

基于石墙控制线的存储器内的 微时序特性分析及应用

@NKID00*

摘 要 本文从一种基于石墙控制线的存储器着手, 通过分析其中使用的微时序特性, 提出了一种可能的微时序特性应用方式.

关键词 基于石墙控制线的存储器 微时序 特性应用

0 引言

基于石墙控制线的存储器 (Wall-based storage, 简称石墙存储器) 是 Minecraft 中的一种新兴的高密度数电存储器.^[1] 存储器的性能由输入输出时必须等待的时延决定. 图1是一种石墙存储器的设计方案的一个单元, 其中输入和输出部分均使用了微时序特性使得比较器能够被侦测器激活, 进而节省了 2 游戏刻 (game tick, Minecraft 中的理论最小时间量子, 在 20 TPS 时等同于 0.05 秒) 的时延, 但同时也带来了输出使能信号无法被正常长度的 (直接由侦测器产生的) 2 游戏刻脉冲激活的问题. 将该石墙存储器单元内各元件进行标注, 以便分析. 注意侦测器 B 和侦测器 C 是两个上下堆叠的不同侦测器.

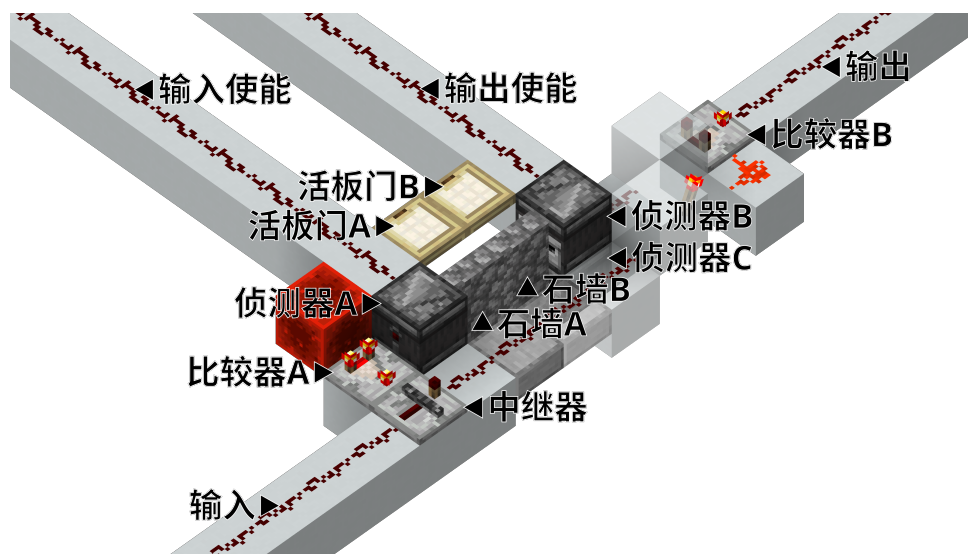


图 1: 石墙存储器及标注

1 分析

在 Minecraft 中¹, 拥有延迟的红石元件 (除活塞外) 都会在接收到红石信号或方块更新时判断并为自己所在的方块位置添加 (schedule, 也可译为安排) 一个对应延迟后的计划刻

*作者邮箱 NKID00@pm.me

¹本文所述内容均基于 Minecraft: Java Edition 1.17.1 版本, 但除石墙电路相关结论仅适用于 Minecraft: Java Edition 1.16.x-1.17.x 版本外^[2], 其他相关结论也应同样适用于 Minecraft: Java Edition 1.13.x-1.17.x 版本.^[3]

(tile tick), 等计划刻被执行时再改变自身状态. 同一位置的方块在每个游戏刻内只能拥有一个计划刻, 重复安排的计划刻会被忽略. 在同一游戏刻内的计划刻先按照优先级 (priority) 数值从小到大的顺序依次执行, 优先级相同的计划刻再按照添加顺序执行. 需要注意的是, 计划刻并没有附加数据的功能. 红石元件执行计划刻时仅通过当前状态来判断要改变为目标状态. 例如熄灭的中继器执行计划刻时就会亮起, 亮起的中继器执行计划刻时就会熄灭. 但比较器等元件 (不包括中继器) 在执行计划刻并改变状态前会先判断当前是否应该改变状态, 如果此时不满足改变状态的条件, 计划刻就会被忽略. [4] [5]

先对 2 游戏刻脉冲激活输入控制时钟信号线时的情况进行微时序分析, 如图2. 品红色线条代表元件被激活的时刻. 标有星号的一处微时序步骤是至关重要的. 在 Minecraft 中, 指向红石二极管 (`net.minecraft.block.AbstractRedstoneGateBlock` 类 [5]), 简称二极管, 包括红石中继器和红石比较器) 且该二极管非反向对顶的比较器遇到能量变化时, 会向自己添加优先级为-1 的计划刻. [4] [5] 该计划刻会比优先级为 0 的侦测器 A 熄灭的计划刻更早执行, 因此在比较器 A 判断是否应该被激活时侦测器 A 尚未熄灭, 比较器 A 能够被从侧面激活 (即被熄灭).

然而, 如图3对 2 游戏刻脉冲激活输出控制时钟信号线时的情况进行微时序分析, 可得知比较器 B 将不可能被激活. 比较器 B 并未指向二极管, 只能添加优先级为 0 的计划刻. 这个计划刻的添加时刻又晚于侦测器 B 和侦测器 C 的计划刻 (标有星号的两处), 因此会在侦测器 B 和侦测器 C 的计划刻执行后 (即熄灭后) 再执行. 此时比较器 B 判断没有满足应该被激活的条件, 不能够被激活.

在不延长脉冲至 3 游戏刻 (即只使用延长的 2 游戏刻脉冲) 的条件下能够使比较器 B 能被激活, 微时序分析如图4. 比较器 B 虽然只能添加优先级为 0 的计划刻, 但这个计划刻的添加时刻恰好位于侦测器 B 和侦测器 C 的计划刻 (标有星号的两处) 之间, 因此会在侦测器 B 和侦测器 C 的计划刻之间执行. 此时侦测器 B 已经熄灭, 但侦测器 C 尚未熄灭, 比较器 B 满足应该被激活的条件, 能够被激活.

2 应用

延长 2 游戏刻脉冲的一种方法是使用如图5所示的利用了微时序特性的信号延长器. 该信号延长器的微时序分析如图6.

需要注意的是, 这种信号延长器因为利用了中继器和侦测器计划刻优先级的差异, 不能延长经过了单独中继器的脉冲. 接受经过了单独中继器的 2 游戏刻脉冲时, 该信号延长器的微时序分析如图7. 这两种情况的关键区别在于输入信号熄灭的计划刻优先级不同, 导致了侦测器的计划刻与下游的计划刻 (标有星号的两处) 添加顺序不同, 接着导致了下游的计划刻执行时输出信号是否处于激活状态的差异, 最终导致下游计划刻执行成功或失败. 因此, 在使用图1所示石墙存储器且必须使用 2 游戏刻脉冲激活输出时钟信号线时, 一种可行的方法是利用侦测器等计划刻优先级不低于 0 的元件生成脉冲并采用多个图5所示的信号延长器串联进行脉冲中继和延长.

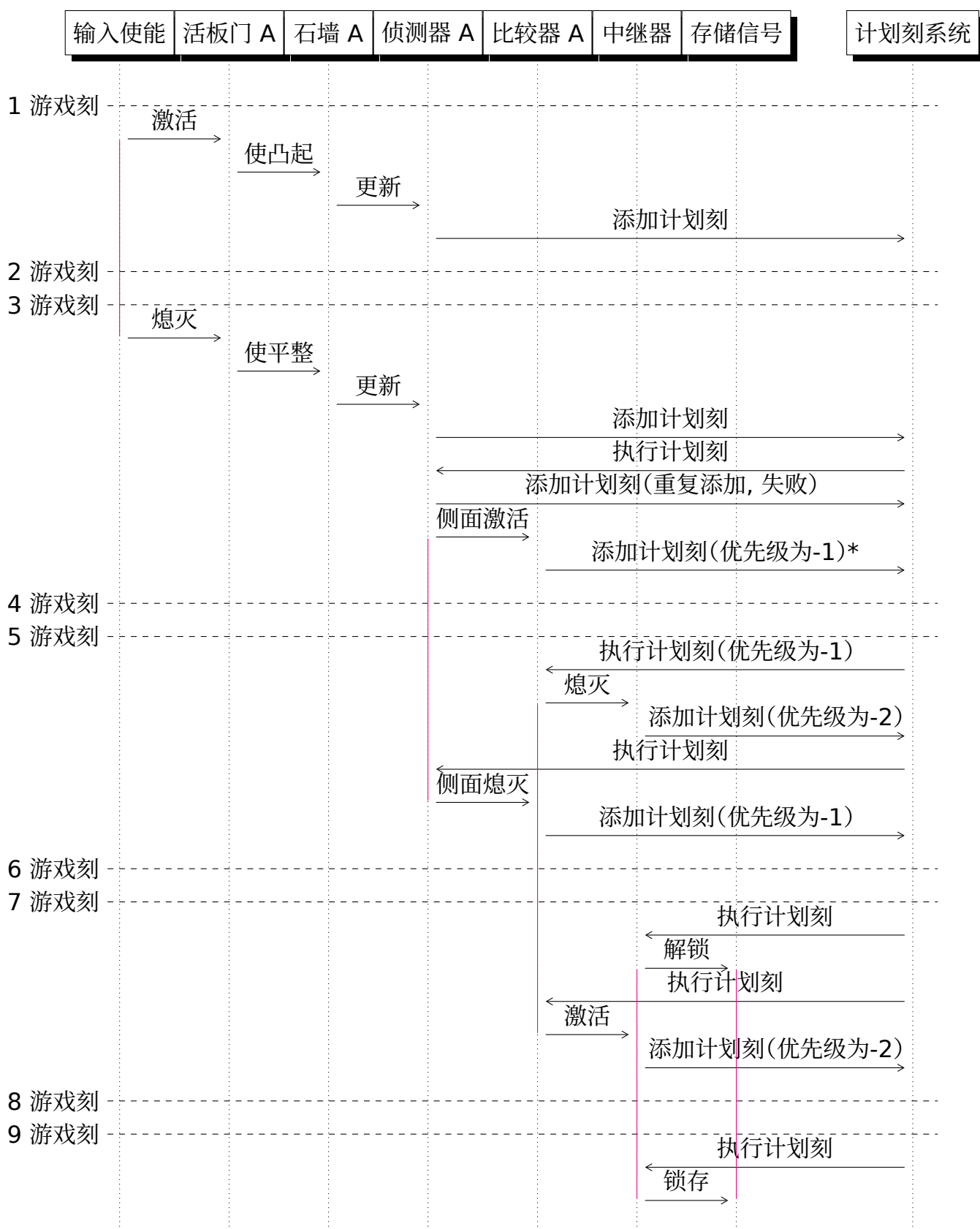


图 2: 2 游戏刻脉冲激活输入控制时钟信号线时各元件的微时序状态

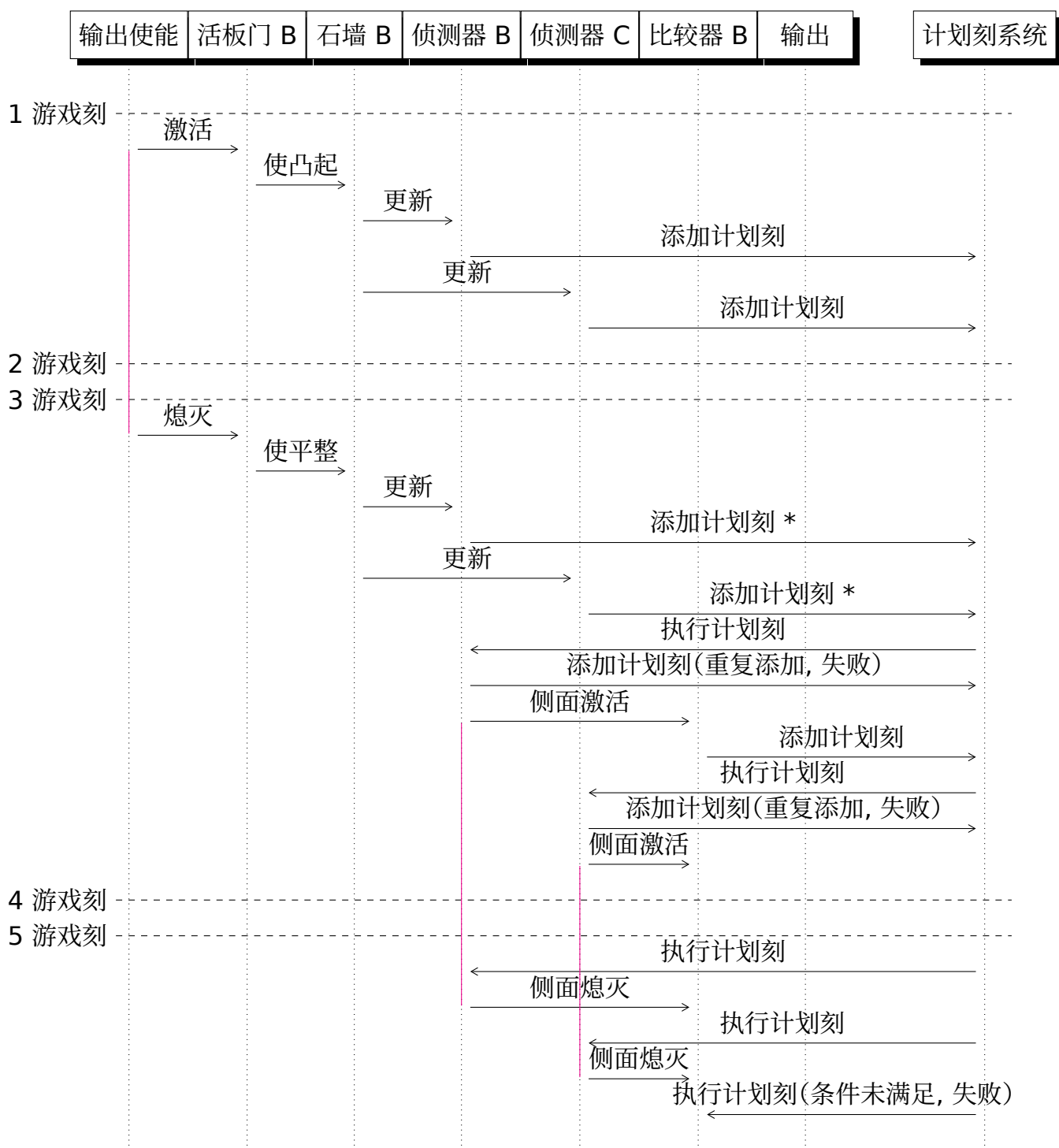


图 3: 2 游戏刻脉冲激活输出控制时钟信号线时各元件的微时序状态

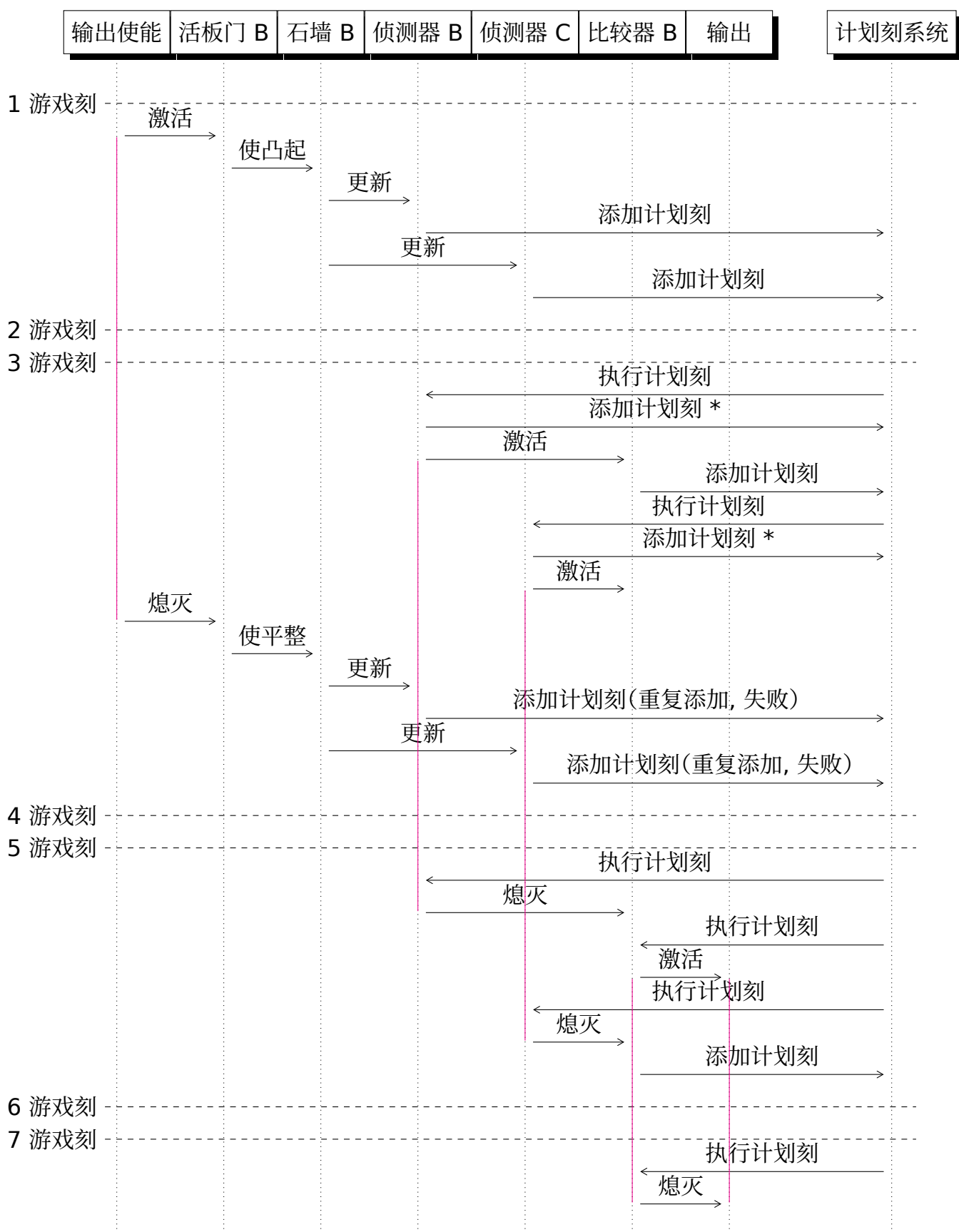


图 4: 延长的 2 游戏刻脉冲激活输出控制时钟信号线时各元件的微时序状态

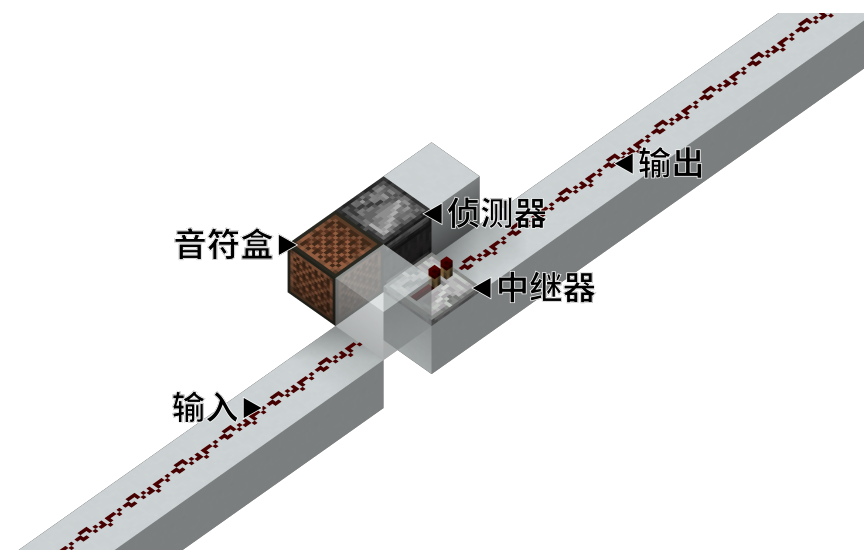


图 5: 一种信号延长器的设计方案

3 结语

图1所示石墙存储器在减少时延的同时因使用了微时序特性而带来了不稳定性. 而这种不稳定性可以通过进一步利用微时序特性来消除. 合理运用微时序特性能在一些方面有效提高红石装置的性能, 许多微时序特性也是多版本通用的. 微时序不只是令人感到疑惑的一种特性, 也能在某些领域发挥它的作用.

主要参考文献

- [1] 辰占鳌头. 基于石墙电路的随机存储器 [J/OL]. 红石数电评论, 2021, 10.
- [2] 森森. 关于“墙”在 Minecraft 红石数字电路中的应用 [J/OL]. 红色的粉末, 2020. <https://zhuanlan.zhihu.com/p/375262793>.
- [3] Fallen_Breath. 计划刻 [J/OL]. 深度剖析 Minecraft, 2020, 3. https://fallenbreath.me/2020/09/16/deeply-dissecting-minecraft_3/.
- [4] Fallen_Breath. 计划刻元件 [J/OL]. 深度剖析 Minecraft, 2020, 3.4. https://fallenbreath.me/2020/09/17/deeply-dissecting-minecraft_3.4/.
- [5] yarn[CP/OL]. 版本 1.17.1+build.63. <https://github.com/FabricMC/yarn/actions/runs/3974649221>.

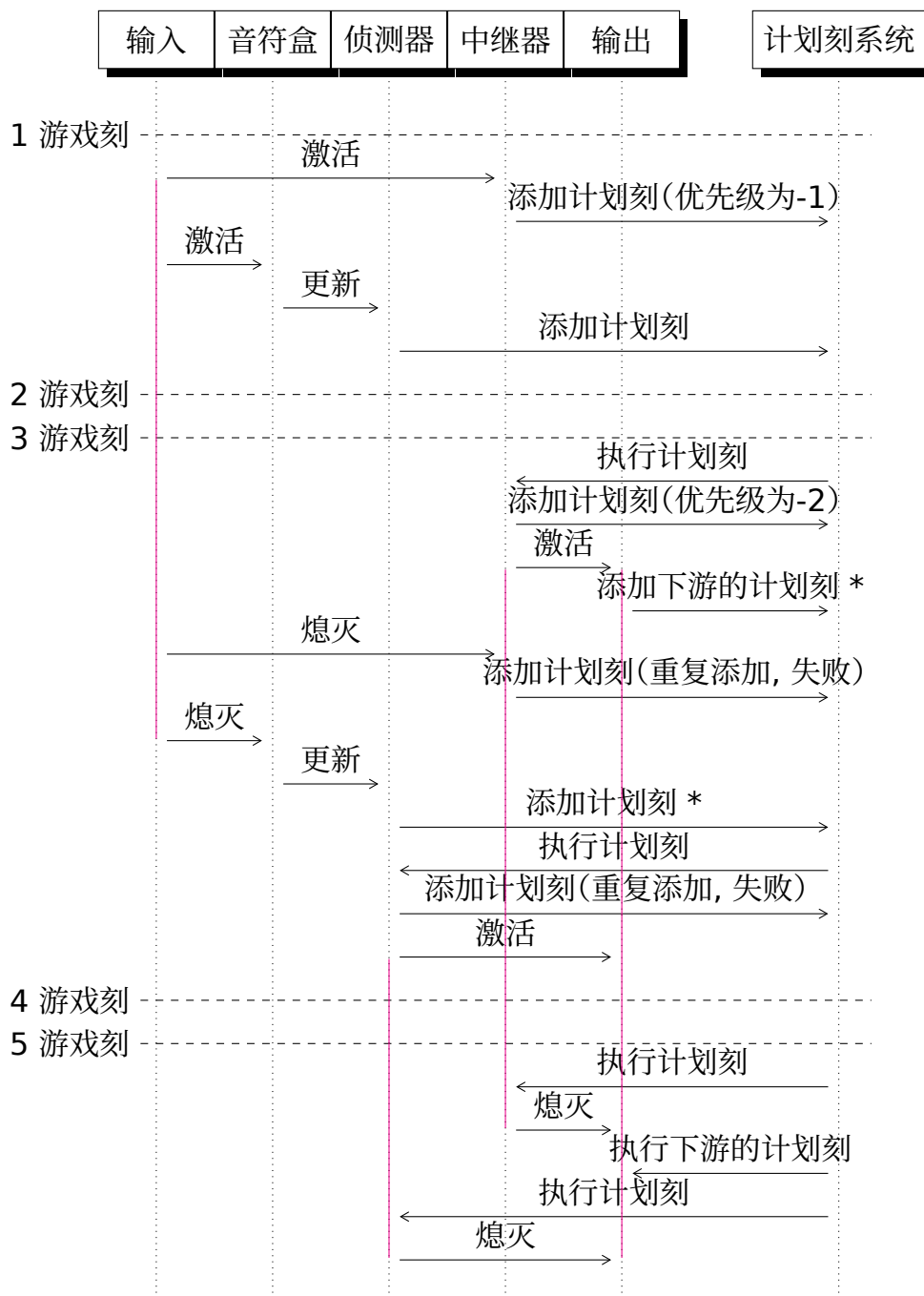


图 6: 信号延长器接受来自侦测器的 2 游戏刻脉冲时各元件的微时序状态

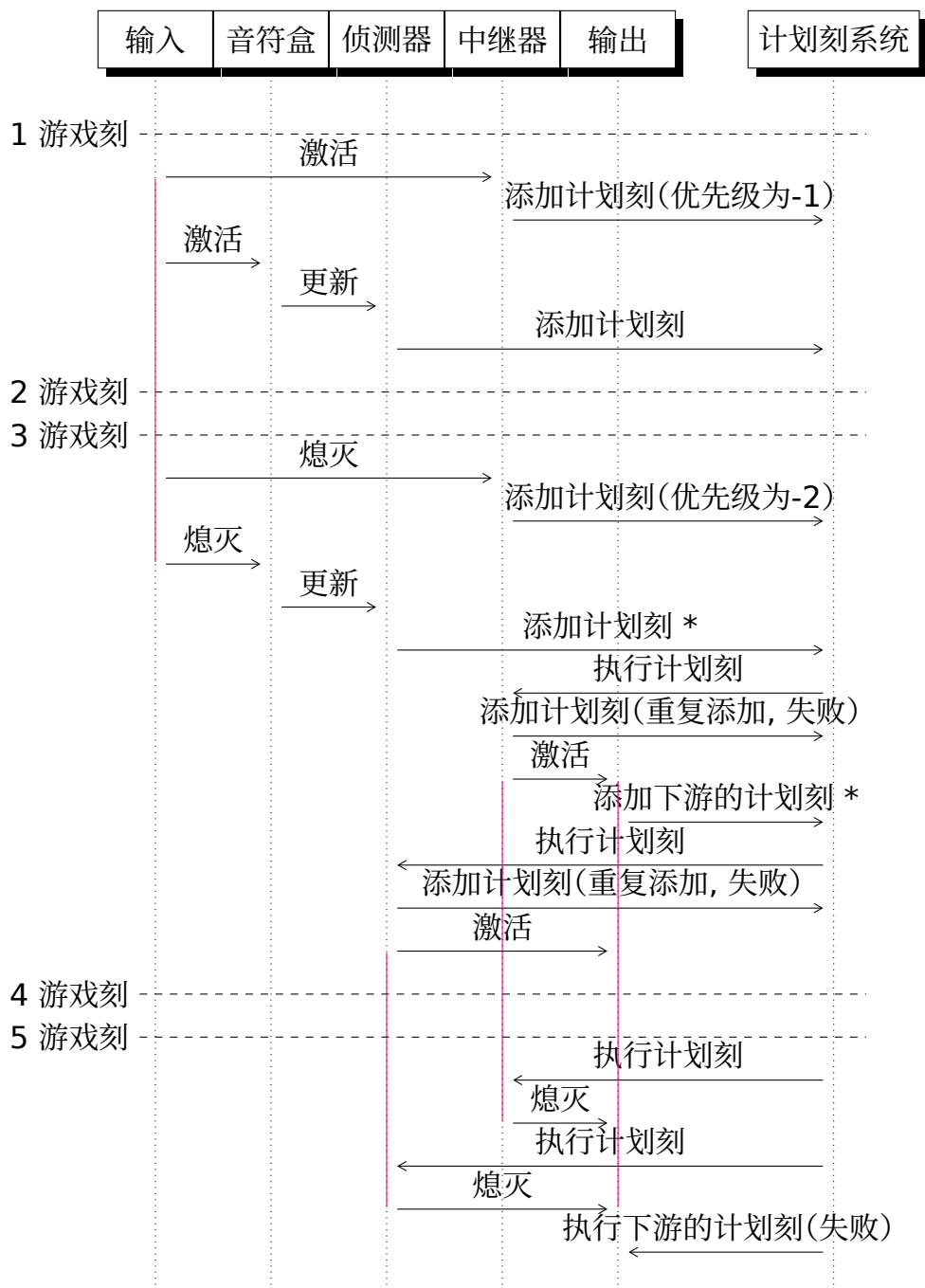


图 7: 信号延长器接受经过了单独中继器的 2 游戏刻脉冲时各元件的微时序状态