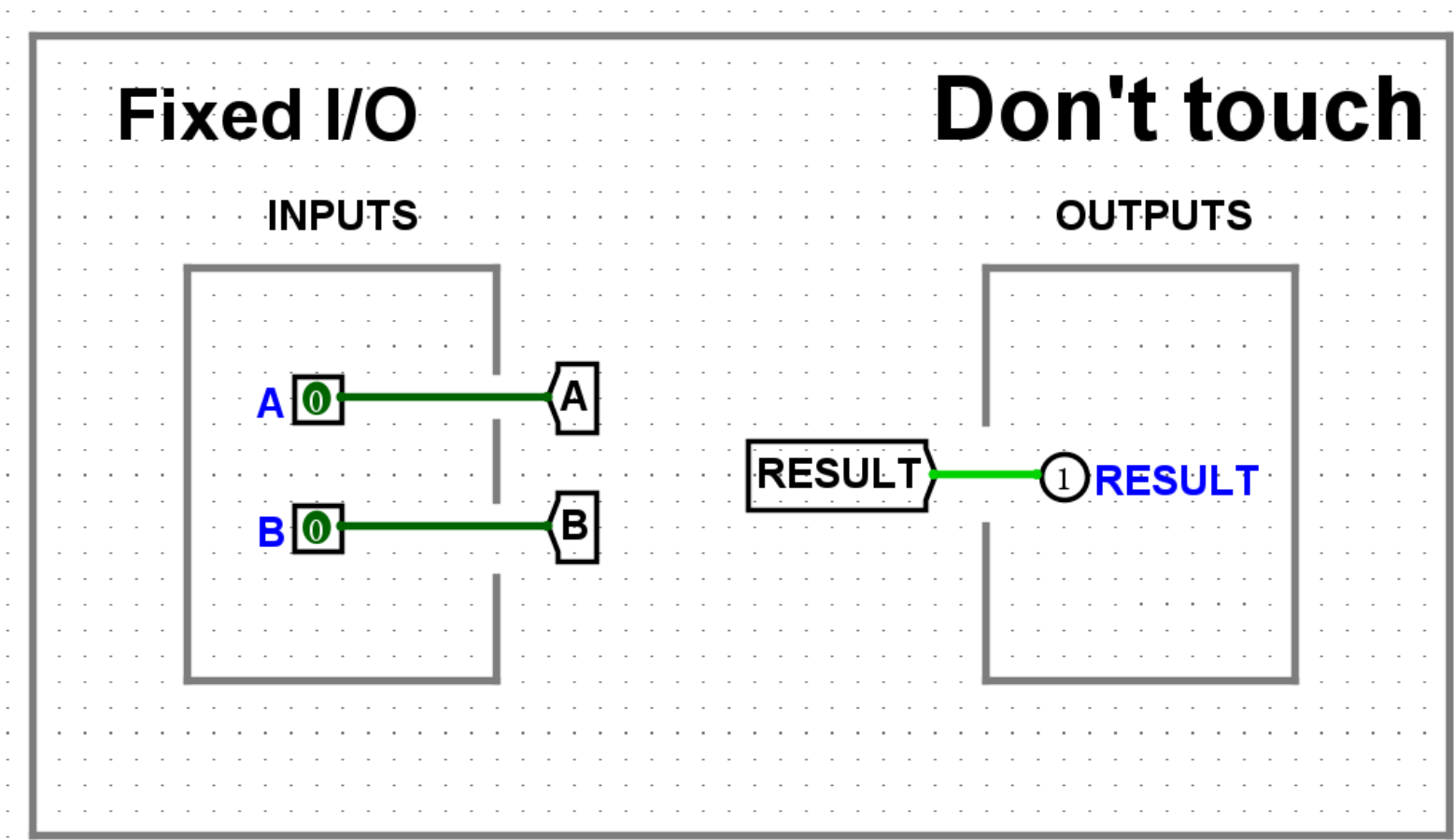


Homework 1

About "Don't Touch"



在作业中，我们有一些标注了 Don't Touch 的电路，这部分是固定位置的输入输出，用于进行作业的结果检测。

- 你可以**复制**它们，但请不要以任何形式**移动**它们。
- 如果因为移动 IO 部分导致测试不通过，TA 有权利不进行补测事宜。
 - 当然，如果你及时发现了问题，可以选择重新下载模板文件。

Exercise 1: Binary Adder

Exercise 1.1: Binary Half Adder

对于一个 1 Bit 的二进制（0 or 1）来说，它的加法被定义为：

- $0 + 0 = 0$
- $1 + 0 = 1$
- $0 + 1 = 1$
- $1 + 1 = (1)0$

其中，在 $1+1$ 中出现的 (1) 被称之为“进位”标志。

因此，在硬件上，我们定义半加器有两个输入 a 和 b 和两个输出 sum （表示 1 Bit 加法的结果）和 $cout$ （表示进位），其公式为：

$$sum = a \oplus b$$

$$cout = a \& b$$

其中 \oplus 为异或（XOR）操作。如果你不知道它是什么，请自行学习~

Exercise 1.2: Binary Full Adder

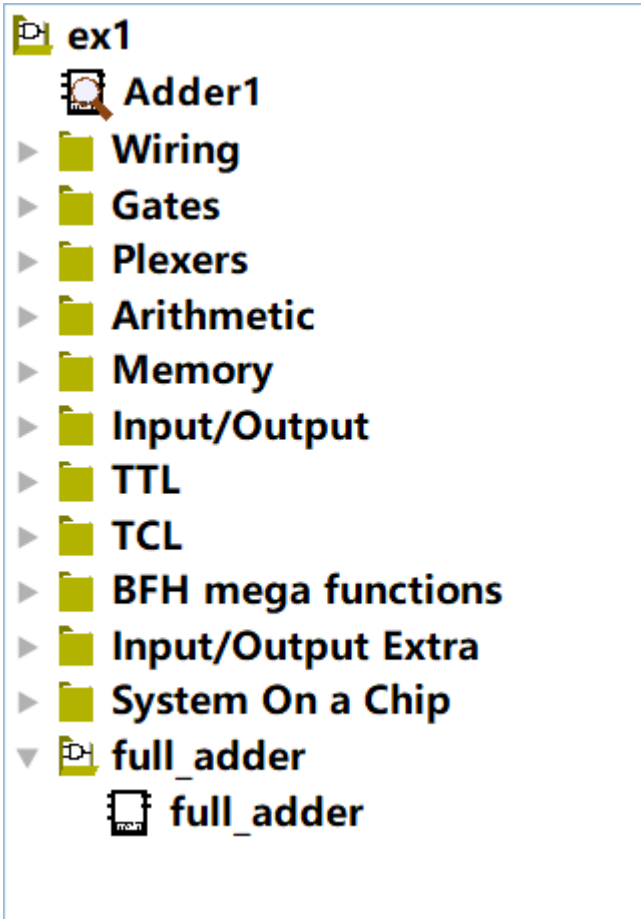
半加器只能执行 1 Bit 的加法运算，这显然不够“给力”。我们需要进一步分析加法的逻辑，来构建更加有用的加法单元：全加器。

全加器的核心是考虑了进位（被记为 `cin`）。对于某一位来说，我们除了考虑两个加数的对应位数 `a` 和 `b`，还要考虑是否有前一位传来的进位信息。因此，我们的输入就有三个数：`a`，`b`，`cin`。我们需要构建这三个数到输出 `sum` 和 `cout` 的逻辑运算。

现在，尝试写出这个真值表，并将它在 Logisim 中构建出来吧！

Exercise 2: 4-Bit Adder and Subtractor

请使用 Exercise 1 子电路的方式构建新的电路。



Exercise 2.1: 4-Bit Adder

在 Exercise 1 中，我们设计了一位的全加器。这种全加器引入了 `Cin` 和 `Cout` 的概念，可以很方便的扩展到多位全加器，从而实现我们常规认知意义上的加法运算。

现在，尝试将 1-Bit 的全加器扩展到四位吧！

注意，在加法运算中，我们忽略可能出现的“多”的一位（称之为“溢出”位，体现为 `Cout` 的值），例如：

$$0b1111(15) + 0b1111(15) = 0b11110(30)$$

这里我们的加法器输出应该是 `(Cout)1 (Result)1110`，我们只需要输出 `1110` 部分即可。

Exercise 2.2: 4-Bit Subtractor

在数字电路中，如果我们使用补码实现数字的表示，则“对一个数取负”是非常简单的：

$$-x = \sim x + 1$$

其中 \sim 是按位取反操作。比如，数字 3 可以表示为 `0011`，则 -3 就可以表示为 `1101`。

如果你不知道什么是补码，请自行学习相关资料；在本次作业中，你不需要理解其中的原理，也不需要考虑任何的“溢出”问题（如果你不知道什么是溢出，请忽略这个概念），你只需要知道我们在进行减法运算时该如何操作即可：

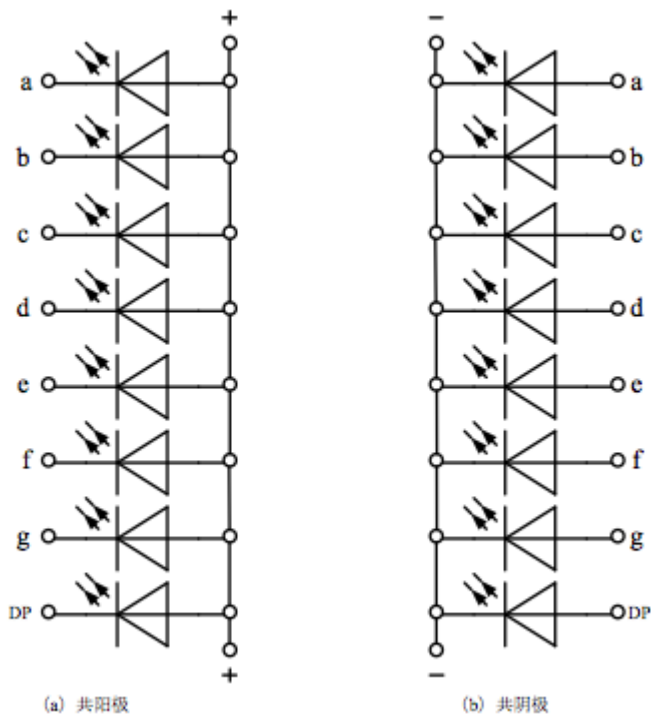
使用了 Cin 和 Cout 结构的全加器可以很方便的实现减法：我们把最低位的 Cin 接到常数 1 上（表示 +1；还记得我们在 Logisim 使用的教程里讲过 Constant 吗？是时候使用它了！），并把输入 B 取反，就得到了一个减法器。

现在，尝试实现一个四位减法器吧！

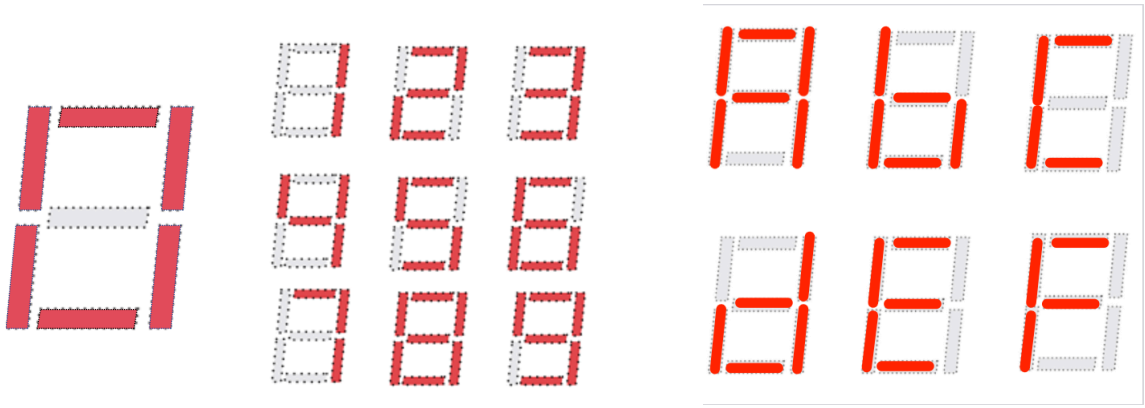
注意，和加法器一样，我们的减法器也不需要考虑最高位 Cout 的输出。

Exercise 3: 七段数码管的解码器

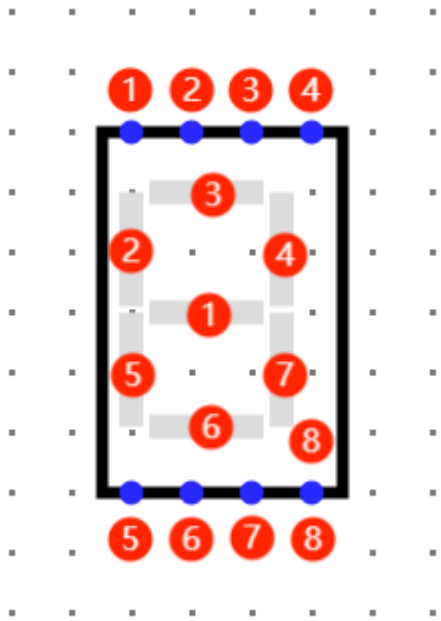
七段 LED 数码管是一种常用的显示元件，常应用于手表、计算器等仪器中，用于显示数值。下图是数码管的原理图，数码管分为共阴极和共阳极两种类型，共阴极就是将七个 LED 的阴极连在一起，让其接低电平。这样给任何一个 LED 的另一端高电平，它便能点亮。而共阳极就是将七个 LED 的阳极连在一起，让其接高电平。这样，给任何一个 LED 的另一端低电平，它就能点亮。这里的 LED 是共阴极 LED。



数字 0~9 及十六进制的 a~f 在数码管上的显示方式如下图所示：



其中不同位置对应的引脚如图所示：



其中引脚输入为 1 的时候，对应的 LED 数码管被点亮。

针对每一个引脚，我们都需要写出它所对应的真值表，然后使用逻辑电路去构建它。请自行设计数码管的对应编码，并实现它。

- 你可以使用“比较器”，也可以对输入的数的每一位进行拆分，然后构建对应的组合电路。

ACTION ITEM & Submission

ACTION ITEM

- **请不要使用 Arithmetic 文件夹中的内置 Adder Subtractor 等模块**（即只使用选择工具图标旁边提供的 AND 、 OR 、 NOT 和 XOR 门
- **提交时请不要更改你的电路文件名称**

Submission

1. Exercise 1: 在 Logisim 中构建 1-Bit Half Adder 和 Full Adder
2. Exercise 2: 将 1 Bit 的 Fall-Adder 扩展到 4-Bit Adder 和 4-Bit Subtractor。
3. 构建七段数码管的解码器
4. 直接将电路文件提交至Gradescope

About Homework

如果对作业题目有疑问，或者发现了作业题目的任何问题，请及时联系助教。

Good luck and have fun!