- Un procesador ARC ejecuta la instrucción assembler: srl %r10, 12, %r7.

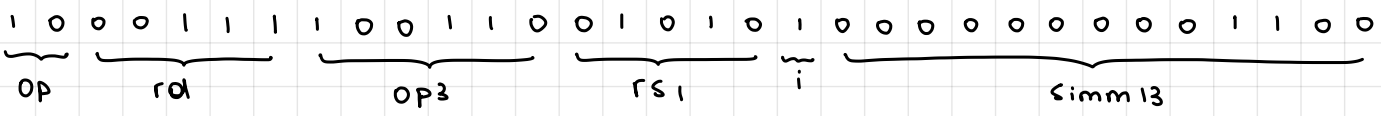
Se pide:

a) Escribir un microcódigo que implemente esta instrucción de assembler indicando en qué direcciones de la memoria de control estará almacenado.

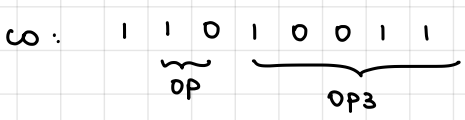
b) Para las siguientes líneas del microcódigo propuesto indicar los valores binarios (o su equivalente en decimal o hexadecimal) en los siguientes puntos de la microarquitectura:

- Entradas y salidas de la lógica de control de saltos
- Líneas de datos del bus A
- Todos los bits del registro de microinstrucciones
- CS Address MUX

Construyo el formato del registro %ir:



Decodifico: 1 1 0 1 0 0 1 1 0 0 0 = 1688_{10} → el código comienza en 1688.



1688: if [IR[13]] THEN GOTO 1690;

1690: R[tempØ] ← SIMM13 (R[ir]);

1691: R[rd] ← SRL (R[rs1], R[tempØ]); GOTO 2047;

2047: R[pc] ← INCPC (R[pc]); GOTO Ø;

Ø: R[ir] ← AND (R[pc], R[pc]); READ;

1: DECODE;

1690‖:

A = 1 0 0 1 0 1 → registro %ir ≡ r37

AMUX = 0 → uso el registro que guarda el campo A de la MIR y no rs1

B = X
BMUX = X
} no se usa ⇒ no importa qué info vengo

C = 1 0 0 0 0 1 → registro %tempØ ≡ r33

CMUX = 0 → uso el registro que guarda el campo C de la MIR y no rd

RD = 0
WR = 0
} no leo ni escribo la memoria principal

ALU = 1 0 1 1 → el código en ALU para la operación SIMM13

COND = 0 0 0 → la siguiente línea a ejecutar es la consecutiva

JAMP ADDR = X → no hay salto condicional ⇒ no importa qué info. vengo