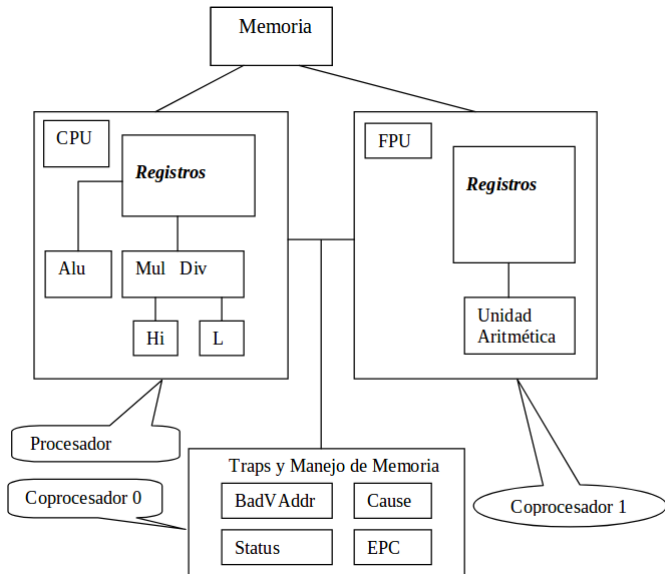


Arquitecturas Computacionales

Repertorio MIPS

Facultad de Ingeniería / Escuela de Informática
Universidad Andrés Bello, Viña del Mar.

Procesador MIPS



Formato de instrucciones MIPS

- Es un procesador RISC, donde las instrucciones ocupan solo una palabra
- MIPS posee tres formatos: R, I y J
- Formato R (para operaciones):

Op 6	Rs 5	Rt 5	Rd 5	Shamnt 5	Funct 6
------	------	------	------	----------	---------

- Fuentes de datos: Rs, Rt
- Resultado: Rd

- Para instrucciones inmediatas, transferencias y bifurcaciones

Op 6	Rs 5	Rt 5	Inmediato 16
------	------	------	--------------

- en operaciones inmediatas:
 - Rs es el otro operando
 - Rt donde se guarda el resultado

- Para instrucciones inmediatas, transferencias y bifurcaciones

Op 6	Rs 5	Rt 5	Inmediato 16
------	------	------	--------------

- en transferencias:
 - Rs es el registro base
 - el valor inmediato es el desplazamiento (con signo)
 - Rt especifica el registro donde cargar un dato desde memoria en `load`, o el valor escrito en una palabra de la memoria en `store`

- Para instrucciones inmediatas, transferencias y bifurcaciones

Op 6	Rs 5	Rt 5	Inmediato 16
------	------	------	--------------

- en bifurcaciones:
 - registros especifican operandos que serán comparados
 - el campo de 16 bits se interpreta como desplazamiento (con signo), relativo al PC (program counter)

- Para saltos

Op 6	Dirección 26
------	--------------

- el campo de 26 bits se usa para configurar una dirección de salto

Instrucciones aritméticas y lógicas

Instrucción	Ejemplo	Significado en C.
add	add \$t1, \$t2,\$t3	$\$t1 = \$t2 + \$t3$
add unsigned	addu \$t1,\$t2,\$t3	$\$t1 = \$t2 + \$t3$
subtract	sub \$t1,\$t2,\$t3	$\$t1 = \$t2 - \$t3$
subtract unsigned	subu \$t1,\$t2,\$t3	$\$t1 = \$t2 - \$t3$
and	and \$t1,\$t2,\$t3	$\$t1 = \$t2 \& \$t3$
or	or \$t1,\$t2,\$t3	$\$t1 = \$t2 \$t3$
xor	xor \$t1,\$t2,\$t3	$\$t1 = \$t2 \wedge \$t3$
nor	nor \$t1,\$t2,\$t3	$\$t1 = \sim(\$t2 \$t3)$
set on less than	slt \$t1,\$t2,\$t3	if ($\$t2 < \$t3$) $\$t1=1$; else $\$t1=0$;
set less than unsigned	sltu \$t1,\$t2,\$t3	if ($\$t2 < \$t3$) $\$t1=1$; else $\$t1=0$;

Instrucciones aritméticas y lógicas (Inmediatas)

Instrucción	Ejemplo	Significado en C.
add immediate	addi \$t1,\$t2,100	\$t1 = \$t2 + 100
add imm. unsigned	addiu \$t1,\$t2,100	\$t1 = \$t2 + 100
and immediate	andi \$t1,\$t2,10	\$t1 = \$t2 & 10
or immediate	ori \$t1,\$t2,10	\$t1 = \$t2 10
xor immediate	xori \$t1,\$t2,10	\$t1 = \$t2 ^ 10
set less than imm	slti \$t1,\$t2,100	if (\$t2 < 100) \$t1=1; else \$t1=0;
set less than imm. uns.	sltiu \$t1,\$t2,100	if (\$t2 < 100) \$t1=1; else \$t1=0
load upper immediate	lui \$t1, 40	

Transferencia de datos

Instrucción	Ejemplo	Significado en C.
Load byte	LB \$t1, 40(\$t2)	$\$t1 = M[(\$t2) + 40]$
Load byte unsigned	LBU \$t1, 40(\$t2)	$\$t1 = M[(\$t2) + 40]$
Load halfword	LH \$t1, 40(\$t2)	$\$t1 = M[(\$t2) + 40]$
Load halfword unsigned	LHU \$t1, 40(\$t2)	$\$t1 = M[(\$t2) + 40]$
Load word	LW \$t1, 40(\$t2)	$\$t1 = M[(\$t2) + 40]$
Store byte	SB \$t1, 41(\$t2)	$M[(\$t2) + 41] = \$t1$
Store half	SH \$t1, 42(\$t2)	$M[(\$t2) + 42] = \$t1$
Store word	SW \$t1, 100(\$t2)	$M[(\$t2) + 100] = \$t1$

Comparaciones entre registros y bifurcación

Instrucción	Ejemplo	Significado en C.
Branch on equal	beq \$t1,\$t2,100	if (\$t1 == \$t2) PC =(PC+4)+100*4;
branch on not eq.	bne \$t1,\$t2,100	if (\$t1!= \$t2) PC =(PC+4)+100*4;

- Desarrolle un programa con instrucciones para un procesador MIPS que haga la resta de dos números y verifique si el resultado es cero o no.

- Desarrolle un programa con instrucciones para un procesador MIPS que verifique si un número es par o no.