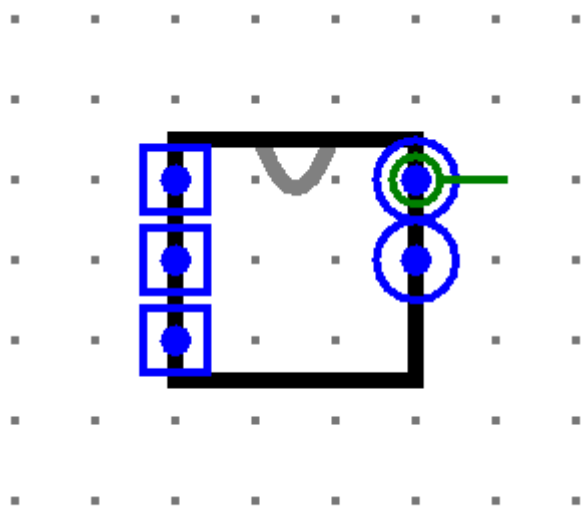


左上角按钮)



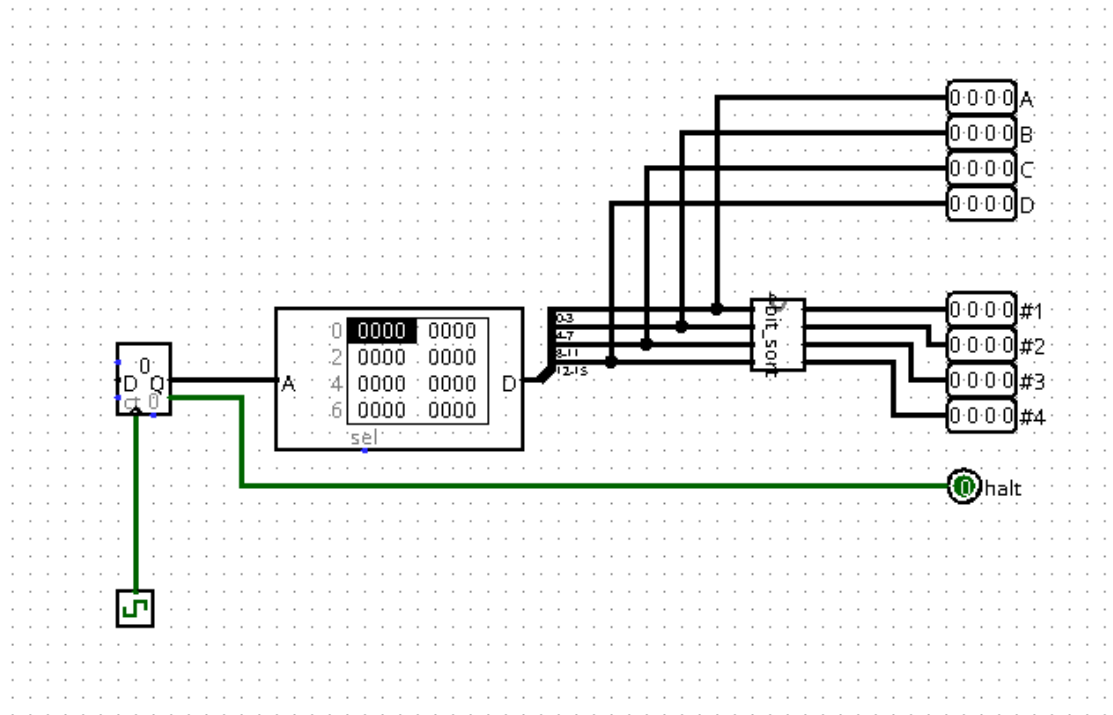
排序电路

在前面的学习过程中，我们搭建了一个 1 位的 **swap** 电路。现在需要我们使用之前的 1 位 **swap** 电路来搭建一个 4 位 4 输入的排序电路。

提交要求

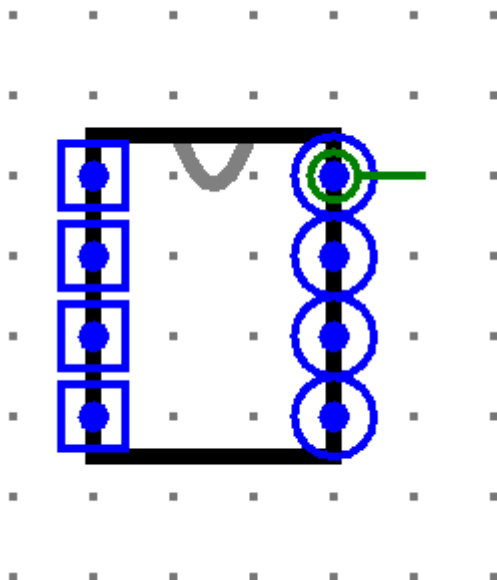
先使用 1 位的 **swap** 搭建 4 位的 **swap**，再使用 4 位的 **swap** 模块和 Logisim 内置的 **comparator** 元件搭建排序电路(请不要使用 **Plexers** 类元件)

- **功能描述:**该电路具有 4 个 4 位的二进制数字作为输入和 4 个 4 位的二进制数字作为输出。它的功能是，将 4 个输入的二进制数字进行排序，从上往下数第一个输出端口输出的是 4 个数字中最小的，第二个输出端口输出的是第二小的，以此类推。
- **输入:** A,B,C,D(4bit)
- **输出:** #1,#2,#3,#4(4bit)(#1 对应第一个输出端口，以此类推)
- 文件内 **1 位 swap** 模块名: 1bit_swap
- 文件内 **4 位 swap** 模块名: 4bit_swap
- 文件内 **排序电路**模块名: 4bit_sort
- **Hint:** 所有的二进制数字均看做是无符号的
- **测试电路图:**



- 注意:请保证模块的 **appearance** 与下图完全一致, 否则有可能造成评测错误(查看模块 appearance 方法:在 Logisim 中打开相应模块后点击

左上角  按钮)



$2^n \bmod 5$

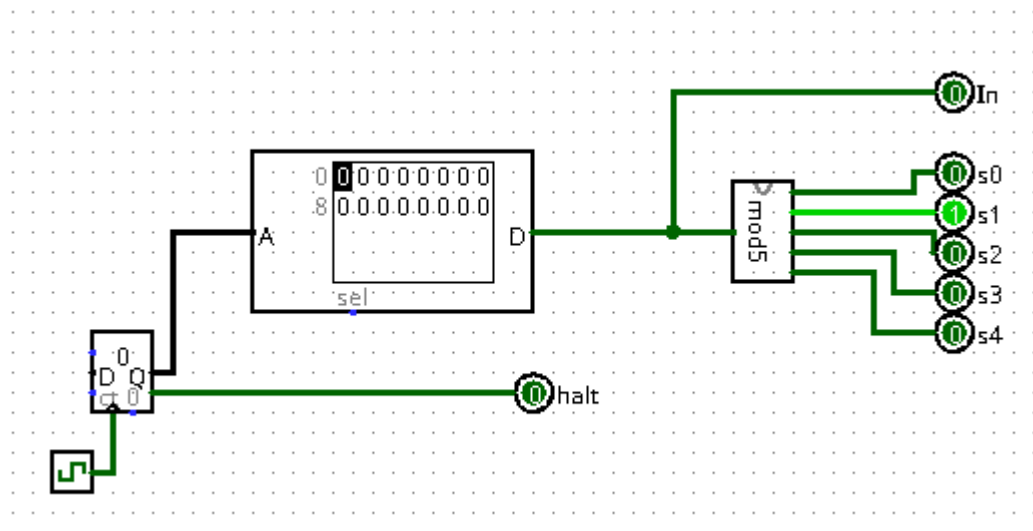
提交要求

使用 Logisim 搭建电路，该电路串行输入一个二进制无符号数 B (先从高位输入, 每输入一个数字就相当于之前输入的数左移一位再加上当前输入的数字), 输出“2 的 B 次幂”模 5 的余数的电路并提交。

- 输入：In (1bit 串行输入)
- 输出：S0, S1, S2, S3, S4 (独热编码，当 S_x 为 1 表示 $2^{\text{In}} \equiv x \pmod{5}$)
- 文件内模块名：mod5
- 注意：切勿使用内置算术器件（如加法器、除法等）！请搭建有限状态机！
- 输入输出样例：
输入输出样例中每一行表示相邻上升沿之间的开区间时间内的输入和期望输出。

1	In	s0	s1	s2	s3	s4
2	0	0	1	0	0	0
3	1	0	0	1	0	0
4	0	0	0	0	0	1
5	0	0	1	0	0	0
6	1	0	0	1	0	0
7	0	0	0	0	0	1
8	0	0	1	0	0	0
9	1	0	0	1	0	0
10	0	0	0	0	0	1
11	0	0	1	0	0	0
12	1	0	0	1	0	0
13	0	0	0	0	0	1
14	0	0	1	0	0	0
15	1	0	0	1	0	0
16	0	0	0	0	0	1

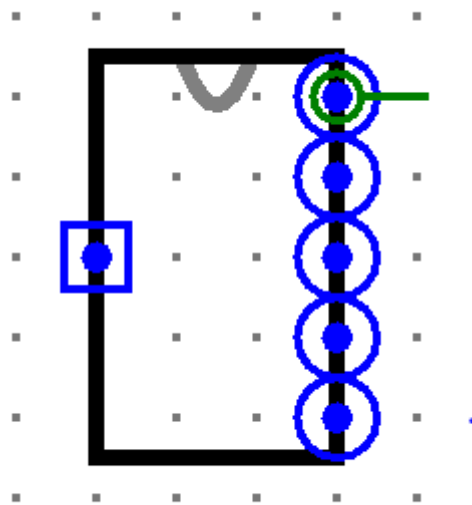
- 测试电路图：



- 注意:请保证模块的 **appearance** 与下图完全一致, 否则有可能造成评测错误(查看模块 appearance 方法:在 Logisim 中打开相应模块后点击



左上角 按钮)



斐波那契数列

黄小板同学在沙河东门半导体公司的工作终于到了尾声，这一次，它面临的任务是设计一个电路在规定的时间内去计算斐波那契数。这时，公司负责人提出了自己的看法，他提出完全可以使用简单的时序电路去计算斐波那契数，说着，他就开始搭建电路

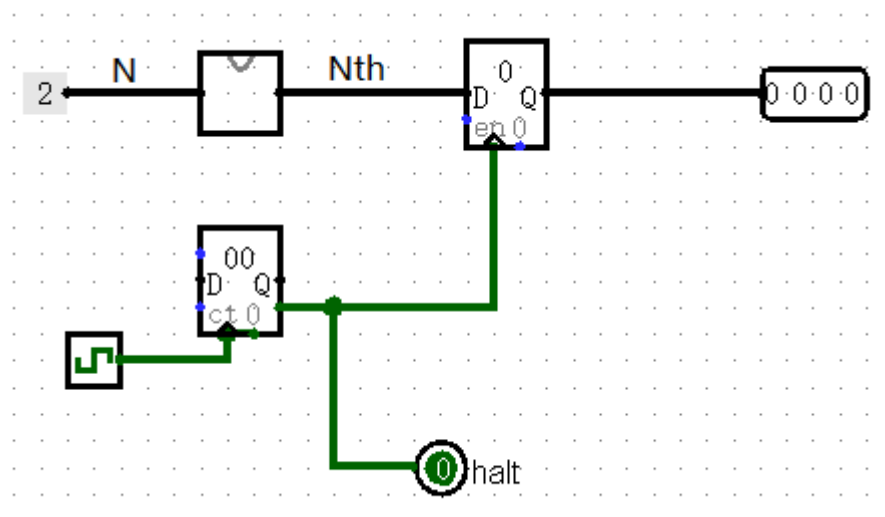
公司负责人开始紧张地思考...

TO BE CONTINUED...

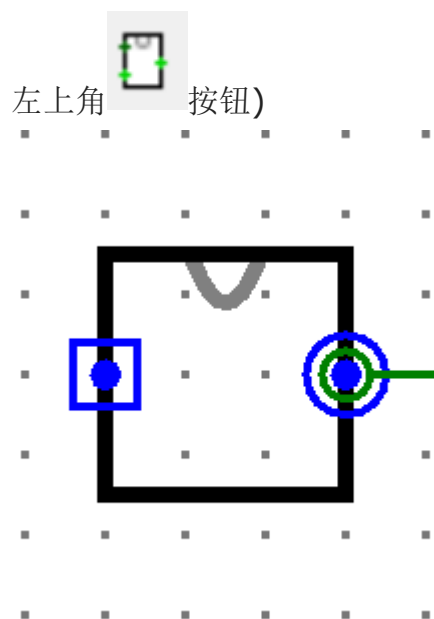
提交要求

使用 Logisim 搭建一个根据输入序号 x 计算对应序号斐波那契数 $\text{fib}[x]$ 的电路（输入序号 **0** 对应输出数 **0**，输入序号 **1** 对应输出数 **1**，输入序号 **2** 对应输出数 **1**，以此类推）并提交。

- 输入： N (3bit)
- 输出： Nth (4bit)
- 文件内模块名： `main`
- 测试要求： 每次给定一个固定输入保持不变，电路在 64 个周期内计算出结果并稳定输出，在结果未计算出之前输出端口输出 0
- 测试电路图：



- 注意: 请保证模块的 **appearance** 与下图完全一致，否则有可能造成评测错误(查看模块 `appearance` 方法: 在 Logisim 中打开相应模块后点击



- 请使用时序逻辑完成本题目。未按照要求（或采用捷径）完成并提交通过的同学，请重新提交。