria y E/S

# Control del procesador

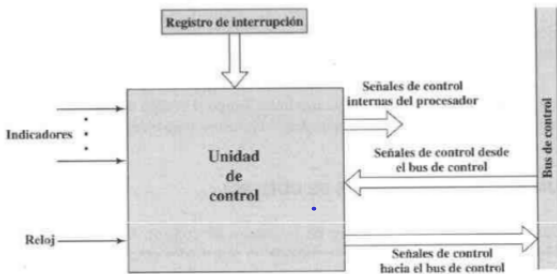
RAN



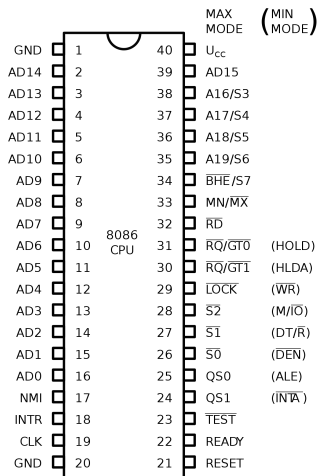
# Control del procesador

## Señales de control

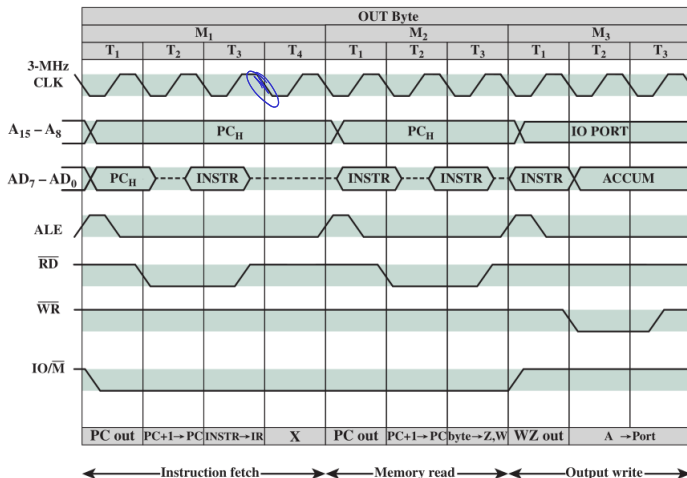
Esquema de señales de control:



# Control del procesador



# Control del procesador



# Contenido

- 1 Microoperaciones
- 2 Control del procesador
- 3 Implementación cableada**

# Implementación cableada

## Organización interna del procesador

Para que la unidad de control sea funcional, deben conectarse todas las señales hacia los elementos de la CPU y el bus de control. Existen dos tipos de implementación:

1. Implementación cableada
2. Implementación microprogramada

# Implementación cableada

$$n \quad 32 \quad \begin{array}{c} 1 \quad 1 \quad 1 \quad 1 \\ 0 \quad 0 \quad 0 \quad 0 \\ \hline 2 \quad 2 \quad 2 \quad 2 \end{array} \quad \dots \quad 2^{32}$$

## Entradas de la unidad de control

Las entradas de la unidad de control son el reloj, los indicadores, el registro de instrucción y las líneas de control del bus. Estas pueden ser vistas como una secuencia de bits y estas pueden ser codificadas como una línea de  $n$  bits. Por ello, es útil utilizar un decodificador de  $n$  entradas y  $2^n$  salidas. Con ello se garantiza:

1. Sólo se activa una línea en la salida ante un estímulo de entrada
2. Se pueden identificar las instrucciones claramente

$$\begin{array}{cccc} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & \end{array} \left. \vphantom{\begin{array}{cccc} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & \end{array}} \right\} \begin{array}{cccc} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array}$$



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

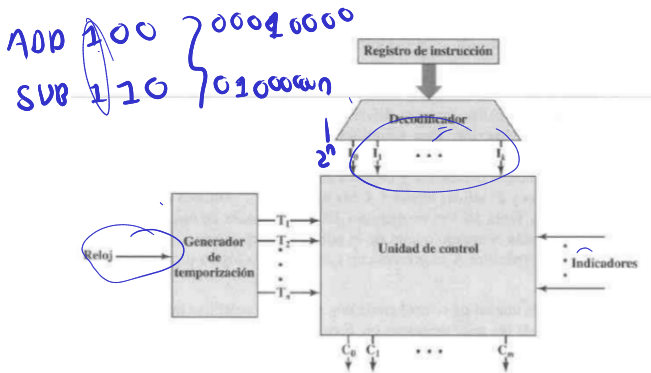
3 → 8



# Implementación cableada

## Entradas de la unidad de control

Esquema de cableado de señales de control:



# Preguntas

¿Preguntas?

Siguiente tema:  
Control microprogramado