

REPERTORIO DE INSTRUCCIONES MÁQUINA

En nuestro ejemplo, sólo tendremos dos instrucciones máquina:

ASIGNAR *cargará un registro con una constante*

Código de operación: **H'10**

Formato: **ASIGNAR reg,#cte**

Ejemplo de uso: **ASIGNAR .0,#23**

Codificación hexadecimal: **[10X0] [0023]**

Operaciones elementales:

IRB <- m[CP]

CP <- CP + 1

SUMAR *hará la suma entre dos registros (se almacenará en el segundo)*

Código de operación: **H'11**

Formato: **SUMAR reg1,reg2**

Ejemplo de uso: **SUMAR .8,.3**

Codificación hexadecimal: **[1183]**

Operaciones elementales:

IRB <- IRB + IRA

MICROCÓDIGO DE LA MEMORIA DE CONTROL

CP EQU .E ; indicamos qué registro usaremos como contador de programa
 SP EQU .F ; y como puntero de pila en la máquina que estamos definiendo

;inicialización de registros (sólo se hará al principio una vez)

```
ORG H' 0
CLR CP
BR H' 20
```

;fase de captación de instrucción

```
ORG H' 20
LD IR, [CP]
INC CP
JMAP
```

;codificación de la instrucción máquina ASIGNAR (códop=10 -> direcc. H'40)

```
ORG H' 40
LD IRB, [CP]
INC CP
BR H' 20 ;fin de la codificación de la instrucción -> salto a la fase de captación
```

;codificación de la instrucción máquina SUMAR (códop=11 -> direcc. H'44)

```
ORG H' 44
ADD IRB, IRA & LD_MFL & BR H' 20
;fin de la codificación de la instrucción -> salto a la fase de captación
```

END

PROGRAMA ENSAMBLADOR DE EJEMPLO

Para probar nuestra máquina, vamos a hacer un programa en ensamblador para ella, y lo probaremos en el emulador.

Simplemente vamos a *asignar valores a dos registros y los sumaremos* (no se puede hacer mucho más con ese juego de instrucciones).

```
ASIGNAR .1, #3
ASIGNAR .2, #6
SUMAR .1, .2
```

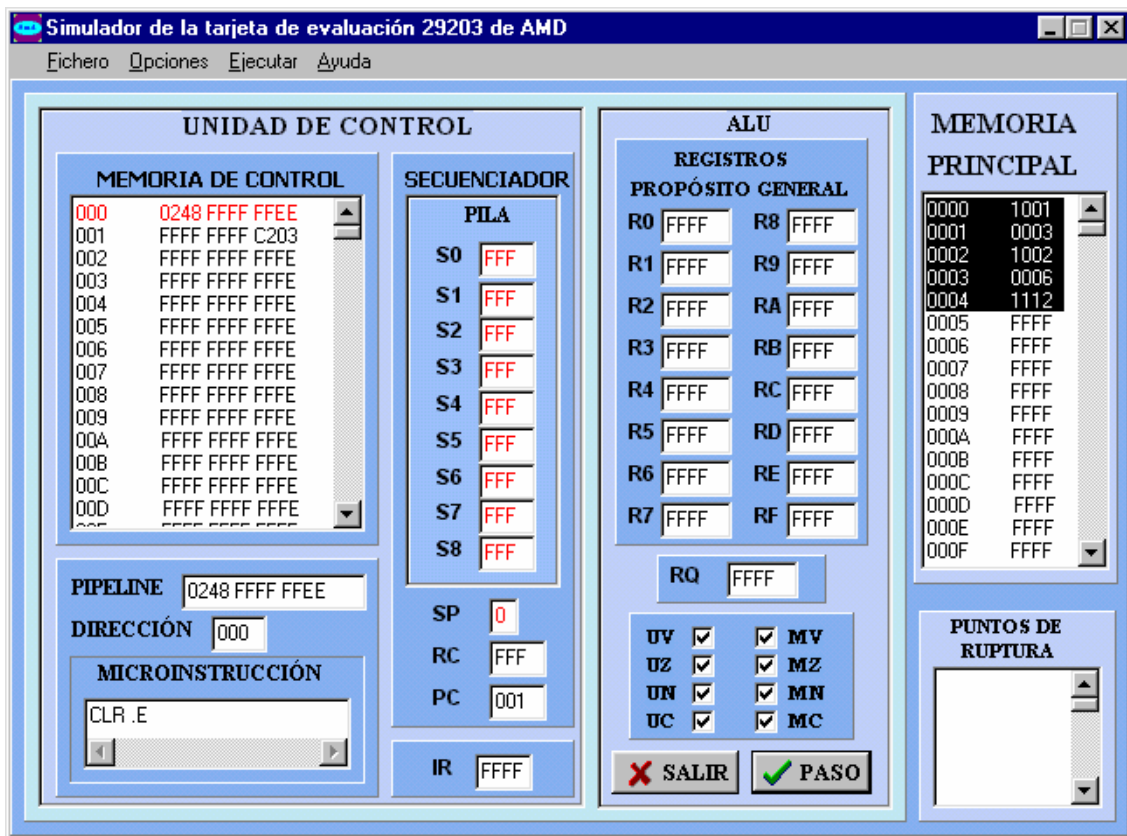
La codificación en hexadecimal que introduciremos en la memoria principal (en las 5 posiciones de memoria que ocupa ese programa) es:

```
10X1 ; asignar al registro 1 la cte de la siguiente posición de memoria
0003 ; cte de la instrucción de asignación
10X2 ; asignar al registro 2 la cte de la siguiente posición de memoria
0006 ; cte de la instrucción de asignación
1112 ; sumar el contenido de .1 y .2, almacenando en .2
```

EJECUCIÓN DEL PROGRAMA ENSAMBLADOR EN LA MÁQUINA DEFINIDA

Deberemos cargar la memoria de control obtenida (fichero .amd) al compilar antes nuestro ejemplo de programa microensamblador (Fichero->Cargar memoria de control). Veremos que las microinstrucciones que codificamos aparecen en las posiciones de la memoria de control que nosotros indicamos en el microprograma. Y la primera a ejecutar aparece resaltada en rojo.

A continuación, cargamos la codificación de nuestro programa ensamblador en la memoria principal (resaltado en la siguiente imagen)



A continuación, vamos pulsando el botón de “*PASO*” y veremos que se van ejecutando una a una cada microinstrucción de la fase de captación (con lo que los registros irán cambiando) y también las de la fase de ejecución correspondiente a la instrucción máquina que estemos ejecutando tras la ejecución de la microinstrucción JMAP. Deberemos mover la barra de desplazamiento de la memoria de control conforme se produzcan los saltos.

Instante en el que se acaba de cargar en IR la primera instrucción máquina y se ha incrementado el CP. A continuación se va a ejecutar la microinstrucción JMAP (situada en la dirección 22 de la memoria de control).

