

Tema 2: El procesador

1 Instrucciones en MIPS

MIPS tiene 32 bits y 32 registros.

Solo puede tener 2^{32} Bytes = 4 GBy de memoria

Por lo tanto como máximo habrán $\frac{2^{32}}{4} = 2^{30}$ palabras.

La memoria solo lee palabras (1 palabra = 4 Bytes).

1.1 Identificación de los registros

Del registro 8 al 15 se identifican con \$t0 a \$t7

Del registro 16 a 23 se identifican con \$s0 a \$s7

También se pueden identificar de \$0 a \$31. El \$0 siempre vale 0

1.2 Instrucciones MIPS

- **Tipo R**(Registro) Operan exclusivamente en registros.(**rs**, **rt**, **rd**)

`add rd, rs, rt`: Suma el contenido de los registros **rs** y **rt** y almacena el resultado en **rd**.

`sub rd, rs, rt`: Resta el contenido de **rt** y **rs**, y guarda el resultado en **rd**.

`and rd, rs, rt`: Realiza una operación lógica AND entre los contenidos de **rs** y **rt**, y almacena el resultado en **rd**.

- **Tipo I**(Inmediato) Incluyen un valor inmediato como un operando

`lw rt, inmed(rs)`: Carga un valor desde la memoria en la dirección **rs + inmed** y lo almacena en **rt**.

`sw rt, inmed(rs)`: Guarda el contenido de **rt** en la memoria en la dirección **rs + inmed**.

`beq rs, rt, etiqueta`: Si los registros **rs** y **rt** son iguales, realiza un salto a la etiqueta especificada.

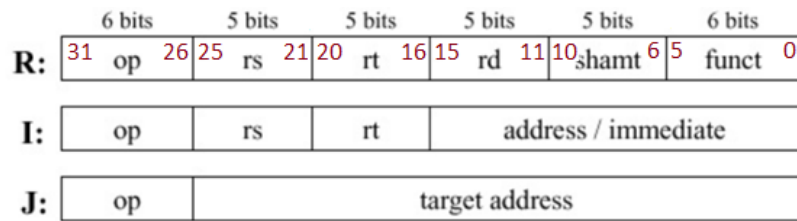
- **Tipo J**(Salto incondicional) Solo contienen una dirección de destino

1.3 Ejemplos

Instrucción MIPS	Equivalencia en ARM
add \$1,\$2,\$3	add \$1,\$2,\$3
lw \$1,20(\$3)	ldr \$1,20(\$3)
sw \$3,1(\$4)	str \$3,1(\$4)
beq \$1,\$2,fin	cmp \$1,\$2 beq fin

add $[rs] + [rt] \implies rd$
lw $[rs] + \text{Inmediato} \implies rt$
sw $[rt] \implies [rs] + \text{inmediato}$
beq $[rs] - [rt]$

1.4 Representación de instrucciones en el computador



- Valor de **op** según instrucción:

	dec.	hexa.	binario
tipo R	0	0x0	000000
lw	35	0x23	100011
sw	43	0x2B	101011
beq	4	0x04	000010

- Valor de **funct** según la operación aritmética:

	dec.	hexa.	binario
add (suma)	32	0x20	100000
sub (resta)	34	0x22	100010
and	43	0x24	100100
or	44	0x25	100101

- El campo **shamt** especifica cuántos bits se desplazarán a la izquierda

1.5 Señales de control

Se activan cuando:

- **MemWrite**: Escribe en memoria
- **MemRead**: Lee de memoria
- **Cero**: Cuando ALU=0
- **Branch**: Compara y redirige
- **RegWrite**: Cuando se hace una operación de escritura en un registro