

Ejercicio

MICROINSTRUCCIÓN

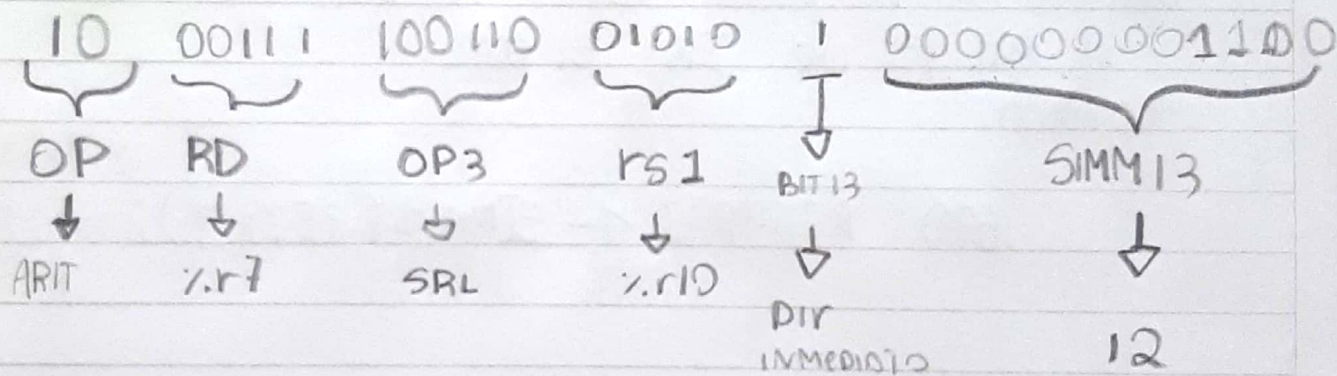
15-02-2022

* PROPONER un microcódigo que implemente LA INSTRUCCIÓN:

SRL $\frac{1}{2}$ r10, 12, $\frac{1}{2}$ r7

JUSTIFICAR LAS DIRECCIONES DE MEMORIA DONDE SE ALMACENA EL CÓDIGO PROPUESTO.

- LA INSTRUCCIÓN es de FORMATO ARITMETICO y tiene direccionamiento inmediato...



* Con esto podemos deducir la dirección en la ROM donde se almacena

Decode

1 10 100 110 00 = 1688

OP + OP3

$$OP + OP3$$

* 1688 es donde debe empezar nuestro micro video

* CODIGO Propuesto

```
1688 : IF IR[13] THEN GOTO 1690;  
1689 : R[rd] ← SRL (R[rs1], R[rs2]);  
      GOTO 2047;  
1690 : R[Temp0] ← SIMM13(R[IR]);  
1691 : R[rd] ← SRL (R[rs1], R[Temp0]);  
      GOTO 2047;
```

- TRAS el SEGUNDO CLOCK, INDICAR SALIDAS Y ENTRADAS ...

- en el segundo CLOCK se estaría ejecutando LA LINEA 1690, YA QUE el BIT 13 de IR SI ESTÁ encendido.

1690: R[Temp0] ← SIMM13(R[IR]);

* CBL

- ENTRADAS

* FLAGS %PSR → 0000

* BIT 13 IR → 1

~~* ALU MIR → 1011 → SIMM13~~

* COND MIR → 000 → NEXT ADDR

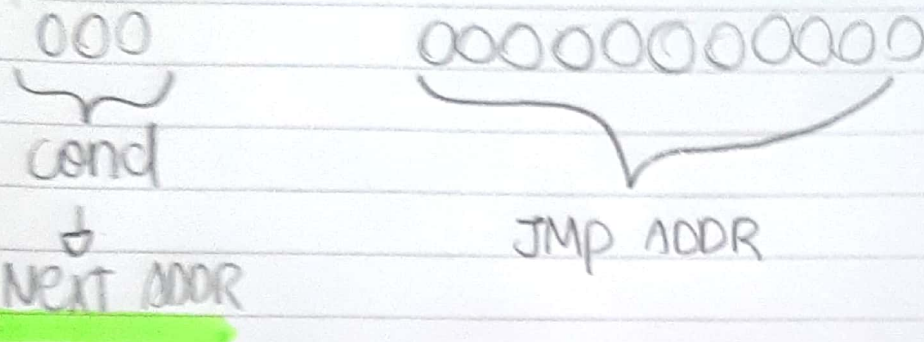
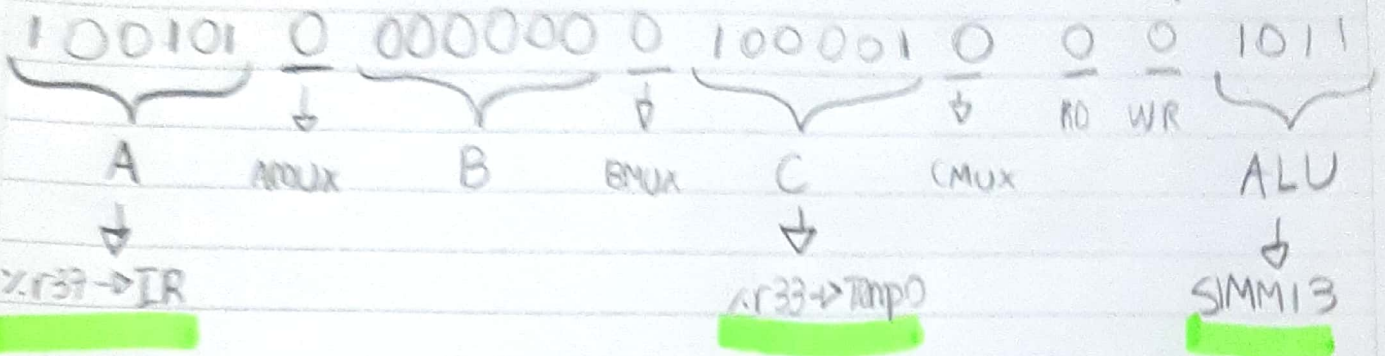
- SALIDAS

ÚNICA SALIDA

en 00 = NEXT

* 41 BITS DEL MIR

NO HAY
ACCESO A MEM



* ARMUX } Le hacemos caso al MIR y
* CMUX } NO AL IR.

* Línea de datos del BUS A.

- Como tenemos en el BUS A al IR.
El valor del BUS A es igual a la
instrucción de assembly = SRL x.r10, 12, x.r7.

BUS A = 10 00111 100110 01010 1 0000001100

* CS ADDRESS MUX

- ENTRADAS

* NEXT = 1691

* Decode = 10100110 (IR)

* CBL = 00 (next addr)

* Jump = 000000000000

- SALIDA

Salida única con

16A1 ya que

CBL mandó un 00.

USA NEXT ADDR.