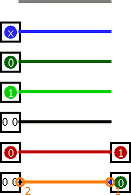
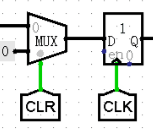
P0课下作业个人小结

1. 浮空值、高阻值与未知值。
   1. 浮空值floating在logisim中被定义为高阻值，相当于断路设计。

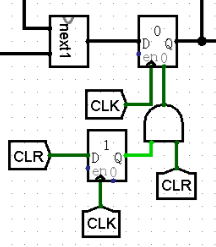


* 1. 浮空（高阻）值的官方解释：当前没有值被“载入”在这条线上。
  2. 未知值在Verilog中用x表示，而在logisim中x表示浮空值，未知值在logisim中应为Error状态。

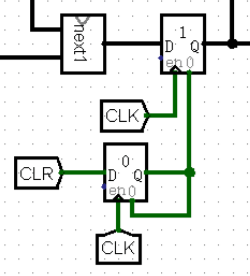
1. Logisim电路中浮空值输出方法。
   1. Disable output ：部分器件disable时输出浮空值，将即当disable=1时，若设置disable output为floating，则会输出浮空值（理解为将这条线切断了）。
   2. Three state ：如输入、分配器、译码器等有此参数配置，一般默认为No，若选择Yes，则当输出非1的时候，会输出floating浮空值。
   3. P0具体实例：GRF 通用寄存器阵列 寄存器单元更新
      1. 寄存器的EN使能接口控制寄存器能否在时钟上升沿更新，通过二进制-热值译码器确定各寄存器EN的值。Three-state = No（EN = floating情况下寄存器默认激活）, disable\_output = 0。
      2. 寄存器的输入端口用于接受更新数据，通过数据分配器将数据分配至对应的寄存器。Three-state = Yes（切断不是目标的寄存器的连接，若为No可能其他寄存器会更新0），disable\_output = floating.
      3. 注：以上1,2双保险，disable时的output设置主要考虑封装嫁接时使用。
2. 时序电路同步复位的几种方法。
   1. 选择器置默认值法。

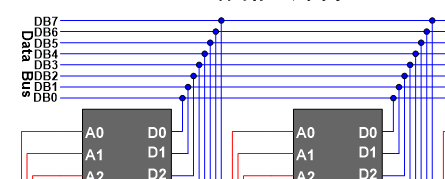


* 1. 复位信号“水匝”传递。缺点：“水匝”寄存器存在T\_oc时间，在上升沿到来后的T\_oc时间内，CLR不能由0->1，否则将提前在本次上升沿复位。（寄存器上升沿变成0期望的值是需要时间的）



* 1. Clr仅在一瞬间有效（不用step simulation则看不出来差异）



1. 阵列型logisim组件的代码级别实现。
2. GRF文件中实现指定某几个寄存器同步读和写的方法。
   1. 思考角度：寄存器接口分析法。
   2. 输入：
      1. 方法一：指定寄存器EN端口置1，其余置0.（译码器）
      2. 方法二：指定寄存器能接受到时钟上升沿变化，其余不可行。（分配器，不推荐）
      3. 方法三：指定寄存器能接收到输入的信息，其余寄存器接收到floating。（分配器）
      4. 注：三种方法均需要在WE有效时才可用，需要设置译码器/分配器的disable输出，以上三个方法可叠加使用。
   3. 输出：
      1. 方法一：寄存器输出值，多路选择器选择。
   4. 多级封装注意事项：
      1. 输出处理目前想到的方法可用多路选择器实现（多级亦可），若额外增加片选信号SC可以实现像存储器一样共享输出通路（指定的有输出，其余的高阻值）。
      2. 
      3. 写入时应采用片选信号SC（本题中称WE），信号作用部位为子电路的分配器、译码器、选择器的enable端口，并注意disable时不同需求所需要的值。
3. 超前进位ALU的注意要点
   1. 每位的进位计算电路不要手动实现，另建子电路模块，写表达式实现。
   2. 减法（逐位取反+1后按加法计算）由于参与预算为~B，所以超前进位公式不同，在子电路中要判断。
4. Moore状态机转Mealy状态机的简易方法。
   1. 将输出模块连接的电路从“寄存器之后”转移到“寄存器之前，状态转移逻辑之后”。即可提前一个周期输出期望的值，并且输出=当前状态+输入。
5. 有关Mealy状态机在相邻上升沿之间输出因输入而改变的输出次数判断。
   1. Mealy状态机，n个上升沿 = n个输入（同步时序电路的优势在于将任何时候考虑输入->最后触发沿时刻的输入），在第n个上升沿后，输入中存在的值（无论是否和第n个一样）就是第n+1个输入值的待选项（在第n+1上升沿被固定）。
   2. P0实例：字符串检测
      1. Baa 这个字符串在第3次输入才应显示1，但是采用Mealy字符机会在“固定”ba后便输出1，这是因为第三个输入恰好和第二个相同，此时改变输入，判断输出会发生变化。
      2. 这样的现象是正确的，在理解时两个角度：Mealy机基本框架+Moore机输入一样可变只是输出不显示出来。
6. Logisim阵列型电路连接心得
   1. **核心思想**：提前细致谋划、demo先行验证、可移植性减少重复工作（保证源器件设置100%正确）。
   2. 分配器、选择器、译码器上的标签“重叠”标签连接
      1. 无需wire连线，直接端口相接。
      2. 当前空的端口，用上一个复制粘贴（因为相邻上一个粘贴自动生成的位置合适）。
      3. 连接完后，选择整体，“扯”出wire连线，查看是否有错位。
      4. 可用temp\_0,temp\_1,xxxx,temp\_31作为临时命名标签，而后在文件中批量替换，减少工作量。
   3. 阵列编号后需要检查的地方：
      1. 每个阵列单元的管道的命名。
      2. 选择器、分配器、译码器是否线错位。
7. Logisim源文件该电路心得
   1. 不建议通过C程序生成阵列元器件，考虑的东西过多。
   2. 部分“高度相似分布”器件可先建立雏形，而后在源代码中批量替换。
   3. 修改完的代码的**文本不可立即关闭**，谨防代码错误logisim无法打开，无法撤销。
   4. 中文注释在代码编辑保存时可能存在问题（ex：少一个”），避免使用中文注释。
8. Multiplexer无用输入端口的处理
   1. Multiplexer若选择到无用端口只能输出浮空值，可能造成后续电路发生错误。
      1. 方法一：将无用的输入端口置成0.
      2. 方法二：用条件语句，在输出无用端口时取消enable。
      3. 方法三：用pull resister – 将浮空值输出为期望的0、1、error值。
9. 有限状态机做题步骤
   1. 明确使用Moore型或Mealy型。
   2. 确定状态总量，写出每个状态对应的关系（可能涉及多个状态机）。
   3. 画出状态转移图。
   4. 复杂时做出转移真值表（可省略）。
   5. 检查
   6. 写状态转移逻辑（尽量用表达式/真值表处理）->用splitter/mul来整合端口。
   7. 写输出转移逻辑。