

Especificación del lenguaje ensamblador usado durante el curso

Características generales de la arquitectura	
Propiedad	Valor
Unidad mínima direccionable (byte)	8 bits
Tamaño de una palabra	16 bits
Tamaño de una dirección	16 bits
Codificación de datos	Little endian a nivel de bytes
Codificación de enteros	Complemento a dos
Alineación en memoria de datos	Opcional

Registros disponibles	
Nombre del registro	Código máquina
R0	000
R1	001
R2	010
R3	011
R4	100
R5	101
RC (registro de estado o condición)	110
RP (registro de instrucción del programa)	111

Sentencias del ensamblador	
Sintaxis	Descripción
OFFSET <i>entero</i>	Posicionamiento en la dirección de memoria indicada por <i>entero</i>
ALIGN <i>entero</i>	Posicionamiento en la siguiente dirección de memoria múltiplo de <i>entero</i>
ALLOC <i>entero</i>	Apartar <i>entero</i> bytes de memoria

Conjunto de instrucciones		
Sintaxis	Código máquina	Descripción
NOP	00000000	Instrucción vacía
ADD <i>ra, rb, rc</i>	0001000aaabbbccc	Suma entre registros
SUB <i>ra, rb, rc</i>	0001001aaabbbccc	Resta entre registros
MUL <i>ra, rb, rc</i>	0001010aaabbbccc	Multiplicación entre registros
DIV <i>ra, rb, rc</i>	0001011aaabbbccc	División entre registros
REM <i>ra, rb, rc</i>	0001100aaabbbccc	Residuo entre registros
AND <i>ra, rb, rc</i>	0001101aaabbbccc	AND a nivel bit entre registros
IOR <i>ra, rb, rc</i>	0001110aaabbbccc	OR inclusiva a nivel bit entre registros
XOR <i>ra, rb, rc</i>	0001111aaabbbccc	OR exclusiva a nivel bit entre registros
NEG <i>ra, rb</i>	0000010000aaabbb	Negación aritmética de registro
COM <i>ra, rb</i>	0000010001aaabbb	Complemento a nivel bit de registro
LST <i>ra, rb</i>	0000010010aaabbb	Comparación "menor que" entre registros
GRT <i>ra, rb</i>	0000010011aaabbb	Comparación "mayor que" entre registros
LSE <i>ra, rb</i>	0000010100aaabbb	Comparación "menor o igual" entre registros
GRE <i>ra, rb</i>	0000010101aaabbb	Comparación "mayor o igual" entre registros
EQU <i>ra, rb</i>	0000010110aaabbb	Comparación de igualdad entre registros
NEQ <i>ra, rb</i>	0000010111aaabbb	Comparación de desigualdad entre registros
MOV <i>ra, rb</i>	0000011000aaabbb	Transferencia entre registros
MOV <i>ra, entero</i>	00001aaaaeeeeeeeeeeeeeeee	Asignación a registro (direccionamiento inmediato)
MOV <i>ra, WORD [entero]</i>	00100aaaaeeeeeeeeeeeeeeee	Lectura de una palabra en memoria (direccionamiento directo)
MOV <i>ra, BYTE [entero]</i>	00101aaaaeeeeeeeeeeeeeeee	Lectura de un byte en memoria (direccionamiento directo)
MOV WORD [<i>entero</i>], <i>ra</i>	00110aaaaeeeeeeeeeeeeeeee	Escritura de una palabra en memoria (direccionamiento directo)

MOV BYTE [<i>entero</i>], <i>ra</i>	00111 <i>aaaa</i> eeeeeeeeeeeeeeee	Escritura de un byte en memoria (direccionamiento directo)
MOV <i>ra</i> , WORD [<i>rb</i>]	0100000000 <i>aaabbb</i>	Lectura de una palabra en memoria (direccionamiento indirecto)
MOV <i>ra</i> , BYTE [<i>rb</i>]	0100000001 <i>aaabbb</i>	Lectura de un byte en memoria (direccionamiento indirecto)
MOV WORD [<i>ra</i>], <i>rb</i>	0100000010 <i>aaabbb</i>	Escritura de una palabra en memoria (direccionamiento indirecto)
MOV BYTE [<i>ra</i>], <i>rb</i>	0100000011 <i>aaabbb</i>	Escritura de un byte en memoria (direccionamiento indirecto)
MOV <i>ra</i> , WORD [<i>rb</i> + <i>entero</i>]	0100000100 <i>aaabbb</i> eeeeeeeeeeeeeeee	Lectura de una palabra en memoria (direccionamiento relativo a registro base)
MOV <i>ra</i> , BYTE [<i>rb</i> + <i>entero</i>]	0100000101 <i>aaabbb</i> eeeeeeeeeeeeeeee	Lectura de un byte en memoria (direccionamiento relativo a registro base)
MOV WORD [<i>ra</i> + <i>entero</i>], <i>rb</i>	0100000110 <i>aaabbb</i> eeeeeeeeeeeeeeee	Escritura de una palabra en memoria (direccionamiento relativo a registro base)
MOV BYTE [<i>ra</i> + <i>entero</i>], <i>rb</i>	0100000111 <i>aaabbb</i> eeeeeeeeeeeeeeee	Escritura de un byte en memoria (direccionamiento relativo a registro base)
JMP <i>entero</i>	10000000eeeeeeee	Salto (direccionamiento relativo a RP, $-128 \leq \text{entero} \leq +127$)
JIF <i>entero</i>	10000001eeeeeeee	Salto condicional (direccionamiento relativo a RP, $-128 \leq \text{entero} \leq +127$)
JMP <i>entero</i>	10000010eeeeeeeeeeeeeeee	Salto (direccionamiento absoluto)
JIF <i>entero</i>	10000011eeeeeeeeeeeeeeee	Salto condicional (direccionamiento absoluto)
JMP <i>ra</i>	11000 <i>aaa</i>	Salto (direccionamiento mediante registro)
JIF <i>ra</i>	11001 <i>aaa</i>	Salto condicional (direccionamiento mediante registro)

Ejemplo de sintaxis y uso de etiquetas

```
OFFSET 5  
ALIGN 2  
var: ALLOC 2
```

```
OFFSET 30  
MOV WORD [var], 10    # comentario, poniendo un 10 en var  
JMP 30
```