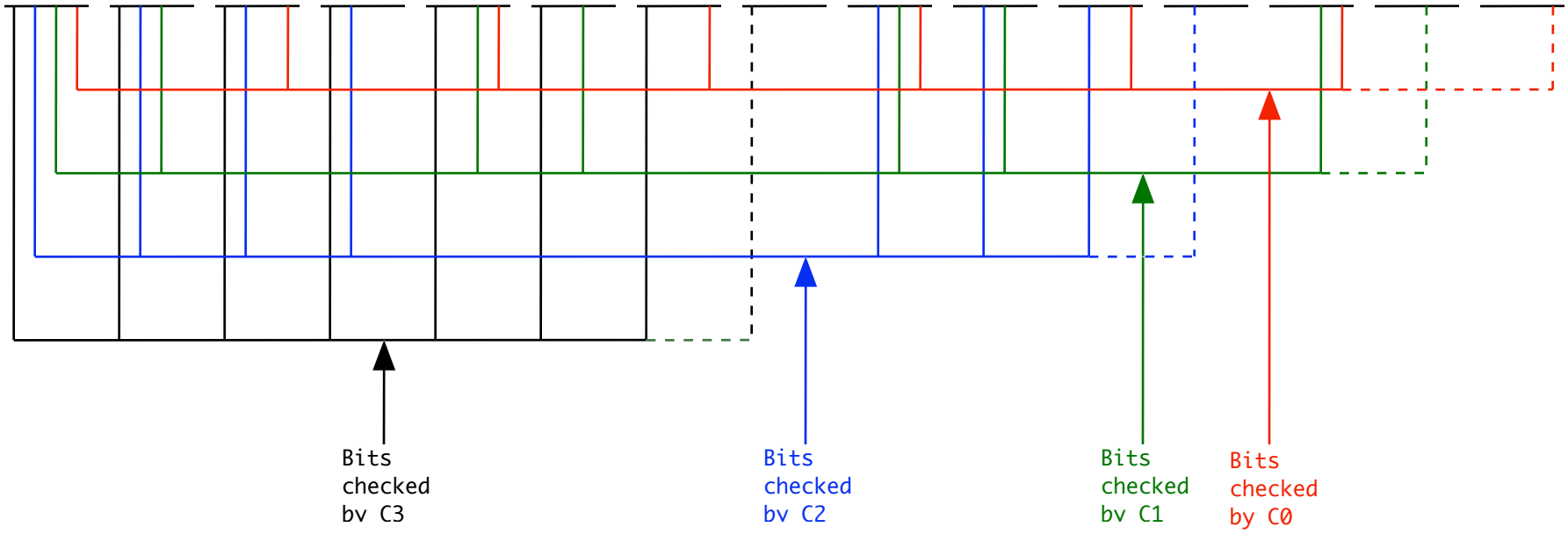
KEY

Bit Position (decimal) Bit Position (binary) Contents: Check bit (C) or Data bit (D)

15 1111_2 D10	14 1110_2 D9	13 1101_2 D8	12 1100_2 D7	11 1011_2 D6	10 1010_2 D5	9 1001_2 D4	8 1000_2 C3	7 0111_2 D3	6 0110_2 D2	5 0101_2 D1	4 0100_2 C2	3 0011_2 D0	2 0010_2 C1	1 0001_2 C0



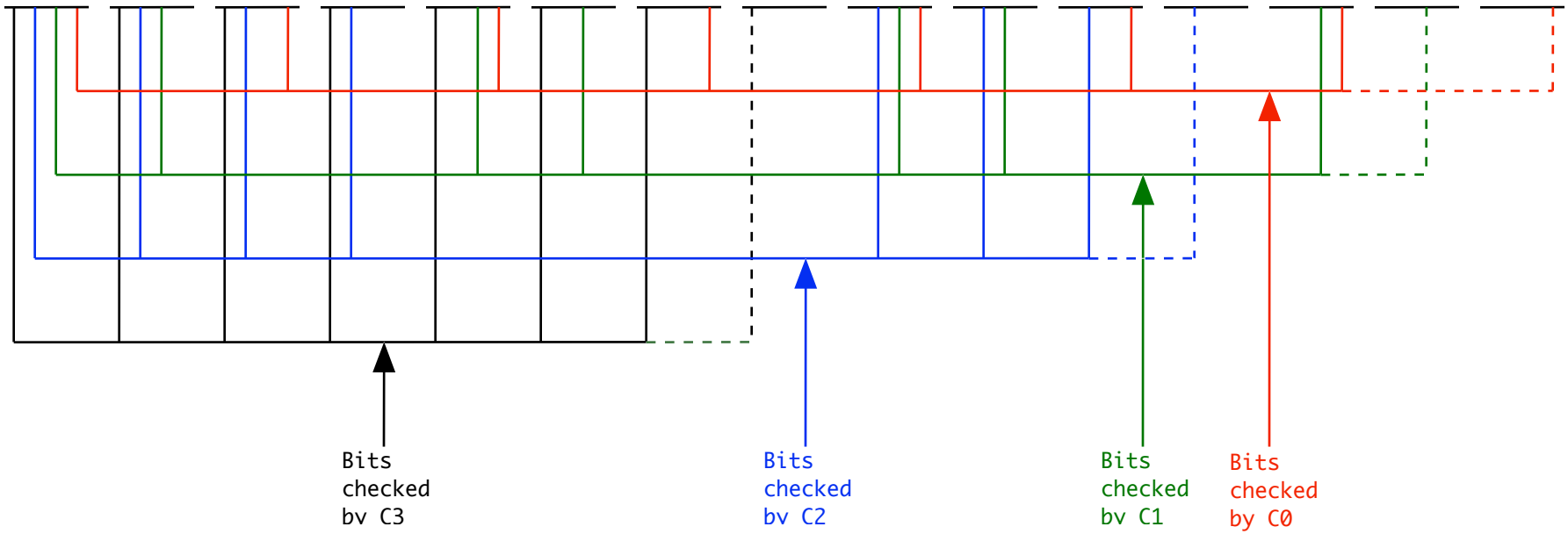
Suppose we wish to store or transmit
the data 10101010101

KEY

Bit Position (decimal) Bit Position (binary) Contents: Check bit (C) or Data bit (D)

15 1111_2 D10	14 1110_2 D9	13 1101_2 D8	12 1100_2 D7	11 1011_2 D6	10 1010_2 D5	9 1001_2 D4	8 1000_2 C3	7 0111_2 D3	6 0110_2 D2	5 0101_2 D1	4 0100_2 C2	3 0011_2 D0	2 0010_2 C1	1 0001_2 C0

1 0 1 0 1 0 1 0 1 0 1



Calculate values for the checking bits:

Choose C0 to make 1 1 1 1 0 0 1 ? have odd parity, so C0 = 0
Choose C1 to make 1 0 1 0 0 1 1 ? have odd parity, so C1 = 1
Choose C2 to make 1 0 1 0 0 1 0 ? have odd parity, so C2 = 0
Choose C3 to make 1 0 1 0 1 0 1 ? have odd parity, so C3 = 1

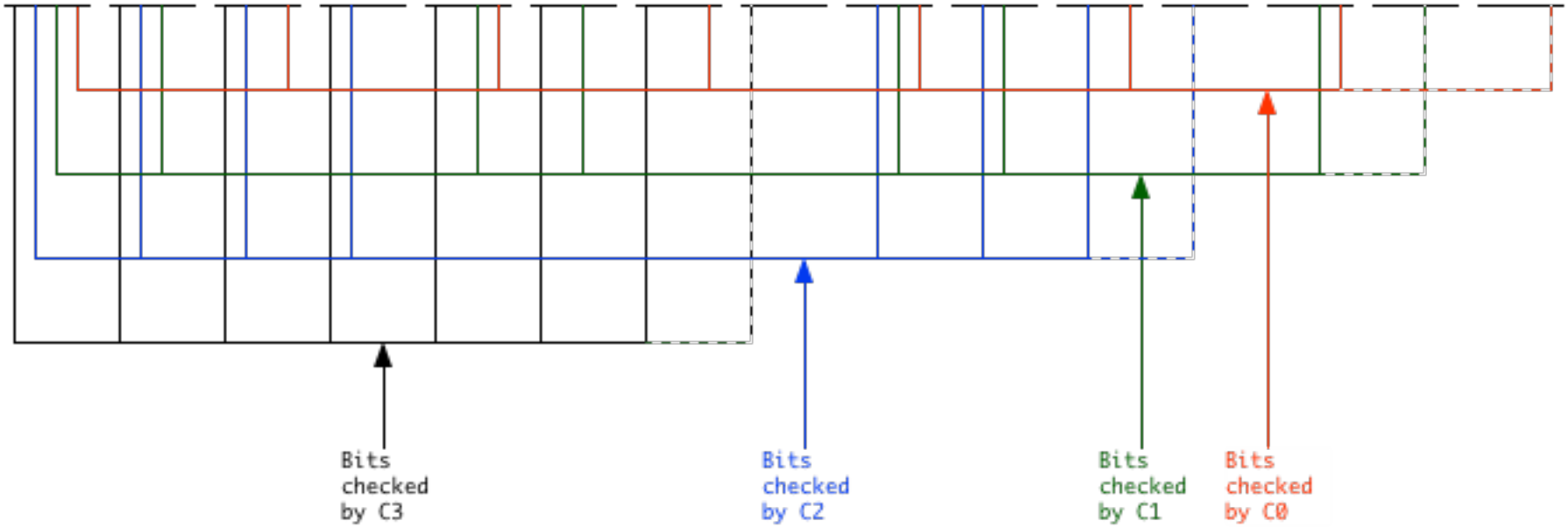
Suppose the data bit 4 (which is at position 9 in the stored or transmitted message) is corrupted, so that the value read or received is 1010100101001l0

As a result of the corruption, we think that the original data was 10101000101, which is, of course, not correct

KEY

Bit Position (decimal) Bit Position (binary) Contents: Check bit (C) or Data bit (D)

15 1111_2 D10	14 1110_2 D9	13 1101_2 D8	12 1100_2 D7	11 1011_2 D6	10 1010_2 D5	9 1001_2 D4	8 1000_2 C3	7 0111_2 D3	6 0110_2 D2	5 0101_2 D1	4 0100_2 C2	3 0011_2 D0	2 0010_2 C1	1 0001_2 C0



Calculate expected values for the checking bits:

We expect C0 to make 1 1 1 0 0 0 1 ? have odd parity, so C0 should be 1
We expect C1 to make 1 0 1 0 0 1 1 ? have odd parity, so C1 should be 1
We expect C2 to make 1 0 1 0 0 1 0 ? have odd parity, so C2 should be 0
We expect C3 to make 1 0 1 0 1 0 0 ? have odd parity, so C3 should be 0

Checking bits received = 1010
Checking bits expected = 0011

Doing a bitwise exclusive-or yields 1001,
which tells us that the bit in position 9 is
corrupt.

Since the received value in this position
is 0, the correct value must be 1

Hence, the original data was really
101010101