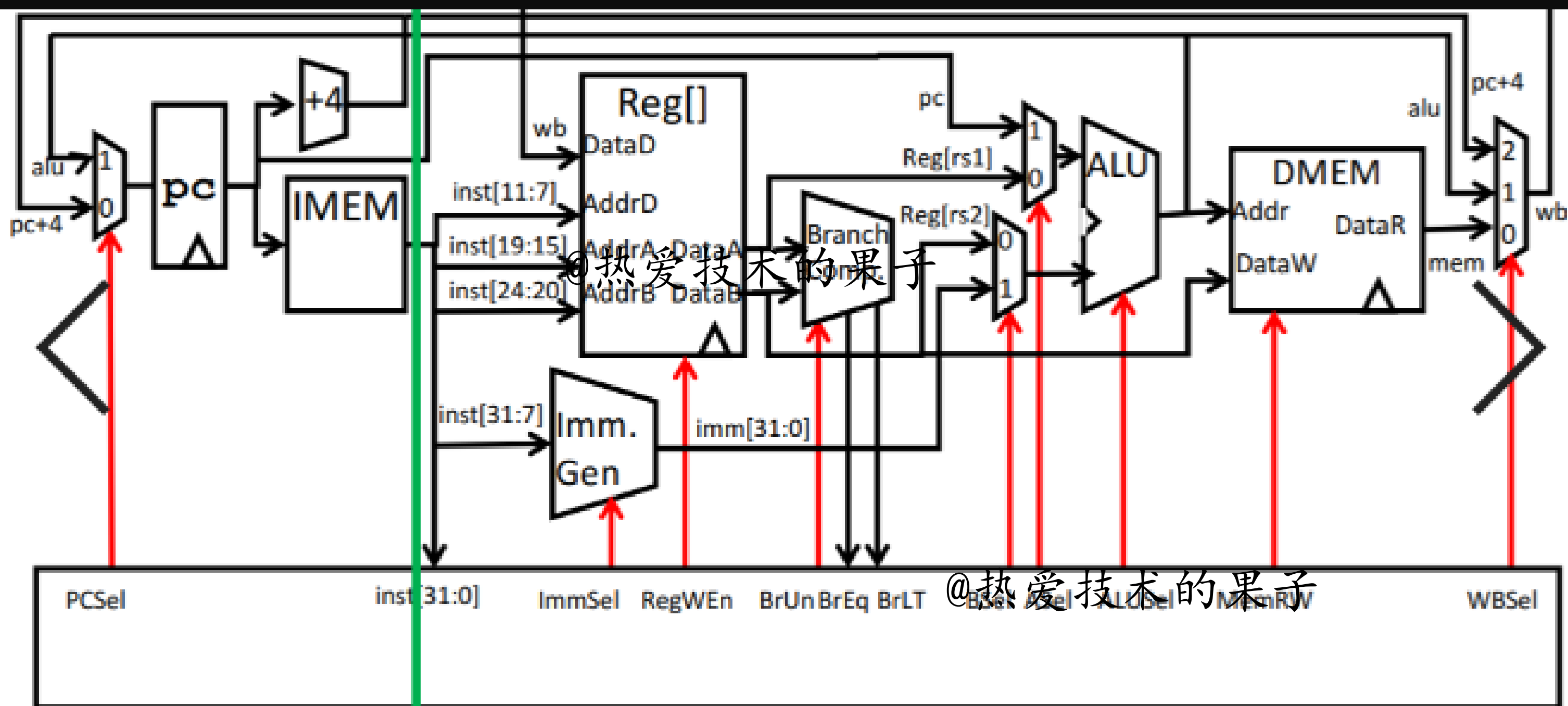


# 果子老师硬核分享 教你搭建自己的 CPU之三：寄存器

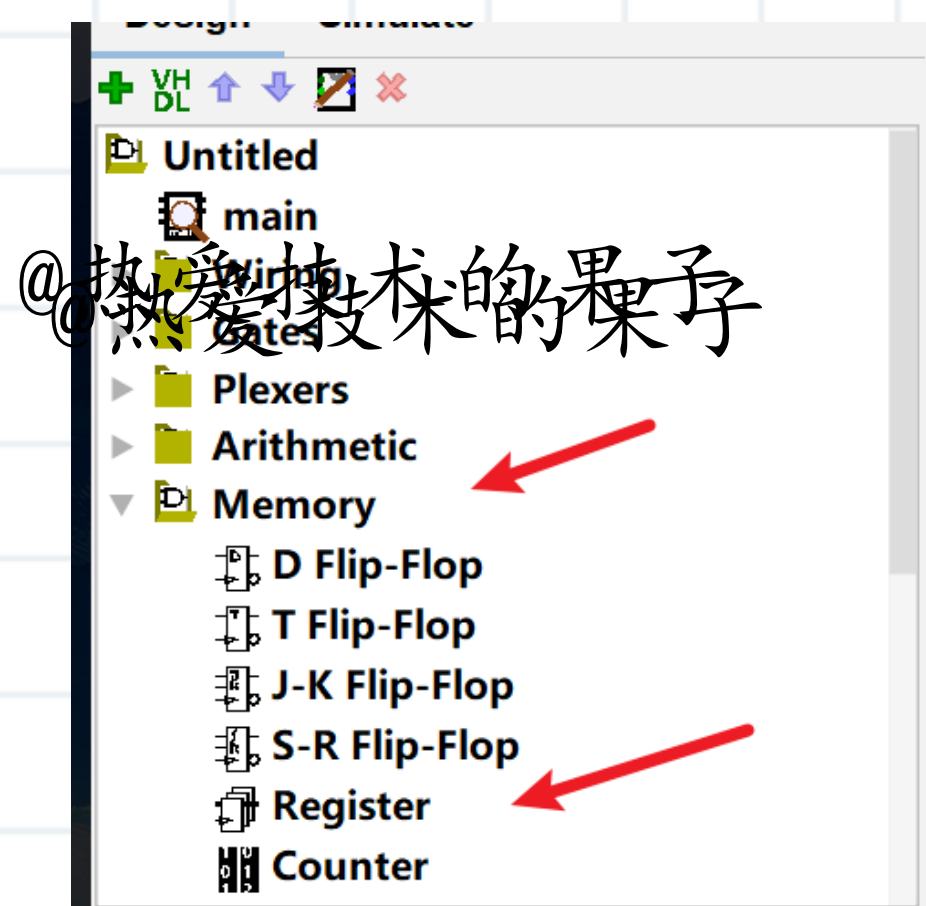


# 寄存器作用

本实验所有电路图都是使用 Logisim Evolution v3.8 完成。

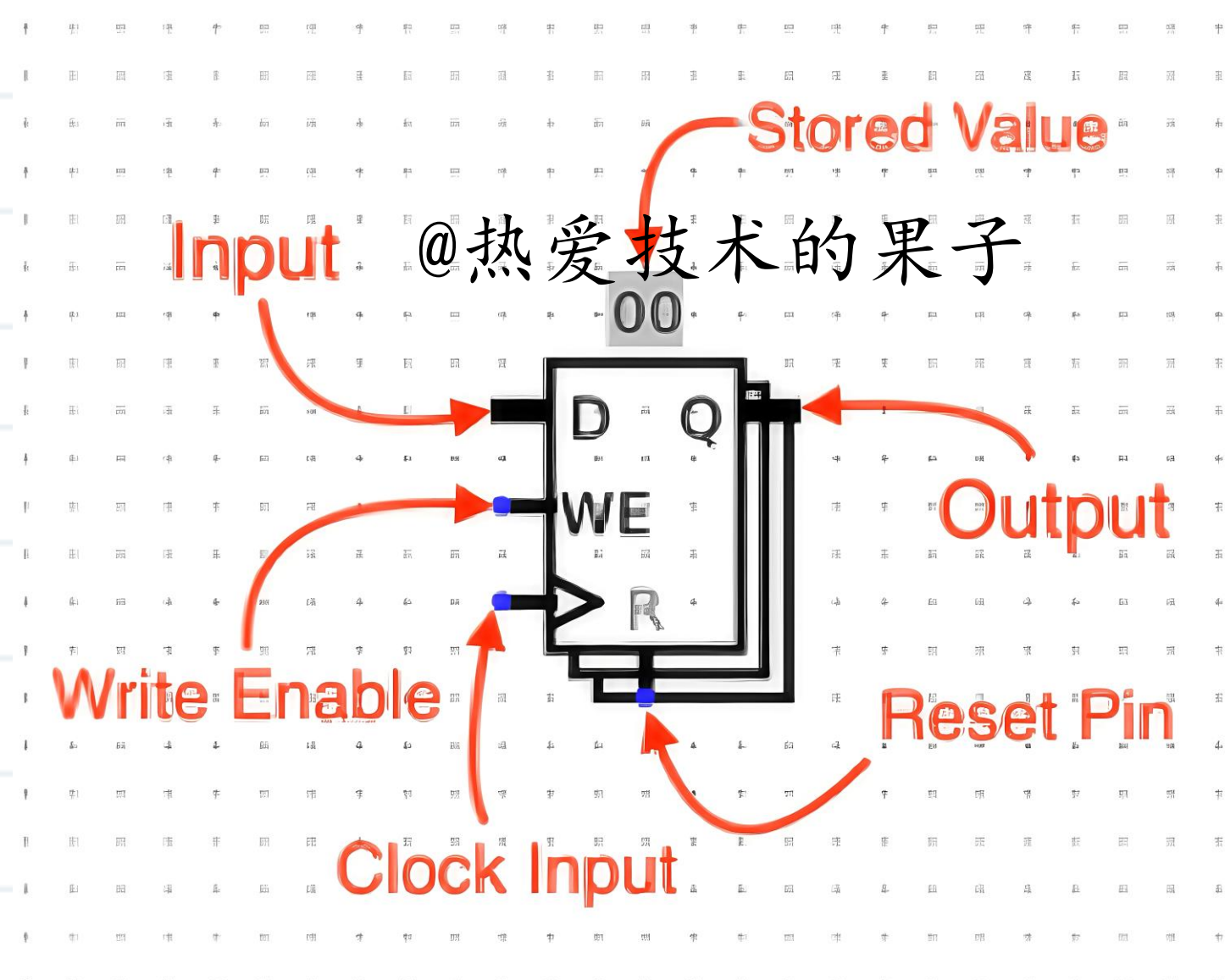
在CPU中，寄存器（Register）是位于处理器内部、速度极快的小容量存储单元，用来临时保存数据和指令执行过程中的中间结果。它们直接与运算单元相连，能在一个时钟周期内完成读写，比访问内存快得多。

## 寄存器元器件位置：



Logisim使用的是带写使能（Write Enable）和复位（Reset）功能的寄存器，各个管脚功能如下：

1. Input (D)数据输入端（Data）
2. Write Enable (WE)写使能信号端，当该信号为高电平时，寄存器在时钟触发时会将输入端（D）的数据写入存储单元；如果为低电平，则不会更新存储值。
3. Clock Input (Clock)时钟输入端，用于控制寄存器的更新时机。通常在上升沿（或下降沿）时，根据写使能信号的状态决定是否将输入数据存入寄存器。
4. Reset Pin (R)复位端，当该信号有效时，会将寄存器存储的值清零（或设为初始值），不受时钟和写使能的影响。
5. Stored Value寄存器内部当前存储的值，是通过输入数据和时钟更新后保留下来的。
6. Output (Q)数据输出端，会实时输出寄存器中存储的值，供CPU或其他电路使用





# 寄存器使用和连线

让我给你演示，实现一个重复递增值的电路。

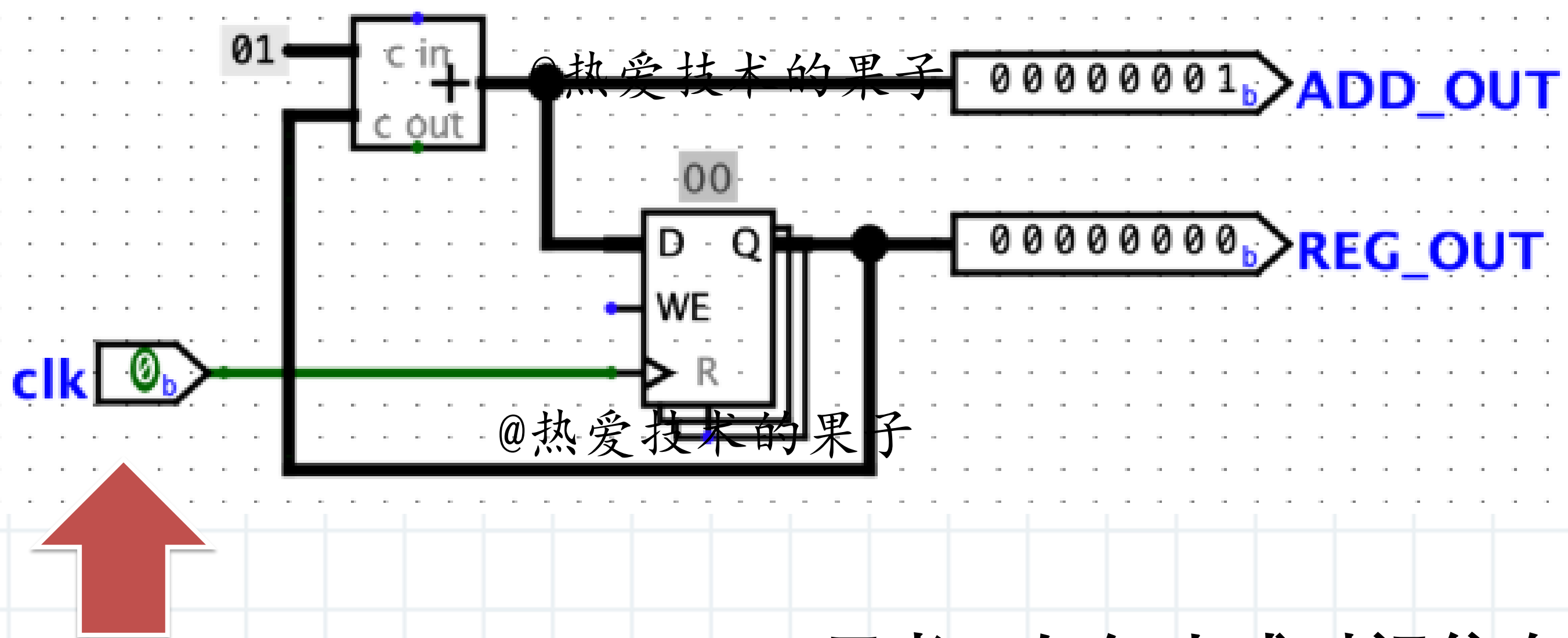
将加法器的输出连接到寄存器的输入端，将寄存器的输出连接到加法器的输入端。

将值为 1 的 8 位常量连接到加法器的第二个输入。您可以在 Wirng 库中找到常数电路元素。要将其值更改为 1，只需为 Value 属性键入 1 并按 Enter 键。您现在应该看到该值为 0x1

（Logisim 会自动将您输入的十进制值转换为十六进制）。

将两个输出引脚连接到电路，以便您可以监控加法器和寄存器的输出。

常量（注意位数要统一!）



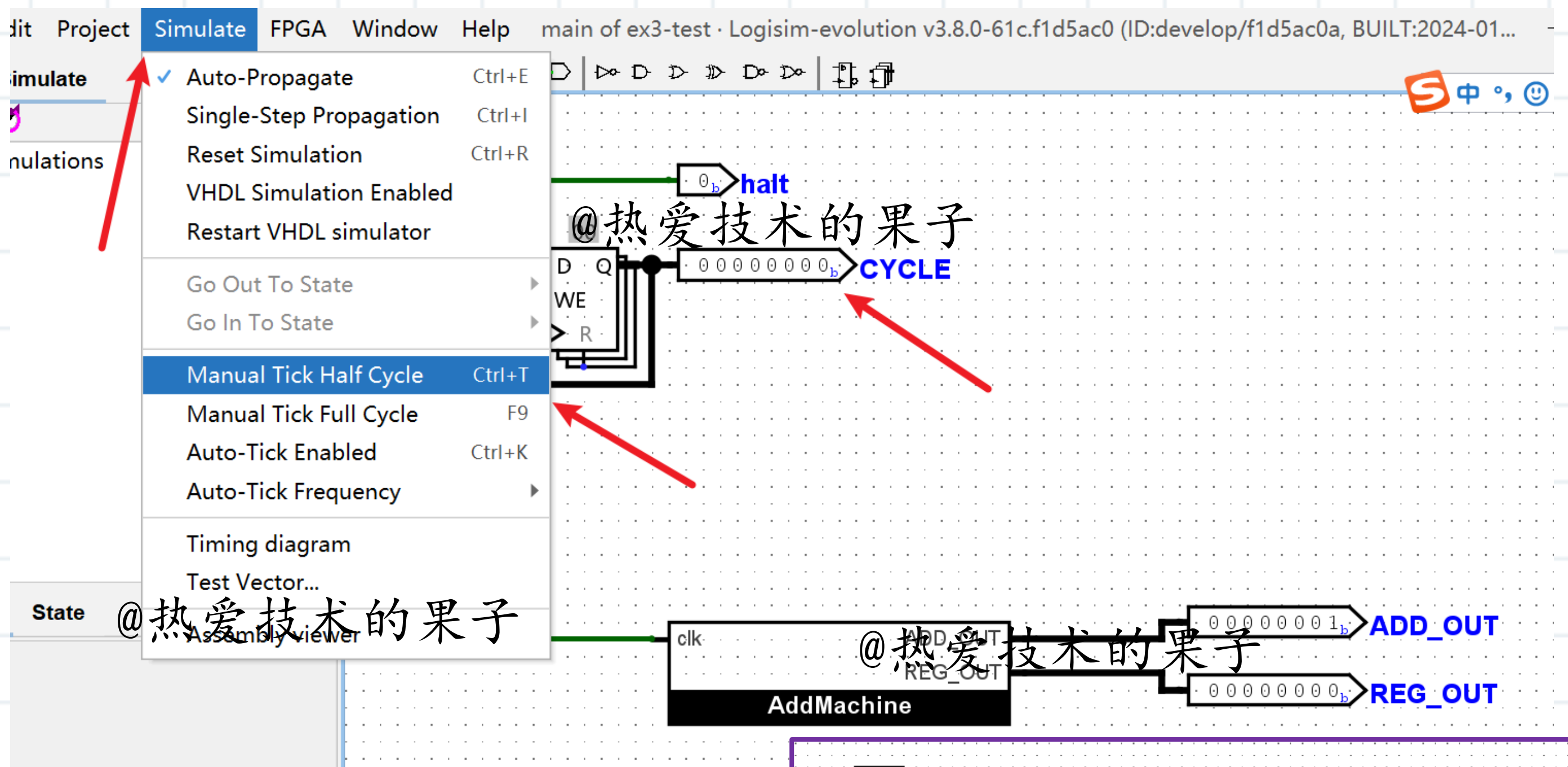
必须要有时钟

思考：如何生成时间信息触发寄存器运行，从而看到累加效果？

# 模拟调试 ☆

使用Simulate -> Tick Half Cycle (Command/Control + T)，勾选一次就能发现寄存器值发生了一次变化。如果您再次勾选半循环，您会注意到时钟变回深绿色，但数字没有变化。这是因为默认情况下，寄存器是上升沿触发的（即在从低时钟信号变为高信号时触发）。

当然如果使用Simulate -> ticks enabled/enable clock ticks (command/control + k) 会自动不断触发时间信号，可以看到寄存器一直累加！



电路图：

