

Formatos de Instrucción

Para empezar a diseñar un microprocesador debemos especificar el formato de las instrucciones. Este formato representa la forma como cada instrucción es almacenada en la memoria de programa del microprocesador.

El conjunto de instrucciones del ESCOMIPS tienen un formato de 25 bits, con el que podemos realizar instrucciones con 3 operandos. Básicamente se tienen 3 formatos de instrucciones que son: Formato tipo R, formato tipo I y formato tipo J. Estos formatos de instrucción permiten manejar los modos de direccionamiento inmediato, por registro, directo e indirecto.

Formato de Instrucción tipo R (Register):

Este formato lo tienen todas las instrucciones cuyos operandos son todos registros. Los 25 bits de la instrucción se distribuyen en 6 campos de la siguiente manera:

24.....20	19.....16	15.....12	11.....8	7.....4	3.....0
OP	Rd	Rt	Rs	SHAMT	FUNCT
5 bits	4 bits	4 bits	4 bits	4 bits	4 bits

El significado de cada uno de los campos es:

OP: Es el código de operación.

Rd: Es el registro operando destino.

Rt: Es el primer registro operando fuente

Rs: Es el segundo registro operando fuente

SHAMT: Cantidad de bits a desplazar en las instrucciones de corrimiento.

FUNCT: Este campo selecciona una variante de la instrucción especificada en el opcode. También se le denomina código de función.

Formato de Instrucción tipo I (Immediate):

Este formato lo tienen todas las instrucciones donde uno de los operandos es un número de 12 o 16 bits que representa una constante o dirección. Los 25 bits de la instrucción se distribuyen en 3 campos cuando la constante o dirección es de 16 bits y en 4 campos cuando es de 12 bits, tal como se muestra a continuación:

24.....20	19.....16	15.....0
OP	Rd	Constante o Dirección
5 bits	4 bits	16 bits

El significado de cada uno de los campos es:

OP: Es el código de operación.

Rd: Es el registro operando destino.

Constante: dato de 16 bits que representa un número inmediato o dirección.

24.....20	19.....16	15.....12	11.....0
OP	Rd	Rt	Constante o Dirección
5 bits	4 bits	4 bits	12 bits

El significado de cada uno de los campos es:

OP: Es el código de operación.

Rd: Es el registro operando destino.

Rt: Es el primer registro operando fuente

Dirección: dato de 12 bits que representan un número inmediato o dirección.

Formato de Instrucción tipo J (Jump):

Este formato lo tienen las instrucciones de salto. Los 25 bits de la instrucción se distribuyen en 3 campos, tal como se muestra a continuación:

24.....20	19.....16	15.....0
OP	Sin uso	Constante o dirección
5 bits	4 bits	16 bits

El significado de cada uno de los campos es:

OP: Es el código de operación.

Constante o dirección: Representa la dirección de 16 bits en las instrucciones de salto.

Con estos formatos de instrucción se diseñan todas las instrucciones del microprocesador. Dichas instrucciones se describen a continuación.

CONJUNTO DE INSTRUCCIONES.

Las instrucciones propuestas en el ESCOMIPS son instrucciones que se pueden agrupar en las siguientes categorías:

- Instrucciones de carga y almacenamiento
- Instrucciones aritméticas
- Instrucciones lógicas
- Instrucciones de corrimiento
- Instrucciones de brincos condicionales e incondicionales.
- Instrucciones de manejo de subrutinas.

Estas instrucciones se muestran en la tabla 1.

AUTOR: VICTOR HUGO GARCÍA ORTEGA

INSTRUCCIONES DE CARGA Y ALMACENAMIENTO										
Instr.	Ejemplo	Significado	Formato de instrucción				Formato		Banderas	
LI	LI Rd, #Slit16	Rd = Slit16	01	Rd	Slit16			I	Ninguna	
LWI	LWI Rd, lit16	Rd = Mem[lit16]	02	Rd	lit16			I	Ninguna	
LW	LW Rd,lit12(Rt)	Rd = Mem[Rt+lit12]	23	Rd	Rt	lit12		I	C N Z OV	
SWI	SWI Rd, lit16	Mem[lit16] = Rd	03	Rd	lit16			I	Ninguna	
SW	SW Rd, lit12(Rt)	Mem[Rt+lit12] = Rd	04	Rd	Rt	lit12		I	C N Z OV	
INSTRUCCIONES ARITMÉTICAS										
ADD	ADD Rd,Rt,Rs	Rd = Rt+Rs	00	Rd	Rt	Rs	S/U	00	R	C N Z OV
SUB	SUB Rd,Rt,Rs	Rd = Rt-Rs	00	Rd	Rt	Rs	S/U	01	R	C N Z OV
ADDI	ADDI Rd,Rt,#Slit12	Rd = Rt+Slit12	05	Rd	Rt	Slit12		I	C N Z OV	
SUBI	SUBI Rd,Rt,#Slit12	Rd = Rt-Slit12	06	Rd	Rt	Slit12		I	C N Z OV	
INSTRUCCIONES LÓGICAS										
AND	AND Rd,Rt,Rs	Rd=Rt&Rs	00	Rd	Rt	Rs	S/U	02	R	N Z
OR	OR Rd,Rt,Rs	Rd=Rt Rs	00	Rd	Rt	Rs	S/U	03	R	N Z
XOR	XOR Rd,Rt,Rs	Rd=Rt ^ Rs	00	Rd	Rt	Rs	S/U	04	R	N Z
NAND	NAND Rd,Rt,Rs	Rd=~(Rt & Rs)	00	Rd	Rt	Rs	S/U	05	R	N Z
NOR	NOR Rd,Rt,Rs	Rd=~(Rt Rs)	00	Rd	Rt	Rs	S/U	06	R	N Z
XNOR	NOR Rd,Rt,Rs	Rd=~(Rt ^ Rs)	00	Rd	Rt	Rs	S/U	07	R	N Z
NOT	NOT Rd, Rs	Rd = ~Rs	00	Rd	Rs	Rs	S/U	08	R	N Z
ANDI	ANDI Rd,Rt,#lit12	Rd=Rt & lit12	07	Rd	Rt	lit12		I	N Z	
ORI	ORI Rd,Rt,#lit12	Rd=Rt lit12	08	Rd	Rt	lit12		I	N Z	
XORI	XORI Rd,Rt,#lit12	Rd=Rt ^ lit12	09	Rd	Rt	lit12		I	N Z	
NANDI	NANDI Rd,Rt,#lit12	Rd=~(Rt & lit12)	10	Rd	Rt	lit12		I	N Z	

AUTOR: VICTOR HUGO GARCÍA ORTEGA

NORI	NORI Rd,Rt,#lit12	Rd=~(Rt lit12)	11	Rd	Rt	lit12			I	N Z
XNORI	XNORI Rd,Rt,#lit12	Rd=~(Rt ^ lit12)	12	Rd	Rt	lit12			I	N Z
INSTRUCCIONES DE CORRIMIENTO										
SLL	SLL Rd,Rt,#lit4	Rd=Rt<<lit4	00	Rd	Rt	S/U	lit4	09	R	N Z
SRL	SRL Rd,Rt,#lit4	Rd=Rt>>lit4	00	Rd	Rt	S/U	lit4	10	R	N Z
INSTRUCCIONES DE SALTOS CONDICIONALES E INCONDICIONALES										
BEQI	BEQI Rd,Rt,Slit12	If(Rt==Rd) goto Slit12 PC = PC + Slit12	13	Rd	Rt	Slit12			I	C N Z OV
BNEI	BNEI Rd,Rt,Slit12	If(Rt!=Rd) goto Slit12 PC = PC + Slit12	14	Rd	Rt	Slit12			I	C N Z OV
BLTI	BLTI Rd,Rt,Slit12	If(Rt<Rd) goto Slit12 PC = PC + Slit12	15	Rd	Rt	Slit12			I	C N Z OV
BLETI	BLETI Rd,Rt,Slit12	If(Rt<=Rd) goto Slit12 PC = PC + Slit12	16	Rd	Rt	Slit12			I	C N Z OV
BGTI	BGTI Rd,Rt,Slit12	If(Rt>Rd) goto Slit12 PC = PC + Slit12	17	Rd	Rt	Slit12			I	C N Z OV
BGETI	BGETI Rd,Rt,Slit12	If(Rt>=Rd) goto Slit12 PC = PC + Slit12	18	Rd	Rt	Slit12			I	C N Z OV
B	B lit16	PC = lit16	19	S/U	lit16			J	Ninguna	
INSTRUCCIONES DE MANEJO DE SUBROUTINAS										
CALL	CALL #lit16	SP++ PC(SP) = lit16	20	S/U	lit16			J	Ninguna	
RET	RET	SP-- PC = PC(SP)	21	S/U	S/U	S/U	S/U	S/U		Ninguna
OTRAS INSTRUCCIONES										
NOP	NOP		22	S/U	S/U	S/U	S/U	S/U		Ninguna

Tabla 1 Instrucciones del ESCOMIPS

AUTOR: VICTOR HUGO GARCÍA ORTEGA

Instrucciones de carga y almacenamiento.

Estas instrucciones se encargan de transferir datos de la memoria a los registros y viceversa. La instrucción LI carga un valor inmediato a cualquier registro. Las instrucciones LWI y SWI cargan un valor de memoria a registro y registro a memoria respectivamente. Las instrucciones LW y SW también cargan un valor de memoria a registro y registro a memoria respectivamente, pero la dirección de memoria se obtiene sumando una literal de 12 bits con un registro. Esto sirve para manejar arreglos. Estas instrucciones soportan direccionamiento inmediato.

Instrucciones aritméticas

Estas instrucciones consisten en las operaciones aritméticas suma y resta. Estas instrucciones soportan direccionamiento inmediato y por registro.

Instrucciones lógicas

Estas instrucciones consisten en las operaciones lógicas: AND, OR, XOR, NAND, NOR, XNOR y NOT. Estas instrucciones soportan direccionamiento inmediato y por registro.

Instrucciones de corrimiento

Estas instrucciones consisten en las operaciones de corrimiento a la izquierda y a la derecha (*SLL* – Shift Left Logical, *SRL* – Shift Right Logical). El número de bits a desplazarse se especifica en el campo SHAMT dentro del formato tipo R. Estas instrucciones soportan direccionamiento por registro.

Instrucciones de brincos condicionales e incondicionales.

Las instrucciones de brinco condicionales inmediatas (*BEQI* – Branch Equal Immediate, *BNEI* – Branch Not Equal Immediate, *BLTI* – Branch Less Than Immediate, *BLETI* – Branch Less Equal Than Immediate, *BGTI* – Branch Greater Than Immediate, *BGETI* – Branch Greater Equal Than Immediate) realizan la comparación de los registros y su respectivo salto en un ciclo de reloj. En estas instrucciones la dirección de salto es de 12 bits permitiendo lo que conocemos como saltos relativos en una ventana de 2K hacia adelante o hacia atrás a partir de la posición actual del registro contador de programa.

La instrucción de brinco incondicional (*B*) realiza el salto a la dirección de memoria de 16 bits especificada en la instrucción.

Instrucciones de manejo de subrutinas.

Estas instrucciones permiten manejar funciones. La instrucción CALL permite llamar a una función o subrutina. Para esto, primero se debe incrementar el registro STACK POINTER con lo que se selecciona el siguiente registro PC disponible, en este registro se carga la dirección donde se encuentra la función. Al terminar la ejecución de la función se debe colocar la instrucción RET, con esta instrucción se decrementa el registro STACK POINTER para regresar al registro PC anterior y se incrementa el contenido de este registro PC. Con esto, regresamos a la siguiente instrucción

AUTOR: VICTOR HUGO GARCÍA ORTEGA

después del llamado de la función.