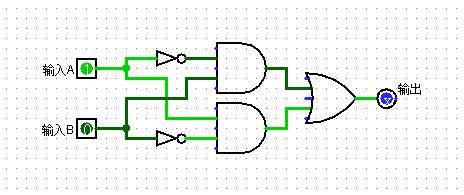
**异或门和全加器最简结构设计原理**

一．异或门XOR

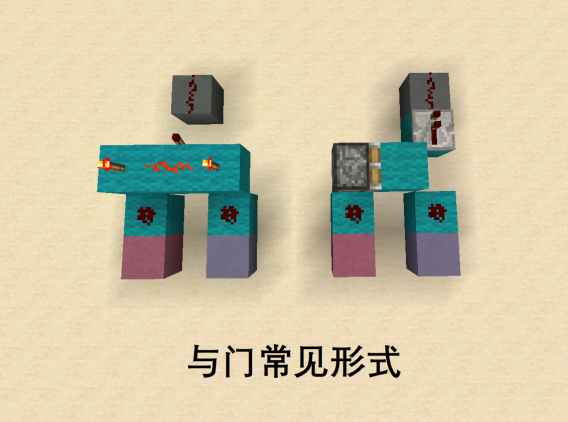
异或门最简单的逻辑代数表达式如下：

表达式1：A⊕B = A·B’+B·A’

其对应的逻辑门电路结构图如下：



但是这种最简单的表达式在游戏中对应的电路结构占用空间较大，因为游戏中与门不是基础逻辑门，需要用或门和非门进行组合才能建造出与门，而该电路结构用到的与门有两个并且难以优化空间压缩体积，故舍弃。



理论上异或门存在很多种表达式，我们需要从中挑选出那些用红石电路搭建起来体积比较小的方案。

现给出如下四种异或门表达式（还存在其他表达式，此处取如下四种为例）：

表达式2：A⊕B = (A·B)’·A+(A·B)’·B

表达式3：A⊕B = (((A·B)’·A)’·((A·B)’·B)’)’)’

表达式4：A⊕B = (A’+A·B)’+(B’+A·B)’

表达式5：A⊕B = (A·B+A’·B’)’

可以证明这四种表达式与表达式1是等价的，比如表达式4和5：

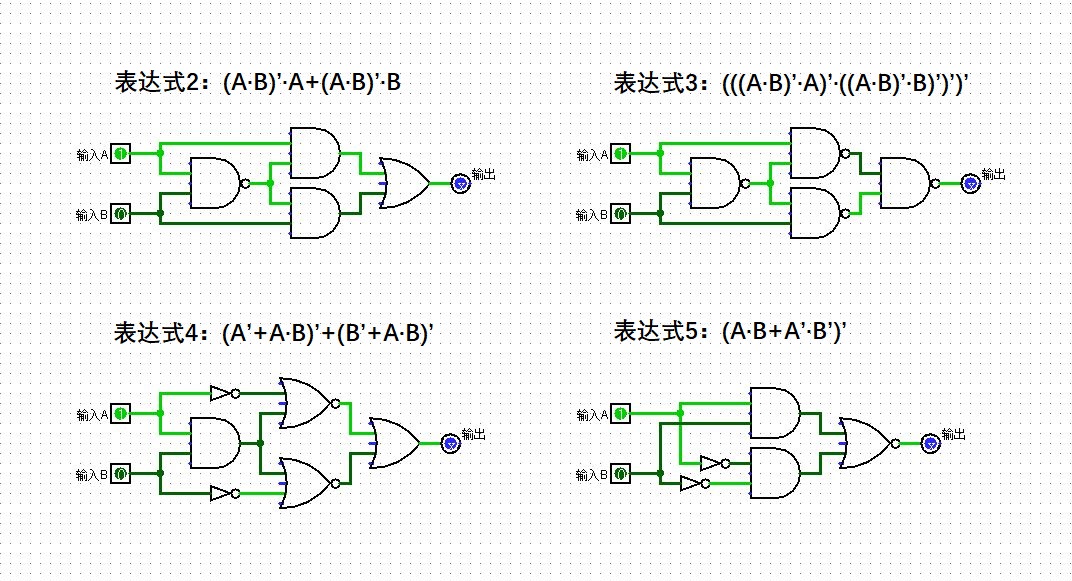
表达式4：A⊕B =

(A’+A·B)’+(B’+A·B)’ = A·(AB)’+B·(AB)’ = A·(A’+B’)+B·(A’+B’) = A·A’+A·B’+B·B’+B’·A’ = 0+A·B’+0+B’·A’ = A·B’+B·A’

表达式5：A⊕B =

(A·B+A’·B’)’ = (A·B)’·(A’·B’)’ = (A·B)’·(A+B) = A·(A·B)’+B·(A·B)’ = A·(A’+B’)+B·(A’+B’) = A·A’+A·B’+B·B’+B’·A’ = 0+A·B’+0+B’·A’ = A·B’+B·A’

上述四种表达式对应的电路结构图如下：



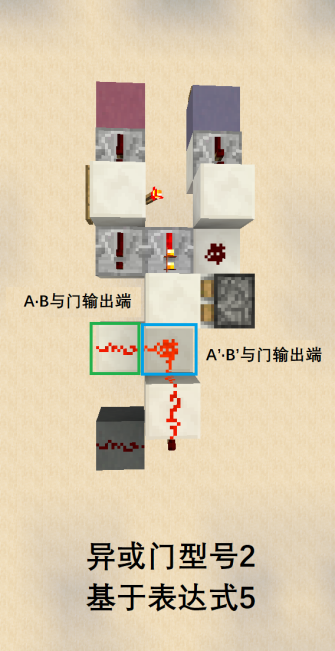
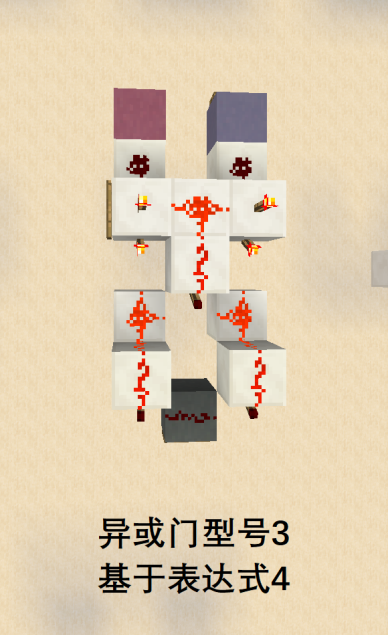
可以看出：

①表达式2和表达式3存在三个及以上的与门或者与非门，难以压缩结构，故舍弃。

②表达式4里的两个A·B的输出可以**共用**，所以等于只有一个与门，很适合用来做异或门。

③表达式5虽然和表达式1一样也有两个与门，但是有玩家用粘性活塞的一些特性做成非常紧凑的双与门结构可以用于表达式5。仔细观察可知，表达式5的A’·B’实际上等价于一个或非门。但是电路中不直接选择或非门，仍然使用上述结构的原因就是为了配合粘性活塞建造出紧凑的双与门。

我们把同种功能电路不同的实现方式称为**异构体**，把结构压缩到最小的实现方式称为**最简结构**。即使基于同一种逻辑代数表达式，实现出的最简结构也会有一些差异，往往表现在输入输出的朝向不同或者立体的长宽高不同。教学存档中给出了3种最简的异或门的异构体，其中型号1和型号3是基于表达式4的，型号2是基于表达式5的。



对多种异或门表达式对应的红石电路进行测试，实验证明基于表达式4能够实现体积最小且运行最稳定的异或门，因为基于表达式5的型号2异或门中带有粘性活塞的结构，而粘性活塞由于1.5t的激活特性导致活塞推拉在高频信号的场景中容易出错。

1. 全加器Full Adder

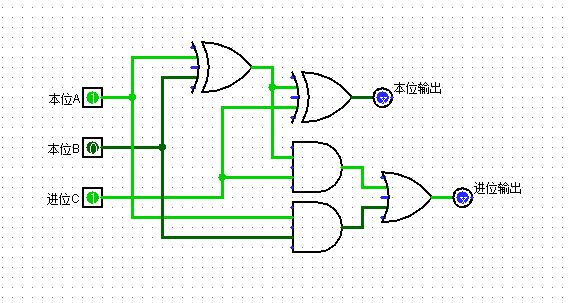
全加器的本位输出和进位输出表达式如下：

本位输出 = A⊕B⊕C

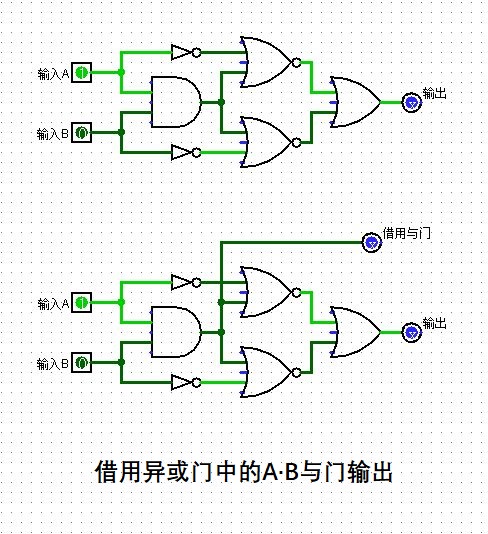
进位输出 = A·B+(A⊕B)·C

其中A和B是本位输入，C是上一级产生的进位输入。

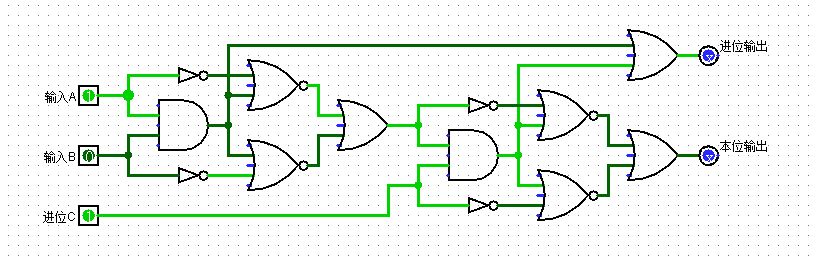
全加器逻辑门电路结构图如下：



红石全加器的电路压缩采用了如下技巧：两个异或门朝向同样的方向一前一后地摆放，两者之间留有一定的空间，为的是利用这个空隙处来实现进位端的运算，并且使得进位端的输入输出在一条直线上满足级联的需求（即上一级的进位输出正好对准下一级的进位输入）。异或门中已经有一个与门结构，两个进位端运算需要用到的与门可以**借用**这两个异或门中各自的与门（见下图），所以最终实际上不需要建造额外的两个与门，这是减小体积**最关键**的一步。另外还需要注意串线的问题，Quiz的第六道题里就需要用诸如方块隔挡和半砖纵向单向传输的特点，来节省布线占用的空间以及避免距离很近的两条线路互相干扰。



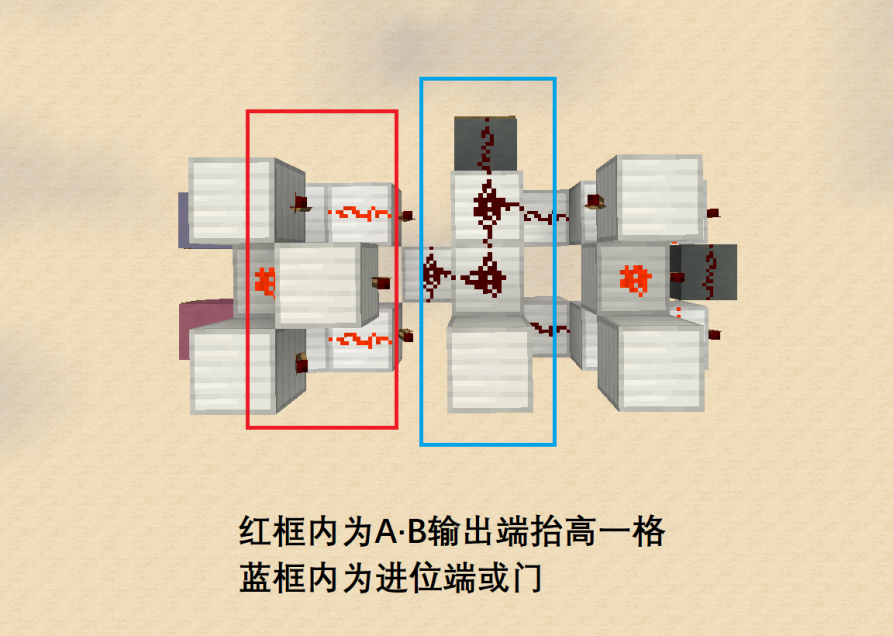
基于表达式4的全加器电路结构图如下：



表达式4的异构体中，型号1最适合用于设计能够级联的全加器，因为基于型号1设计出的全加器体积是最小的，这主要得益于它紧凑的5·3·4立体结构，比型号3这种平式躺设计更能有效利用两个拼在一起的异或门之间的空间。

最简全加器一前一后摆放的两个异或门中的第一个异或门和异或门型号1有略微差异，但是差别不大仍可以看作是异或门型号1，它将A·B与门输出端的红石火把抬高了一格，为的是被借用的与门的输出端能够输出信号到中间进位端的或门，而原来的输出端太低了没办法将这个信号传上去。

下图为最简全加器的结构，是所有ALU结构的基石：



费很大的精力将电路压缩到最简结构，一方面是一种乐趣，另一方面当我们建造大规模电路的时候，单位结构体积的微小差异会被不断放大。了解了存档中给出的几种异或门和全加器的设计理念，就差不多掌握了Minecraft数字电路的核心思路。

存档中有一个文件“saves/logisim异或门和全加器.circ”，可以用数字电路仿真软件logisim打开，里面包含了本文档里列举的若干数字电路的结构。