CSL 2080

Lab Exam 1

Time: 1 hour **MM: 10x2 = 20**

Note: Submit your code and circuit file as <roll_number_array>.s and <roll_number_adder>.circ

1. Write a RISC-V assembly program to reverse an array of integers in place.
   Hint:
   a. Load the base address of the array into a register.
   b. Initialise two indices, one pointing to the beginning of the array (start index) and the other pointing to the end of the array (end index).
   c. Swap the elements at the start and end indices iteratively until the start index surpasses the end index.
   d. Increment the start index and decrement the end index after each swap operation.
   e. Repeat steps 3 and 4 until the start index surpasses the end index, indicating that the entire array has been reversed.

2. Design a 2-bit full adder circuit using only primitive logic gates (AND, OR, NOT) along with XOR gates.

   Recall our discussions on half and full adders, where half adders are used for adding only 2 bits(A0 and B0) and full adders are required to add subsequent bits due to possible carry-in. Utilising this knowledge, your circuit should take two 2-bit inputs, A (A1A0) and B (B1B0), and produce outputs S0S1 representing the sum of the two inputs, along with a carry-out (Cout) indicating any carry generated during addition. Your design should adhere to the principles of digital logic design and implement the necessary logic gates efficiently.