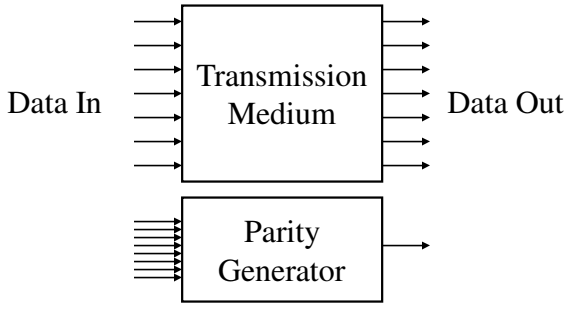
Error Detection with Parity

Error Detection with Parity



Error Correcting Code – by Design

Start with Bit Position = Column Number

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Error Correcting Code – by Design

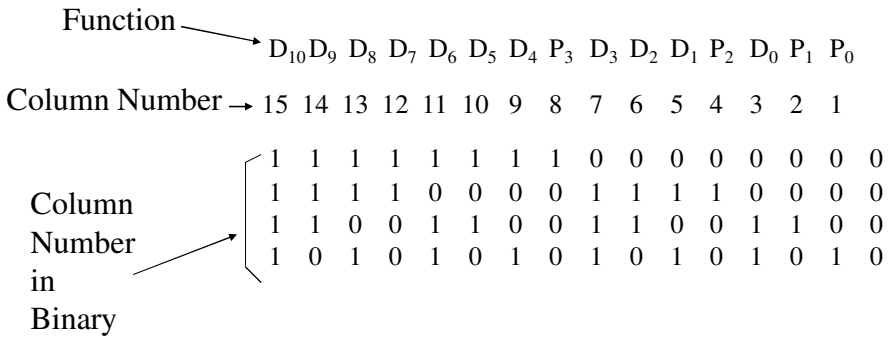
Start with Bit Position = Column Number

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

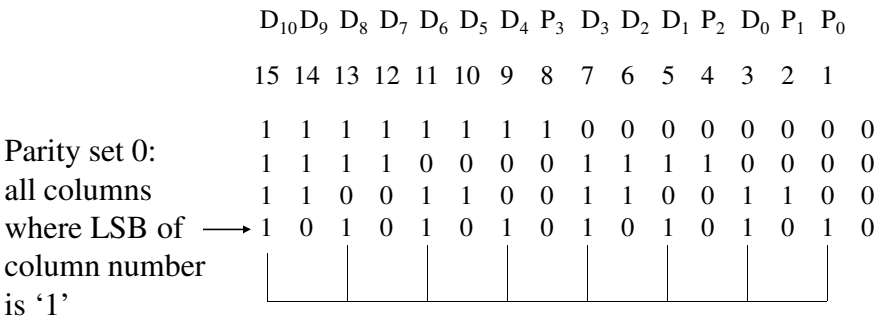
Error Correcting Code – by Design

Start with Bit Position = Column Number

D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	P_3	D_3	D_2	D_1	P_2	D_0	P_1	P_0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

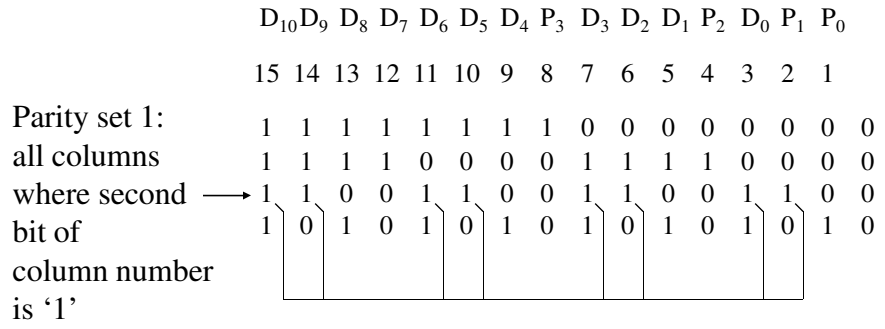


Four Bit Column Identifier? Then build four parity sets



Columns 15, 13, 11, 9, 7, 5, 3, 1

Four Bit Column Identifier? Then build four parity sets



Columns 15, 14, 11, 10, 7, 6, 3, 2

Four Bit Column Identifier? Then build four parity sets

	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	P_3	D_3	D_2	D_1	P_2	D_0	P_1	P_0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
→	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Parity set 2: all columns where third bit of column number is '1'

Columns 15, 14, 13, 12, 7, 6, 5, 4

Four Bit Column Identifier? Then build four parity sets

	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	P_3	D_3	D_2	D_1	P_2	D_0	P_1	P_0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
→	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Parity set 3: all columns where fourth bit of column number is '1'

Columns 15, 14, 13, 12, 11, 10, 9, 8

Finally – Add Parity Across Entire Data Set

Parity set PT: all columns with data or parity bits →

D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	P_3	D_3	D_2	D_1	P_2	D_0	P_1	P_0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

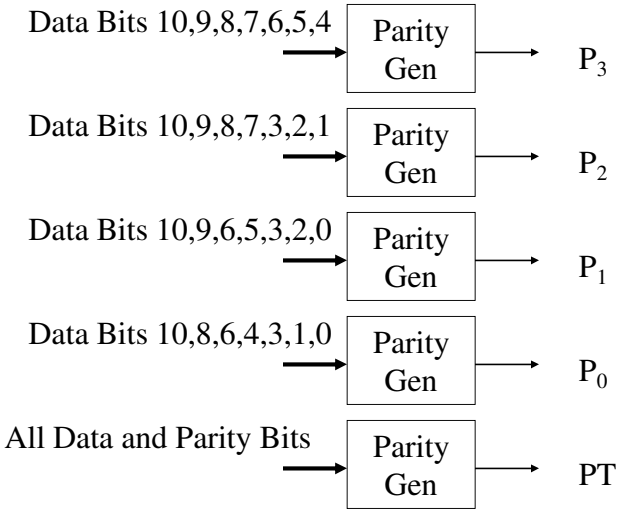
Columns 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

SECDED Coding Scheme

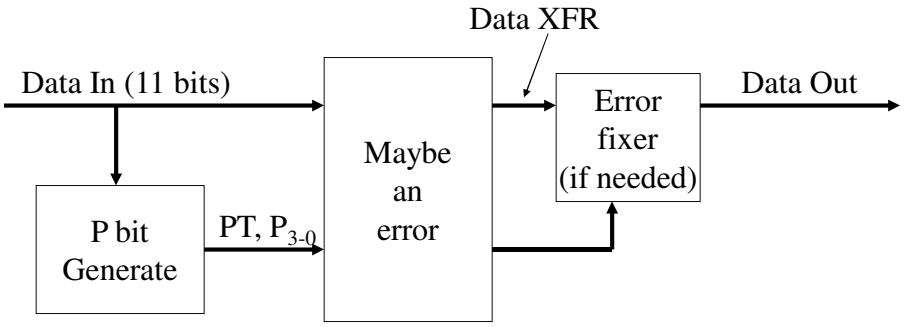
PT – parity across whole word
Pk – parity bits on set basis (any Pk nonzero = error)

PT	PK	Situation
Good	Good	No error – accept data as passed
Good	Bad	Two bits in error
Bad	Good	PT in error
Bad	Bad	Single bit error – corrected (Error in column $P_3P_2P_1P_0$)

To Create Parity Bits for Scheme



Test System



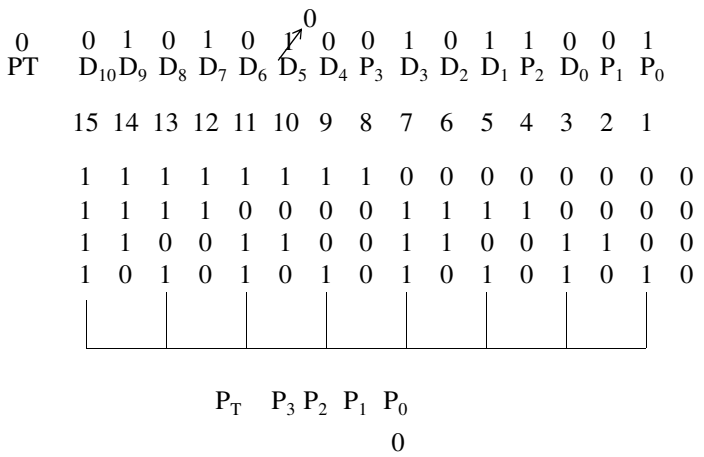
Start point: set up correct values in parity bits, data bits

0	0	1	0	1	0	1	0	0	1	0	1	1	0	0	1
PT	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	P_3	D_3	D_2	D_1	P_2	D_0	P_1	P_0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

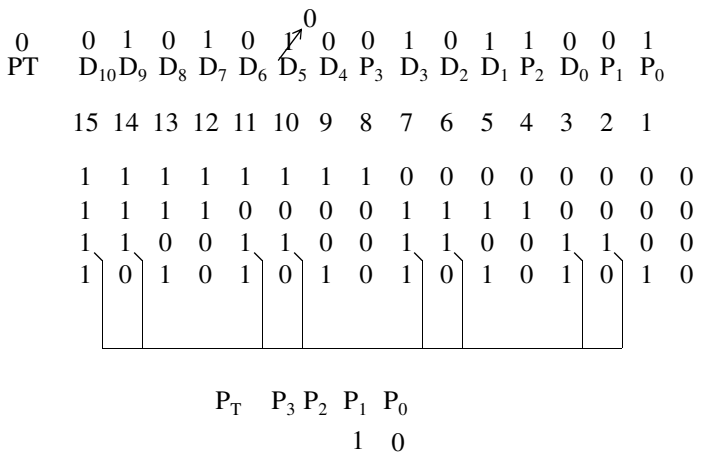
For demonstration: make one bit wrong

0	0	1	0	1	0	1 (0)	0	0	1	0	1	1	0	0	1
PT	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	P_3	D_3	D_2	D_1	P_2	D_0	P_1	P_0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Check Parity zero for correctness



Check Parity one for correctness



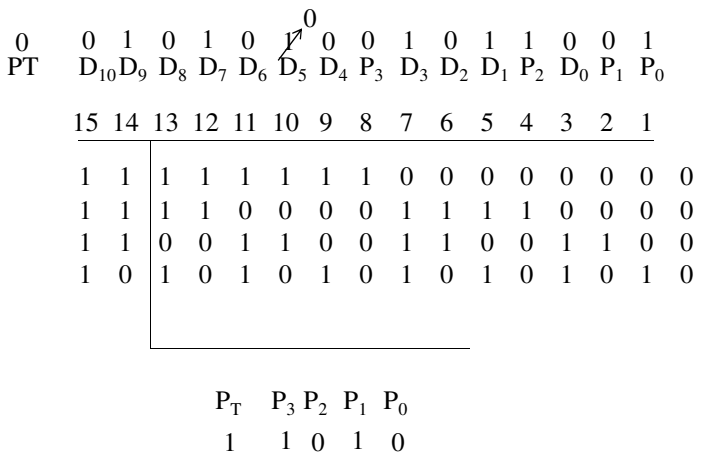
Check Parity two for correctness

0	0	1	0	1	0	1	0	0	1	0	1	1	0	0	1
PT	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	P_3	D_3	D_2	D_1	P_2	D_0	P_1	P_0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

P_T	P_3	P_2	P_1	P_0
	0	1	0	

Check Parity three for correctness

0	0	1	0	1	0	1	0	0	1	0	1	1	0	0	1
PT	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	P_3	D_3	D_2	D_1	P_2	D_0	P_1	P_0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

P_T	P_3	P_2	P_1	P_0
	1	0	1	0

Check Parity - Total for correctness