



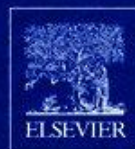
A Nifty Tool for Studying Program and System Behaviors

Pete Sanderson, Otterbein College

Ken Vollmar, Missouri State University

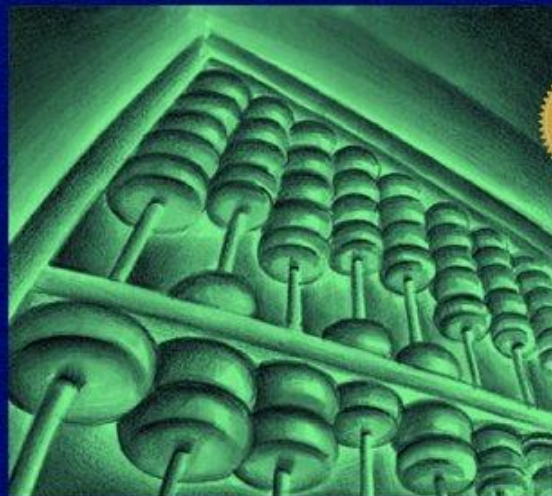
www.cs.missouristate.edu/MARS/

29 September 2007



COMPUTER ORGANIZATION AND DESIGN

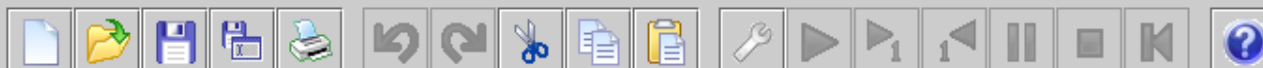
THE HARDWARE/SOFTWARE INTERFACE



DAVID A. PATTERSON
JOHN L. HENNESSY



File Edit Run Settings Tools Help



Run speed at max (no interaction)



Edit Execute

```

1 # Compute first twelve Fibonacci numbers and put in array, then print
2     .data
3     fibs: .word    0 : 12      # "array" of 12 words to contain fib values
4     size: .word    12         # size of "array"
5     .text
6     la    $t0, fibs          # load address of array
7     la    $t5, size          # load address of size variable
8     lw    $t5, 0($t5)        # load array size
9     li    $t2, 1             # 1 is first and second Fib. number
10    sw    $t2, 0($t0)         # F[0] = 1
11    sw    $t2, 4($t0)         # F[1] = F[0] = 1
12    addi   $t1, $t5, -2       # Counter for loop, will execute (size-2) times
13 loop: lw    $t3, 0($t0)      # Get value from array F[n]
14        lw    $t4, 4($t0)      # Get value from array F[n+1]
15        add    $t2, $t3, $t4    # $t2 = F[n] + F[n+1]
16        sw    $t2, 8($t0)      # Store F[n+2] = F[n] + F[n+1] in array
17        addi   $t0, $t0, 4      # increment address of Fib. number source
18        addi   $t1, $t1, -1     # decrement loop counter
19        bgtz   $t1, loop        # repeat if not finished yet.
20        la    $a0, fibs        # first argument for print (array)
21        add    $a1, $zero, $t5  # second argument for print (size)
22        jal    print           # call print routine.
23        li    $v0, 10          # system call for exit
24        syscall                # we are out of here.
25

```

Line: 1 Column: 1 ☒ Show Line Numbers

Mars Messages

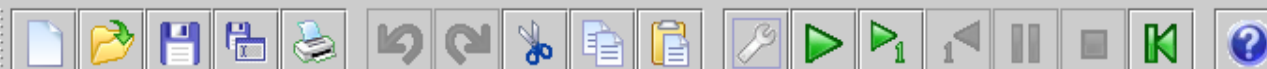
Run I/O

Clear

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
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

File Edit Run Settings Tools Help



Run speed at max (no interaction)



Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x3c011001	lui \$1,4097	6: la \$t0, fibs # load address of array
<input type="checkbox"/>	0x00400004	0x34280000	ori \$8,\$1,0	
<input type="checkbox"/>	0x00400008	0x3c011001	lui \$1,4097	7: la \$t5, size # load address of siz...
<input type="checkbox"/>	0x0040000c	0x342d0030	ori \$13,\$1,48	
<input type="checkbox"/>	0x00400010	0x8dad0000	lw \$13,0(\$13)	8: lw \$t5, 0(\$t5) # load array size
<input type="checkbox"/>	0x00400014	0x3c010000	lui \$1,0	9: li \$t2, 1 # 1 is first and seco...
<input type="checkbox"/>	0x00400018	0x342a0001	ori \$10,\$1,1	
<input type="checkbox"/>	0x0040001c	0xad0a0000	sw \$10,0(\$8)	10: sw \$t2, 0(\$t0) # F[0] = 1
<input type="checkbox"/>	0x00400020	0xad0a0004	sw \$10,4(\$8)	11: sw \$t2, 4(\$t0) # F[1] = F[0] = 1
<input type="checkbox"/>	0x00400024	0x21a9fffe	addi \$9,\$13,-2	12: addi \$t1, \$t5, -2 # Counter for loop, ...
<input type="checkbox"/>	0x00400028	0x840b0000	lw \$11,0(\$8)	13: loop: lw \$t3, 0(\$t0) # Get value from arr...

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x0000000c	0x68540020	0x69462065	0x616e6f62
0x10010040	0x20696363	0x626d756e	0x20737265	0x3a657261	0x0000000a	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

☒ Hexadecimal Addresses
 ☒ Hexadecimal Values

Mars Messages

Run I/O

Assemble: assembling Fibonacci.asm

Clear

Assemble: operation completed successfully.

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
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Text Segment

 Data Segment

data

Simulate and illustrate data cache performance

Cache Organization

8

4

128


Cache Performance

■ = miss

Close

Run I/O

Assembly: operation completed successfully.



Text Segment

 Data Segment

data

Run I/O

Clear

 Data Cache Simulator

Simulate and illustrate data cache performance

Cache Organization

Placement Policy

Direct Mapping ▼

Number of blocks

8

Block Replacement Policy

N/A

Cache block size (words)

4

Cachable addresses

all of data segment

Cache size (bytes)

128

Cache Performance

Memory Access Count

12

Cache Block Table

Cache Hit Count

9

(block 0 at top)


☐ = empty

Cache Miss Count

3

Cache Hit Rate

75%

 = miss

Disconnect from MIPS program

Reset Counts and Cache

Close

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	0x1001000c
\$t1	9	0x00000007
\$t2	10	0x00000005
\$t3	11	0x00000002
\$t4	12	0x00000003
\$t5	13	0x0000000c
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400040
hi		0x00000000
lo		0x00000000