

Formato de Instrucción

Ejercicio en grupo: creación de un formato de instrucción

Se quiere diseñar el formato de instrucción para un procesador con las siguientes características:

- Memoria de 256 posiciones
- Banco de 16 registros de propósito general (r0, r1, ... r15)
- Registro Acumulador (AC) de propósito específico

Diseña el formato de instrucción para las siguientes **6 instrucciones**, teniendo en cuenta que la longitud de todas las instrucciones debe ser de **16 bits**. Para cada una de las instrucciones tienes que indicar su nemotécnico (cómo lo usaría el programador, ej. "addi rd, rs, inm") y el formato binario de la instrucción (cadena binaria y tamaño en bits de cada uno de los campos que la compongan).

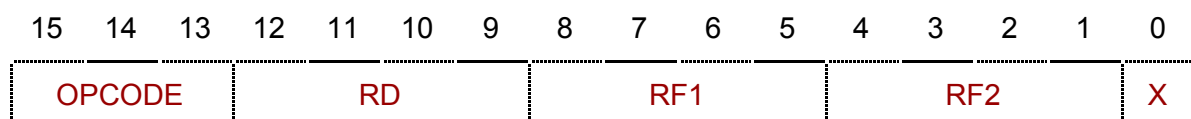
Puesto que el conjunto de instrucciones es de 6, todas las instrucciones tienen que tener un campo común (OPCODE) que sirva para identificarlas unívocamente. Cada una de las seis tendrá un código distinto, por lo que necesitamos 6 códigos distintos. Con 3 bits tenemos suficiente para crear 6 códigos binarios distintos ($2^3 = 8 > 6$).

Por tanto **OPCODE = 3 bits**. Los tres bits que representan el OPCODE en cada instrucción deberán estar siempre en la misma posición (por ejemplo los 3 bits más significativos de los 16). De esta forma, cuando se lea una instrucción de la memoria, se sabe qué bits mirar para determinar la instrucción leída y a partir de saber qué instrucción es (decodificación), se podrán interpretar el resto de bits (los 13 restantes) según el formato de dicha instrucción.

1. Suma del contenido de un registro fuente 1 (RF1) con un registro fuente 2 (RF2) y el resultado se almacena en un registro destino (RD). (Tanto RF1, RF2 y RD pueden ser cualquiera de los 16 registros del banco de registros y vendrán especificados explícitamente en la instrucción).

Nemotécnico: **SUMA RD, RF1, RF2**

Formato de instrucción:



Campos que componen el formato:

Nombre del campo (como aparece en el Formato de instrucción)	Descripción
OPCODE = 000	Código de la instrucción SUMA. Se ha elegido el 000 (sin ningún motivo especial)
RD	Registro destino: 4 bits ya que especifica un registro y hay 16 registros ($2^4 = 16$)
RF1	Registro fuente1: 4 bits ya que especifica un registro y hay 16 registros ($2^4 = 16$)
RF2	Registro fuente2: 4 bits ya que especifica un registro y hay 16 registros ($2^4 = 16$)

Sobra un bit (el bit 0) y lo indicamos con X para saber que ese bit puede tomar cualquier valor.

2. Carga el contenido de una posición de memoria (M) especificada en la instrucción en un registro destino (RD) también especificado en la instrucción.

Nemotécnico: **CARGA RD, M**

Formato de instrucción:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE				RD				M							X

Campos que componen el formato:

Nombre del campo (como aparece en el Formato de instrucción)	Descripción
OPCODE = 001	Código de la instrucción CARGA. Se ha elegido el 001 (sin ningún motivo especial, excepto que sea distinto a los códigos ya asignados)
RD	Registro destino: 4 bits ya que especifica un registro y hay 16 registros ($2^4 = 16$)
M	Dirección de memoria: 8 bits ya que especifica una dirección de memoria y hay 256 ($2^8 = 256$)

- Almacena el contenido de un registro fuente (RF) especificado en la instrucción en una posición de memoria (M) especificada en la instrucción.

Nemotécnico: **ALMACENA M, RF**

Formato de instrucción:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE				RF				M							X

Campos que componen el formato:

Nombre del campo (como aparece en el Formato de instrucción)	Descripción
OPCODE = 010	Código de la instrucción ALMACENA. Se ha elegido el 010 (sin ningún motivo especial, excepto que sea distinto a los códigos ya asignados)
RF	Registro fuente: 4 bits ya que especifica un registro y hay 16 registros ($2^4 = 16$)
M	Dirección de memoria: 8 bits ya que especifica una dirección de memoria y hay 256 ($2^8 = 256$)

- Mueve el contenido de un registro fuente (RF) a otro registro (RD), ambos especificados en la instrucción.

Nemotécnico: **MUEVE RD, RF**

Formato de instrucción:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE				RD				RF				X	X	X	X

Campos que componen el formato:

Nombre del campo (como aparece en el Formato de instrucción)	Descripción
OPCODE = 011	Código de la instrucción MUEVE. Se ha elegido el 011 (sin ningún motivo especial, excepto que sea distinto a los códigos ya asignados)

RD	Registro destino: 4 bits ya que especifica un registro y hay 16 registros ($2^4 = 16$)
RF1	Registro fuente: 4 bits ya que especifica un registro y hay 16 registros ($2^4 = 16$)

5. Añade al contenido del registro acumulador (AC) (registro implícito) el contenido de la posición de memoria (M) especificada en la instrucción.

Nemotécnico: **SUMAAC M**

Formato de instrucción:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE				X	X	X	X	M							X

Campos que componen el formato:

Nombre del campo (como aparece en el Formato de instrucción)	Descripción
OPCODE = 100	Código de la instrucción SUMAAC. Se ha elegido el 100 (sin ningún motivo especial, excepto que sea distinto a los códigos ya asignados)
M	Dirección de memoria: 8 bits ya que especifica una dirección de memoria y hay 256 ($2^8 = 256$)

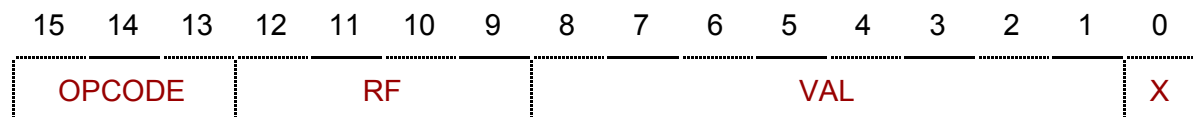
No se tiene que especificar explícitamente en el formato de instrucción el registro destino AC ya que va implícito en la instrucción. Es decir, al leer el opcode 0100, se sabe que es una instrucción SUMAAC y también se sabe que el registro destino es AC, no hay que especificarlo (con otro campo en el formato de instrucción) y por tanto tampoco se puede cambiar (no se puede especificar otro registro distinto de AC como destino de la instrucción).

La posición del campo M podría ser cualquier otra (respetando los bits del OPCODE). Se ha elegido esa por homogeneidad con las otras instrucciones que usan también campo M, pero NO ES OBLIGATORIO QUE COINCIDA LA POSICIÓN DE M EN DISTINTAS INSTRUCCIONES, lo único que debe coincidir forzosamente es el OPCODE.

6. Realiza la operación AND del contenido de un registro fuente (RF) con un valor de 8 bits dado en la instrucción (VAL) y el resultado se almacena en el propio registro fuente (RF).

Nemotécnico: **AND RF, VAL**

Formato de instrucción:



Campos que componen el formato:

Nombre del campo (como aparece en el Formato de instrucción)	Descripción
OPCODE = 101	Código de la instrucción AND. Se ha elegido el 101 (sin ningún motivo especial, excepto que sea distinto a los códigos ya asignados)
RF	Registro fuente: 4 bits ya que especifica un registro y hay 16 registros ($2^4 = 16$)
VAL	Valor de 8 bits que va en la propia instrucción

Ejercicio en grupo: codificación de instrucciones

De entre las 6 instrucciones para las cuales habéis creado el formato de instrucción, elegid 4 y codificadlas siguiendo dicho formato, con sus correspondientes operandos.

Debéis usar operandos concretos, es decir, si el operando es un registro, elige uno concreto, ej. r3, si es una posición de memoria especificad su valor numérico, ej. 38, y de igual forma si se trata de un operando valor numérico. Un posible ejemplo de instrucción sería: SUMA R5, R2, R4.

Instrucción 1: AND R8, 15

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cod. binaria	1	0	1	1	0	0	0	0	0	0	0	1	1	1	1	0
Cod. hexa.	B				0				1				E			

Instrucción 2: SUMAAC 20

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cod. binaria	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
Cod. hexa.	8				0				2				8			

Instrucción 3: ALMACENA 128, R3

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cod. binaria	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Cod. hexa.	4				7				0				0			

Instrucción 4: SUMA R3, R5, R7

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cod. binaria	0	0	0	0	0	1	1	0	1	0	1	0	1	1	1	0
Cod. hexa.	0				6				A				E			

Una vez elegidas las cuatro instrucciones, codificadas en binario y posteriormente en hexadecimal, rellenad la tabla denominada “Codificación hexadecimal de 4 instrucciones” del ejercicio anterior (página 4) sólomente con los valores hexadecimales obtenidos para las 4 instrucciones.

Cuando tengáis la tabla rellena, tenéis que entregar al grupo compañero que os ha tocado, las páginas con la especificación del formato que habéis creado, así como la tabla rellena con las 4 instrucciones codificadas en hexadecimal (páginas 1-4).

Ejercicio en grupo: decodificación de instrucciones

Estudid el formato de instrucción que os ha facilitado vuestro grupo compañero y anotad todos los comentarios que creáis convenientes para luego discutirlos con ellos (dudas, justificación de decisiones, errores, etc.).

Una vez estudiado el formato implementado, usadlo para decodificar las 4 instrucciones que vuestros compañeros os han pasado codificadas en hexadecimal (página 4), y rellenad la siguiente tabla con las instrucciones decodificadas (tenéis que indicar el nemotécnico de la instrucción así como los operandos concretos que esta usa).

- Instrucción 1:

Cod. hexa.	3				A				1				A			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cod. binaria	0	0	1	1	1	0	1	0	0	0	0	1	1	0	1	0
Instrucción:	CARGA R13, 13															

- Instrucción 2:

Cod. hexa.	6				1				F				F			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cod. binaria	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1
Instrucción:	MUEVE R0, R15															

- Instrucción 3:

Cod. hexa.	8				0				0				0			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cod. binaria	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Instrucción:	SUMAAC 0															

- Instrucción 4:

Cod. hexa.	B				E				E				F			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cod. binaria	1	0	1	1	1	1	1	0	1	1	1	0	1	1	1	1
Instrucción:	AND R15, 119															

Debatid con el grupo compañero vuestra solución comparada con la codificación de sus cuatro instrucciones elegidas. Detectad los errores, llegad a un acuerdo sobre su solución (error en la creación del formato, error en la codificación, error en la decodificación, ...). Elaborad un informe con los errores y las soluciones consensuadas para cada una de las instrucciones (tanto las 4 vuestras como las 4 de vuestro grupo compañero).