State formulae for De Morgan’s Law and the Distributive Rule

De Morgan’s Law 德摩根定律 对偶律

Not(x And y) = Not(x) Or Not(y)

Not(x Or y) = Not(x) And Not(y)

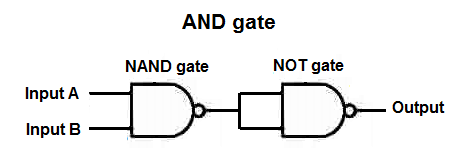
the Distributive Rule 分配规则

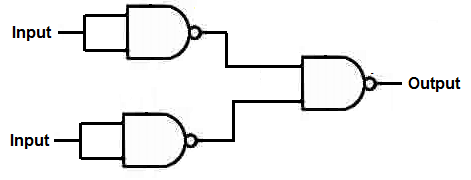
x and (y or z) = (x and y) or ( x and z)

x or (y and z) = (x or y) and ( x or z)

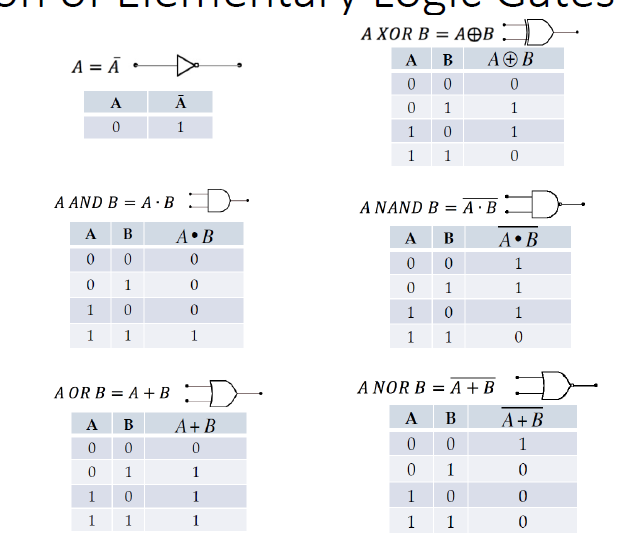
12个逻辑定律

Draw gate diagrams for the **Or** gate and the **Not** gate using only **Nand** chips.





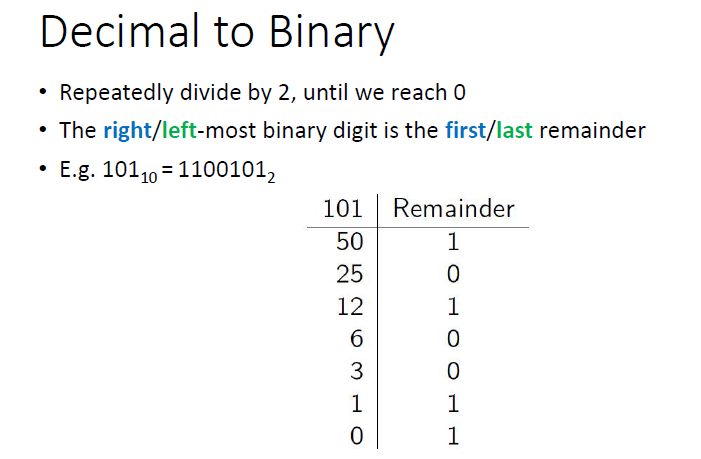
判断图题

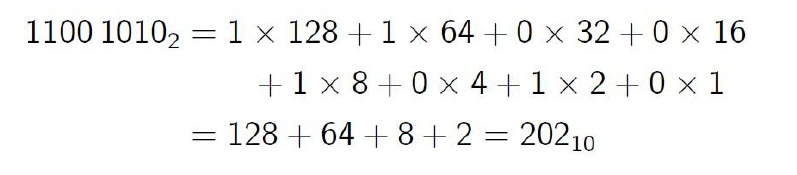


****

第一种方法:以真值表内输出端“1”为准  
第一步:从真值表内找输出端为“1”的各行,把每行的输入变量写成乘积形式;遇到“0”的输入变量上加非号。  
第二步:把各乘积项相加,即得逻辑函数的表达式。  
  
第二种方法:以真值表内输出端“0”为准  
第一步:从真值表内找输出端为“0”的各行,把每行的输入变量写成求和的形式,遇到“1”的输入变量上加非号。  
第二步:把各求和项相乘,即得逻辑函数表达式。

Convert the signed decimal number -120 to Octal, Hexadecimal and 8-bit 2s complement binary numbers.





Dec to octal

120 / 8 15 0

15/8 1 7

1/8 0 1

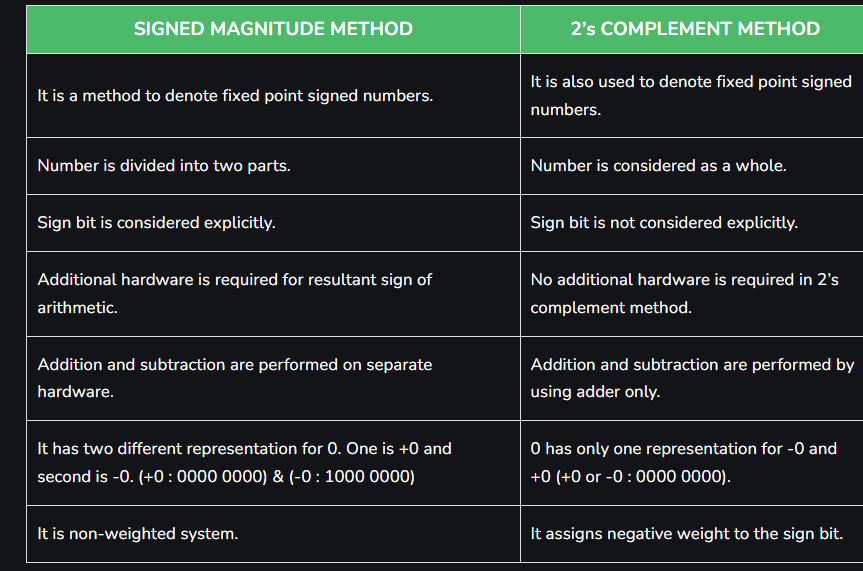
-1708

Hexadecimal (base 16)

120 / 16 7 8

7 / 16 0 7   
-78 16

Briefly explain the ‘2s complement’ method of representing negative numbers and the ‘Sign and Magnitude’ approach, discuss the merits of each



Convert the binary number 11011001 to an 8 bit, 2s complement decimal number

**sign and magnitude**

**leftmost** bit to be the **sign**

0 ⇒+, 1 ⇒–

magnitude can range from 0000000 (0) to 1111111 (127)

One’s Complement

E.g. the ones' complement form of 00101011 (4310) becomes 11010100 (−4310)

Convert the binary number 11011001 to an 8 bit, 2s complement decimal number

步骤1：确定最高有效位（MSB）是0还是1。

最高有效位是1，表示一个负数在二进制补码表示中。

步骤2：对剩下的7位（不包括最高有效位）进行取反（1's complement）。

剩下的7位是1011001。为了得到1's complement，我们将每一位取反，得到0100110。

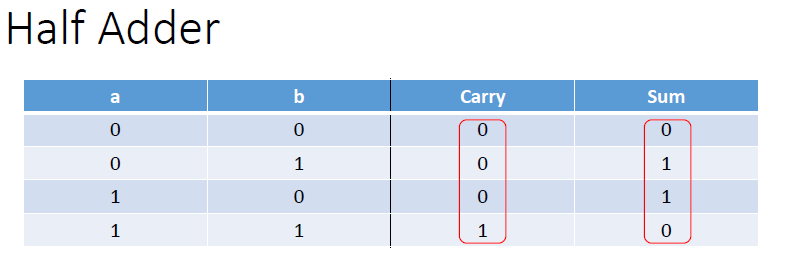
步骤3：在步骤2得到的1's complement上加1。

0100110 + 1 = 0100111

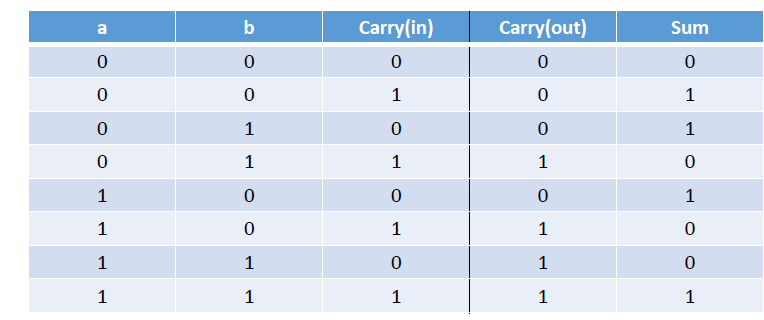
Half adder: adds two bits

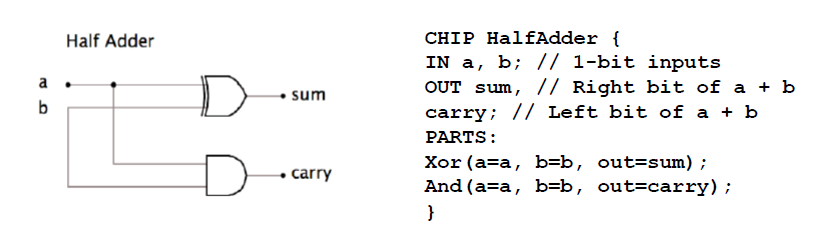
•Full adder: adds three bits

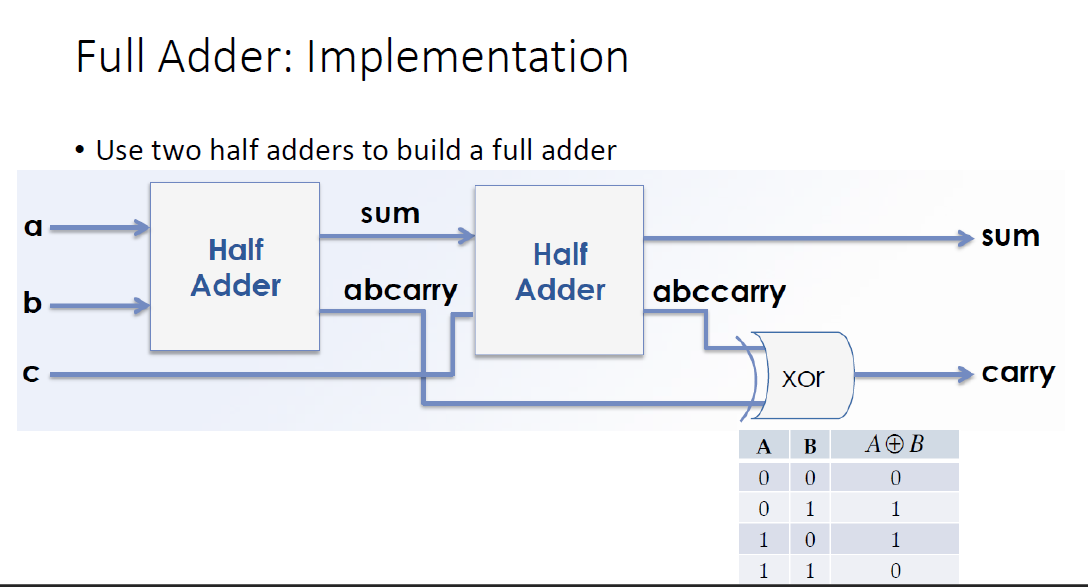
•Adder: adds two integers



Draw the truth table for a 1 bit full adder







图灵机能够执行各种计算任务，被认为是一种通用的计算模型。它可以模拟任何可以用算法形式描述的算法或计算过程。纸带被用作机器的内存，读写头可以读取和写入纸带上的符号，并且可以向左或向右移动。

冯·诺依曼体系结构（von Neumann architecture）它由四个主要组成部分构成：中央处理单元（CPU）、存储器、输入输出设备和控制单元。

冯·诺依曼体系结构与图灵机有一些相似之处。两者都使用存储器来存储指令和数据。在冯·诺依曼体系结构中，存储器通常是使用随机存取存储器（RAM）实现的。冯·诺依曼计算机的中央处理单元从存储器中获取指令、执行指令，并将结果存回存储器。

存储程序计算机的概念，即将指令和数据存储在同一存储器中，是冯·诺依曼体系结构的一个基本特点。这使得程序可以被当作数据处理，通过将不同的指令加载到存储器中，实现执行不同程序的能力。

尽管图灵机和冯·诺依曼体系结构在存储器和计算方面存在一些相似之处，但冯·诺依曼体系结构将图灵机模型扩展为包括其他组件和特性，以适应实际的计算系统，例如输入输出设备和分层存储器结构。冯·诺依曼体系结构是对图灵机思想的实际实现，并成为大多数现代计算机的基础。

一个1位寄存器的关键组成部分包括锁存器（Latch）和触发器（Flip-flop）。

锁存器是一种存储器件，用于存储和保持输入信号的状态。它由两个反馈连接的逻辑门组成，通常是两个交叉连接的 NOR 门或 NAND 门。这两个门的输出互相连接到对方的输入，形成了一个闭环。锁存器有两个输入端：数据输入（D）和时钟输入（CLK），以及一个输出端（Q）。

在1位寄存器中，锁存器的输出与触发器的输出相连。触发器是一种时序电路元件，根据时钟信号的变化来改变输出。常见的触发器有 D 触发器（D Flip-flop）、RS 触发器（RS Flip-flop）、JK 触发器（JK Flip-flop）等。

对于一个基本的1位寄存器，其输入和输出关系如下：

数据输入（D）：用于输入要存储的位值。在时钟上升沿（或下降沿）到达时，数据输入的值被存储在寄存器中。

时钟输入（CLK）：用于控制寄存器的存储操作。时钟输入信号的上升沿（或下降沿）触发寄存器的存储动作，将数据输入的值存储到输出端。

输出（Q）：在时钟上升沿（或下降沿）到达时，寄存器中存储的数据被输出到输出端。输出值保持不变，直到下一个时钟信号触发。

在每个时钟间隔中，输入信号的值（D）被存储到寄存器中，并在时钟上升沿（或下降沿）触发时从寄存器输出（Q）。这样，在下一个时钟信号到来之前，输出值将保持不变。

NOT（非）：拥有最高的优先级，它应用于单个操作数，将其逻辑值取反。

AND（与）：其次是逻辑与运算符。当表达式中同时使用多个 AND 运算符时，它们从左到右依次计算。

OR（或）：具有最低的优先级，它用于组合两个或多个逻辑表达式。当表达式中同时使用多个 OR 运算符时，它们从左到右依次计算。

Discuss how different types of delays effect the performance of a ripple carry adder.

级联进位加法器（ripple carry adder）是一种用于执行二进制数的算术加法的数字电路。它由多个全加器按级联方式连接而成，其中每个加法器的进位输出与下一个加法器的进位输入相连。

不同类型的延迟会显著影响级联进位加法器的性能。下面我们将讨论两种重要的延迟类型：传播延迟和进位传播延迟。

传播延迟：

传播延迟是信号从逻辑门的输入传播到输出所需的时间。在级联进位加法器中，每个全加器都有一个与之相关的传播延迟。传播延迟决定了在提供输入后多快可以获得输出结果。如果传播延迟较高，将增加获得输出所需的总延迟。

当级联进位加法器的输入发生变化时，进位信号必须通过每个级别传播才能到达最终输出。传播延迟随着进位信号在每个加法器级别中传播而累积。因此，随着要相加的位数增加，获得正确输出所需的总延迟也会增加。

进位传播延迟：

进位传播延迟专指进位信号在加法器的每个级别中传播所需的时间。在级联进位加法器中，每个加法器级别的进位输出与下一个级别的进位输入相连。由于进位信号必须按顺序在每个级别传播，进位传播延迟会影响加法器的整体性能。

Discuss how different types of delays within a chip effects the maximum possible clock speeds

传输延迟（Transit Delay）：

传输延迟是信号在芯片内部传输的时间。它取决于信号通过电路路径所需的时间，包括电线、晶体管和其他逻辑门等元件的传输时间。较长的传输延迟会增加信号从一个位置到另一个位置所需的时间，从而限制了时钟信号的频率。

响应延迟（Response Delay）：

响应延迟是芯片对输入变化做出响应的时间。它包括各种逻辑门、寄存器和其他组件的响应时间。较长的响应延迟会导致输入信号在产生变化后，芯片内部的电路需要一定时间才能稳定下来并产生正确的输出。这限制了时钟信号的频率，因为时钟信号的周期必须足够长，以确保芯片内的电路有足够的时间来响应输入变化。

存储延迟（Memory Delay）：

存储延迟指的是从读取或写入存储器（如寄存器、缓存或内存）中获取数据或将数据写入存储器的时间。存储延迟可能受到存储器的访问速度、读写电路的复杂性以及数据传输路径的长度等因素的影响。较长的存储延迟会降低读写操作的速度，从而限制芯片的时钟速度。