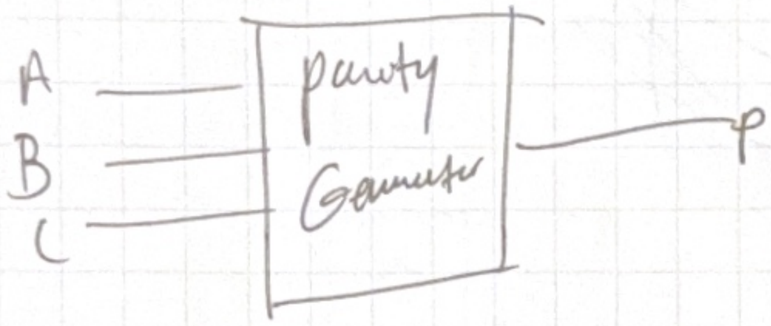
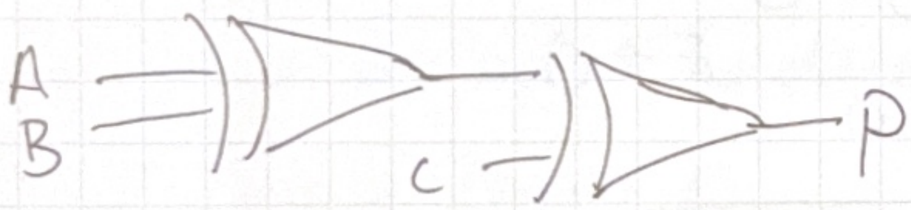
Odd Parity Generator

A	B	C	P
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

3-bit





0 = even

1 = odd.

even parity Generator

LED	A	B	C	P	odd parity
~~LED~~ ON	0	0	0	0	1
OFF	0	0	1	1	0
OFF	0	1	0	1	0
ON	0	1	1	0	1
OFF	1	0	0	1	0
ON	1	0	1	0	1
ON	1	1	0	0	1
OFF	1	1	1	1	0

even

odd

odd

odd

odd

Odd parity Generator

IS $\Rightarrow$ $P = (A \oplus B \oplus C)'$

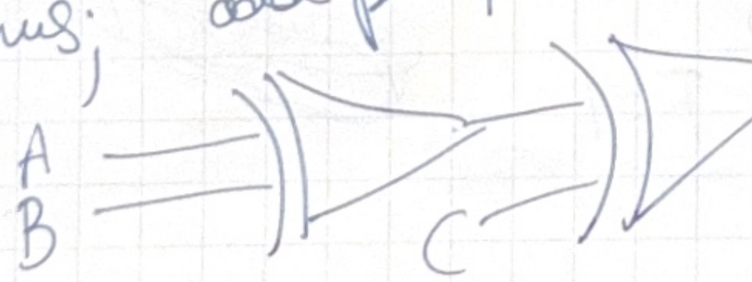
$P = A'B'C + A'BC' + AB'C' + ABC$

This Truth Table "P" is flipped.

$= A'(B'C + BC') + A(B'C' + BC)$

$= A'(B \oplus C) + A(B \oplus C)'$

Thus; odd parity Generator



$P = A \oplus B \oplus C$

~~A=Blue~~

~~B=~~

A=white
~~B=Blue~~
~~C=Green~~
~~D=Yellow.~~

This entire Circuit is simply a 1-bit parity bit.

8-4-2-1
0 0 0 P

A=Blue
B=Green
C=yellow
D=~~Blue~~ Blue

Raw XOR Gate for 3-input.

A	B	C	F		
0	0	0	0	off	even
0	0	1	1	ON	odd
0	1	0	1	ON	odd
0	1	1	0	OFF	even
1	0	0	1	ON	odd
1	0	1	0	OFF	even
1	1	0	0	OFF	even
1	1	1	1	ON.	odd

(LED lights up any time the parity is odd.

↓

to turn the raw XOR itself into an even parity Generator, I will have to ~~add~~ add an ~~in~~ inverter at output.)

↓

To now append the parity bit as (P) and add it as a 4-bit binary String: [a,b,c,P]

↓

This will now total # of 1's across all 4-bits to be always even.

for n-inputs, for data stream Parity Generator Requires n-inputs through XOR so as to "Register" the inputs to be accounted for.

A⊕B = 1 if odd # of 1's among A,B.

A⊕B⊕C = 1 iff odd # of 1's among A,B,C

XOR counts parity, but only for the inputs you include.