

1. Se tivermos 2 inteiros, cada um com 32 bits, quantos bits podemos esperar para o produto?

C. 64

2. Quais os registradores que armazenam os resultados na multiplicação?

B. hi e lo

3. Qual a operação usada para multiplicar inteiros em comp. de dois?

A. mult

4. Qual instrução move os bits menos significativos da multiplicação para o reg. 8?

C. mflo \$8

5. Se tivermos dois inteiros, cada um com 32 bits, quantos bits deveremos estar preparados para receber no quociente?

B. 32

6. Após a instrução div, qual registrador possui o quociente?

A. lo

7. Qual a inst. Usada para dividir dois inteiros em comp. de dois?

D. div

8. Faça um arithmetic shift right de dois no seguinte padrão de bits: 1001 1011

Como não sei se você está considerando que esse valor está guardado em um registrador do MIPS, vou dar duas respostas:

Para registradores de 32 bits, ou seja, para o MIPS:

B. 0010 0110

Para registradores de 8 bits, ou seja, uma máquina imaginária com as mesmas regras do MIPS:

A. 1110 0110

9. Qual o efeito de um arithmetic shift right de uma posição?

C. Se o inteiro for unsigned, o shift pode ocasionar um valor errado. Se o inteiro for signed, o shift o divide por 2.

10. Qual sequencia de instruções avalia  $3x+7$ , onde x é iniciado no reg. \$8 e o resultado armazenado em \$9?

A.

ori \$3,\$0,3

mult \$8,\$3

mflo \$9

addi \$9,\$9,7

The screenshot displays the MARS 4.5 MIPS assembler simulator interface. The main window shows assembly code for a MIPS program. The 'Registers' panel on the right displays the state of MIPS registers, with \$a0 highlighted. The 'Mars Messages' panel at the bottom shows a message indicating the program has finished running.

**Assembly Code (Text Segment):**

```

11: lui    $t1, 0x1001          # Coloca o endereço base da memória no registrador...
13: lw     $s0, 0($t1)          # $s0 = MEM[0x10010000] = A
15: slt    $t2, $s0, $zero      # $t2 = A < 0 ? 1 : 0
16: beq    $t2, $zero, save_A    # A >= 0 ?      jump save_A
17: sub    $s0, $zero, $s0      # $t2 = 0 - A = -A
20: sw     $s0, 0($t1)          # MEM[0x10010000] = |A|

```

**Registers Panel:**

| Name   | Number | Value      |
|--------|--------|------------|
| \$zero | 0      | 0x00000000 |
| \$at   | 1      | 0x00000000 |
| \$v0   | 2      | 0x00000000 |
| \$v1   | 3      | 0x00000000 |
| \$a0   | 4      | 0x00000000 |
| \$a1   | 5      | 0x00000000 |
| \$a2   | 6      | 0x00000000 |
| \$a3   | 7      | 0x00000000 |
| \$t0   | 8      | 0x00000000 |
| \$t1   | 9      | 0x10010000 |
| \$t2   | 10     | 0x00000001 |
| \$t3   | 11     | 0x00000000 |
| \$t4   | 12     | 0x00000000 |
| \$t5   | 13     | 0x00000000 |
| \$t6   | 14     | 0x00000000 |
| \$t7   | 15     | 0x00000000 |
| \$a0   | 16     | 0x00000000 |
| \$a1   | 17     | 0x00000000 |
| \$a2   | 18     | 0x00000000 |
| \$a3   | 19     | 0x00000000 |
| \$s4   | 20     | 0x00000000 |
| \$s5   | 21     | 0x00000000 |
| \$s6   | 22     | 0x00000000 |
| \$s7   | 23     | 0x00000000 |
| \$s8   | 24     | 0x00000000 |
| \$s9   | 25     | 0x00000000 |
| \$k0   | 26     | 0x00000000 |
| \$k1   | 27     | 0x00000000 |
| \$x0   | 28     | 0x10008000 |
| \$sp   | 29     | 0x7ffffefc |
| \$fp   | 30     | 0x00000000 |
| \$ra   | 31     | 0x00000000 |
| pc     |        | 0x00400018 |
| hi     |        | 0x00000000 |
| lo     |        | 0x00000000 |

**Mars Messages:**

```

-- program is finished running (dropped off bottom) --

```

The screenshot displays the Mars Mips Assembler/Debugger interface. The main window is divided into several panes:

- Top Bar:** Shows the file path "C:\Users\A1137910\Downloads\Relatório15 - LACmp2asm.asm - MARS 2.5.0" and standard menu options (File, Edit, Run, Settings, Tools, Help).
- Assembly Code Pane:** Displays assembly instructions with their binary, code, basic, and source representations. The code is as follows:
 

| Bkpt                     | Address    | Code       | Basic                        | Source  |
|--------------------------|------------|------------|------------------------------|---|
| <input type="checkbox"/> | 0x00400000 | 0x3c101001 | lui \$t6, 0x000001001        | 9: lui \$t6, 0x1001 # calcula o endereço \$t6                                   |
| <input type="checkbox"/> | 0x00400004 | 0xe1100000 | lw \$t7, 0x00000000(\$t6)    | 11: lw \$t7, 0 (\$t6) # carrega o primeiro valor na memória                     |
| <input type="checkbox"/> | 0x00400008 | 0x2a28001e | slli \$t0, \$t1, 0x00000001e | 13: slli \$t0, \$t1, 30 # \$t0 = (\$t1 < 30 ? 1 : 0)                            |
| <input type="checkbox"/> | 0x0040000c | 0x15000002 | bne \$t0, \$t0, 0x000000002  | 14: bne \$t0, \$zero, flag0 # vai para flag0 se \$t0 != 0, ou seja, se \$t0 ... |
| <input type="checkbox"/> | 0x00400010 | 0x2a280033 | slli \$t1, \$t1, 0x000000033 | 15: slli \$t1, \$t1, 51 # \$t0 = (\$t1 < 51 ? 1 : 0)                            |
| <input type="checkbox"/> | 0x00400014 | 0x15000002 | bne \$t0, \$t0, 0x000000002  | 16: bne \$t0, \$zero, flag1 # vai para flag1 se \$t0 != 0, ou seja, se \$t0 ... |
| <input type="checkbox"/> | 0x00400018 | 0x00004020 | add \$t0, \$t0, \$t0         | 19: add \$t0, \$zero, \$zero  |
| <input type="checkbox"/> | 0x0040001c | 0x08100009 | j 0x004000024                | 20: j store_flag  |
| <input type="checkbox"/> | 0x00400020 | 0x20080001 | addi \$t0, \$t0, 0x000000001 | 23: addi \$t0, \$zero, 1  |
| <input type="checkbox"/> | 0x00400024 | 0xae080004 | sw \$t0, 0x000000004(\$t6)   | 26: sw \$t0, 4 (\$t6)   |
- Data Segment Pane:** Shows memory addresses and their corresponding values in hexadecimal. The values are mostly 0x00000000, with some non-zero values at the end of the segment.
- Registers Pane:** Displays the current state of registers. The registers are numbered 0 to 31, and their values are shown in hexadecimal. The registers are:
 

| Registers | Coproc 1 | Coproc 0 | Name | Number | Value      |
|-----------|----------|----------|------|--------|------------|
| \$zero    |          |          |      | 0      | 0x00000000 |
| \$at      |          |          |      | 1      | 0x00000000 |
| \$v0      |          |          |      | 2      | 0x00000000 |
| \$v1      |          |          |      | 3      | 0x00000000 |
| \$a0      |          |          |      | 4      | 0x00000000 |
| \$a1      |          |          |      | 5      | 0x00000000 |
| \$a2      |          |          |      | 6      | 0x00000000 |
| \$a3      |          |          |      | 7      | 0x00000000 |
| \$t0      |          |          |      | 8      | 0x00000000 |
| \$t1      |          |          |      | 9      | 0x00000000 |
| \$t2      |          |          |      | 10     | 0x00000000 |
| \$t3      |          |          |      | 11     | 0x00000000 |
| \$t4      |          |          |      | 12     | 0x00000000 |
| \$t5      |          |          |      | 13     | 0x00000000 |
| \$t6      |          |          |      | 14     | 0x00000000 |
| \$t7      |          |          |      | 15     | 0x00000000 |
| \$t8      |          |          |      | 16     | 0x00000000 |
| \$t9      |          |          |      | 17     | 0x00000000 |
| \$s0      |          |          |      | 18     | 0x00000000 |
| \$s1      |          |          |      | 19     | 0x00000000 |
| \$s2      |          |          |      | 20     | 0x00000000 |
| \$s3      |          |          |      | 21     | 0x00000000 |
| \$s4      |          |          |      | 22     | 0x00000000 |
| \$s5      |          |          |      | 23     | 0x00000000 |
| \$s6      |          |          |      | 24     | 0x00000000 |
| \$s7      |          |          |      | 25     | 0x00000000 |
| \$s8      |          |          |      | 26     | 0x00000000 |
| \$s9      |          |          |      | 27     | 0x00000000 |
| \$sp      |          |          |      | 28     | 0x00000000 |
| \$fp      |          |          |      | 29     | 0x00000000 |
| \$ra      |          |          |      | 30     | 0x00000000 |
| \$pc      |          |          |      | 31     | 0x00000000 |
| \$hi      |          |          |      |        | 0x00000000 |
| \$lo      |          |          |      |        | 0x00000000 |
- Mars Messages Pane:** Shows the output of the program. The message is:
 

```
-- program is finished running (dropped off bottom) --
```

## Programa 15

C:\Users\1137910\Downloads\Relatório10 - LAC\mips3.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed 30 inst/sec

**Text Segment**

| Bkpt | Address    | Code       | Basic                       | Source                       |
|------|------------|------------|-----------------------------|------------------------------|
|      | 0x00400004 | 0x02209820 | addi \$t0, \$t0, 1          | 14: add \$t2, \$t1, \$zero   |
|      | 0x00400008 | 0x02010000 | addi \$t1, \$t0, 0          | 15: addi \$t0, \$zero, 0     |
|      | 0x0040000c | 0x00009820 | addi \$t1, \$t0, 0          | 16: add \$t3, \$zero, \$zero |
|      | 0x00400010 | 0x02140064 | addi \$t0, \$t0, 0x00000064 | 17: addi \$t4, \$zero, 100   |
|      | 0x00400014 | 0x02104020 | addi \$t0, \$t0, 1          | 20: add \$t0, \$t0, \$t0     |
|      | 0x00400018 | 0x02108001 | addi \$t0, \$t0, 1          | 21: addi \$t0, \$t0, 1       |
|      | 0x0040001c | 0x0ae48000 | sw \$t0, 0x00000000(\$t0)   | 22: sw \$t0, 0(\$t2)         |
|      | 0x00400020 | 0x8e480000 | lw \$t0, 0x00000000(\$t0)   | 23: lw \$t0, 0(\$t2)         |
|      | 0x00400024 | 0x22520004 | addi \$t2, \$t2, 4          | 24: addi \$t2, \$t2, 4       |
|      | 0x00400028 | 0x22100001 | addi \$t2, \$t2, 0x00000001 | 25: addi \$t0, \$t0, 1       |
|      | 0x0040002c | 0x0269820  | addi \$t1, \$t1, 8          | 26: add \$t3, \$t3, \$t0     |
|      | 0x00400030 | 0x1614ffff | bne \$t0, \$t4, \$zero      | 27: bne \$t0, \$t4, \$zero   |
|      | 0x00400034 | 0x0ae53000 | sw \$t1, 0x00000000(\$t1)   | 29: sw \$t3, 0(\$t2)         |

**Data Segment**

| Address    | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x100100a0 | 0x00000051 | 0x00000053 | 0x00000055 | 0x00000057 | 0x00000059  | 0x0000005b  | 0x0000005d  | 0x0000005f  |
| 0x100100c0 | 0x00000061 | 0x00000063 | 0x00000065 | 0x00000067 | 0x00000069  | 0x0000006b  | 0x0000006d  | 0x0000006f  |
| 0x100100e0 | 0x00000071 | 0x00000073 | 0x00000075 | 0x00000077 | 0x00000079  | 0x0000007b  | 0x0000007d  | 0x0000007f  |
| 0x10010100 | 0x00000081 | 0x00000083 | 0x00000085 | 0x00000087 | 0x00000089  | 0x0000008b  | 0x0000008d  | 0x0000008f  |
| 0x10010120 | 0x00000091 | 0x00000093 | 0x00000095 | 0x00000097 | 0x00000099  | 0x0000009b  | 0x0000009d  | 0x0000009f  |
| 0x10010140 | 0x000000a1 | 0x000000a3 | 0x000000a5 | 0x000000a7 | 0x000000a9  | 0x000000ab  | 0x000000ad  | 0x000000af  |
| 0x10010160 | 0x000000b1 | 0x000000b3 | 0x000000b5 | 0x000000b7 | 0x000000b9  | 0x000000bb  | 0x000000bd  | 0x000000bf  |
| 0x10010180 | 0x000000c1 | 0x000000c3 | 0x000000c5 | 0x000000c7 | 0x000000c9  | 0x000000cb  | 0x000000cd  | 0x000000cf  |
| 0x100101a0 | 0x000000d1 | 0x000000d3 | 0x000000d5 | 0x000000d7 | 0x000000d9  | 0x000000db  | 0x000000dd  | 0x000000df  |
| 0x100101c0 | 0x000000e1 | 0x000000e3 | 0x000000e5 | 0x000000e7 | 0x000000e9  | 0x000000eb  | 0x000000ed  | 0x000000ef  |
| 0x100101e0 | 0x000000f1 | 0x000000f3 | 0x000000f5 | 0x000000f7 | 0x000000f9  | 0x000000fb  | 0x000000fd  | 0x000000ff  |

**Mars Messages**

Run I/O

Reset: reset completed.

Clear

-- program is finished running (dropped off bottom) --

Registers

| Name   | Number | Value       |
|--------|--------|-------------|
| \$zero | 0      | 0x00000000  |
| \$at   | 1      | 0x00000000  |
| \$v0   | 2      | 0x00000000  |
| \$v1   | 3      | 0x00000000  |
| \$a0   | 4      | 0x10010000  |
| \$a1   | 5      | 0x00000000  |
| \$a2   | 6      | 0x00000000  |
| \$a3   | 7      | 0x00000000  |
| \$t0   | 8      | 0x00000000  |
| \$t1   | 9      | 0x00000000  |
| \$t2   | 10     | 0x00000000  |
| \$t3   | 11     | 0x00000000  |
| \$t4   | 12     | 0x00000000  |
| \$t5   | 13     | 0x00000000  |
| \$t6   | 14     | 0x00000000  |
| \$t7   | 15     | 0x00000000  |
| \$s0   | 16     | 0x00000000  |
| \$s1   | 17     | 0x10010000  |
| \$s2   | 18     | 0x10010190  |
| \$s3   | 19     | 0x00002710  |
| \$s4   | 20     | 0x00000064  |
| \$s5   | 21     | 0x00000000  |
| \$s6   | 22     | 0x00000000  |
| \$s7   | 23     | 0x00000000  |
| \$s8   | 24     | 0x00000000  |
| \$s9   | 25     | 0x00000000  |
| \$k0   | 26     | 0x00000000  |
| \$k1   | 27     | 0x00000000  |
| \$gp   | 28     | 0x10008000  |
| \$sp   | 29     | 0x7ffffeffc |
| \$fp   | 30     | 0x00000000  |
| \$ra   | 31     | 0x00000000  |
| pc     |        | 0x00400038  |
| hi     |        | 0x00000000  |
| lo     |        | 0x00000000  |

## Programa 16

C:\Users\1137910\Downloads\Relatório10 - LAC\mips3.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

**Text Segment**

| Bkpt | Address    | Code       | Basic                       | Source                     |
|------|------------|------------|-----------------------------|----------------------------|
|      | 0x00400070 | 0x21080001 | addi \$t0, \$t0, 1          | 58: addi \$t0, \$t0, 1     |
|      | 0x00400074 | 0x0ae48000 | sw \$t0, 0x00000000(\$t0)   | 59: sw \$t0, 0(\$t2)       |
|      | 0x00400078 | 0x0ae48000 | sw \$t0, 0x00000000(\$t0)   | 60: lw \$t0, 0(\$t2)       |
|      | 0x0040007c | 0x22520004 | addi \$t2, \$t2, 4          | 61: addi \$t2, \$t2, 4     |
|      | 0x00400080 | 0x22100001 | addi \$t2, \$t2, 0x00000001 | 62: addi \$t0, \$t0, 1     |
|      | 0x00400084 | 0x0269820  | addi \$t1, \$t1, 8          | 63: add \$t3, \$t3, \$t0   |
|      | 0x00400088 | 0x1614ffff | bne \$t0, \$t4, \$zero      | 64: bne \$t0, \$t4, \$zero |
|      | 0x0040008c | 0x3c041001 | lui \$t4, 0x00000101        | 66: lui \$a0, 0x1001       |
|      | 0x00400090 | 0x20050000 | addi \$t0, \$t0, 0x00000000 | 67: addi \$a1, \$zero, 0   |
|      | 0x00400094 | 0x20060064 | addi \$t0, \$t0, 0x00000064 | 68: addi \$a2, \$zero, 100 |
|      | 0x00400098 | 0x0c100001 | jai 0x00400004              | 69: jai bubblesort         |
|      | 0x0040009c | 0x8c890004 | lw \$t0, 0x00000004(\$t0)   | 71: lw \$t1, 4(\$a0)       |
|      | 0x004000a0 | 0x0ac89000 | sw \$t0, 0x0000000c(\$t0)   | 72: sw \$t1, 12(\$a0)      |

**Data Segment**

| Address    | Value (+0)   | Value (+4)   | Value (+8)   | Value (+c)   | Value (+10)  | Value (+14)  | Value (+18)  | Value (+1c)  |
|------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 0x10010000 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x10010020 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x10010040 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x10010060 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x10010080 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x100100a0 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x100100c0 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x100100e0 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x10010100 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x10010120 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |
| 0x10010140 | 0xffffffff3b | 0xffffffff3d | 0xffffffff3f | 0xffffffff41 | 0xffffffff43 | 0xffffffff45 | 0xffffffff47 | 0xffffffff49 |

**Mars Messages**

Run I/O

-- program is finished running (dropped off bottom) --

Clear

-- program is finished running (dropped off bottom) --

Registers

| Name   | Number | Value        |
|--------|--------|--------------|
| \$zero | 0      | 0x00000000   |
| \$at   | 1      | 0x00000000   |
| \$v0   | 2      | 0x00000000   |
| \$v1   | 3      | 0x00000000   |
| \$a0   | 4      | 0x10010000   |
| \$a1   | 5      | 0x00000000   |
| \$a2   | 6      | 0x00000000   |
| \$a3   | 7      | 0x00000000   |
| \$t0   | 8      | 0xffffffff3b |
| \$t1   | 9      | 0xffffffff3d |
| \$t2   | 10     | 0x00000000   |
| \$t3   | 11     | 0x00000001   |
| \$t4   | 12     | 0xffffffff   |
| \$t5   | 13     | 0x00000000   |
| \$t6   | 14     | 0x00000063   |
| \$t7   | 15     | 0x00000064   |
| \$s0   | 16     | 0x00000064   |
| \$s1   | 17     | 0x10010000   |
| \$s2   | 18     | 0x10010190   |
| \$s3   | 19     | 0xffffd9b8   |
| \$s4   | 20     | 0x00000064   |
| \$s5   | 21     | 0x00000000   |
| \$s6   | 22     | 0x00000000   |
| \$s7   | 23     | 0x00000000   |
| \$s8   | 24     | 0x1001018e   |
| \$s9   | 25     | 0x00000064   |
| \$k0   | 26     | 0x00000000   |
| \$k1   | 27     | 0x00000000   |
| \$gp   | 28     | 0x10008000   |
| \$sp   | 29     | 0x7ffffeffc  |
| \$fp   | 30     | 0x00000000   |
| \$ra   | 31     | 0x0040009c   |
| pc     |        | 0x004000a4   |
| hi     |        | 0x00000000   |
| lo     |        | 0x00000000   |