

# ACTIVIDAD 1

**Enunciado:** Ayudándote del simulador, piensa y rellena los siguientes datos relativos a esta CPU:

Tamaño en bits de una instrucción	16 bits
Tamaño en bits de la unidad direccionable de la memoria.	16 bits
Tamaño en bits del PC.	16 bits
Tamaño en bits de los registros R0, R1...R7	16 bits
Número de bits que se usarán dentro del formato de una instrucción para indicar un registro (si hay ocho registros...)	3 bits
Número de bits del bus de datos de la memoria.	16 bits
Número de bits del bus de direcciones de la memoria.	16 bits

# ACTIVIDAD 2

**Enunciado:** Explica qué hace cada una de las siguientes órdenes (utiliza la ayuda del simulador).

- CLR R0: Esta orden pone a 0 el registro R0.
- MOV R0, R1: Esta orden hace una transferencia de datos entre el registro R0 y R1.
- MOV [R0], R1: Esta orden copia el contenido del registro R1 en la posición de memoria cuya dirección está en R0.
- INC R0: Esta orden incrementa el contenido del registro R0 en una unidad.
- ADD R1, R1, R1: Esta orden suma el contenido del registro R1 2 veces y lo almacena en R1.
- SB R1, R2, R3: Esta orden resta el contenido de R2 y R3 y almacena el resultado en R1.

# ACTIVIDAD 3

**Enunciado: ¿Qué hace el programa pot? ¿Dónde almacena los resultados?**

El programa pot utiliza los registros R0 y R1 para calcular las potencias sucesivas de 2, hasta en 6 ocasiones.

Los datos del programa pot se almacenan en el registro que marca el valor en hexadecimal del registro R0. Así cada potencia está en un registro diferente, todos ellos seleccionados por los valores del registro R0.

## ACTIVIDAD 4

**Enunciado: Carga ahora el programa impares ¿Qué hace el programa? ¿Qué crees que hace la instrucción BRNS –5?**

-El programa impares muestra los sucesivos números impares desde el 99 (decimal) hasta el 0, repitiendo el proceso una y otra vez ayudándose de una especie de bucle para ahorrar memoria y almacenamiento.

-Una vez el número del registro R2 sea negativo, el programa se detiene. Además, almacena los diferentes números impares en las casillas dependiendo del valor del registro R0.

-Lo que hace BRNS –5 es hacer un retroceso de 5 instrucciones contándose a sí misma, mientras que el resultado de la operación anterior carezca de signo (ya sea positivo o cero).

## ACTIVIDAD 5

**Enunciado: Utilizando la ayuda del simulador (fichero usuario.hlp -> Juego de instrucciones), indique porqué la instrucción MOV [R0], R2 es en código máquina 0x1040.**

-Porque el 0x significa que todo lo que hay a la izquierda de la x, se trata de un número hexadecimal, por lo tanto, hemos de pasar 1040 a binario. Una vez pasado a binario y organizado quedaría 00 010 000 010 0000. La parte azul que vemos corresponde a la

estructura de la instrucción de MOV [R1], R5. Mientras que la parte verde corresponde al número del registro R0 y la parte naranja la del registro R2.

## ACTIVIDAD 6

**Enunciado:** Utilizando la ayuda del simulador (fichero usuario.hlp -> Juego de instrucciones) y la pregunta anterior, indique como sería el código máquina de la instrucción MOV [R1], R7.

-Este código lo que hace es copiar el contenido del registro R7 en la posición de la memoria cuya dirección está en IR (R1).

-El código de la instrucción en código máquina sería:

-00 010 R1 R7 00000

-00 010 001 111 00000

-El código máquina me sale: 0x11E0

## ACTIVIDAD 7

**Enunciado:** Introduce algún cambio en los programas, explicando en qué consiste el cambio y qué instrucciones has añadido o modificado (por ejemplo, que los resultados se guarden en posiciones de memoria distintas o que el programa impares cuente hacia adelante en lugar de hacia atrás. O lo que se te ocurra).

Primero que todo, he cambiado el valor que le otorgamos al registro R2, yo le he dado el valor de 6D, que corresponde a 109 en decimal. Posteriormente lo que hago es mover todas las instrucciones, para añadir otro INC R0, porque para lo que quiere hacer es que los impares que se vayan restando se vayan almacenando en las casillas pares, es decir, 0, 2, 4, 6, 8....

Además, para poder continuar con el bucle que realiza la instrucción BRNS, hemos de retocarla un poco, para que pase de BRNS -5 a BRNS -6 , ya que estamos añadiendo otra instrucción entre medias.

HEXADECIMAL	INSTRUCCIÓN
5800	CLR R0
226D	MOVL R2, 6DH

1040	MOV [R0], R2
5200	INC R0
5200	INC R0
5480	DEC R2
5480	DEC R2
FFFA	BRNS -6

Explicación más detallada: El programa editado por mí, muestra los impares desde el 109 hasta el -1. Sin embargo, en mi programa, los impares se van almacenando en las casillas pares como: 0, 2, 4, 6, 8, 10.... Para ello incremento el número de R0 dos veces, y al añadir una instrucción nueva, he de cambiar el BRNS -5 por un BRNS -6 para que haga el bucle sin ningún error.