Bryan Arnold

CSE 3666

3/9/17

Architecture Assignment 5

a) To check a stuck-at-0 fault, set the value of $at to 1. Then, add $at and $zero into register $t0. If the result is 0, then there is a stuck-at-0 fault here. Another test for stuck-at-0 for the instruction memory, which involves the PC adder to find additional instructions later, is not needed. All the addresses need to be multiples of 4, PC + 4, so the bit is always stuck at 0 anyway so it isn't a problem. The last test for stuck-at-0 is in the ALU if a result is stuck-at-0, would be using addition that satisfies the ALU zero flag. For example, load $at, 15 and load $t0, 1. When adding these two, if the result in the branch multiplexor links back to the PC, a stuck-at-0 fault has occurred.
b) To check a stuck-at-1 fault, set register $at to 1 again. Then, add $zero and $zero into $t0. If the result is not 0, then there is a stuck-at-1 fault here. Another stuck-at-1 fault test is that if the instruction memory, the PC adder component, is stuck at 1, the next instruction will be PC + 4 + 2, making the instructions not in order and will crash. The last test for stuck-at-0 is in the ALU if a result is stuck-at-0, would be using addition that satisfies the ALU zero flag. For example, load $at, 4 and load $t0, 4. When adding these two, if the result in the branch multiplexor links it back to the register memory, a stuck-at-1 fault has occurred.
c) The wire is still usable but will only be able to use certain registers, not all, and must use more memory to compensate for lack of registers and offset each instruction by ½ in case of PC adder stuck at 1 faults.