



INSTRUCCIONES DE BAJO Y ALTO NIVEL

INTRODUCCIÓN

Las instrucciones representan las líneas de código cuando se hace el programa.

Se establecen en instrucciones de:

- Bajo nivel ---Lenguaje Ensamblador
- Alto nivel----Lenguaje de Máquina



EJERCICIO

Convertir a instrucciones de bajo y alto nivel

```
int main() {  
    int X = 3;  
    int Y = 5;  
    B [50] = (X + Y) - (B [90] + X)  
}
```

CONCLUSIONES

*Para iniciar a realizar las conversiones se colocó el páctica 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 el operador OR y con el registro global cero, para obtener resultados exactos y óptimos.

*Para realizar las conversiones debo verificar que operaciones voy a realizar para así saber que tipos de operandos utilizar, en este caso utilice los operandos SUB, ADD, OR, LOAD y STORE.

*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 Load/Store

*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 11 (Load/Store)

*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 Load/Store

*Para diligenciar el formato #3 tuve en cuenta las direcciones de memoria de cada uno para posterior realizar la conversión a lenguaje hexadecimal



¿CÓMO SE REALIZA 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)
- 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 Load/Store, en caso de no realizar las operaciones anteriores se puede utilizar el formato #1 Instrucciones de llamado (CALL) y formato #2 Instrucciones de Salto (SETHI y BRANCH)
- 6-Utilizar el formato OP de las operaciones a realizar, en este caso se utiliza el OP 10 (Aritmético-lógica) y OP 11 (Load/Store)
- 7-Utilizar el formato OP3 de las instrucciones, en este caso se utiliza el OP3 del OR(000010), ADD(000000), SUB(000100), LOAD(000000) y STORE(000100)
- 8-Diligenciar el formato #3 de cada uno.
- 9-Cuando se obtenga la conversión del lenguaje de máquina se procede a realizar la conversión a código hexadecimal, que separar de a cuatro 4 dígitos de derecha a izquierda y obtener el número binario.

DESARROLLO

Lenguaje Ensamblador:

Definición de registros

B = %L0

X = %L1

Y = %L2

Inicialización de Variables

OR %G0, 3, %L1

OR %G0, 5, %L2

Ejecución de operaciones

ADD %L1, %L2, %L2

LD [%L0 + (90 * 4)], %L3

ADD %L3, %L1, %L1

SUB %L2, %L1, %L1

ST %L1, [%L0, (50 * 4)]

Lenguaje de Máquina:

DIRECCIONES	OP	RD	OP3	RS1	i	Unused/zero	RS2
OX0000	10	10001	000010	00000	1	00000000000011	
OX0004	10	10010	000010	00000	1	00000000000101	
OX0008	10	10010	000000	10001	0	00000000	10010
OX000C	11	10011	000000	10000	1	00001011010000	
OX0010	10	10001	000000	10011	0	00000000	10001
OX0014	10	10001	000100	10010	0	00000000	10001
OX0018	11	10001	000100	10000	1	00000110010000	

Lenguaje de Máquina a Código Hexadecimal:

DIRECCIONES	CONVERSIÓN HEXADECIMAL (OX)
OX0000	A2102003
OX0004	A4102005
OX0008	A4044012
OX000C	E6042168
OX0010	A204C011
OX0014	A2248011
OX0018	E22420C8