3350 Computer Architecture Assignment 2

Justin Grant

250787131

PROBLEM 1

1.1 List the final state of the cache, with each valid entry represented as a record of:<index, tag, data block(s)>.

Decimal Address	Tag	Index	Offset	Hit or Miss
29	…0000 0000 00	00 0001	1101	Miss
45	…0000 0000 00	00 0010	1101	Miss
9	…0000 0000 00	00 0000	1001	Miss
25	…0000 0000 00	00 0001	1001	Hit
6	…0000 0000 00	00 0000	0110	Hit
33	…0000 0000 00	00 0010	0001	Hit
22	…0000 0000 00	00 0001	0110	Hit
51	…0000 0000 00	00 0011	0011	Miss
41	…0000 0000 00	00 0010	1001	Hit
54	…0000 0000 00	00 0011	0110	Hit
5	…0000 0000 00	00 0000	0101	Hit
23	…0000 0000 00	00 0001	0111	Hit
30	…0000 0000 00	00 0001	1110	Hit
47	…0000 0000 00	00 0010	1111	Hit
21	…0000 0000 00	00 0001	0101	Hit

1.2 What is the cache hit ratio?

The Cache Hit Ratio = 11/ 15 = 0.7333

PROBLEM 2

2.1 If the memory address in $s0 stores an integer 5, what is the value in $s1 after the execution of the above MIPS code? What does the code compute?

The values of $s1 after execution will be three. The code computes the number of digit places the number stored by the address in $s0 contains in binary, not counting any leading zeros. The number 5 is 101 in binary, which is three digits. If it was instead 8, or 1000, then $s1 would contain 4.

2.2

Add $s1, $0, $0 - 80004	0	0	0	17	0	32	Register addressing
Lw $t2, 0($s0) - 80008	35	16	10	0	-	-	Base addressing
Loop: Beq $t2, $0, Exit - 80012	4	10	0	3	-	-	PC-Relative addressing
Srl $t2, $t2, 1 - 80016	0	0	10	10	1	2	Register Addressing
Addi $t2, $t2, 1 - 80020	8	10	10	1	-	-	Immediate Addressing
J Loop - 80024	2	20001	-	-	-	-	Pseudodirect addressing
Exit: 80028	…	-	-	-	-	-	-

PROBLEM 3

4.1 Implement the above function in MIPS assembly. Try to use a minimum number of MIPS instructions as possible.

4.2 Write a complete MIPS code and run it in QTSPIM to compute A (1, 3). Show the final value of registers you used.

<<<SEE Assignment2Part1.s>>>

PROBLEM 4

3.1 Implement the above function Merge in MIPS assembly. Try to use a minimum number of MIPS instructions as possible.

3.2 Implement the above function MergeSort in MIPS assembly. Try to use a minimum number of MIPS instructions as possible.

3.3 Write a complete MIPS code and run it in QtSpim to sort the following array of 8 elements: [29, 45, 9, 25, 6, 33, 22, 51] Show the final values of registers you used.

<<<SEE Assingment2Part2.s>>>