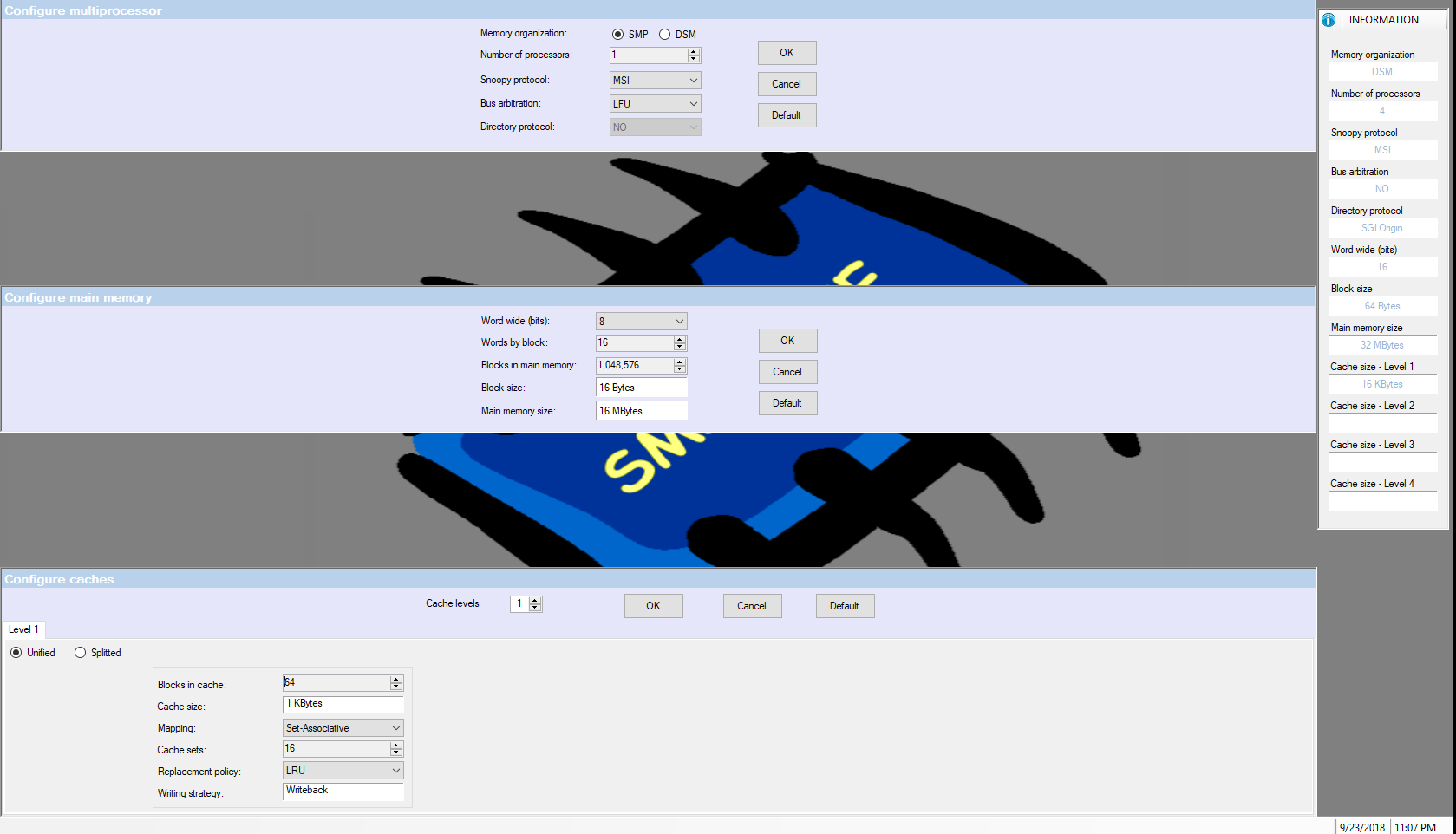
Tyler Nichols

CST-307

Dr. Citro

November 23, 2018



These were my selections for my SMPCache. I then used the burst trace file and the non-burst file and collected the following information

|  |  |
| --- | --- |
| Non-Burst | Burst |
| Readings | Readings |
| 61.795% hits | 98.521% hits |
| 38.205% misses | 1.479% misses |
| Writings | Writings |
| 25.215% hits | 97.421% hits |
| 74.784% misses | 2.579% misses |
| Global | Global |
| 56.843% hits | 98.122% hits |
| 43.157% misses | 1.878% misses |

We can find the average miss penalty by taking the miss penalty and multiplying it to the miss rate. The miss penalty is 20 clock cycles. For the non-burst Readings we have a .38205 x 20 = 7.641 clock cycles. For non-burst Writings .74784 x 20 = 14.9568 clock cycles. For non-burst Global we have .43157 x 20 = 8.6314 clock cycles. For the burst transfer there is an 8 penalty. So, the calculations are as follows; Readings 0.01479 x 8 = 0.11832 clock cycles. Writings 0.02579 x 8 = 0.20632 clock cycles. Global 0.01878 x 8 = 0.15024 clock cycles

QTSPIM Code:

.data

prompt1: .asciiz "\n\n Enter the first integer please:"

prompt2: .asciiz "Enter the second integer please:"

result: .asciiz "The result is:"

.text

main:

#t0-to hold first integer

#t1-to hold second integer

#t2- used to hold the sum of t$1 and t$2

#first number

li $v0, 4 #syscall to print string

la $a0, prompt1 #address of string to print

syscall

#

li $v0, 5 #syscall to read an integer

syscall

move $t0, $v0 #move the number to read into $t0

#second number

li $v0, 4

la $a0, prompt2

syscall

#

li $v0,5

syscall

move $t1,$v0

#

#print out sum of $t2

li $v0, 4

la $a0, result

syscall

#

add $a0, $t1, $t0 #compute the sum

li $v0, 1

syscall

#

li $v0, 10

syscall