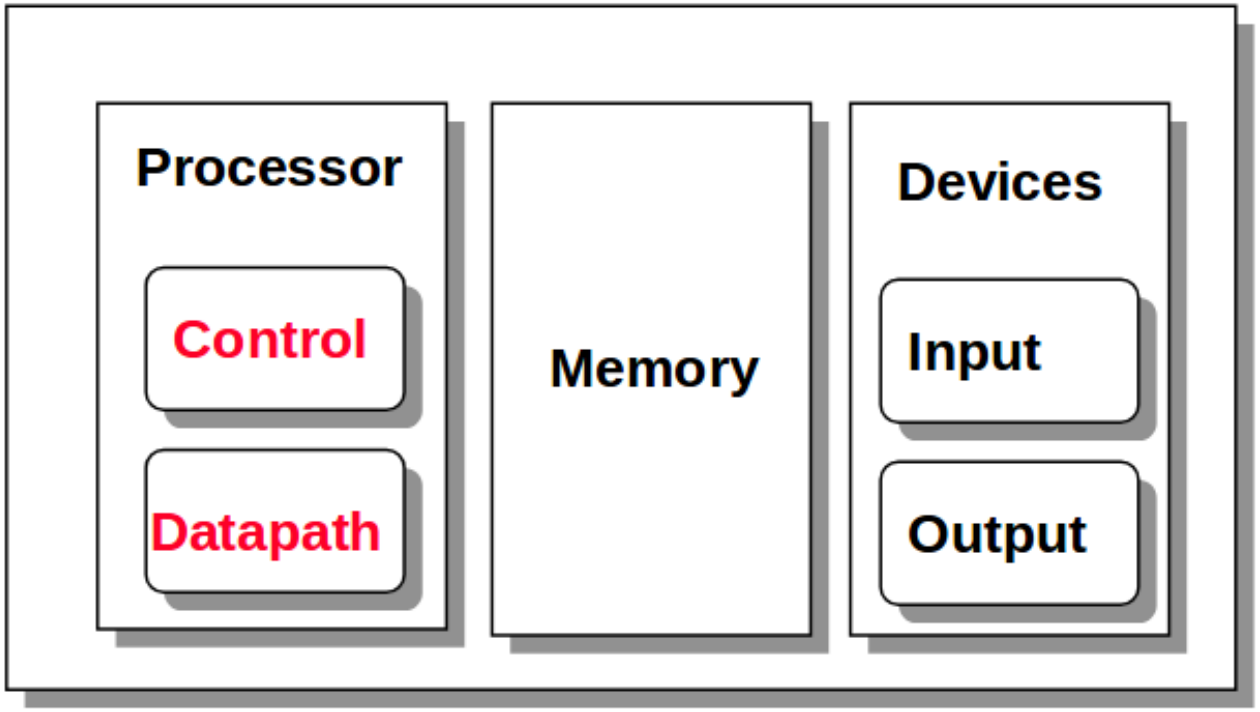
Arquitecturas Computacionales

Arreglos

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

Registros vs Memoria

- Operandos de instrucciones aritméticas deben ser registros (solo 32 disponibles)



- Los compiladores asocian variables con registros

Accediendo a la Memoria

- MIPS tiene dos instrucciones básicas de transferencia de datos para acceder a la memoria:
  - lw $t0, 4($s3) : carga palabra desde memoria
  - sw $t0, 8($s3) : guarda palabra en memoria
- La instrucción de transferencia de datos debe especificar:
  - la dirección de memoria (desde donde leer o hacia donde escribir)
  - en que registros cargar información o cuales leer

UNIVERSIDAD ANDRES BELLO

Accediendo a la Memoria

- MIPS tiene dos instrucciones básicas de transferencia de datos para acceder a la memoria:
  - lw $t0, 4($s3) : carga palabra desde memoria
  - sw $t0, 8($s3) : guarda palabra en memoria
- La dirección de memoria se forma al sumar la dirección constante de la instrucción y los contenidos del segundo registro.

UNIVERSIDAD ANDRES BELLO

Ejercicio

Suponga que la variable `h` está asociada con el registro `$s2`, y la dirección base del arreglo `A` está en `$s3`, indique el equivalente en assembler de:

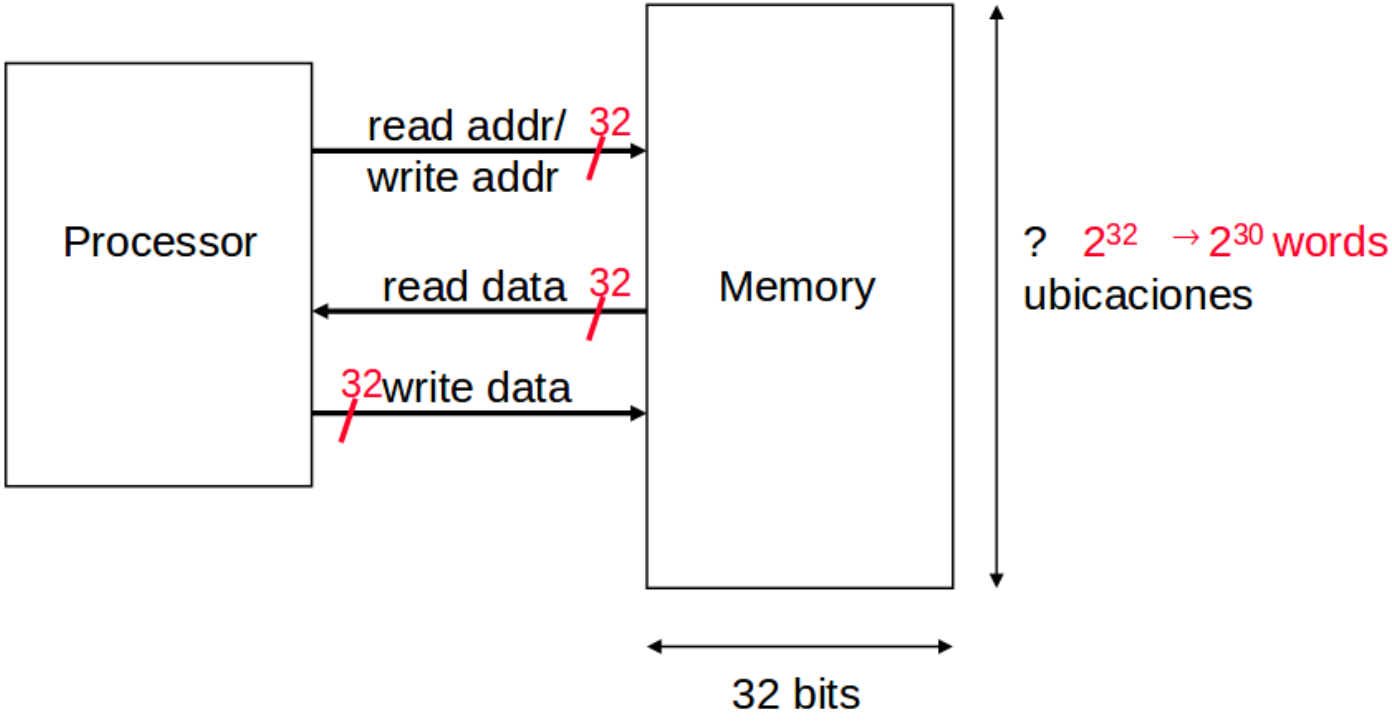
```
A[12] = h + A[8]
```

Solución

```
lw $t0, 32($s3)
add $t0, $s2, $t0
sw $t0, 48($s3)
```

Interconexión del Procesador y Memoria

- La memoria se ve como un arreglo de ubicaciones de almacenamiento con direcciones
- Una dirección de memoria es un índice para el arreglo



Tipos de datos MIPS

- Bit: 0, 1
- Bit string: secuencia de bits de un largo particular
  - 4 bits es un nibble
  - 8 bits es un byte
  - 16 bits es media palabra
  - 32 bits (4 bytes) es una palabra
  - 64 bits es una palabra doble

Tipos de datos MIPS

- Caracter: código ASCII de 7 bits
- Decimal: digitos 0$\rightarrow$9
  - codificados como 0000b a 1001b
  - dos dígitos decimales por cada byte de 8 bits
- enteros: en complemento 2
- punto flotante

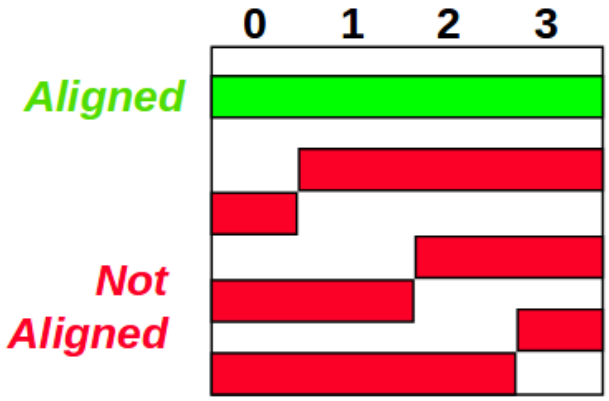
Direcciones de bytes

- la mayoría de las arquitecturas direccionan bytes individuales de la memoria
- la dirección de memoria de una palabra tiene que ser múltiplo de 4 (restricción de alineamiento)
- Procesadores MIPS generalmente pueden ser configurados como Big o Little Endian
  - Big Endian: byte de la izquierda es dirección de `word`
  - Little Endian: byte de la derecha es dirección de `word`

Direcciones de bytes

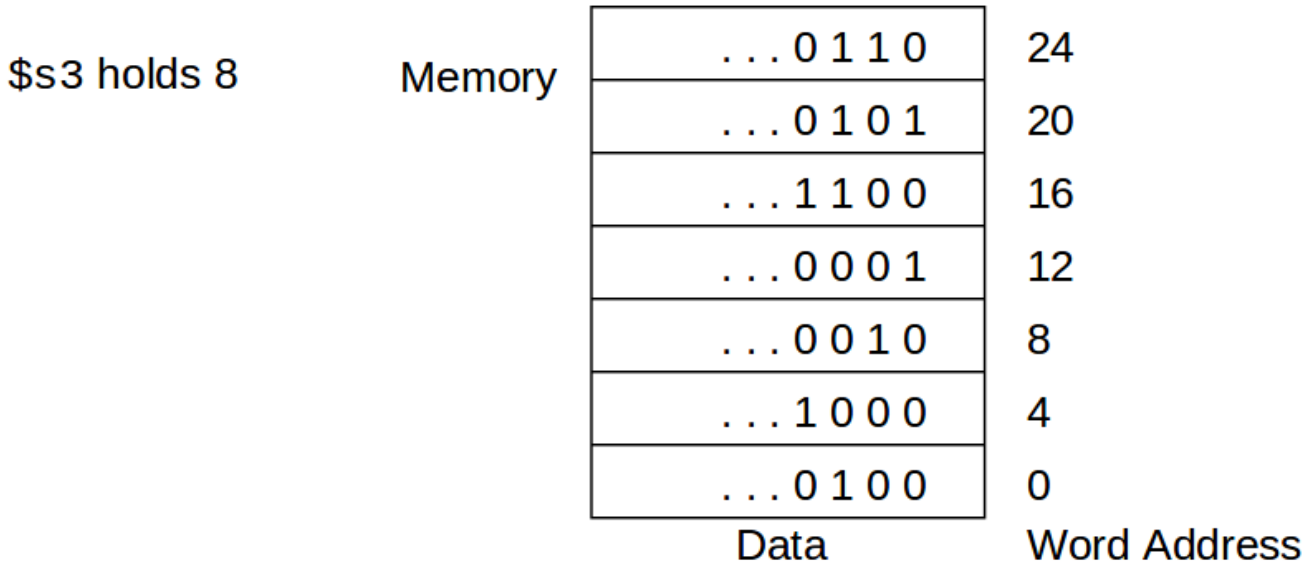
- Procesadores MIPS generalmente pueden ser configurados como Big o Little Endian
  - Big Endian: byte de la izquierda es dirección de `word`
  - Little Endian: byte de la derecha es dirección de `word`

big endian



Direccionamiento de memoria en MIPS

- La dirección de memoria se forma sumando la parte constante de la instrucción y el contenido del segundo registro (base) (Suponer Little Endian)

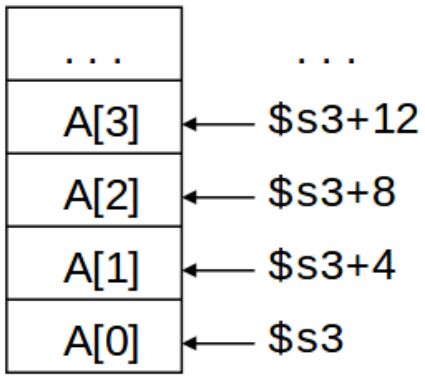


- `lw $t0,4($s3)` : qué se carga en $t0 ?
- `sw $t0,8($s3)` : $t0 se almacena dónde?

Compilando con Loads y Stores

- Si la variable `b` se almacena en `$s2` y la dirección base de un arreglo `A` está en `$s3`, cuál es el código assembler MIPS para el siguiente código en C?

```
A[8] = A[2] - b
```



Solución

```
lw $t0, 8($s3)
sub $t0, $t0, $s2
sw $t0, 32($s3)
```

UNIVERSIDAD ANDRES BELLO

Compilando con un índice de un arreglo

- Asumiendo que `A` es un arreglo de 50 elementos para los cuales su base está en `$s4`, y que las variables `b`, `c`, `i` están en `$s1`, `$s2` y `$s3` respectivamente. Indique el código MIPS para la siguiente línea en C:
  `c = A[i] -b`

Solución

```
add $t1, $s3, $s3
add $t1, $t1, $t1
add $t1, $t1, $s4
lw $t0, 0($t1)
sub $s2, $t0, $s1
```