

Resumen MIPS 32

Nombre: Martin Natera C.I:30445341 Carrera: Computacion

Lenguaje ensamblador MIPS:

Operandos MIPS:

Aritmetica:

Instrucción	Ejemplo	Significado	Comentarios
add:	<i>add \$s1, \$s2, \$s3</i>	$\$s1 = \$s2 + \$s3$	Tres operandos; datos en registros.
subtract:	<i>sub \$s1, \$s2, \$s3</i>	$\$s1 = \$s2 - \$s3$	Tres operandos; datos en registros
add immediate:	<i>addi \$s1, \$s2, 100</i>	$\$s1 = \$s2 + 100$	Usado para sumar constantes

Transferencia de datos:

Instrucción	Ejemplo	Comentarios	Comentarios
load word	<i>lw \$s1, 100(\$s2)</i>	$\$s1 = \text{Memory}[\$s2 + 100]$	Palabra de memoria a registro
store word	<i>sw \$s1, 100(\$s2)</i>	$\text{Memory}[\$s2 + 100] = \$s1$	Palabra de registro a memoria
load half	<i>lh \$s1, 100(\$s2)</i>	$\$s1 = \text{Memory}[\$s2 + 100]$	Media palabra de memoria a registro
store half	<i>sh \$s1, 100(\$s2)</i>	$\text{Memory}[\$s2 + 100] = \$s1$	Media palabra de registro a memoria
load byte	<i>lb \$s1, 100(\$s2)</i>	$\$s1 = \text{Memory}[\$s2 + 100]$	Byte de memoria a registro
store byte	<i>sb \$s1, 100(\$s2)</i>	$\text{Memory}[\$s2 + 100] = \$s1$	Byte de registro a memoria
load upper immed.	<i>lui \$s1, 100</i>	$\$s1 = 100 * 2^{16}$	Cargar constante en los 16 bits de mayor peso

Lógica:

Instrucción	Ejemplo	Significado	Comentarios
and	<i>and \$s1, \$s2, \$s3</i>	$\$s1 = \$s2 \& \$s3$	Tres registros operandos; AND bit-a-bit
or	<i>or \$s1, \$s2, \$s3</i>	$\$s1 = \$s2 \mid \$s3$	Tres registros operandos; OR bit-a-bit
nor	<i>nor \$s1, \$s2, \$s3</i>	$\$s1 = \sim (\$s2 \mid \$s3)$	Tres registros operandos; NOR bit-a-bit
and immediate	<i>andi \$s1, \$s2, 100</i>	$\$s1 = \$s2 \& 100$	AND Bit-a-bit registro con constante
or immediate	<i>ori \$s1, \$s2, 100</i>	$\$s1 = \$s2 \mid 100$	OR Bit-a-bit registro con constante
shift left logical	<i>sll \$s1, \$s2, 10</i>	$\$s1 = \$s2 \ll 10$	Desplazamiento a la izquierda por constante
shift right logica	<i>srl \$s1, \$s2, 10</i>	$\$s1 = \$s2 \gg 10$	Desplazamiento a la derecha por constante

Salto condicional:

Instrucción	Ejemplo	Significado	Comentarios
branch on equal	beq \$s1,\$s2,L	if (\$s1 == \$s2) go to L PC + 4 + 100	Comprueba igualdad y bifurca relativo al PC
branch on not equal	bne \$s1,\$s2,L	if (\$s1 != \$s2) go to L PC + 4 + 100	Comprueba si no igual y bifurca relativo al PC
set on less than	slt \$s1,\$s2,\$s3	if (\$s2 < \$s3) \$s1 = 1; else \$s1 = 0	Compara si es menor que; usado con beq, bne
set on less than immediate	slt \$s1,\$s2,100	if (\$s2 < 100) \$s1 = 1; else \$s1 = 0	Compara si es menor que una constante

Salto incondicional:

Instrucción	Ejemplo	Significado	Comentarios
jump	j 2500	go to 10000	Salto a la dirección destino
jump register	jr \$ra	go to \$ra	Para retorno de procedimiento
jump and link	jal 2500	\$ra = PC + 4; go to 10000	Para llamada a procedimiento

Registros:

Nombre	Número de registros	Uso	Preservado en llamada
\$zero	0	El valor constante 0	n.a.
\$v0-\$v1	2-3	Valores para los resultados y evaluación de la expresión	no
\$a0-\$a3	4-7	Argumentos	no
\$t0-\$t7	8-15	Temporales	no
\$s0-\$s7	16-23	Guardados	sí
\$t8-\$t9	24-25	Más temporales	no
\$gp	28	Puntero global	sí
\$sp	29	Puntero de la pila	sí
\$fp	30	Puntero de estructura	sí
\$ra	31	Dirección de retorno	sí