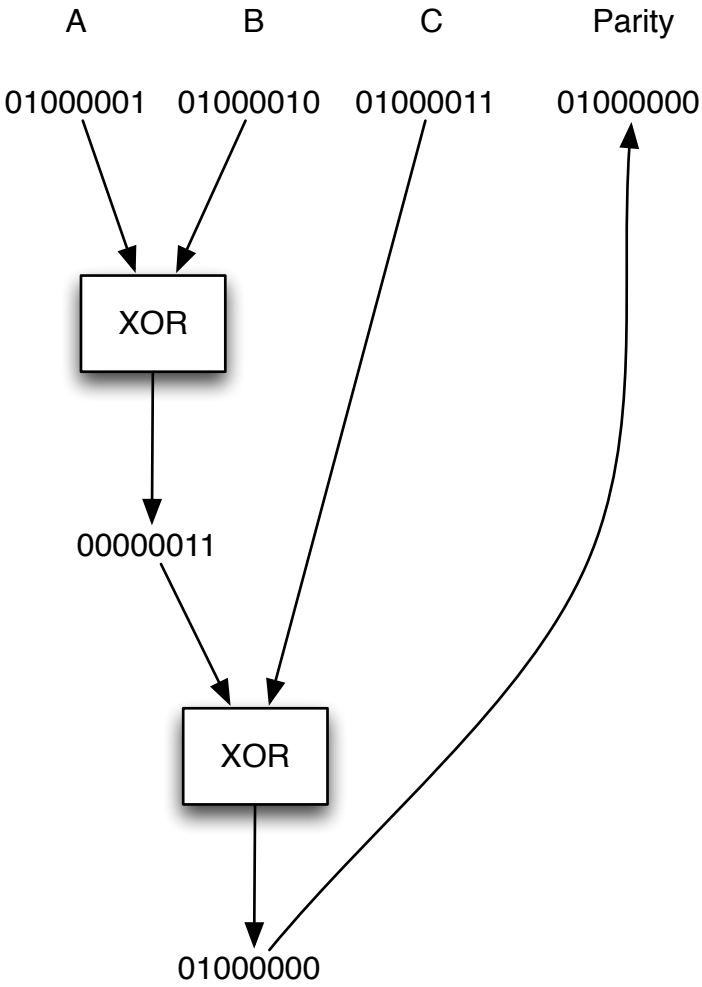
CSG1102 – Workshop 7
RAID 5

Introduction

A very basic parity implementation has been performed for redundancy purposes on some ASCII encoded text. A two stage XOR process has been used in order to simulate RAID 5 redundancy. This process is outlined below



In this example we have three inputs the characters ‘A’, ‘B’ and ‘C’. We have first gotten the result of ‘A’ XOR ‘B’ and then used the output as an input for XOR 'C’. Through this we have a parity value to store for redundancy purposes. If any one of the values ‘A’, ‘B’, or ‘C’ are lost we are able to reconstruct the value through a series of XOR operations

Your Task

Your task is to reconstruct a short message, which has been subject to data loss. In this example an X indicates corrupted data where a 0 or 1 should be. In reality however corrupted data is normally determined through the use of a checksum or other mechanism.

BYTE 1	BYTE 2	BYTE 3	PARITY BYTE
01000111	01101111	011011XX	01000111
01X00100	00100000	01101110	00101010
01100101	01110111	011XXX11	01100001
00XXX000	01100101	01110110	00110011
01100101	01110010	01111001	01101110
01100010	011X1111	01100100	01101001
01111001	0X100000	0010XX00	01111001

Discuss your results with those around you or via the blackboard forums.