**EJERCICIO**

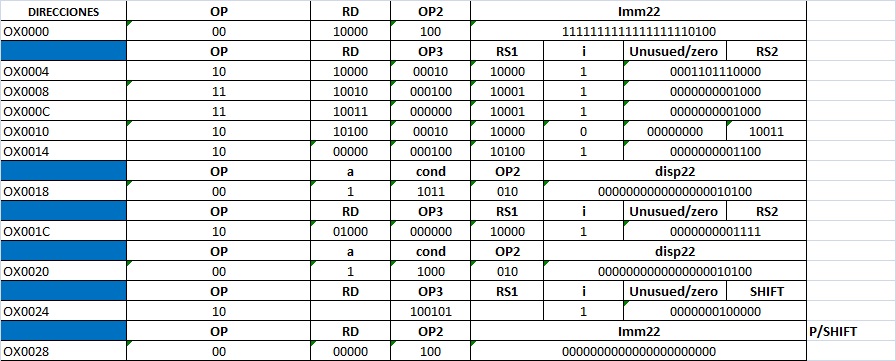
**¿CÓMO SE REALIZA LA IMPLEMENTACIÓN?**

Para realizar la conversión a  instrucciones de bajo y alto nivel debemos hacer uso de los principios de diseño de hadware 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 las operaciones  entre los registros, se utilizo la variable sintética CMP del operador SUB para realizar comparación.  
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 desplazamientos (SHIFT) 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 shift) y OP 00 (Branch, Nop y Sethi).  
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) y el OP2 del BA y BGE (010), NOP (100),SETHI (100).  
8- Diligenciar el formato #3 (ADD, SUB, OR y SHIFT) y formato #2 (BRANCH,NOP y SETHI) de cada uno .

**Profesora está parte del código subrayada en rojo no la entiendo muy bien al realizarlo en lenguaje ensamblador y lenguaje máquina la instrucción SHIFT.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | int main (){ | | int i=-19600; | | b[2]=33; | | **if (i-b[2])>12{** | | **return i+15;** | | **}else{** | | **return i\*32;** | | } | | } | |
| C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0217698.wmf |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**LENGUAJE DE MÁQUINA**



**DESARROLLOC:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0195812.wmf**

**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 BGE %L4 FALSE ADD %L0,15,%O0 BA %L4 EXIT FALSE SLL %L0,32,%O0 EXIT NOP

**CONCLUSIONES**

\*Para iniciar a realizar las conversiones se colocó la práctica de lo aprendido en clase de los principios del diseño del hadware, 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.  
\*Para realizar las conversiones debo verificar que operaciones voy a realizar para así saber que 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 SHIFT 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-shift 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-shitf) 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