

hasta 17:50

Oscar Cabrera
Alejandro
Abraham
102256

e-2) Un procesador ejecuta:

ORCC %r10, %r10, %r7.

Para la microinstrucción 1609:

1608: IF R[IR[13]] then Goto 1610;

1609: R[ra] ← ORCC(R[ra], R[tempo]);
Goto 2047;

• Para la microinstrucción 1609:

a) bit contenido en el registro de Microinstrucciones.

de ORCC %r10, %r10, %r7 → Identificamos que.

viene del IR

MIR: A B C RW ALU COND. Jmp Address
000000 0 000000 0 000000 0 0 0 0001 110 1111111111

A, B y C están en el registro
del IR y no son
programados por eso van a 0.

No tenemos
ni en memoria
por eso van a 0.

ALU hace un ORCC → F3 F2 F1 F0
0001

11
21 bits
41 bits

son 41 bits.

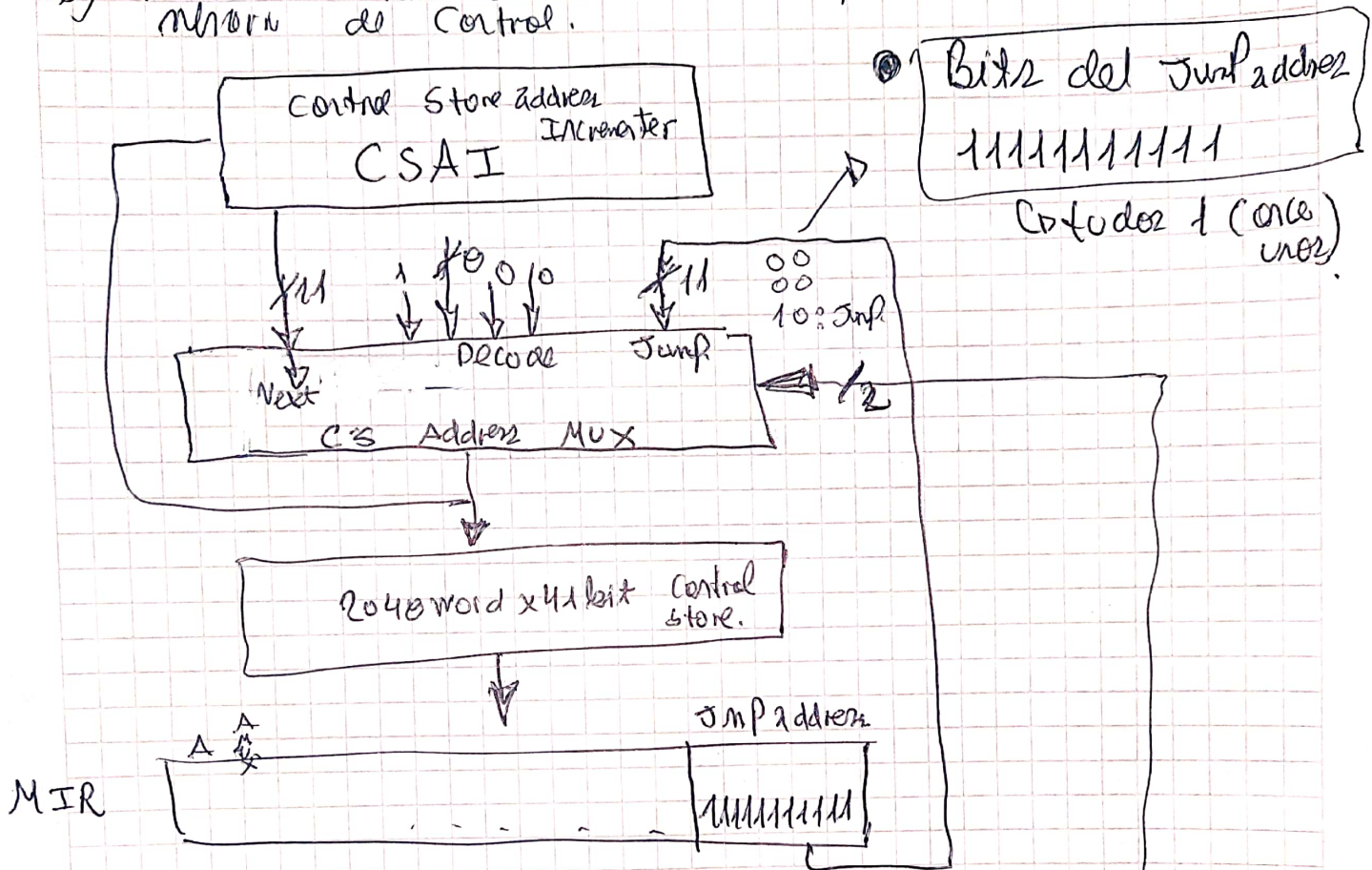
Venimos un
salto
incondicional
a 2047
dirección
2047.
Por esa razón
usamos 110.

son
ON/OFF
un bit

2047
10
El 2047
en
binario.

Oscar Caballero
Alejandro
Abraham
102256

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



• Vamos en NEXT.
Contiene la segt microinstrucciones.
es decrt. 1610
Incrementa en una la actual (1609)

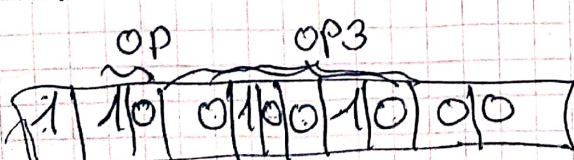
• En Decode decodificamos

↳ Formato aritmética

1 2 3 4 5 6 7 8 9 10

op = 10

op3 = 010010



Entrada Decode: Bits del Decode.
110.

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

Bits del Next

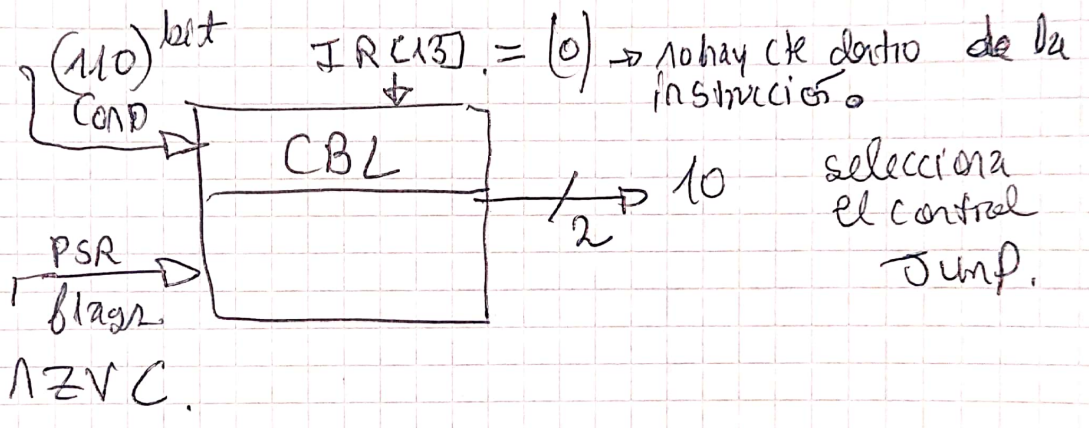
Oscar Cabrera
 Alejandro
 Abraham
 102256

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

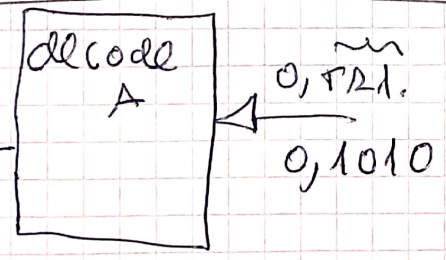
MIR

Cond = 110

bit



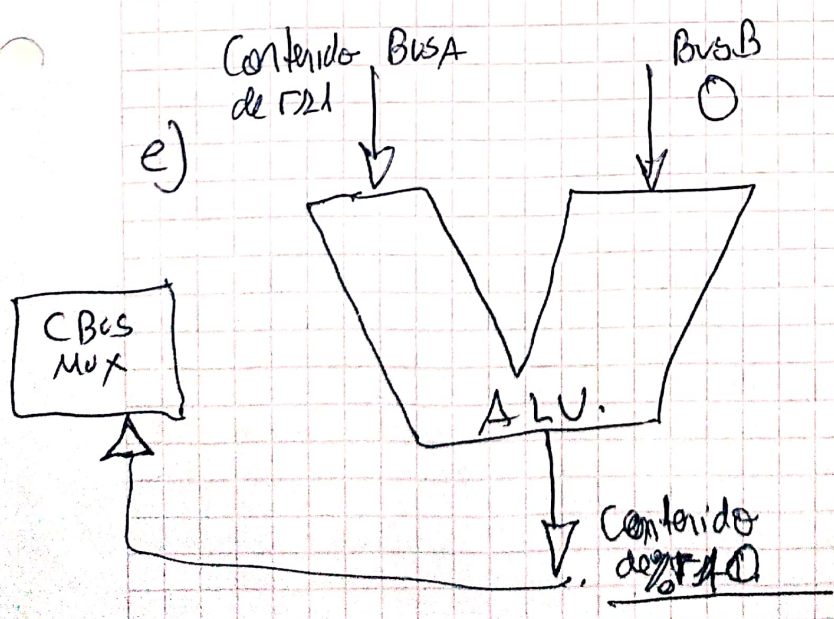
d). selecciona el registro r10 01010



5 bits r21 = 4 3 2 1 0
 0 1 0 1 0

Recibe el 01010.

e) Contenido BusA BusB



En un orcc se como con 960 ex como son queda tal cual.