



INSTRUCCIONES DE BAJO Y ALTO NIVEL

EJERCICIO

```
int main (){
int i=-19600;
b[2]=33;
if (i-b[2])>12{
return i+15;
}else{
return i*32;
}
}
```



¿CÓMO SE REALIZA LA IMPLEMENTACIÓN?

- Para realizar la conversión a instrucciones de bajo y alto nivel debemos hacer uso de los principios de diseño de hardware para desarrollarlo se de tener en cuenta:
- 1-Definir los registros de cada variable
 - 2-Inicializar las variables en cero con la variable global [0] (%G0) utilizando la variable sintética MOV
 - 3- Realizar el complementos a dos al valor de la variable (i), la cual tiene una valor negativo. 4-Cuando se obtenga la conversión de lenguaje ensamblador se procede a realizar la conversión a Lenguaje Máquina (Binario) y luego inicializar la variable con la instrucción Sethi y or.
 - 4-Realizar las operaciones entre los registros, se utilizó la variable sintética CMP del operador SUB para realizar comparación.
 - 5-Reutilizar registros que no voy a volver a utilizar
 - 6-Utilizar el formato #3,el cual se utiliza para las instrucciones aritmético-lógicas y desplazamientos (SHCNT) y utilizó y formato #2 Instrucciones de Salto(SETHI, BRANCH y NOP),en caso de no realizar las operaciones anteriores se puede utilizar el formato #1Instrucciones de llamado (CALL)
 - 7-Utilizar el formato OP de las operaciones a realizar, en este caso se utiliza el OP 10(Aritmético-lógica y shcnt) y OP 00 (Branch, Nop y Sethi).
 - 8-Utilizar el formato OP3 de las instrucciones y OP2, en este caso se utiliza el OP3 del OR-MOV (000010), ADD (000000), SUB (000100), CMP (SUBcc (010100)) y el OP2 del BA y BGE (010), NOP (100), SETHI (100).
 - 9- Diligenciar el formato #3 (ADD, SUB, SUBcc OR y SHNCT) y formato #2 (BRANCH, NOP y SETHI) de cada uno
 - 10-Se utiliza para las comparaciones la variable sintética CMP que es un SUBcc.

DESARROLLO

LENGUAJE ENSAMBLADOR

Definición de registros

```
i=%L0
b=%L1
```

Inicialización de variables

```
SETHI -20,%L0
OR %L0,880,%L0
mov 33,%L2
```

Ejecución de operaciones

```
ST %L2,[ %L1 + (2*4) ]
LD [%L1+(2*4)],%L3
SUB %L0,%L3,%L4
CMP %L4,12,%L5
BLE a SALTO1
ADD %L0,15,%L0
BA a EXIT
SALTO1
SLL %L0,32,%L0
EXIT
NOP
```

LENGUAJE DE MÁQUINA

| DIRECCIONES | OP | RD | OP2 | Imm22 | | | |
|-------------|----|-------|--------|-----------------------|-----------------------|---------------|--------|
| OX0000 | 00 | 10000 | 100 | 11111111111111110100 | | | |
| | OP | RD | OP3 | RS1 | i | Unusued/zero | RS2 |
| OX0004 | 10 | 10000 | 00010 | 10000 | 1 | 0001101110000 | |
| OX0008 | 11 | 10010 | 000100 | 10001 | 1 | 0000000001000 | |
| OX000C | 11 | 10011 | 000000 | 10001 | 1 | 0000000001000 | |
| OX0010 | 10 | 10100 | 00010 | 10000 | 0 | 00000000 | 10011 |
| OX0014 | 10 | 10101 | 010100 | 10100 | 1 | 0000000001100 | |
| | OP | a | cond | OP2 | disp22 | | |
| OX0018 | 00 | 1 | 0010 | 010 | 00000000000000000011 | | |
| | OP | RD | OP3 | RS1 | i | Unusued/zero | RS2 |
| OX001C | 10 | 01000 | 000000 | 10000 | 1 | 0000000001111 | |
| | OP | a | cond | OP2 | disp22 | | |
| OX0020 | 00 | 1 | 1000 | 010 | 000000000000000000010 | | |
| | OP | RD | OP3 | RS1 | i | Unusued/zero | SHCNT |
| OX0024 | 10 | 10000 | 100101 | 10000 | 1 | 00000000 | 100000 |
| | OP | RD | OP2 | Imm22 | | | |
| OX0028 | 00 | 00000 | 100 | 000000000000000000000 | | | |

CONCLUSIONES

- *Para iniciar a realizar las conversiones se colocó la práctica de lo aprendido en clase de los principios del diseño del hardware, los cuales me guiaron paso a paso para realizar dichas conversiones.
- *Inicialmente le asigné a cada variable un registro Local
- *Posterior a la asignación de registro de variable, realicé la inicialización de las variables, en este caso se inicializaron con la variable sintética MOV del operador OR.
- *Se utilizó la variable sintética CMP para realizar las comparaciones.
- *Realicé los complementos a dos al valor de la variable (i), la cual tiene un valor negativo. El completo a dos se utiliza para representar los números negativos, se realiza invirtiendo sus bits y luego sumando uno. Después de realizar la operación se selecciona los 22 bits más significativos de izquierda a derecha y realizó nuevamente complemento a dos para obtener un número más pequeño. Luego de realizar el complemento a dos inicializó la variable con la instrucción SETHI y OR.
- *Para realizar las conversiones debo verificar que operaciones voy a realizar para así saber qué tipos de operandos utilizar, en este caso utilicé los operandos SUB, ADD, OR, variable sintética CMP para comparar, voy a utilizar las instrucciones de BRANCH las cuales son BA y BGE, la instrucción NOP, instrucción de desplazamiento SHCNT e instrucción SETHI.
- *Para realizar la conversión de lenguaje de máquina verifiqué que tipos de instrucciones iba a utilizar para así seleccionar el tipo de formato, en este caso utilicé el formato #3 el cual se utiliza para las instrucciones Aritmético-lógicas-shcnt y el formato #2 el cual se utiliza para las instrucciones BRANCH,NOP y SETHI.
- *Seleccioné el formato OP de las operaciones que voy a realizar en el ejercicio, en este caso utilicé el OP 10 (aritmético-lógicas-shcnt) y OP 00 (Branch, Nop y Sethi)
- *Se aplicó los conocimientos de los tipos de instrucciones y formatos que se deben utilizar para cada caso, para este ejercicio se utilizó el formato OP3 donde se utiliza las instrucciones aritmético-lógica y desplazamiento y OP2 donde se utiliza las instrucciones de salto.