

Hasta 17:50

Oscar Cabrerizo

Alejandro

Abrilhan

102256

ej 2) Un procesador ejecuta:

ORCC %R10, %R0, %R7.

Para la microinstrucción 1609:

1608: iF R [IRE[13]] then Goto 1610,

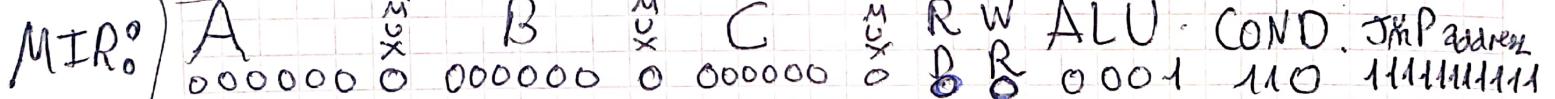
1609: R [rd] ← ORCC (R [r10], R [tempo]) ;

Goto 2047;

• Para la microinstrucción 1609. :

a) bit contenido en el Registro de Microinstrucciones.

de ORCC %R10, %R0, %R7 → Identificación que.
vía del IR → r10 r0 r7
 r21 r22 rd.



A, B y C estan en el registro
vía del IR y accesible al
programador por eso van a ser
programados por el programador.

No tienen memoria

Al memorizar
por eso van
a ser ceros.

Vienen un
salto

incondicional
a 2047

o dirección
2047

2047 → 10
binario.

ALU da como salida ORCC → F₃ F₂ F₁ F₀
0001

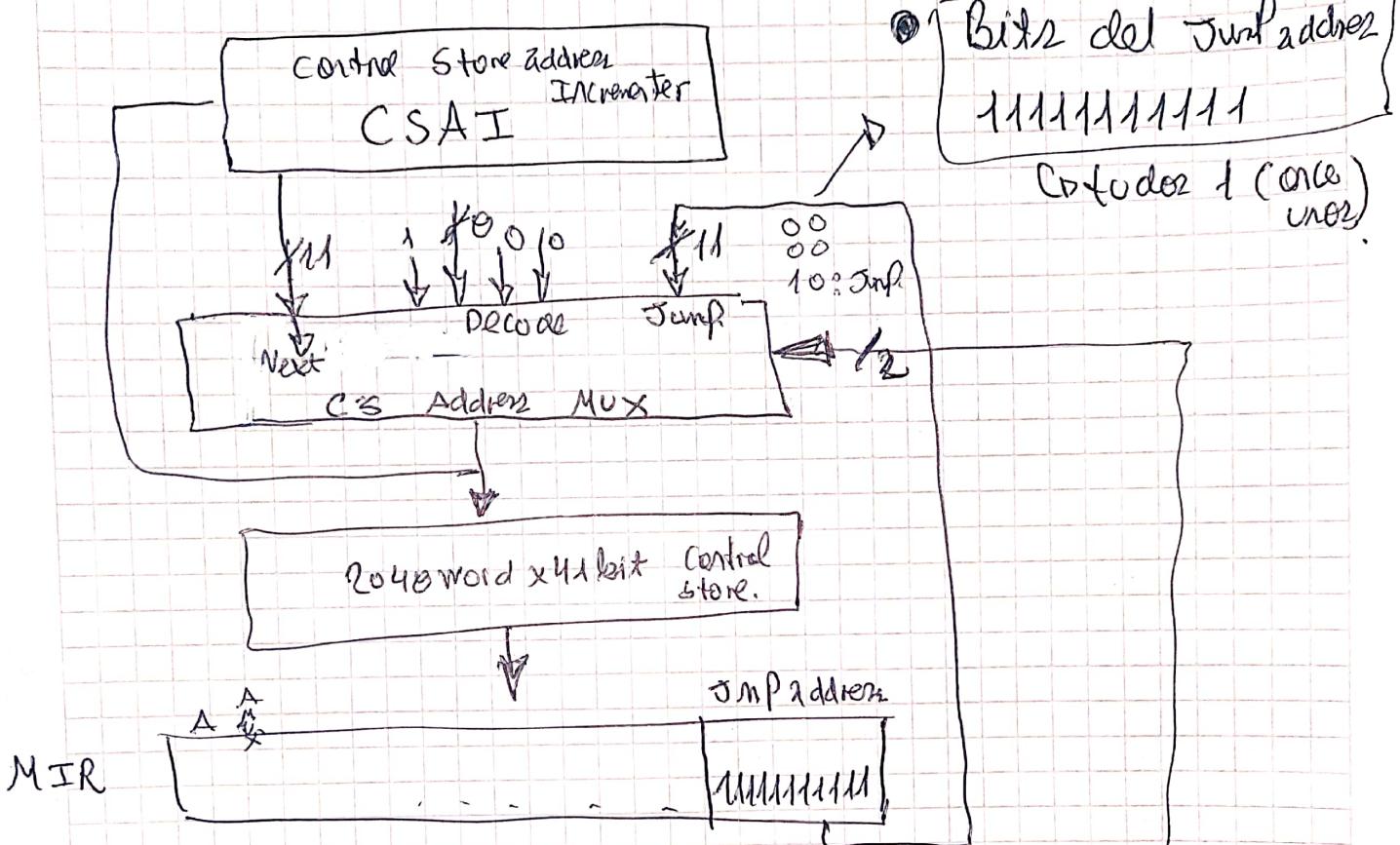
Por esa razón
uso 110.

El 2047
en
binario.

11
21
31
41
507 41 bits.

Oscar Cabral
Alejandro
Abraham
102256

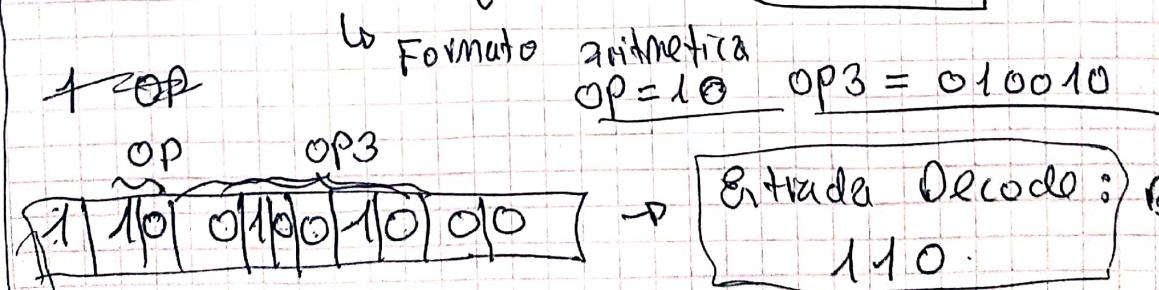
- b) Bits de las entradas del multiplexor de dirección de la memoria de control.



- Valores en **Next**:
Contendrá la sigt microinstrucciones.
oí decir: 1610
Incrementa en uno la actual (1609)

CB2

- En **Decode** decodifiquemos

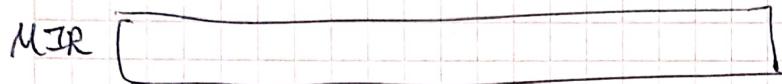


$$\text{Rpta } 1610_{10} = 2^{10} + 2^9 + 2^6 + 2^3 + 2^1 = 11001001010$$

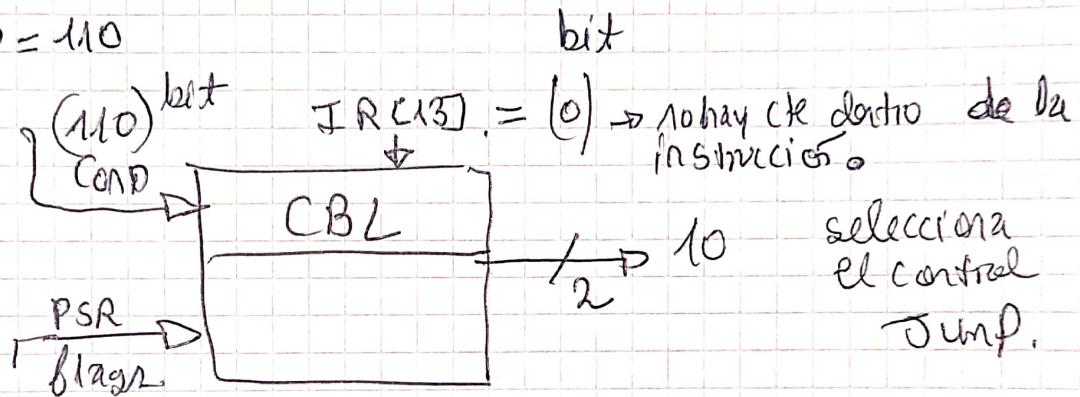
Bits del **Next**

Oscar Cabrerizo
 Alejandro Abrahán
 102256
 L.

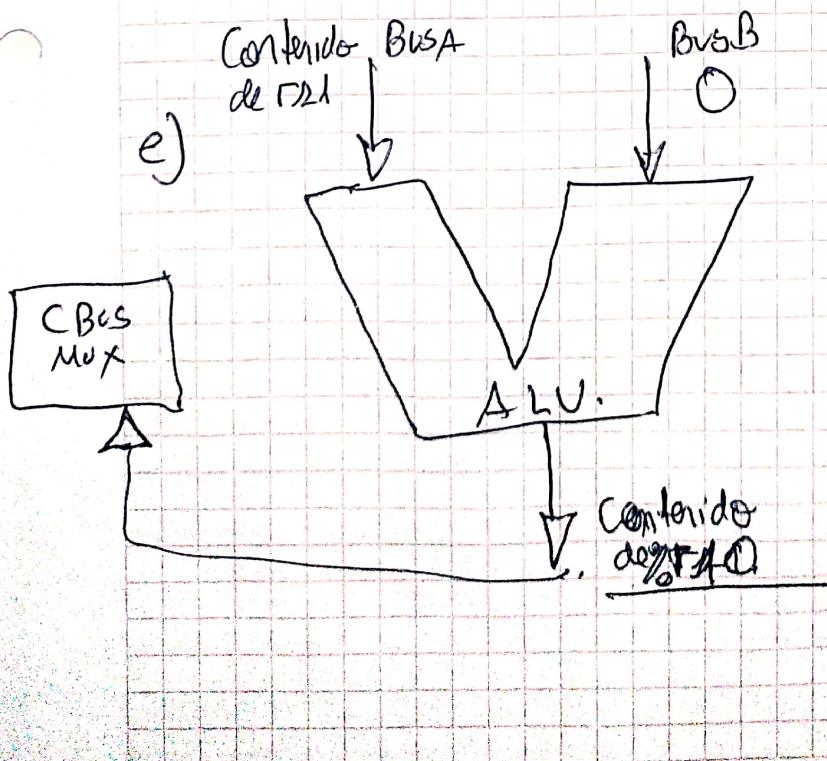
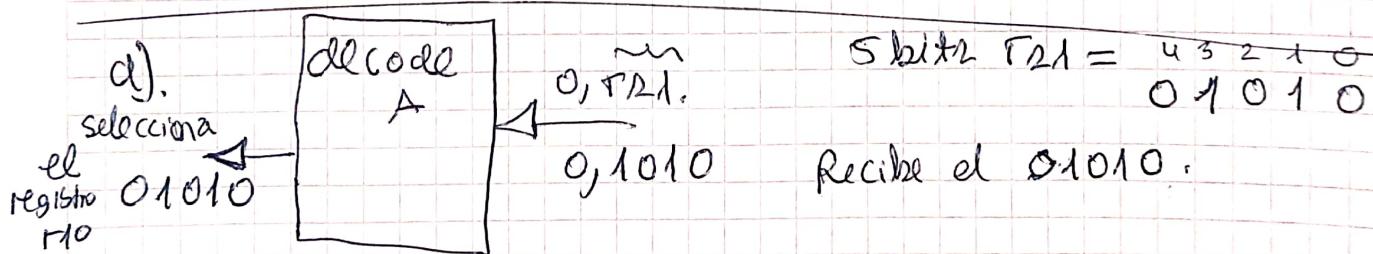
c) Entrada y Salida de la lógica de control de salto. (CBL).



Cuando cond = 110



AZVC.



En un ORCC, Si tomo con zero en cont suel queda tal cual.