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
# Compute first twelve Fibonacci numbers and put in array, then print
.data
fibs:.word 0 : 12  # "array" of 12 words to contain fib values
size:.word 12  # size of "array"
.text
la $t0, fibs  # load address of array
la $t5, size  # load address of size variable
lw $t5, 0($t5)  # load array size
li $t2, 1  # 1 is first and second Fib. number
sw $t2, 0($t0)  # F[0] = 1
sw $t2, 4($t0)  # F[1] = F[0] = 1
add $t1, $t5, -2  # Counter for loop, will execute (size-2) times
loop: lw $t3, 0($t0)  # Get value from array F[n]
lw $t4, 4($t0)  # Get value from array F[n+1]
add $t2, $t3, $t4  # $t2 = F[n] + F[n+1]
sw $t2, 8($t0)  # Store F[n+2] = F[n] + F[n+1] in array
addi $t0, $t0, 4  # increment address of Fib. number source
addi $t1, $t1, -1  # decrement loop counter
bgtz $t1, loop  # repeat if not finished yet.
la $a0, fibs  # first argument for print (array)
add $a1, $zero, $t5  # second argument for print (size)
jal print  # call print routine.
li $v0, 10  # system call for exit
syscall  # we are out of here.
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: 0
- Cache Hit Count: 0
- Cache Miss Count: 0
- Cache Hit Rate: 0%

Connect to MIPS program  Reset Counts and Cache  Close