

Guía de Ejercicios - Procesador

1. ¿Cómo funciona el ciclo de trabajo búsqueda - decodificación - ejecución?

Este ciclo se repite continuamente de una instrucción a la siguiente secuencia.

Busqueda:

- En esta etapa, la CPU obtiene la siguiente instrucción desde la memoria. La dirección de memoria de la próxima instrucción se obtiene desde el contador de programa (PC, program Counter).
- La instrucción se extrae de la memoria y se carga en el registro de instrucción de la CPU.

Decodificación:

- Una vez que la instrucción está en el registro de instrucción, la CPU la decodifica. Esto implica interpretar la instrucción y determinar la operación que debe realizarse, así como identificar las operaciones involucradas.
- La decodificación también puede involucrar la determinación del tipo de instrucción, la identificación de Registros o direcciones de memoria.

Ejecución: → LA ALU es quien devuelve un resultado por un bus interno y lo guarda en un registro.

- En esta etapa, la CPU realiza la operación especificada por la instrucción. Puede implicar operaciones aritméticas, lógicas, transferencia de datos, etc.
- Los resultados de la ejecución se almacenan en registros específicos o en la memoria, según sea necesario.

2. Indicar 5 características de los procesadores CISC.

- Pocos Registros de Procesador (Especializados)
- Muchas Instrucciones para trabajar con Memoria
- MicroArquitectura en software/Hardware Compleja
- Instrucciones Complejas (Más de un ciclo Reloj)
- Varios Modos de direccionamiento
- Muchos tipos de Datos
- Muchos formatos de Instrucción (Variables o Híbridos)
- Orientado al Hardware, Compiladores relativamente simples

3. Indicar 5 características de los procesadores RISC.

- Muchos Registros de Procesador de uso General
- Set de Instrucciones pequeños
- Solo Acceso a la Memoria a través del LOAD/STORE
- Microarquitectura en Hardware Simple
- Instrucciones simples (Un ciclo Reloj)
- Pocos Modos de direccionamiento
- Pocos tipos de Datos
- Pocos formatos de Instrucción (Fijos)
- Orientado al Software, compiladores relativamente complejos

4. Mencionar los nombres de 3 procesadores CISC y 3 RISC.

CISC →

- VAX
- Intel x86 (hasta IA-32)
- Intel 64
- IBM Mainframes
- Motorola 68K

RISC →

- SPARC
- MIPS
- ARM
- Intel Itanium (IA-64)

5. Los dos programas siguientes hacen lo mismo. Indicar cuál tarda menos si el promedio de ciclos de máquina por instrucción de este procesador RISC es de 1,2 y el del CISC es 1,5.

$$CPI(RISC) = 1.2$$

$$CPI(CISC) = 1.5$$

RISC

```
.text                                @ Indica que los siguientes
                                     @ ítems en memoria son
                                     @ instrucciones

start:
- mov r0, #15                       @ Carga el número 15 en r0
- mov r1, #20                       @ Carga el número 20 en r1
- bl func                          @ Llamado a subrutina
- swi 0x11                          @ Fin de programa
func:                               @ Subrutina
- add r0, r0, r1                    @ r0 = r0 + r1
- mov pc, lr                       @ Retornar desde subrutina
.end                                @ Marcar fin de archivo
```

```
.org 100h
.code                                @Indica que los siguientes
                                     @ítems en memoria son
                                     @instrucciones

start:
- mov al,15h                       ;Carga el número 15 en al
- mov ah,20h                       ;Carga el número 20 en ah
- call func                        ;Llamado a subrutina
- ret                              ;Fin de programa
func PROC                          ;Subrutina
- add al, ah                       ;AL = AL + AH
- ret                              ;Retorno de la subrutina
func ENDP                          ;Fin de la subrutina

end                                ;Fin de programa
```

a. Completar la siguiente tabla sin incluir las directivas:

Información Clave e Importante: Ciclo por Instrucción "CPI"

6 instrucciones
el resto
son
directivas

RISC		CISC	
N número de instrucciones	CPI promedio de ciclos de máquina por instrucción	N número de instrucciones	CPI promedio de ciclos de máquina por instrucción
6	1.2	6	1.5

Ahora solo queda hacer

$$N_{\text{Instrucciones}} * CPI \text{ (RISC o CISC)}$$

$$\text{RISC} \rightarrow 6 \cdot 1.2 = 7.2 \text{ ciclos}$$

$$\text{CISC} \rightarrow 6 \cdot 1.5 = 9.0 \text{ ciclos}$$

b. Comparar $N_{\text{RISC}} \cdot CPI_{\text{RISC}}$ con $N_{\text{CISC}} \cdot CPI_{\text{CISC}}$

Las N_{RISC} y CPI_{RISC} tardan menos que las N_{CISC} y CPI_{CISC} .

6. El tiempo de ejecución de un programa en alto nivel es $T = N \times CPI / f_{\text{ck}}$. En esta expresión N es la cantidad de instrucciones ejecutadas, CPI es el número de ciclos de máquina promedio por instrucción y f_{ck} es la frecuencia de reloj del procesador.

El mismo programa se ejecuta en una máquina CISC y en una RISC con un promedio de ciclos de máquina por instrucción del procesador RISC de 1,2 y de 1,5 en el procesador CISC.

- Si las dos máquinas tardan lo mismo en ejecutar el programa y la frecuencia de reloj es la misma en ambos procesadores, ¿qué porcentaje de instrucciones N_{RISC} representa N_{CISC} ?
- ¿Qué porcentaje de instrucciones N_{RISC} representa N_{CISC} si la frecuencia de reloj del procesador RISC es 12% mayor a la frecuencia de reloj del procesador CISC y tardan lo mismo en ejecutar el programa?

q)

$$T_{\text{RISC}} = \frac{N_{\text{RISC}} * CPI_{\text{RISC}}}{f_{\text{ck}}}$$

$$T_{\text{CISC}} = \frac{N_{\text{CISC}} * CPI_{\text{CISC}}}{f_{\text{ck}}}$$

$$T = \frac{N_{\text{RISC}} * CPI_{\text{RISC}}}{f_{\text{ck}}}$$

$$T = \frac{N_{\text{CISC}} * CPI_{\text{CISC}}}{f_{\text{ck}}}$$

$$\frac{N_{\text{RISC}} * CPI_{\text{RISC}}}{f_{\text{ck}}} = \frac{N_{\text{CISC}} * CPI_{\text{CISC}}}{f_{\text{ck}}}$$

$$N_{\text{RISC}} * CPI_{\text{RISC}} = N_{\text{CISC}} * CPI_{\text{CISC}}$$

$$N_{\text{CISC}} = N_{\text{RISC}} * \frac{CPI_{\text{RISC}}}{CPI_{\text{CISC}}}$$

$$N_{\text{CISC}} = N_{\text{RISC}} * \frac{1.2}{1.5}$$

$$N_{\text{CISC}} = 0.8 N_{\text{RISC}}$$

N_{CISC} representa el 80% de N_{RISC}

- b. ¿Qué porcentaje de instrucciones NRISC representa NCISC si la frecuencia de reloj del procesador RISC es 12% mayor a la frecuencia de reloj del procesador CISC y tardan lo mismo en ejecutar el programa?

b)

$$f_{ck(RISC)} = 1.12 * f_{ck(CISC)}$$

$$T_{RISC} = \frac{N_{RISC} * CPI_{RISC}}{f_{ck}}$$

$$T_{CISC} = \frac{N_{CISC} * CPI_{CISC}}{f_{ck}}$$

$$T = \frac{N_{RISC} * CPI_{RISC}}{f_{ck}}$$

$$T = \frac{N_{CISC} * CPI_{CISC}}{f_{ck}}$$

$$\frac{N_{RISC} * CPI_{RISC}}{f_{ck}} = \frac{N_{CISC} * CPI_{CISC}}{f_{ck}}$$

$$N_{RISC} = \frac{f_{ck(RISC)} * N_{CISC} * CPI_{CISC}}{f_{ck(CISC)} * CPI_{RISC}}$$

$$N_{RISC} = \frac{1.12 * \cancel{f_{ck(CISC)}} * N_{CISC} * CPI_{CISC}}{\cancel{f_{ck(CISC)}} * CPI_{RISC}}$$

$$N_{RISC} = \frac{1.12 * N_{CISC} * 1.5}{1.2}$$

$$N_{RISC} = 1.4 N_{CISC}$$

$$\frac{1}{1.4} = \frac{N_{CISC}}{N_{RISC}} \Rightarrow N_{CISC} = 0.714 N_{RISC}$$

N_{CISC} representa el 71,43 % de N_{RISC}

7. Un programa de prueba se está ejecutando en un procesador de 40 MHz. El código objeto consta de 100.000 instrucciones, con el siguiente conjunto de instrucciones y ciclo de reloj:

Tipo de Instrucción	Contador de instrucciones	Ciclo de reloj
Aritméticos enteros	45.000	1
Datos transferidos	32.000	2
Punto flotante	15.000	2
Control transferidos	8.000	2

Determinar el CPI efectivo, la velocidad en MIPS, y el tiempo de ejecución para este programa.

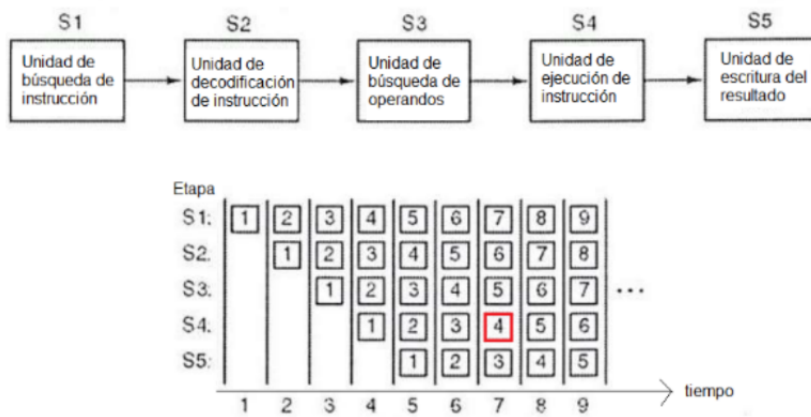
8. En base a la siguiente figura se pide:

a. Explicar el concepto de pipelining.

b. ¿Qué representa el número 4 marcado en rojo en la figura?

c. Indicar todo lo que ocurre durante el tiempo 6.

Pipelining de 5 etapas



a)

Guía de Ejercicios - Memoria

memoria

1. ¿Cómo se llama, en una memoria, el tiempo que transcurre desde que se aplica una dirección en la entrada, hasta que el dato almacenado se tiene en la salida?

Este tiempo se lo conoce como Tiempo de Acceso (Access Time)

Es el lapso que transcurre desde que se envía una solicitud de lectura / escritura a la memoria hasta que el dato está disponible en la salida o ha sido almacenado.

2. Clasificar los siguientes niveles de memoria según su capacidad, tiempo de acceso y costo con 1 para la de mayor capacidad y 5 para la de menor, 1 para la de menor tiempo de acceso y 5 para la de mayor, 1 para la de menor precio y 5 la de mayor

	Registro	Cache	Principal	Unidades de disco	Cinta magnética
Capacidad	5	4	3	2	1
Tiempo de acceso	1	2	3	4	5
Velocidad de transferencia	1	2	3	4	5
Costo	5	4	3	2	1

Capacidad: 1 (Mayor), 5 (Menor)

Tiempo de Acceso: 5 (Mayor), 1 (Menor)

Velocidad de Transferencia: 5 (Mayor), 1 (Menor).

Costo - Precio: 5 (Mayor), 1 (Menor)

Registros: Son las memorias mas rapidas, de menor capacidad y mas cara (por bit), directamente dentro del procesador

Cache: Mas Lenta y de Mayor capacidad al que los registros, pero mucho mas rapida y pequeña que la memoria Principal.

Memoria Principal (RAM): Es la Memoria de trabajo del sistema, más lenta y de mayor capacidad que la cache, pero más rapida que el almacenamiento Secundario

Unidad de Disco (HDD/SSD): Almacenamiento Secundario, mucho mas lento pero con mayor capacidad y menor costo que la principal

Cinta Magnetica: Almacenamiento terciario o de Archivo, el más lento, de mayor capacidad y el mas barato por bit.

3. Dado un circuito integrado de memoria ROM de 512 x 4 indicar:

- a. Cuántas entradas de dirección tiene
- b. Cuántas entradas/salidas de datos tiene
- c. Cuántas palabras y de qué tamaño almacena

a) $N = \text{"Entradas de direcciones"}$ o sea debe cumplir

$$\text{que } 2^N = 512 \Rightarrow \log_2(2^N) = \log_2(512)$$

Tiene 9 entradas
de direcciones

$$N = \log_2(512)$$

$$N = \log_2(2^9)$$

$$\boxed{N = 9}$$

b) tiene un total de 4 entradas/salidas de datos para cada dirección

c) Almacena un valor de 512 palabras y de 4 bits para cada una

ROM de 512×4

Capacidad

Longitud de cada palabra y cantidad de Entrada/Salida de datos

4. Considere la parte del procesador y la RAM del sistema que se muestra a continuación

a. ¿Cuántas palabras contiene la RAM y con qué longitud de bits?

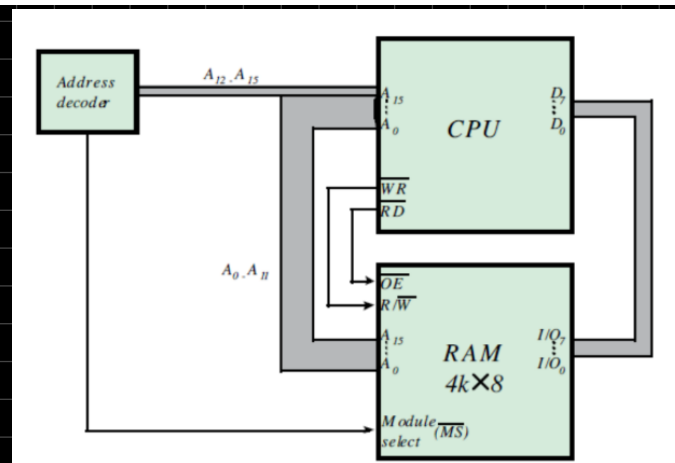
b. Para escribir en la RAM, las entradas de dirección A0 a A11 deben estar activadas, entonces, el selector de módulo de memoria MS debe estar habilitado. ¿La habilitación debería lograrse con un "1" o un "0"?

c. Al mismo tiempo, la única entrada lectura R y escritura W se mantiene a un nivel constante; ¿este nivel debería ser un nivel ALTO o un nivel BAJO?

d. ¿La habilitación de salida OE debe estar en ALTO o BAJO?

e. ¿I/O0 a I/O7 son entradas o salidas para la operación?

f. ¿Qué líneas constituyen el bus de control en la figura?



5. ¿Cuál es la diferencia entre una memoria volátil y una perenne o no volátil?

- Una memoria Volatil es aquella que pierde su contenido (los datos almacenados) cuando se interrumpe el suministro de energía Eléctrica. (Por ejemplo Apagar la computadora). La RAM es un ejemplo perfecto.
- Una memoria perenne o NO Volatil es aquella que retiene su contenido (los datos almacenados) incluso cuando se interrumpe el suministro de energía Eléctrica. Ejemplos: ROM, PROM, EPROM y Flash son ejemplos de Memorias No volátiles.

6. Para las siguientes memorias indicar el significado de las siglas, las características que derivan de su nombre y si es volátil o no volátil:

RAM, ROM, PROM, EPROM, EEPROM, FLASH

- RAM (Random Access Memory - Memoria de Acceso Aleatorio)

* Características: Permite el Acceso Directo (Aleatorio) a cualquier celda de memoria en el mismo tiempo, sin importar su ubicación. Es la memoria principal del sistema, es rápida pero suele costar más que el almacenamiento secundario.

* Volatil

- ROM (Read Only Memory - Memoria de Solo Lectura)

(*) Características: Su contenido se graba durante la fabricación y no puede ser Modificado por el usuario durante la operación normal. Se usa para almacenar firmware o BIOS.

(*) NO Volatil

- PROM (Programmable Only Memory)

* Característica: Es una ROM que puede ser programada una sola vez por el usuario (o Fabricante). Una vez programada, su contenido es permanente.

(*) NO Volatil

- EPROM (Erasable Programmable Read Only Memory)
Memoria de Solo Lectura programable Borrable

(*) Característica: Se puede programar, y su contenido puede borrarse exponiéndola a luz ultravioleta a través de una ventana en su encapsulado. Una vez borrada puede ser reprogramada.

* No Volatil

EEPROM (Electrically Erasable Programmable Read Only Memory)
Memoria de Solo Lectura Programable Borrable

Electricamente

* Característica: Similar a la EPROM, pero su contenido puede borrarse y reprogramarse eléctricamente, sin necesidad de luz ultravioleta. Esto permite la reescritura en el circuito.

* No-Volátil

FLASH (Es un tipo de EEPROM también llamada Flash EEPROM)

* Característica: Permite Borrar y reprogramar Bloques enteros de Memoria (a diferencia de EEPROM que borra byte por byte)

Es muy utilizada en unidades de estado sólido, memorias USB, tarjetos SD, etc., por su alta densidad de Almacenamiento y velocidad.

* No Volátil

7. Diseñar un banco de memoria RAM de 1 GByte. Se dispone de un circuito integrado de 512 M palabras de 4 bits y dos de 256 M palabras de 4 bits. Se pide:

- Indicar cuántos y cuáles circuitos integrados se van a usar.
- Armar un mapa de memoria en el que se vea la primera dirección y la última para cada circuito integrado.
- Decodificar los selectores de integrado (Chip Select).
- Dibujar el circuito.

$$1GB = 1024 MByte = 2^{30} \rightarrow A_{29} \dots A_0$$

$$RAM \rightarrow 1024 MByte = 512 MByte + 2 \times 256 MByte$$

$$\begin{array}{lcl} \text{a)} & \text{IC1} & 512 MBytes \rightarrow 2^9 \times 2^{20} = 2^{29} \rightarrow A_{28} \dots A_0 \\ & \text{IC2} & 256 MBytes \rightarrow 2^8 \times 2^{20} = 2^{28} \rightarrow A_{27} \dots A_0 \\ & \text{IC3} & 256 MByte \rightarrow 2^8 \times 2^{20} = 2^{28} \rightarrow A_{27} \dots A_0 \end{array}$$

Se usaran 3 circuitos integrados 1 de 512 MB y 2 de 256 MByte.

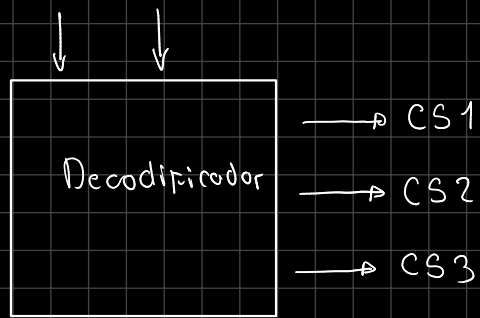
b)

A ₂₉	A ₂₈	A ₂₇	A ₂₆	A ₂₅	A ₂₄	...	A ₀	
0	0	0	0	0	0		0	1° IC1
0	1	1	1	1	1	...	1	Último IC1
1	0	0	0	0	0	...	0	1° IC2
1	0	1	1	1	1	...	1	Último IC2
1	1	0	0	0	0	...	0	1° IC3
1	1	1	1	1	1	...	1	Último IC3

c)

	A29	A28	A27	A26	A25	
IC1	0	x	x	x	x	CS1
IC2	1	0	x	x	x	CS2
IC3	1	1	x	x	x	CS3

Entradas:
2



8. Diseñar un banco de memoria RAM de 768 MBytes y 256 Bytes de ROM. Para la memoria RAM se dispone de dos circuitos integrados de 256 Mbytes y dos de 128 MBytes. Para la memoria ROM se dispone de un circuito integrado de 128 MBytes, uno de 64 MBytes y dos de 32 Mbytes. Se pide:

- Indicar cuántos y cuáles circuitos integrados se van a usar.
- Armar un mapa de memoria en el que se vea la primera dirección y la última para cada circuito integrado.
- Decodificar los selectores de integrado (Chip Select).
- Dibujar el circuito.

a) RAM 768 MBytes $\rightarrow 2 \times 256 + 2 \times 128 = 512 + 256 = 768 \text{ MBytes}$

IC1 256 Mbytes = $2^8 \times 2^{20} = 2^{28} \rightarrow A0 \text{ hasta } A27$
 IC2 256 Mbytes = $2^8 \times 2^{20} = 2^{28} \rightarrow A0 \text{ hasta } A27$
 IC3 128 Mbytes = $2^7 \times 2^{20} = 2^{27} \rightarrow A0 \text{ hasta } A26$
 IC4 128 Mbytes = $2^7 \times 2^{20} = 2^{27} \rightarrow A0 \text{ hasta } A26$

ROM 256 MBytes

IC5 128 MBytes = $2^7 \times 2^{20} = 2^{27} \rightarrow A0 \text{ hasta } A26$
 IC6 64 MBytes = $2^6 \times 2^{20} = 2^{26} \rightarrow A0 \text{ hasta } A25$
 IC7 32 MBytes = $2^5 \times 2^{20} = 2^{25} \rightarrow A0 \text{ hasta } A24$
 IC8 32 MBytes = $2^5 \times 2^{20} = 2^{25} \rightarrow A0 \text{ hasta } A24$

b) $768 + 256 = 1024 \text{ MByte} = 1 \text{ GByte} = 2^{30} \rightarrow A0 \text{ hasta } A29$

A29	A28	A27	A26	A25	A24	...	A0	
0	0	0	0	0	0	...	0	1° dir de IC1
...
0	0	1	1	1	1	...	1	última dir IC1

+1

A29	A28	A27	A26	A25	A24	...	A0	
0	1	0	0	0	0	...	0	1° dir \rightarrow IC2
...
0	1	1	1	1	1	...	1	última dir \rightarrow IC2

+1

A29	A28	A27	A26	A25	A24	...	A0	
1	0	0	0	0	0	...	0	1° dir \rightarrow IC3
...
1	0	0	1	1	1	...	1	última dir \rightarrow IC3

+1

A29	A28	A27	A26	A25	A24	...	A0	
1	0	1	0	0	0	...	0	1° dir \rightarrow IC4
...
1	0	1	1	1	1	...	1	última dir \rightarrow IC4

+1

A29	A28	A27	A26	A25	A24	...	A0	
1	1	0	0	0	0	...	0	1° dir \rightarrow IC5
...
1	1	0	1	1	1	...	1	última IC5

+1

1	1	1	0	0	0	-----	0	1º dir → IC6
⋮	⋮	⋮	⋮	⋮	⋮		⋮	
1	1	1	0	1	1		1	Ultima → IC6

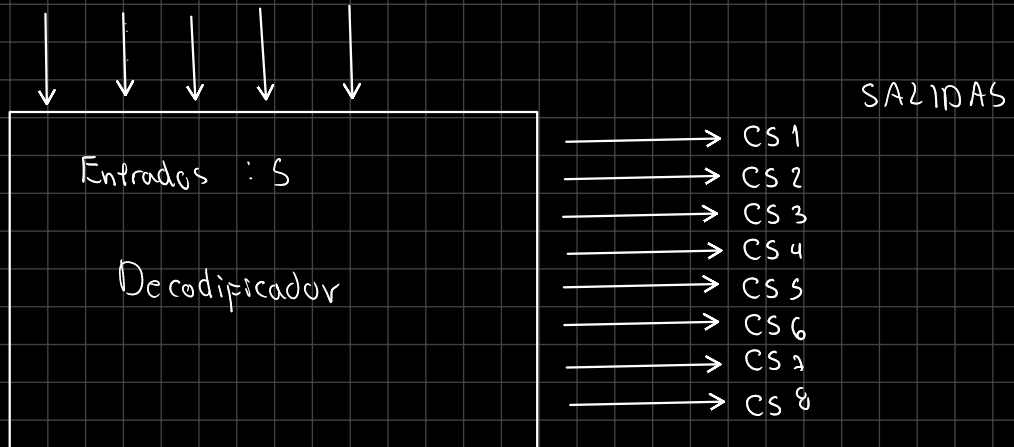
+ 1

1	1	1	1	0	0	-----	0	1º IC7
⋮	⋮	⋮	⋮	⋮	⋮		⋮	
1	1	1	1	0	1	-----	1	Ultima IC7

1	1	1	1	1	0	-----	0	1º IC8
⋮	⋮	⋮	⋮	⋮	⋮		⋮	
1	1	1	1	1	1	-----	1	Ultima IC8

c)

	A29	A28	A27	A26	A25	A24	A23	-----	AO	
IC1	0	0	x	x	x	x	x	-----	x	CS1
IC2	0	1	x	x	x	x	x	-----	x	CS2
IC3	1	0	0	x	x	x	x	-----	x	CS3
IC4	1	0	1	x	x	x	x	-----	x	CS4
IC5	1	1	0	x	x	x	x	-----	x	CS5
IC6	1	1	1	0	x	x	x	-----	x	CS6
IC7	1	1	1	1	0	x	x	-----	x	CS7
IC8	1	1	1	1	1	x	x	-----	x	CS8



Ejemplo para el CS6

IC6 1 1 1 0 x CS6

A29 - - - -
A28 - - - -
A27 - - - -
A26 - - - -



-----> CS6

o Not 26

