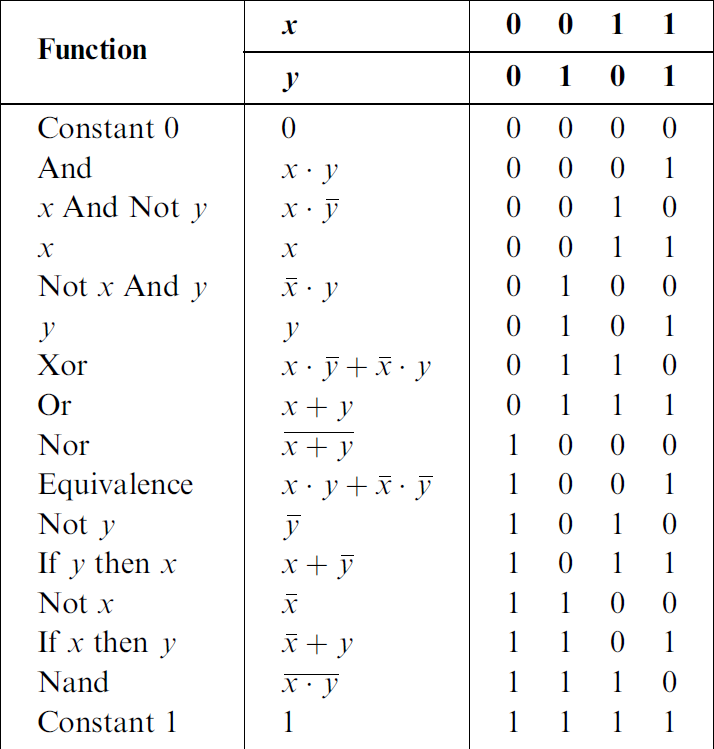
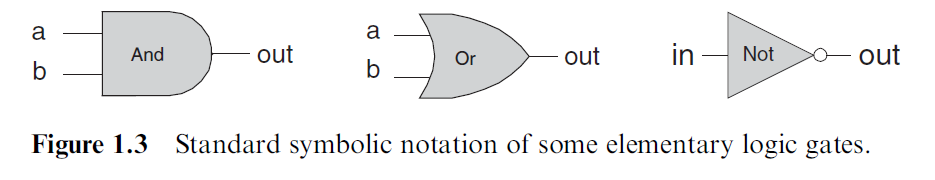
x\*y | x **and** y = 1 when both x and y = 1

x+y | x **or** y = 1 when either x or y or both = 1

x (with line over it) | **not x** = 1 when x = 0





The Nand function (as well as the Nor function) has an interesting theoretical

property: Each one of the operations And, Or, and Not can be constructed from it,

and it alone (e.g., x Or y ¼ ðx Nand xÞ Nand ðy Nand yÞ. And since every Boolean

function can be constructed from And, Or, and Not operations using the canonical

representation method, it follows that every Boolean function can be constructed

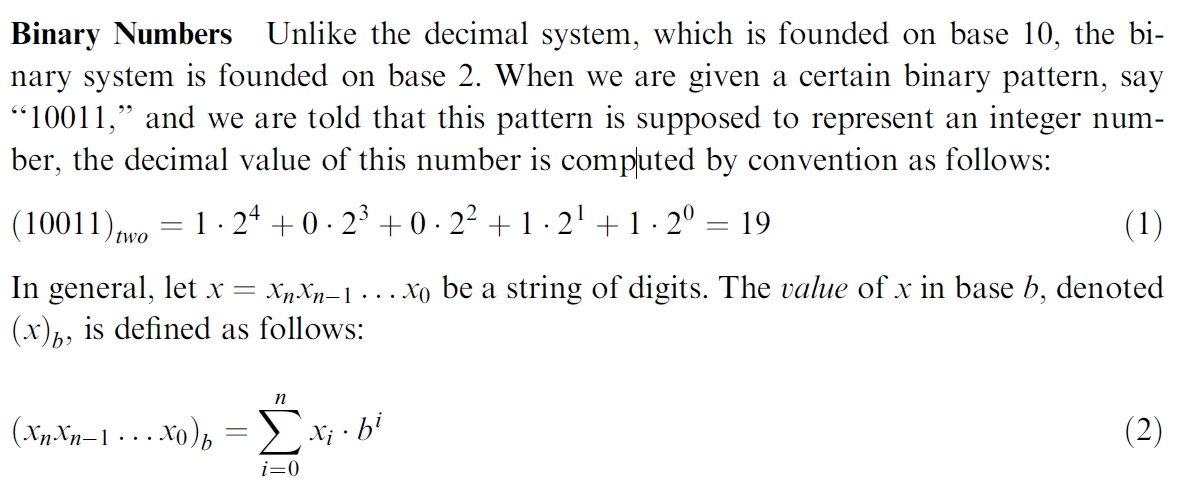
from Nand operations alone.

Chapter 2 – Boolean Arithmetic

**A**rithmetic **L**ogical **U**nit

binary codes and Boolean arithmetic can be used, respectively, to

represent and add signed numbers



To obtain the code of

m -x from the code of x, leave all the trailing (least signiﬁ-

cant) 0’s and the ﬁrst least signiﬁcant 1 intact, then ﬂip all the remaining bits (convert

0’s to 1’s and vice versa). An equivalent shortcut, which is easier to implement in

hardware, is to ﬂip all the bits of x and add 1 to the result.

