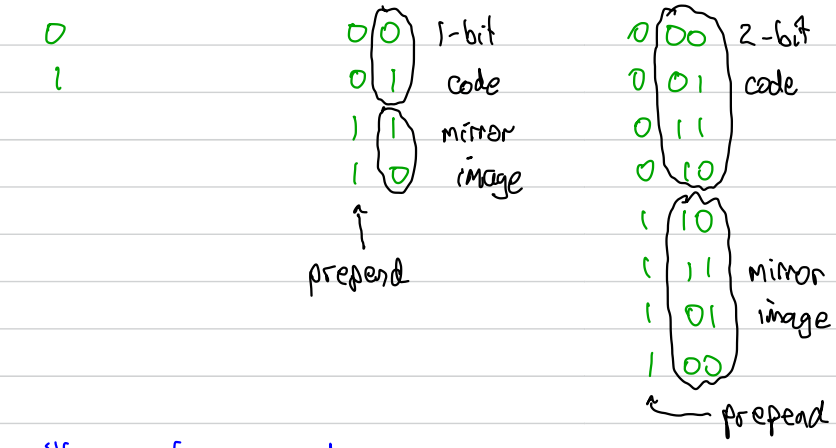
Primary property: When counting, only one bit changes at a time. In binary code, as many as 3 bits can change at a time (e.g., 011 → 100)

A couple of ways to generate Gray code:

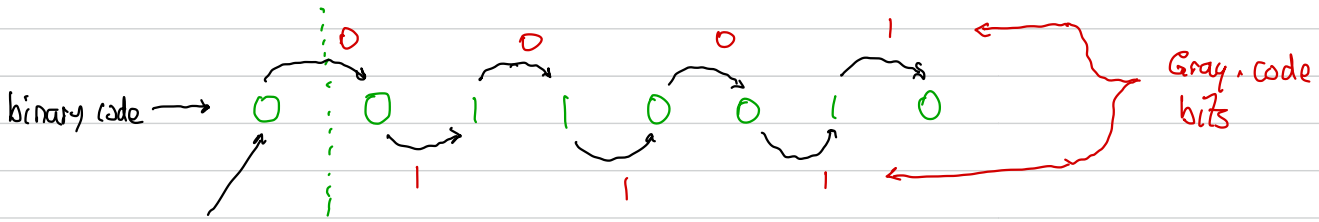
Recursively:



Another way: Start with a binary code

- Scan left-to-right; catalog bit changes (record 1 for change, 0 for no change).

Example: Convert 0110010_2 to Gray code

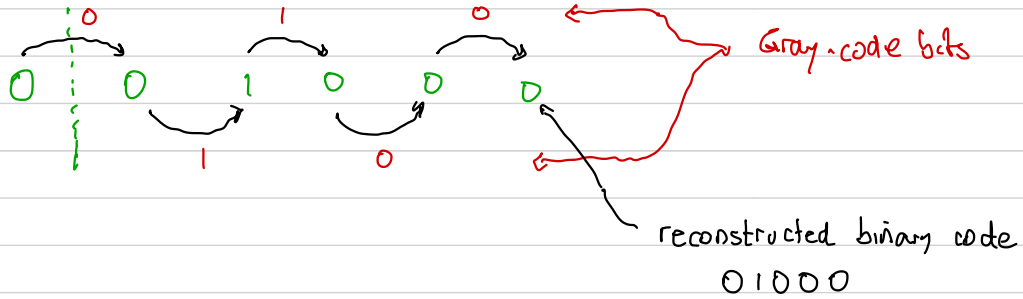


Gray code: 0101011

Going from Gray to binary

- Gray code tells us how to reconstruct a binary code.

Example: convert 01100 to binary



Logic gates

We know how to use binary variables to represent information. We now explore logic circuits. Two important classes:

Combinational circuits: outputs depend only on current values of inputs.

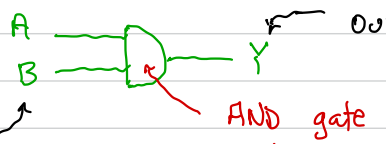
Sequential circuits: same, but also depends on the history of inputs (a later course topic)

Simple combinational circuit:

A key building block in digital circuits is a logic gate.

Logic gate: a combinational circuit with 1 or more inputs, and one output.

for example:



$Y = AB$

Relationship between inputs and outputs are described by a truth table, or Boolean expression

1-input gates

The NOT gate, also called an inverter.

the "bubble" denotes inversion.

$Y = \bar{A}$ "Y equals NOT A"

Notation: The inversion is denoted algebraically with an overbar $Y = \bar{A}$

~ previous ENEL 353 textbooks use $Y = A'$ (apostrophe)

Truth table:

A	Y
0	1
1	0

The buffer gate

Truth table

A	Y
0	0
1	1

Y is simply a copy of A

The buffer behaves as a wire does

- however, it's a useful gate to have to improve reliability of some large circuits.