

中国科学院大学网络空间安全学院

计算机组成原理研讨课

实 验 报 告

实验序号： 2 实验名称： 运算器设计

一、电路图/代码

本次实验我掌握了补码一位乘法运算的基本原理，并掌握了 Logisim 寄存器电路的使用。最后利用前面实验封装好的运算器以及平台现有运算部件，构建了一个 32 位算数运算单元 ALU。

1 32 位快速加法器

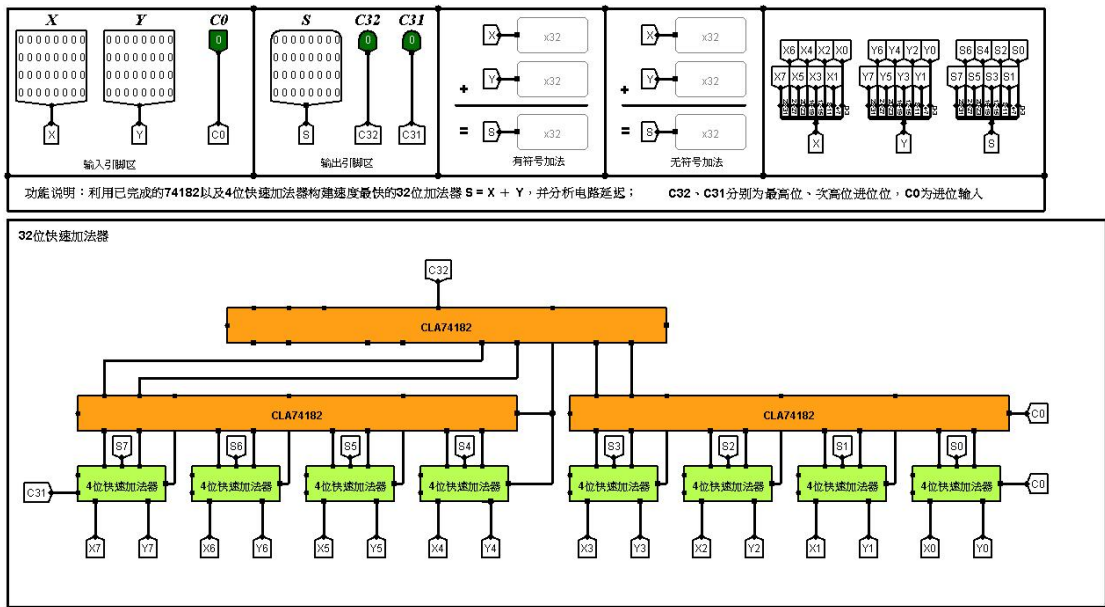


图 1 32 位快速加法器

三个方案：

2 个 16 位加法器直接串联，C16 信号采用下层的进位输出；

2 个 16 位加法器直接串联，C16 进位输入采用上层的进位输出；

在 16 位快速加法器的基础上再增加一级组间先行进位电路，类似 64 位快速加法器的方法；

第一种方案并行进位电路的延迟小于串行电路，因为其组内各位计算不依赖于低位的进位信号。第二种方案在此基础上采用了单级先行，即组内并行，组间串行。我采用的是第三种方案，多级先行，即组内并行，组间并行。如图所示，我在 16 位快速加法器的基础上再增加一级组件先行进位电路，即将 16 位快速加法器的输出 G^* 、 P^* 分别连接 $G1$ 、 $G2$ ， $P1$ 、 $P2$ 。从 $C2$ 处输出 $C32$ 。这种搭建方式更能减小延迟。同理，当计算位数增加（如 64 位）时，还可以再增加一级组建先行进位电路。总体来说它对时间复杂度的优化是对数级别的。

2 5 位无符号乘法流水线

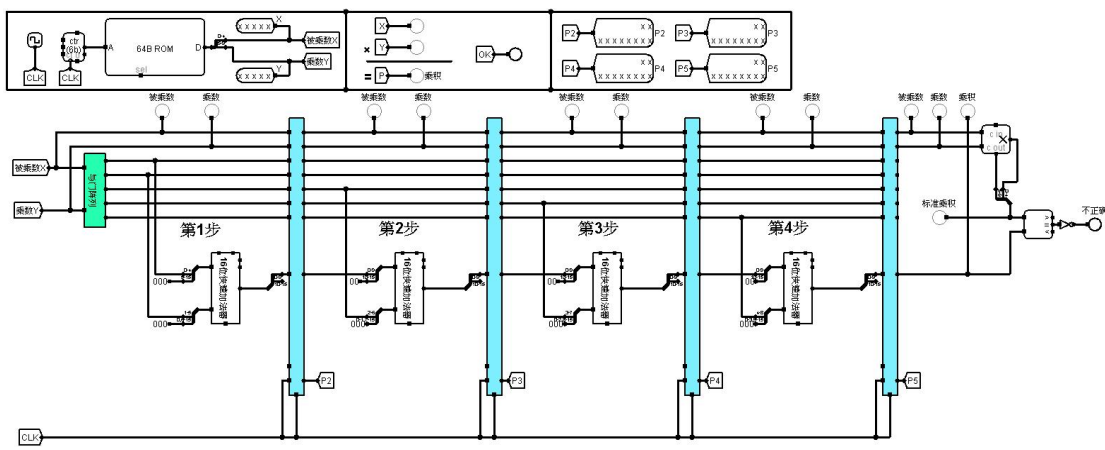
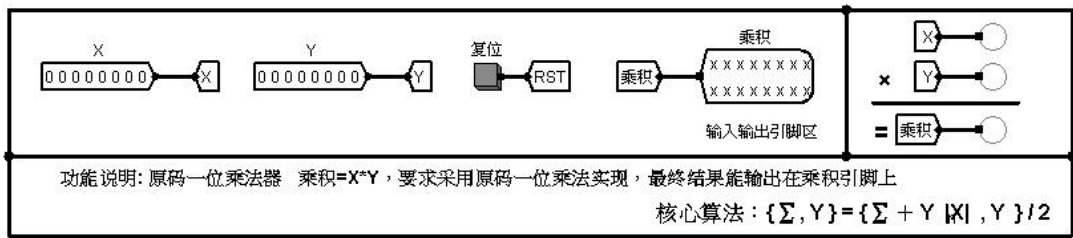


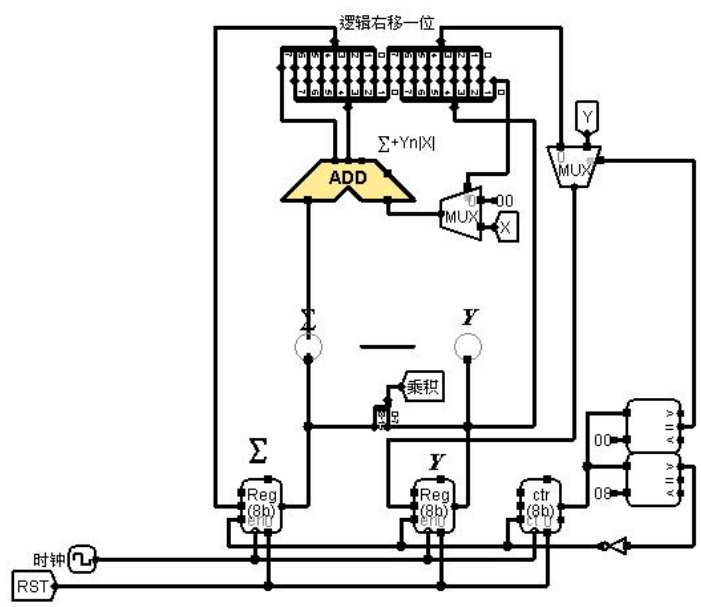
图 2 5 位无符号乘法流水线

乘法流水线将乘法细分成错位相加。判断 Y 为 0/1，对部分积做加 0/加 X 的操作，这里使用了上一关搭建的 16 位快速加法器。连接过程中要注意部分积 P 和加数 X 之间位的对齐，图中使用分线器完成移位操作。

3 原码一位乘法器



8位无符号一位乘法器



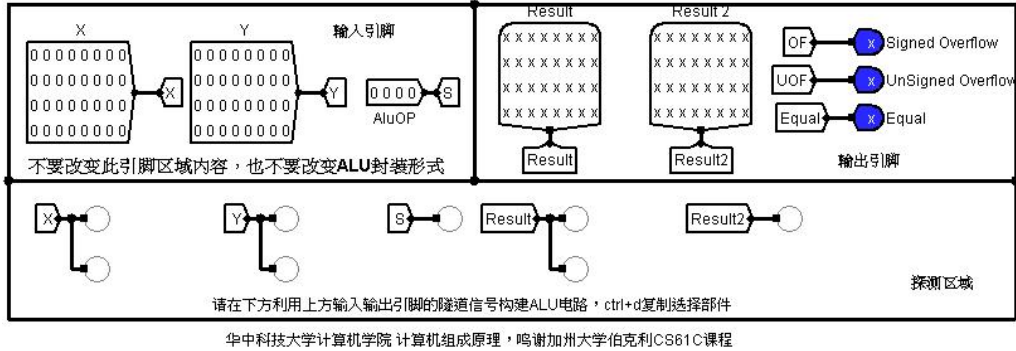
后续所有实验中凡是涉及存储器件，如寄存器、计数器、RAM等，必须增加统一的复位信号RST，方便系统复位

图 3 原码一位乘法器

该电路在收到复位信号后，将乘数 Y 存入右下寄存器中。接下来每到时钟上升沿，将 Y 与部分积相加，加法器上部通过两个分线器实现部分积和 Y 的向右移位，从而保证了错位相加的过程中位与位对齐。另外在完成 8 次相加后计算结果为 $\Sigma + Y$ 共 16 位，节省了空间。

右上有一双路选择器，计数器输出分别进行两次比较来判断是否为初始态（0）和最终态（8），比较结果分别控制 Y 的初始置位和电路自动停止。因为只有第一次时钟上升沿时需要输入 Y ，以后只需将本次运算结果返回寄存器即可，而到了第 9 个周期时 $\Sigma + Y$ 即为最终结果，应停止计算。

4 算术逻辑运算单元 ALU



32位MIPS运算器

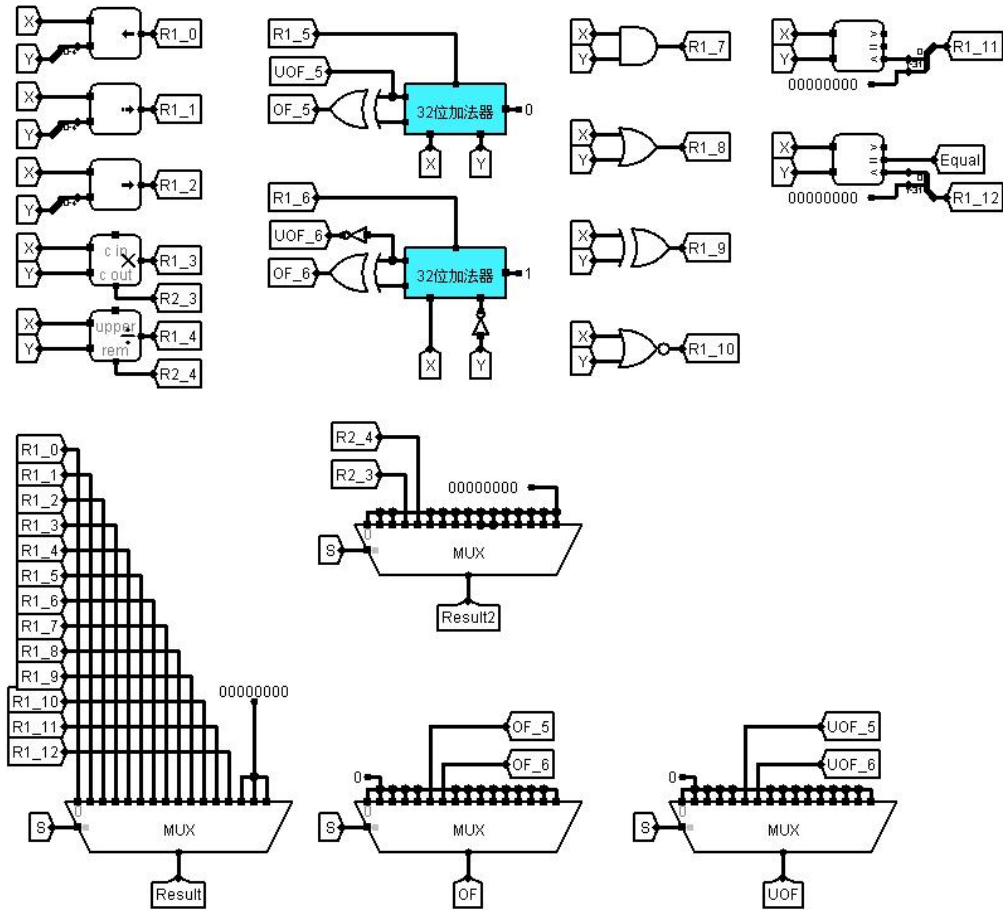


图 4 红绿灯状态机

本题我按照下表，利用隧道标签将 XY 与 12 中运算电路相连并分别得到运算结果（R1_X 等），最后连入 4 个 16 路选择器并用功能码作控制信号得到所需结果。

ALU_OP	十进制	运算功能
0000	0	Result = X << Y 逻辑左移 (Y 取低五位) Result2=0
0001	1	Result = X >>> Y 算术右移 (Y 取低五位) Result2=0
0010	2	Result = X >> Y 逻辑右移 (Y 取低五位) Result2=0
0011	3	Result = (X * Y)[31:0]; Result2 = (X * Y)[63:32] 无符号乘法
0100	4	Result = X/Y; Result2 = X%Y 无符号除法
0101	5	Result = X + Y (Set OF/UOF)
0110	6	Result = X - Y (Set OF/UOF)
0111	7	Result = X & Y 按位与
1000	8	Result = X or Y 按位或
1001	9	Result = X⊕Y 按位异或
1010	10	Result = ~(X or Y) 按位或非
1011	11	Result = (X < Y) ? 1 : 0 符号比较
1100	12	Result = (X < Y) ? 1 : 0 无符号比较

表 1 操作码对应运算功能

需要注意的是位宽要保持一致，隧道标签不要出错。简单分析可得，这个设计虽然直截了当地实现了 ALU，但是它非常占用资源，事实上它完成了 12 种运算，最后只是对输出结果进行了选择。一种更优秀的设计是让操作码成为使能信号，自然得到所需的结果。

二、实验中遇到的问题

对寄存器电路理解不够深刻

本次实验大多为时序逻辑电路，离不开寄存器的使用。在电路的搭建过程中，我常常忽略时钟信号、使能信号、复位信号的时序问题，导致电路无法正常工作。

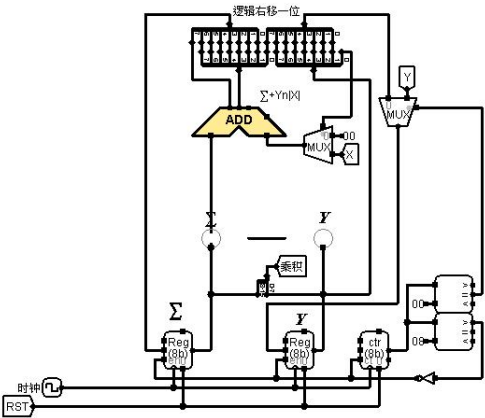


图 5 计数器实现初始置位和电路停止

以搭建原码一位乘法器为例，该电路需要在初始（复位信号来临后）给寄存器置上 Y，后续时钟周期只需将运算结果返回寄存器。如何控制右上位置的双路选择器，又如何控制电路停止，我最开始的设想是通过复位信号控制选择器。但由于复位信号是脉冲信号，而控制

电路进行的时钟信号又晚于复位信号，这导致 Y 无法被输入寄存器中。我也没有想到该如何控制电路停止。

所以在经过思考后我决定引入计数器来记录循环次数，最后解决了这两个问题。

三、实验心得

本次实验所搭建电路的复杂程度明显高于上次。我更深刻地理解了子电路的封装、复用对电路搭建的带来的便利。比如搭建 ALU 的过程中只需要将 XY 与已有的运算器连接，这大大减少了工作量。

本次实验我也体会到了在时序逻辑电路中，统一的复位信号所带来的好处。它能够将电路各寄存器归 0，回到初始状态。也便于在电路出错时清除错误状态。另外，确定的信号要比刚通电时的未知状态更利于电路自启、顺利运行。

在布局布线方面，我也有了更多的经验。例如在连线较简单时，要考虑如何布局以减少线路的交错和拐弯。在线路较为复杂时，则可以利用隧道标签避免线路的大范围“穿梭”，还有利于电路的调试。不过，使用隧道标签也要适时适度，过于简化的电路不易于阅读，难以理解各个子电路的练习。

总之，希望能在以后的实验中积累更多设计电路的技巧。