

1-CÓDIGO

$D[3] = \{ Y + (A - M[2]) \}$
$Y = \%L0$
$A = \%L1$
$M = \%L2$
$D = \%L4$

Primero inicializamos las variables del código

4-CÓDIGO HEXADECIMAL

0X0000	E604A008
0X0004	A2244013
0X0008	A0040011
0X000C	E225200C

El Código hexadecimal lo sacamos del código máquina uniendo de a 4 dígitos

2-LENGUAJE ENSAMBLADOR

LD [%L2 + (2*4)], %L3
SUB %L1, %L3, %L1
ADD %L0, %L1, %L0
ST %L0, [%L4 + (3*4)]

Para el lenguaje ensamblador utilizamos el **LD** para extraer la dirección de memoria, luego un **SUB** para restar los registros y le sigue un **ADD** para sumar ese resultado con un nuevo registro; por último con el **ST** guardamos en memoria

5-CONCLUSIONES

- Para pasar el código a lenguaje ensamblador se debe inicializar variables
- El load y el store son los únicos que tienen acceso a la memoria
- Cuando hay números inmediatos se convierte en un registro de 13bits en el lenguaje máquina
- El OP depende del campo de instrucción que estemos utilizando

3-LENGUAJE MAQUINA

	OP	RD	OP3	RS1	I	UNSIED ZERO	RS2
0X0000	11	10011	000000	10010	1	000000000	1000
0X0004	10	10001	000100	10001	0	00000000	10011
0X0008	10	10000	000000	10000	0	00000000	10001
0X000C	11	10001	000100	10100	1	000000000	1100

Para pasar el lenguaje ensamblador a lenguaje máquina utilizamos la tabla vista en la cual se llena con los datos del ensamblador, pero solo con 1 y 0. Para hallar el **OP** y el **OP3** utilizamos unas tablas; y si el registro **i=1** (que quiere decir que hay números inmediatos) el **unsied** se une con el **registro S2**