Unit-III
Chap-II

CODE CONVERTERS

Created by, Ashish Shah, J. M. Patel College, Goregoan W

Code Converters

- Numbers are usually coded in one form or another so as to represent or use it as required.
- For instance, a number 'nine' is coded in decimal using symbol (9)d. Same is coded in natural-binary as (1001)b.
- In telecommunication, the term **code conversion** has the following meanings:

  Conversion of signals, or groups of signals, in one **code** into corresponding signals, or groups of signals, in another **code**.

Code Converters

- Coding is the process of translating the input information which can be understandable by the machine or a particular device.
- Coding can be used for security purpose to protect the information from steeling or interrupting.
- The code converters are used to convert the information in to the code which we want. These are basically encoders and decoders which converts the data in to an encoded form. The below explains some digital codes used in digital electronics.

The below explains some digital codes used in digital electronics.

- **ASCII:**

  ASCII (American Standard for information exchange): It is a character encoding scheme for computer to represent the English alphabet and to communicate with the devices which uses text as controlling commands. ACSII code was developed a long ago to represent the English letters with the numbers which are understandable by digital equipment. If anyone ask to write a letter in ACSII format means writing only alphabets without formatting. Writing a letter in notepad is one of the examples for ASCII format. We cannot do any formatting in notepad like BOLD, TAB etc.

Excess-3 code:

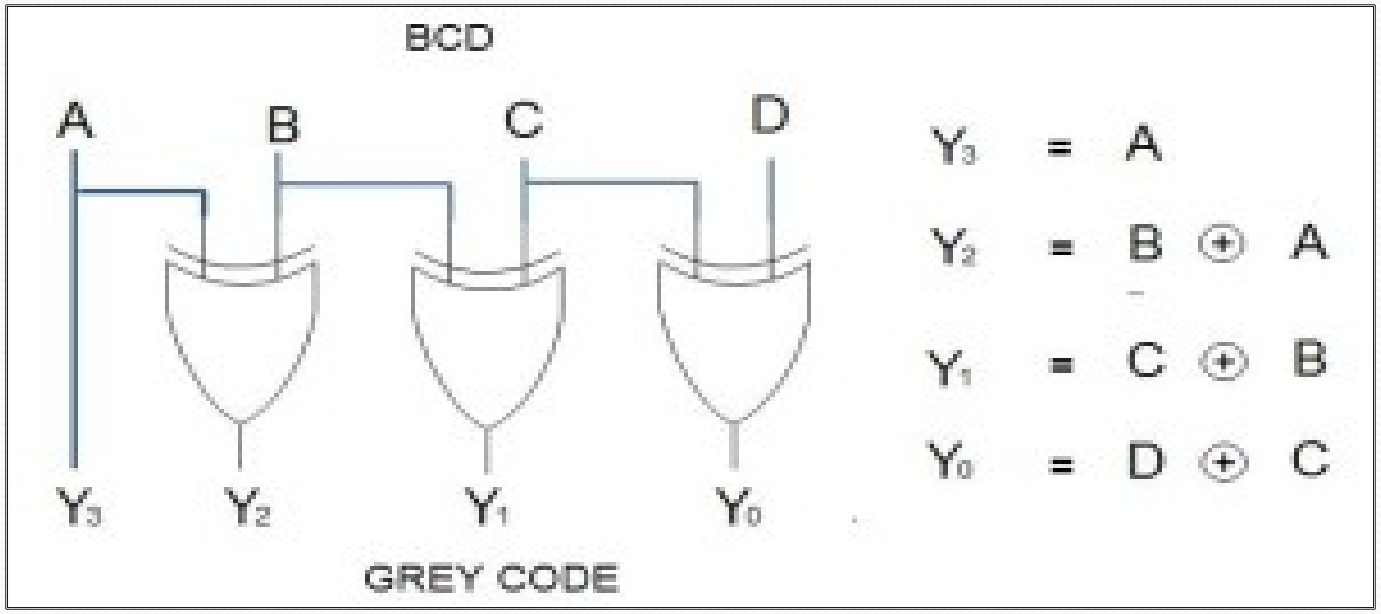
- It is also known as self complementary code as the complement of any number (0-9) will be available within these 10 numbers. As the name implies it is excess of 3 for the regular BCD code i.e. if u add 3(0011) to the BCD Addition u can get Excess-3 code.

Grey code

- It is also called as unit distance code and reflective code word. It is not arithmetic code that there are no specific weights assigned to each bit positions.

Example of code converter

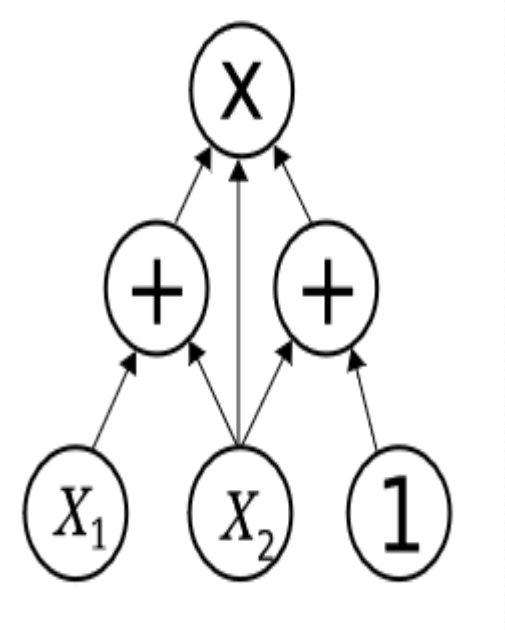
- BCD to Grey Code



Arithmatic Circuit

- In computational complexity theory, *arithmetic circuits* are the standard model for computing polynomials.
- Informally, an arithmetic circuit takes as inputs either variables or numbers, and is allowed to either add or multiply two expressions it already computed. Arithmetic circuits give us a formal way for understanding the complexity of computing polynomials.
- The basic type of question in this line of research is "what is the most efficient way for computing a given polynomial f?"

- for example: the input gates compute (from left to right) x 1 , x 2 and 1 , the sum gates compute x 1 + x 2 and x 2 + 1 , and the product gate computes (x 1 + x 2) * (x 2 + 1) .



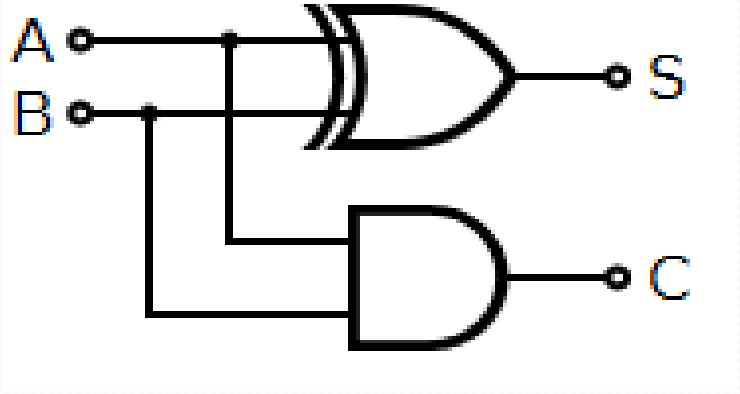
Adder

- An **adder** is a digital circuit that performs addition of numbers. In many computers and other kinds of processors adders are used in the arithmetic logic units.
- They are also utilized in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators, and similar operations.

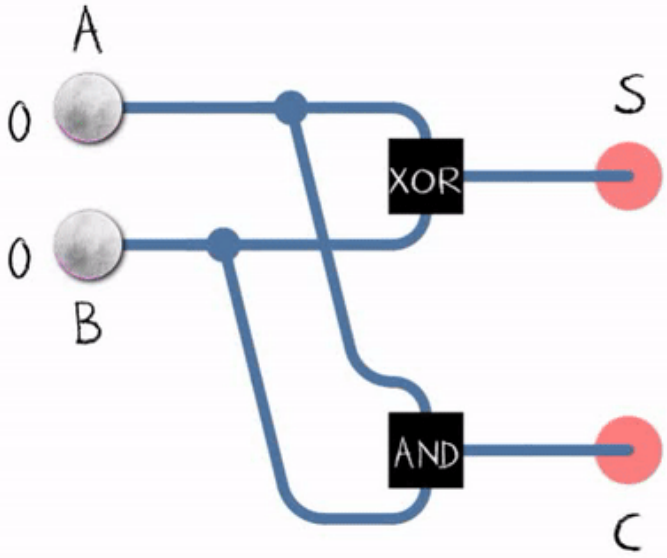
Adder

- Although adders can be constructed for many number representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers.
- In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an adder into an adder–subtractor. Other signed number representations require more logic around the basic adder.

Half Adder



- The **half adder** adds two single binary digits *A* and *B*.
- It has two outputs, sum (*S*) and carry (*C*).



Half Adder

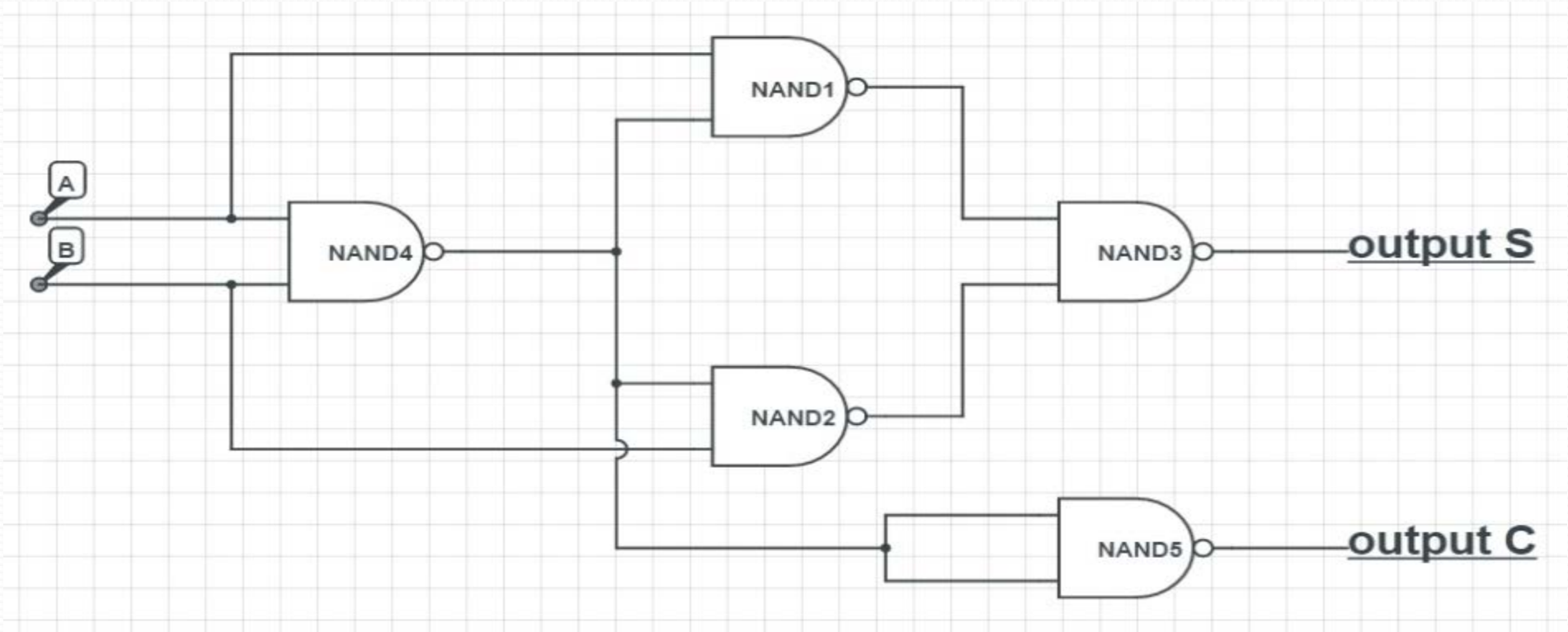
- The simplest half-adder design, pictured on the right, incorporates an XOR gate for *S* and an AND gate for *C*.
- The Boolean logic for the sum (in this case **S**) will be **A'B+AB'** whereas for carry (**C**) will be **AB.**
- With the addition of an OR gate to combine their carry outputs, two half adders can be combined to make a full adder.
- The half adder adds two input bits and generates a carry and sum, which are the two outputs of a half adder.

Half Adder

- The input variables of a half adder are called the augend and addend bits.
- Augend means (arithmetic) A quantity to which another is added. *In "4 + 5", 4 is the* ***augend.*** Addend means a number which is added “5” is addend.
- The output variables are the sum and carry.
- The truth table for the half adder is:

Inputs		Outputs	
A	B	C	S
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

Half adder using NAND gates

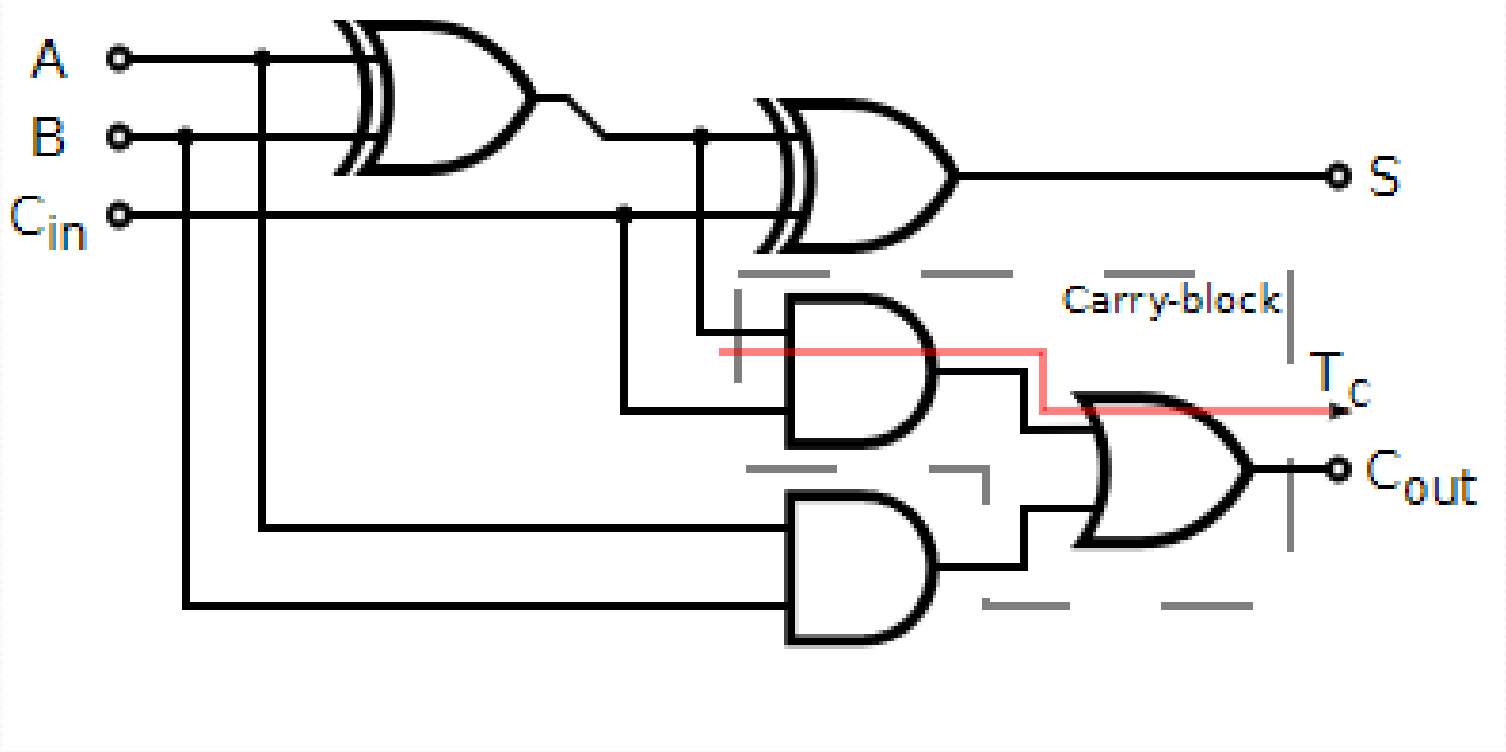


Full adder

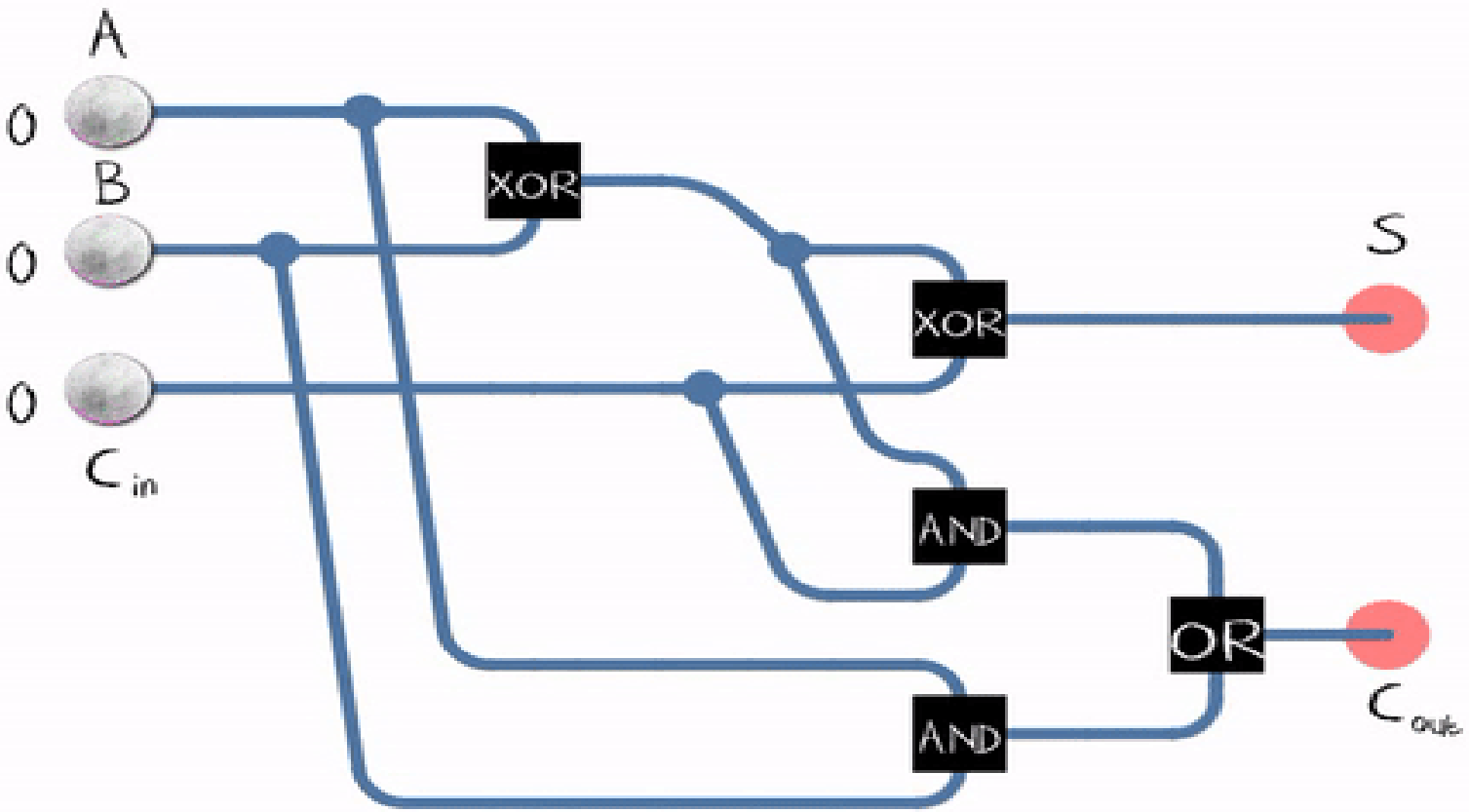
- A **full adder** adds binary numbers and accounts for values carried in as well as out.
- A one-bit full adder adds three one-bit numbers, often written as A, B, and C_{in}; A and B are the operands, and C_{in} is a bit carried in from the previous less-significant stage.
- The full adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. bit binary numbers. The circuit produces a two-bit output.
- Output carry and sum typically represented by the signals Cout and S, where

  sum = 2 × C out + S in decimal system.

Full adder



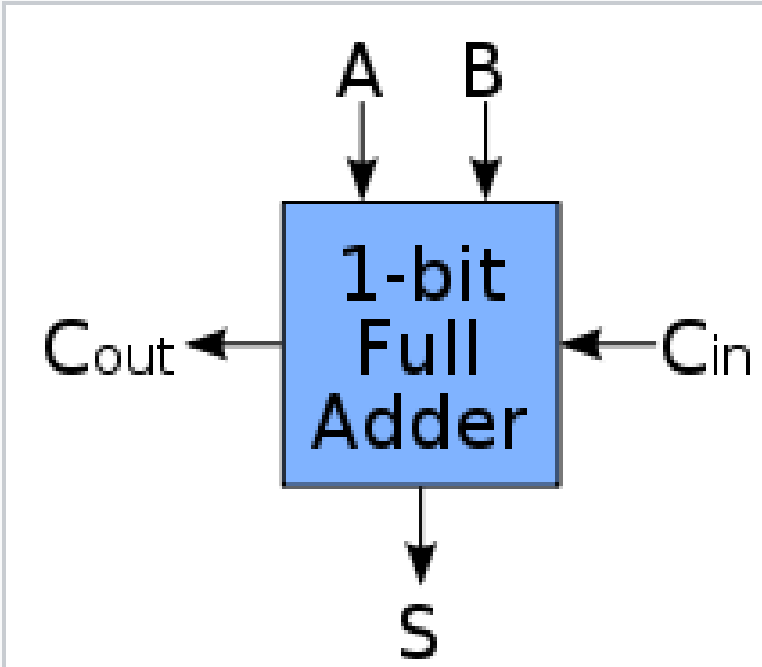
Full adder



Full adder

- The truth table for the full adder is:

Inputs			Outputs	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Schematic symbol for a 1-bit full adder with C_{in} and C_{out} drawn on sides of block to emphasize their use in a multi-bit adder