



Lenguaje de Alto Nivel

```
Int main ()
{
X[3] = ( Y + Z[2] ) - G
return Y
}
```

Lenguaje Emsamblador

```
X[3]: %o0
Y: %L2
Z: %L1
G: %L3

Ld [%L1 + (2x4)], %L4 ---> Load: Carga de memoria
al vector

Add %L2, %L4, %L2 ---> Add :Sumar

Sub %L3, %L2, %L3 ---> Sub: Restar

St %L2 [%O0+ (3x4)]--->Store: Almacena valor en
la memoria
```

Lenguaje Máquina y Hexadecimal

ld %l1, 8, %l4 ----> 0x1D008C008

11	10100	000000	10001	1	000000000	01000
----	-------	--------	-------	---	-----------	-------

add %l2, %l4, %l2 -> 0x290120014

10	10010	000000	10010	0	000000000	10100
----	-------	--------	-------	---	-----------	-------

sub %l3, %l2, %l3 ->0xA6248004

10	10011	000100	10010	0	000000000	10011
----	-------	--------	-------	---	-----------	-------

st %l2, 12, %o0 -> 0x1A049400C

11	01000	000100	10010	1	000000000	01100
----	-------	--------	-------	---	-----------	-------

Conclusiones

- El ejemplo fue inventado para que realizara LOAD el STORE, que inicialicen con OR y hagan operaciones con SUB y ADD.
- De este programa se extrae el lenguaje emsamblador.
- Despues del lenguaje emsamblador se pasa a lenguaje de máquina.
- Para obtener el la conversion de binario a hexadecimal se toman de a 4 bits para poder convertirlos apoyandonos en la tabla de conversion.

