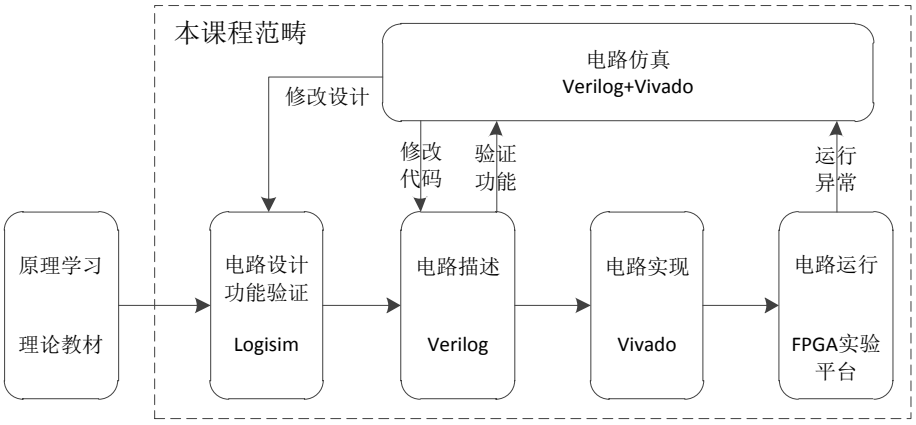
实验 09 竞争冒险及流水线技术

简介



通过前面实验的学习，读者应该已经具备熟练设计组合逻辑电路和时序逻辑电路的能力。不难发现，绝大部分的电路都是同步时序逻辑电路，几乎找不到一款单纯由组合逻辑构成的集成电路，这到底是什么原因呢？本次实验中，我们将从时序相关的几个基本概念入手，解释同步时序逻辑电路的优势，并着重介绍一种提高时序逻辑电路性能的技术——流水线设计。

实验目的

掌握电路时序相关的基本概念

了解同步时序逻辑的优势

掌握流水线的设计方法

掌握流水线发生异常时的处理技巧

实验环境

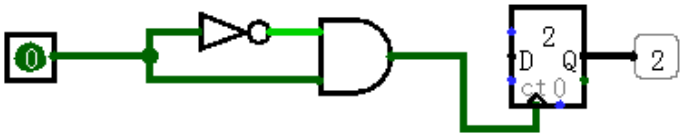
VLAB:vlab.ustc.edu.cn

Logisim

实验步骤

Step1. 组合逻辑电路中的毛刺及竞争冒险

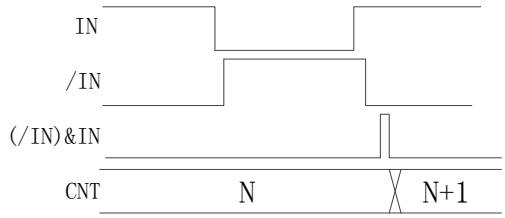
在 Logisim 中搭建如下所示的电路图，包含非门、两输入与门、计数器各一个。



分析电路可以发现，与门的两个输入是始终相反的一对信号，理论上来说，其输出应始终为 0，因此计数器的时钟端口信号应始终为 0，计数器不会进行计数。但实际情况并不是这样，通过鼠标改变输入信号值的时候，计数器在不停计数，这说明计数器的时钟端检测到了电平的变化，这种现象与电路的延时有关系。

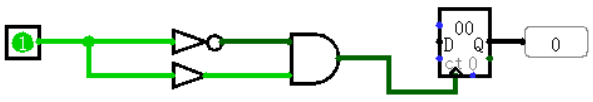
任何逻辑门以及导线传输信号都存在延时，逻辑门的延时要比导线大得多，具体的延时大小由逻辑门的复杂程度和实现工艺有关，以现在的工艺水平一般都在 ps~ns 量级。如下图所示，输入信号变化后，输出信号不会立即变化，而是会有一定时间的延时。

实际上，上图中所展示的也只是一个理想化的模型，在实际电路中，由于寄生电容等因素，信号的变化时连续的，而不是突变,对于这个问题我们此处不做进一步讨论。考虑到逻辑门的传输延时后，前面电路图中信号变化波形图将如下图所示



可以看到，在每次输入信号（IN）由 0 变为 1 时，计数器的时钟端都会出现一个不应当出现的脉冲，使得计数器产生计数。我们将这种由于逻辑门延时产生的不正常的信号波动，称为“毛刺”。同一输入信号通过两条或两条以上的途径传到输出端，由于每条途径延时不同，到达输出端的时间有先有后，这种现象称为“竞争”。不会对电路正确性产生影响的竞争，称为非临界竞争，会引起电路发生瞬间或永久输出错误的，称为临界竞争。由于竞争而可能产生输出干扰脉冲（毛刺）的现象称为“冒险”。竞争是因，冒险是果，由于电路存在竞争，才会导致冒险现象的发生。除了同一信号的竞争冒险会产生毛刺外，多个输入信号同时变化也会在输出端产生毛刺。

在 Logisim 中修改上述电路，在另一分支上加入缓冲器，如下图所示，再次进行仿真，无论输入端如何变化，计数器都不再计数。这是因为在 Logisim 中，认为两个路径的延时相同，不会产生竞争，但实际电路中，逻辑门和导线都会产生延时，基本不可能做到两条路径延时完全相等，因此竞争必然存在，只不过产生的毛刺更细，持续时间更短而已。



通过分析上述电路可以发现，只所以会产生毛刺，是因为逻辑门

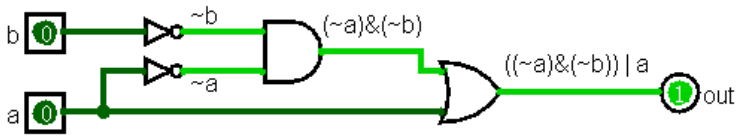
的多个输入端同时变化引起的(例如前面电路中与门的两个输入端)，如果可以修改电路设计，使其输入端信号不会同时变化，便有可能消除毛刺（但很多实际情况下还是无法彻底消除）。

单个输入信号的变化也能引发输出端出现毛刺，如下图所示，左侧为真值表、右侧为卡诺图，如按照该卡诺图的框选办法，其逻辑表达式可写为：out = ~b ~a + a

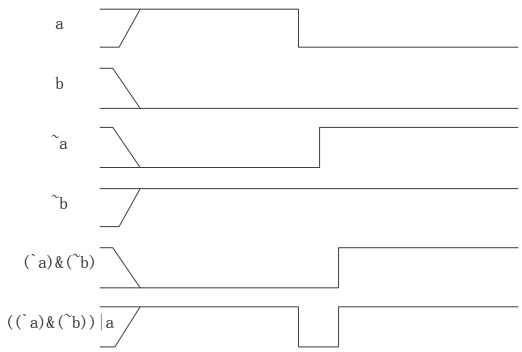
b	a	out
0	0	1
0	1	1
1	0	0
1	1	1

b \ a	0	1
0	1	1
1	0	1

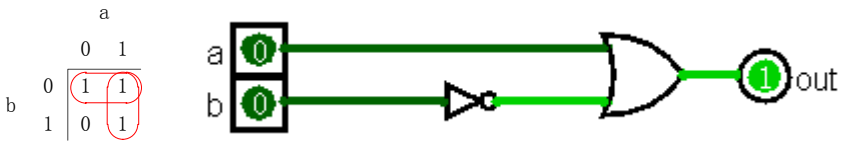
对应的电路图如下所示



当输入信号由“a=1, b=0”变为“a=0, b=0”时，其输出应一直保持为 1，但考虑到器件延时，输出端会产生一个负向脉冲，如下图所示。

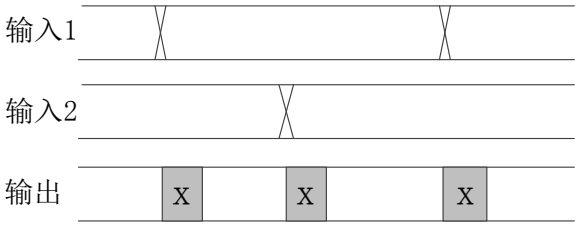


如果我们修改卡诺图的框选方式，如下图所示



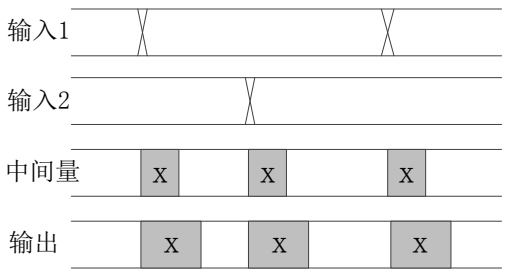
可以很容易的看出，对于这种电路，当输入从“a=1, b=0”变为“a=0, b=0”时，就不会产生毛刺。

Step2. 同步时序逻辑的优势

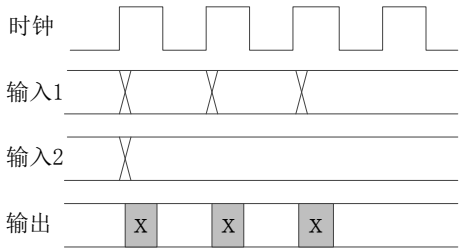


对于纯组合逻辑电路，每当有输入信号变化时，输出信号端产生竞争，导致暂时性的输出结果不可靠，我们称为不确定状态，用“X”来标识，如上图所示。可以看出，不确定状态的长短由各信号路径的延时差决定，为缩短这种不确定状态，有两种办法：缩短信号传输延时时间（提高半导体工艺，或简化逻辑复杂度以减少逻辑门级联），尽量使所有路径的延时时间相同(在较短延时路径上插入延时逻辑)。另外，如果输入端信号可在任意时刻改变的话，我们就无法得知输出端信号什么时间可靠（非 X 状态），因此我们需要知道输入端信号何时变化。此外，为了减少 X 状态的出现频率，最好所有的输入信号同时改变。

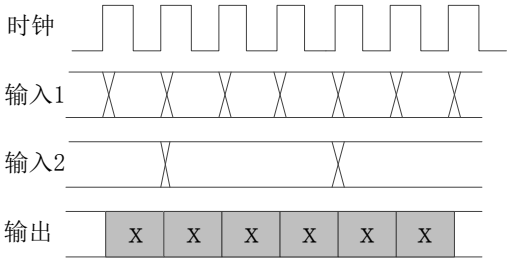
如果逻辑功能比较复杂，输出信号又是后续电路的输入信号，那这种不确定状态会继续传递下去，如下图所示，情况严重的话，输出信号可能在任意时刻都是不可靠的。



为避免纯组合逻辑电路的上述问题，我们引入时钟信号，所有的信号统一在时钟边沿（上升沿）同步变换，这样输出端的 X 状态便确定下来，始终处于时钟边沿之后的一段时间。电路可在下一个时钟信号的上升沿读取输出信号，这便是同步时序逻辑电路。



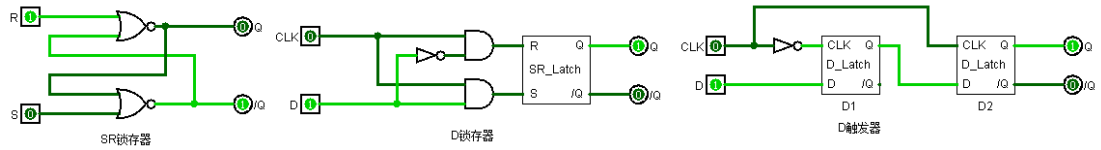
如果时钟信号变化的太快，或者不确定状态持续时间太长，在下一个时钟上升沿到来的时候，输出还处于不确定状态，那就可能输出一个错误的数据，最终导致电路工作出错。因此，同步时序逻辑电路的时钟频率不能无限增加，它受其内部组合逻辑延时最大部分的限制。



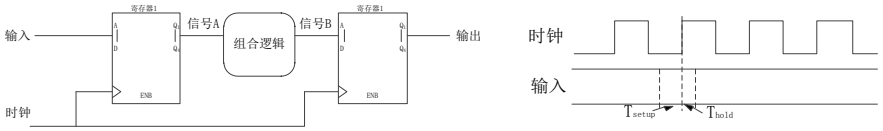
综上所述，同步时序逻辑相对于组合逻辑来说，具有非常明显的优势，使其更适合用于实现较大规模的功能电路。首先，由于采用的统一的时钟作为同步，其信号改变的时间点是确定的，其不确定状态出现的时间点也是确定的，电路更加清晰。其次，由于各信号只在时

钟信号边沿被采样，在其它时刻受到干扰不会对结果产生影响，电路抗干扰能力更强。输出信号的不确定状态不会叠加到下一级，有利于构建更加复杂更大规模的电路系统。

Step3.亚稳态和建立保持时间



我们知道，同步时序逻辑电路的核心器件是 D 触发器，其本质上也是有组合逻辑门电路构成的，但其电路结构决定了时钟信号本身不会使触发器的输出信号产生冒险或毛刺，但如果时钟信号与其它信号同时变化的话，其电路行为就不可控了，有可能产生不确定状态，甚至产生一种非零非一的亚稳态。因此，应当确保在时钟信号的边沿，其它所有相关信号都应当处于确定的状态，并保持不变。



信号在时钟上升沿之前保持稳定的最小持续时间称为建立时间（图中 T_{setup} 所示），在时钟上升沿之后保持稳定的最小持续时间称为保持时间（图中 T_{hold} 所示），建立时间和保持时间是触发器（或寄存器）的固有属性，与外部的组合逻辑延时无关。

上图是一个典型的同步时序电路模型，为使所有的触发器（或寄存器）都能够满足建立时间和保持时间的时序要求，则应当满足以下公式：

$$T_{时钟周期} + T_{时钟信号传输延时} > T_{hold} + T_{组合逻辑最大延时} + T_{setup}$$

其中 $T_{时钟信号传输延时}$指的是时钟信号到达两个级联寄存器（图中的 1,2 号寄存器）的时间差，相比于组合逻辑延时来说，这段延时很小，为简化模型，我们将其忽略不计，则得到如下公式

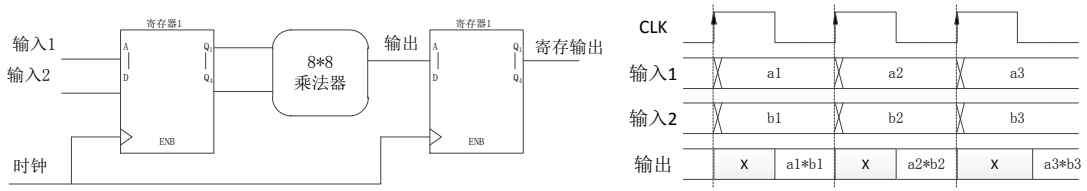
$$T_{时钟周期} > T_{hold} + T_{组合逻辑最大延时} + T_{setup}$$

建立保持时间是寄存器的固有属性，用户无法修改，组合逻辑最大延时则与用户的具体设计有关。因此，在相同工艺下，电路可正常工作的最大时钟频率与目标电路的组合逻辑最大延时有关。

Step4. 提高电路性能的几种手段

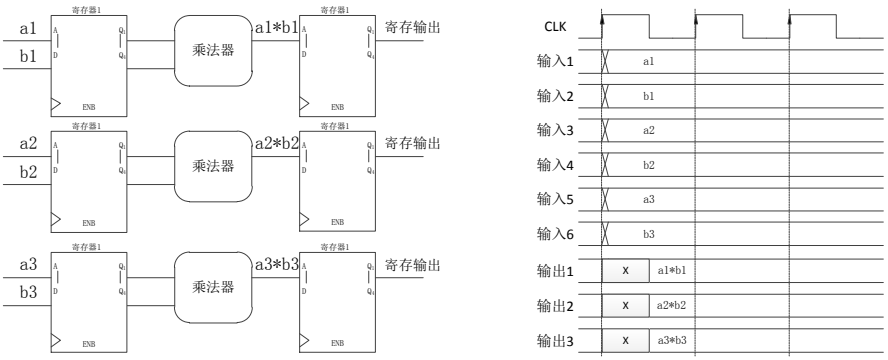
考虑如下问题：a1,a2,a3,b1,b2,b3 都是 4bit 位宽的信号，现要求出 a1*b1、a2*b2、a3*b3 三组乘法的结果,应如何实现？

第一种方式，可以通过一个 8*8 的乘法器通过来实现，如下图所示，这种方式使用的器件数量最少，电路面积也最小，但电路速度不快，耗时最长，

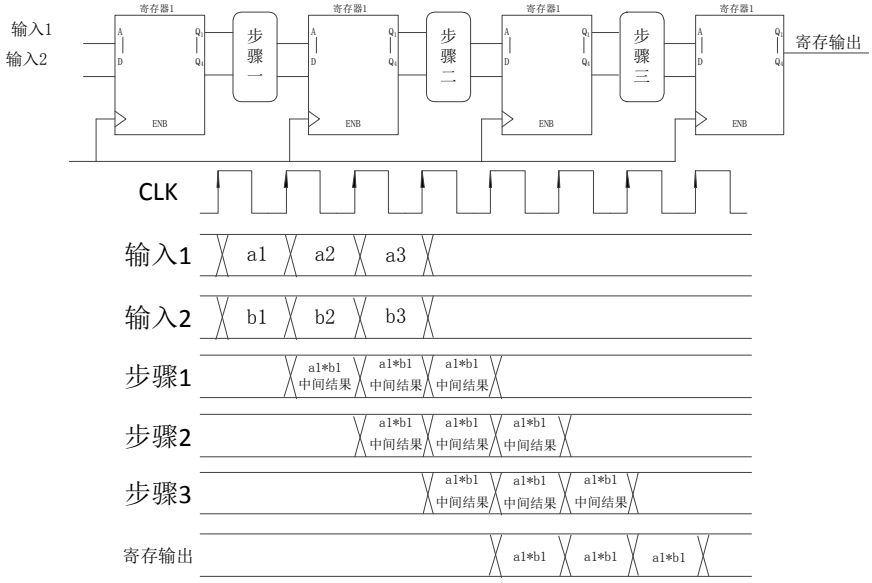


第二种方式，通过提高器件工艺，可以降低组合逻辑的延时，以及寄存器固有的建立保持时间，从而可以提高时钟频率，最终降低所用时间，提升电路性能。

第三种方式,可以增加功能器件数量,在电路中使用三个乘法器，如下图所示，对比第一种方式，电路计算能力提升了约三倍，但电路面积也相应增加了，这是一种典型的以空间换时间的方法，在数字电路系统的设计中经常用到。



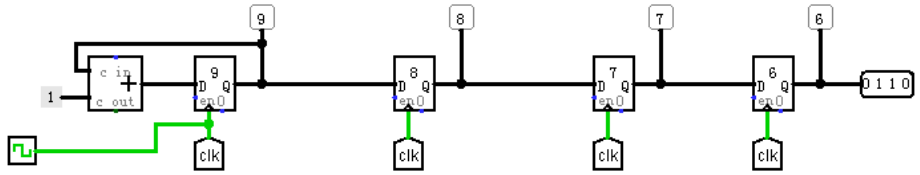
乘法运算是一项比较耗时的工作，想要实现一个 4*4 的乘法运算单元，消耗的逻辑门数量也相当可观，我们知道组合逻辑的延时与逻辑门的级联数量直接相关，逻辑越复杂，级联越多，延时就越大。乘法器的延时成为提升时钟频率最大的限制因素。如果能把乘法运算拆成多个步骤，并分摊到不同的时钟周期完成，便可以降低相邻两级寄存器间的逻辑复杂度，从而提高电路的最大工作频率。



通过把复杂的逻辑分解到不同时钟周期，可以降低组合逻辑延时，极大的提高电路最高工作频率。上图中，虽然每个乘法运算需要花三个时钟周期，但实际上每个周期都有一个结果输出。从电路的处理能力来看，可以认为是每个周期可以完成一个乘法运算的结果，并且由于组合逻辑的拆分，最高工作频率能够得到大幅提升，最终使电路性

能得到明显提升。这种通过将较复杂的组合逻辑拆分成多个步骤，在多个时钟周期完成，从而提升最高工作频率，增大系统吞吐量的做法，称为流水线技术。流水线技术是提升电路性能的一种重要技术手段。

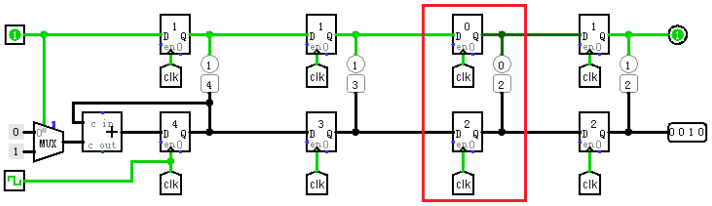
Step5. 理想流水线示例



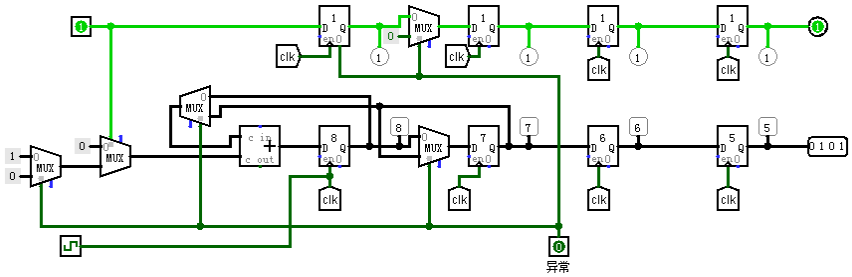
如上图所示，为在 Logisim 中搭建的一个理想的三级流水线，最左侧为一 4bit 循环计数器，从 0~15 循环计数，后面级联了三级寄存器，可以看到，计数值在每个时钟周期都会向后移动一级。每个计数值都在三个时钟周期之后出现在最后一级寄存器。理想流水线正常工作时，每个时钟周期都会输出一组数据，中间不会有阻塞或中断

Step6. 带有冲突检测的流水线

在实际使用流水线时，难免会遇到各种异常状况，导致流水线阻塞、中断或清空。而且我们知道，各级流水线都对应整个数据处理流程中的某个个环节，各环节出现的问题都不尽相同，电路的相应也有所不同。如下图所示，我们为计数器增加使能端，当使能为 0 时计数器不再计数，此时整个流水线应处于阻塞状态，并应当有专门的信号来标识输出数据是否有效。每一级流水线都有对应的数据有效信号，当有效信号为 1 时，对应数据有效，否则无效，流水线被阻塞。



当流水线中间级出现异常情况时，需要特别注意，流水线内的数据可分为两种，一种是在该级流水线之前的数据，另一种是尚未到达该级流水线的数据，对于前一种数据，应按照正常流程继续处理，对于后一种数据则应该从流水线清除，并从发生异常的数据开始重新开始。如下图所示，在流水线中进一步加入了异常处理逻辑，当异常信号有效时，会清空流水线前半段的内容，异常信号撤销后，计数逻辑会从发生异常的数值重新开始计数，在输出端观察，有效信号标示出的是连续的计数结果。



在多级流水线中，每级流水线有可能发生异常，对于不同的功能电路，其应对策略也会有所不同，所以流水线中的异常处理逻辑是重点也是难点，需要读者特别注意。

实验练习

题目 1 在 Logisim 中使用若干加法器及必要的附加电路搭建出一个纯组合逻辑实现的 4*4 乘法器，两个输入信号均为 4bit，输出结果为 8bit。

题目 2 修改题目一中的电路，插入必要的寄存器，将其改造成一个多级流水线电路（推荐使用 3 级流水线），写出 Verilog 代码，并对其进行仿真。

题目 3 将 10 组乘数与被乘数存放在一个 ROM 中，每个时钟周期读取

一组数据，使用题目 2 设计的三级流水线进行计算，要求加入异常处理逻辑，在第二级流水线发生异常时，第一、二级流水线清空，第三级数据正常输出，并重新从 ROM 读取数据，最终完成所有数据的乘法运算，输出数据用有效标志信号标记。

总结与思考

1.请总结本次实验的收获

2.请评价本次实验的难易程度

3.请评价本次实验的任务量

4.请为本次实验提供改进建议