



INSTRUCCIONES DE BAJO Y ALTO NIVEL

EJERCICIO

```
int main (){
    int i=20;
    int j=11;
    if (i!=j){
        i++;
    }else{
        i--;
    }
    return i+j;
}
```



¿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 debe 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 las operaciones entre los registros
 - 3-Reutilizar registros que no voy a volver a utilizar
 - 4-Cuando se obtenga la conversión de lenguaje ensamblador se procede a realizar la conversión a Lenguaje Máquina (Binario)
 - 5-Utilizar el formato #3, el cual se utiliza para las instrucciones aritmético-lógicas y utilizar el formato #2 para las instrucciones de Salto (SETHI y BRANCH), en caso de no realizar las operaciones anteriores se puede utilizar el formato #1 para las instrucciones de Llamado (CALL)
 - 6-Utilizar el formato OP de las operaciones a realizar, en este caso se utiliza el OP 10 (Aritmético-lógica) y OP 00 (Branch y Nop).
 - 7-Utilizar el formato OP3 de las instrucciones y OP2, en este caso se utiliza el OP3 del OR-MOV (000010), ADD (000000), SUB (000100), SUBcc-CMP (010100) y el OP2 del BA y BE (010), NOP (100).
 - 8- Diligenciar el formato #3 (ADD, SUB, OR) y formato #2 (BRANCH y NOP) de cada uno.
 - 9-Se utiliza para las comparaciones la variable sintética CMP que es un SUBcc.

DESARROLLO

LENGUAJE ENSAMBLADOR

Definición de registros

```
i=%L0
j=%L1
```

Inicialización de variables

```
mov 20,%L0
mov 11,%L1
```

Ejecución de operaciones

```
CMP %L0,%L1,%L2
BE a SALTO1
ADD %L0,1,%L0
BA a EXIT
SALTO1
SUB %L0,1,%L0
EXIT
ADD %L0,%L1,%O0
EXIT
NOP
```

LENGUAJE DE MÁQUINA

| DIRECCIONES | OP | RD | OP3 | RS1 | i | Unused/zero | RS2 |
|-------------|----|-------|--------|-------|--------|-----------------------|-------|
| OX0000 | 10 | 10000 | 000010 | 00000 | 1 | 00000000101000 | |
| OX0004 | 10 | 10001 | 000010 | 00000 | 1 | 000000001011 | |
| OX0008 | 10 | 10010 | 010100 | 10000 | 0 | 00000000 | 10001 |
| | OP | a | cond | OP2 | disp22 | | |
| OX000C | 00 | 1 | 0001 | 0010 | | 000000000000000010100 | |
| | OP | RD | OP3 | RS1 | i | Unused/zero | RS2 |
| OX0010 | 10 | 10000 | 000000 | 10000 | 1 | 00000000000001 | |
| | OP | a | cond | OP2 | disp22 | | |
| OX0014 | 00 | 1 | 1000 | 0010 | | 000000000000000010100 | |
| | OP | RD | OP3 | RS1 | i | Unused/zero | RS2 |
| OX0018 | 10 | 10000 | 000100 | 10000 | 1 | 00000000000001 | |
| OX001C | 10 | 1000 | 000000 | 10000 | 0 | 00000000 | 10001 |
| | OP | RD | OP2 | Imm22 | | | |
| OX0020 | 00 | 00000 | 100 | | | 00000000000000000000 | |

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.
- *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, CMP (SUBcc) ADD, OR (MOV), voy a utilizar las instrucciones de BRANCH las cuales son BA y BE, la instrucción NOP.
- *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 y el formato #2 el cual se utiliza para las instrucciones BRANCH y NOP.
- *Seleccioné el formato OP de las operaciones que voy a realizar en el ejercicio, en este caso utilicé el OP 10 (aritmético-lógicas) y OP 00 (Branch y Nop)
- *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 OP2 donde se utiliza las instrucciones de salto