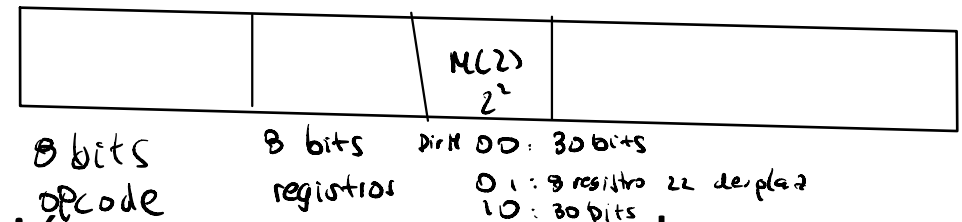


# Problema 1:

- Sea un procesador con las siguientes características:
  - Conjunto de 150 instrucciones
  - Banco de 256 registros
  - Memoria de 1 Giga-posiciones
  - Cada instrucción tiene dos operandos explícitos:
    - 1<sup>er</sup> operando: Indica siempre un registro (direccionamiento directo a registro)
    - 2<sup>o</sup> operando: Indica siempre una posición de memoria
  - Los posibles modos de direccionamiento para el 2<sup>o</sup> operando son:
    - Directo a memoria
    - Relativo (registro explícito)
    - Indirecto por memoria



- Diseña el formato de instrucción para este procesador, indicando los campos que lo componen así como la longitud en bits de cada uno.

# Problema 2b:

- Diseña el formato de instrucción de un procesador:
  - El tamaño de la instrucción debe ser 16 bits.
  - El formato debe admitir un máximo de 64 instrucciones.  $\approx 2^6$
  - El procesador cuenta con un banco de 32 registros.  $= 2^5$
  - Todas las instrucciones tienen un único operando explícito.
  - Habrá tres tipos de instrucciones distintas según sea el modo de direccionamiento de su operando explícito:
    - Tipo1: operando con modo de direccionamiento directo registro.
    - Tipo2: operando con modo de direccionamiento relativo con registro explícito.
    - Tipo3: operando con modo de direccionamiento directo a memoria.

Tipo1	6 bits opcode	5 bits registro	XXXXX
Tipo2	6 bits opcode	5 bits registro	5 bits desplz.
Tipo3	6 bits opcode	10 bits memoria	

- ¿Máximo desplz. en instrucciones Tipo2 (Natural)?  $11111 = 31$
- ¿Capacidad de memoria direccionable Tipo3 (KB)?  $2^{10} = 1024$   
1 KB

# Problema 5:

[2 PUNTO(S)] Sea un procesador que posee una instrucción suma con dos operandos explícitos: **ADD OP1, OP2**, cuya acción es:  $OP1 \leftarrow OP1 + OP2$ . El formato de instrucción es de 16 bits repartidos como se muestra a continuación:

Formato de instrucción		
Opcode	OP1	OP2
0110		
4 bits	6 bits	6 bits

El modo de direccionamiento de ambos operandos (OP1 y OP2) es el mismo. Dado el contenido (en hexadecimal) de las primeras posiciones de memoria del procesador que aparece en la tabla anterior, y el valor almacenado en **PC = 0** y en el registro **Base = 5**, rellena la siguiente tabla indicando, para la primera instrucción que se ejecuta, dónde escribe dicha instrucción y el valor que escribe (en hexadecimal) teniendo en cuenta los siguientes modos de direccionamiento de los operandos. En la columna **Lugar** indica si escribe en memoria (M) o registro (R) además de la posición (para memoria) o número de registro (para registro) (por ejemplo M(20) para posición 20 de memoria ó R20 para registro 20).

Memoria principal

Pos.	Contenido (hex)
0	60C2 <sub>(16)</sub>
1	0010 <sub>(16)</sub>
2	0008 <sub>(16)</sub>
3	0001 <sub>(16)</sub>
4	001A <sub>(16)</sub>
5	F000 <sub>(16)</sub>
6	5555 <sub>(16)</sub>
7	BAFE <sub>(16)</sub>
8	1000 <sub>(16)</sub>

OP1 OP2

Instrucción ejecutada:

ADD	3	2
-----	---	---

M.Direccionamiento de OP1 y OP2	Lugar	Valor (hex)
Directo a Memoria	M(3)	60C2
Indirecto por memoria	M(1)	1010 <sub>(16)</sub>
Relativo a PC (recuerda que PC se incrementa en el fetching)	M(4)	001B
Relativo a Base	M(8)	CAFE

# Problema 6:

- Dados los siguientes formatos de instrucciones (tabla 1) y sabiendo que el estado de memoria es el mostrado en la figura, obtener el nuevo estado de memoria tras la ejecución de las instrucciones del programa.

Tabla 1

Formato	Direccionamiento (fte1,fte2,dest)
SUMA1 <i>fte1</i> , # <i>fte2</i> , <i>dest</i>	Direc.mem., Inmediato, Direc.mem.
SUMA2 # <i>fte1</i> , # <i>fte2</i> , ( <i>dest</i> )	Inmediato, Inmediato, Indirecto
SUMA3 # <i>fte1</i> , ( <i>fte2</i> ), <i>dest</i>	Inmediato, Indirecto, Direc.mem.

...	POSICION DE MEMORIA
40	20
...	
60	40
...	
100	60
...	
- 4	100
...	

## Programa

SUMA1 20, #30, 40 →  $Mem(40) \leftarrow Mem(20) + 30 = 70$   
 $Mem(100) \leftarrow 10$  ← SUMA2 #20, #0, (60) →  $Mem(Mem(60)) \leftarrow 20 + 0$   
 SUMA1 40, # -50, 100 →  $Mem(100) \leftarrow Mem(40) - 50 = 20$   
 SUMA3 #100, (100), 100 →  $Mem(100) \leftarrow 100 + Mem(Mem(100)) = 140$