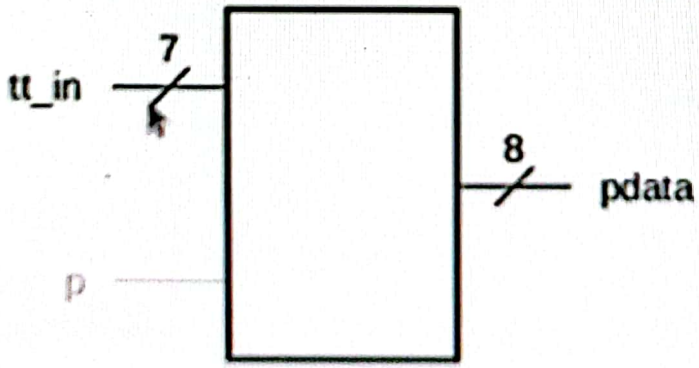
Design examples

Example 8

We need to design an interface module for an old teletype system in order to connect it to a standard serial computer port. The teletype generates 7-bit ASCII character codes while our computer can only receive 8-bit words (bytes) that must have even parity.

a) Design a module that reads 7-bit words through a 'tt_in' input signal and generates 8-bit words with leading even parity bit in output signal 'pdata' (bit 7 in pdata is the parity bit and the rest of the bits are copied from tt_in)

b) Add a control signal 'p' to the system to select the parity of the generated output so that the output will be even when p=0 and odd when p=1.



→ Parity is used to check ~~data~~ error in data

→ XOR gives 1 when input has ~~more ones else 0~~ odd no of ones else 0

Even Parity

→ We add or make no of 1's even

e.g

A	B	C	Parity bit
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

1's even we make by adding Parity bit.

→ We have even 1's

Odd Parity

A	B	C	Parity Generator
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Opposite to even parity.

- Just invert XOR output.

We also check or side - parity detect on receiver

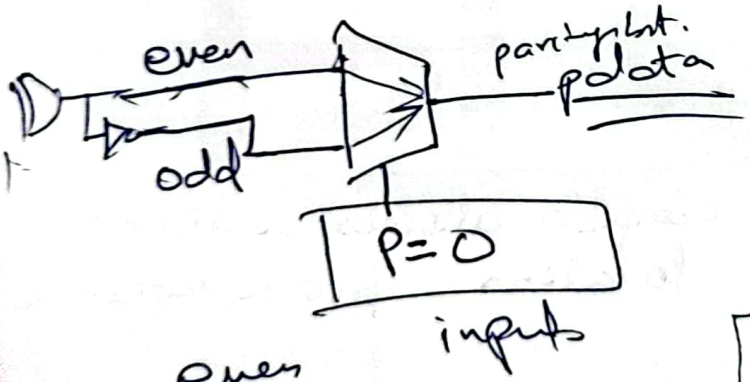
- It is always given in paper whether we are using even or odd parity. So according to that we generate parity bit and then we send our data including parity bit.

→ ODD parity keeps no of ONES odd. If you generated output for even parity then just add Not after XOR to make it odd parity.

For parity check

→ We just has to add XOR all data bits on Receiver end.

→ For even parity check XOR all bits including parity bit if result is 1 then it means data recieved is not correct. If it is 0 then we can say that data recieved is correct.



even
if P = 0

odd
if P = 1.