

实验 4 和 5：数据通路及有限状态机设计

一、实验目的

综合应用掌握的简单组合电路和时序电路的设计方法。完成一个简单的数据通路的设计，掌握数据通路的基本设计方法；为该数据通路配上一个控制器（有限状态机），掌握有限状态机的基本设计方法；将所有的实验综合起来，实现一个自动运算电路。

二、实验内容及步骤

1. 数据通路设计

根据图 1 给出的数据通路（图中 SUM 和 NEXT 是寄存器，Memory 是存储器，+是加法器，==0 是比较器，其它则是多路选择器），完成相应的 Verilog 程序设计。具体要求如下：

- 图中数据线的宽度和各个器件的数据线宽度初始设计时均为 8 位，要求构成数据通路时可以扩充至 16 位或者是 32 位；
- 设计的数据通路能够正确综合，Vivado 所示的电路原理图与图 1 给出的一致。

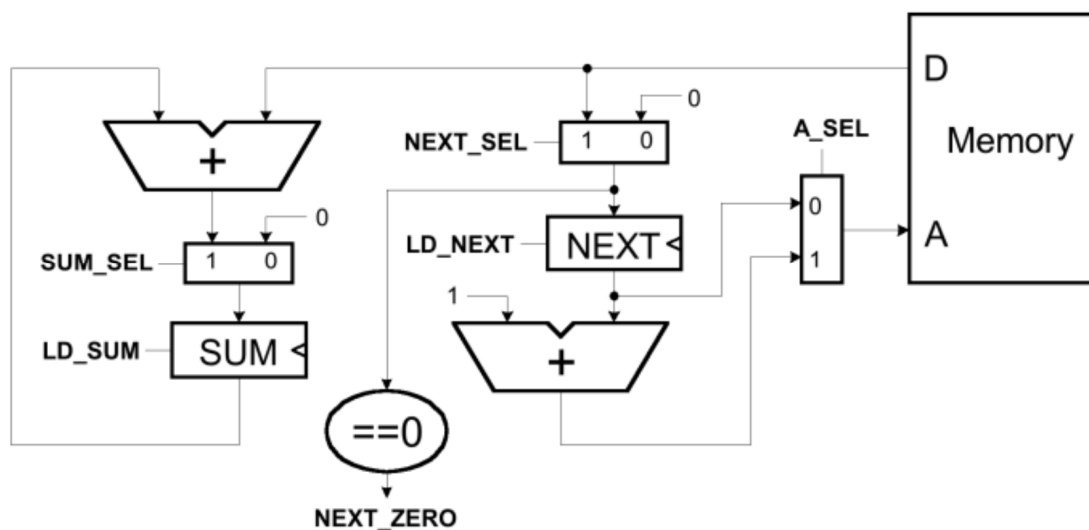


图 1 数据通路图

【提示】

(1) 分别设计 n 位加法器模块, n 位 2 选 1 多路选择器模块, n 位比较器模块。(用 `parameter` 传参来扩展)

(2) 设计一个含同步复位 `rst` 和加载 `load` 端的 n 位寄存器模块

当 `load=1` 时,对输入的 n 位数据进行同步寄存,即让输入 `D` 的值赋给输出 `Q`;

(3) 设计一个 n 位存储器模块,存储器中存放有一条至少包含 3 个节点的数据链表(如图 3 所示),链表第 1 个节点在 0 号地址。存放的链表可以参考后面所示的文本文件用系统函数 `$readmemh` 进行初始化,也可以自行设计数据。

(4) 根据图 1 实例化以上模块完成数据通路模块的设计

输入端口有: 时钟 `clk`, 复位 `rst`, 加载信号 `LD_SUM`, `LD_NEXT`,

输出端口有: 链尾标志 `NEXT_ZERO`

2. 有限状态机设计

假设有限状态机的状态转移图如图 2 所示。根据状态转移图,按照有限状态机(FSM)标准的实现模式来编写 Verilog 程序代码。具体要求如下:

如果要在图 1 所示的数据通路中完成图 3 所示的数据链表的求和运算,需要一个控制器,用我们课上讲的有限状态机(FSM)来实现该控制器。具体步骤要求如下:

- 设计的有限状态机(FSM)能够正确综合;
- 编写有限状态机的仿真程序,完成有限状态机(FSM)的功能仿真,有限状态机功能仿真正确。

【提示】 该控制器模块的端口有:

输入端口: 时钟 `clk`, 复位 `rst`, 启动求和 `start`, 链尾标志 `next_zero`

输出端口: `LD_SUM`, `LD_NEXT`,, 求和结束 `done`

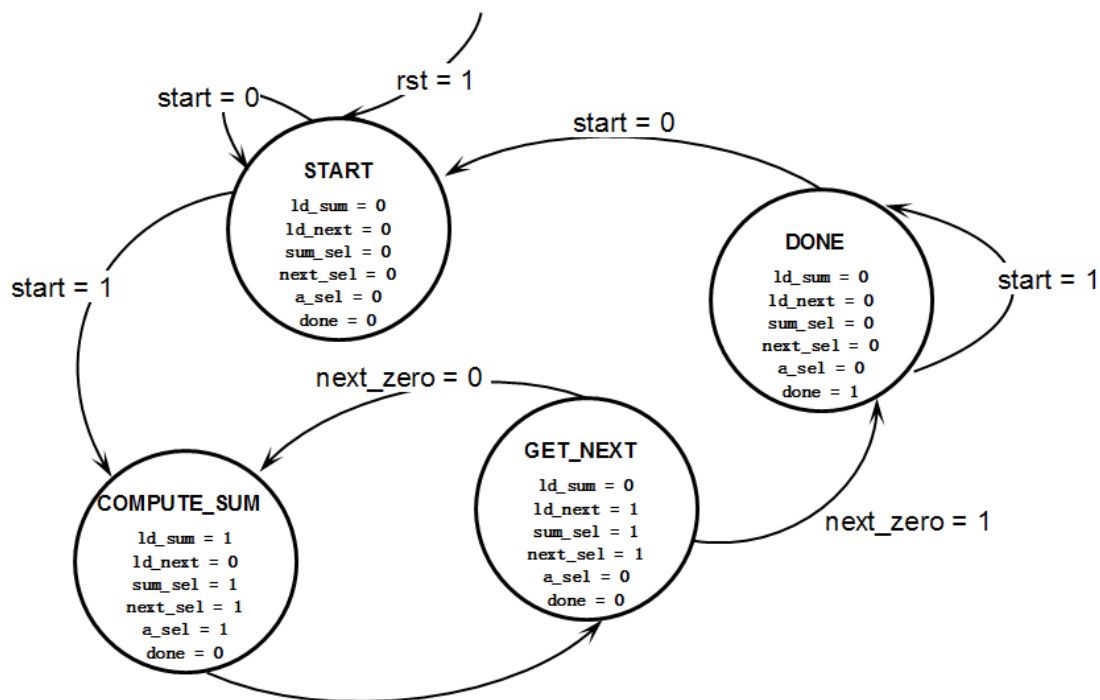
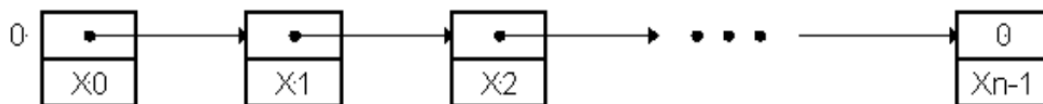


图 2 状态转移图

3. 自动运算电路的设计

将实验步骤 1 实现的数据通路与实验步骤 2 实现的有限状态机（FSM）结合起来，可以进行以链表方式存储的数据的求和运算。

在存储器中存放的数据链表其结构如下图 3 所示，链表的各个节点在存储器中不是连续存放，各节点的第一个地址存放下一个节点的地址，各节点的第二个地址中存放着要进行求和运算的数据，当下一个节点的地址为 0 时，表示到达链表的结尾，求和运算结束。



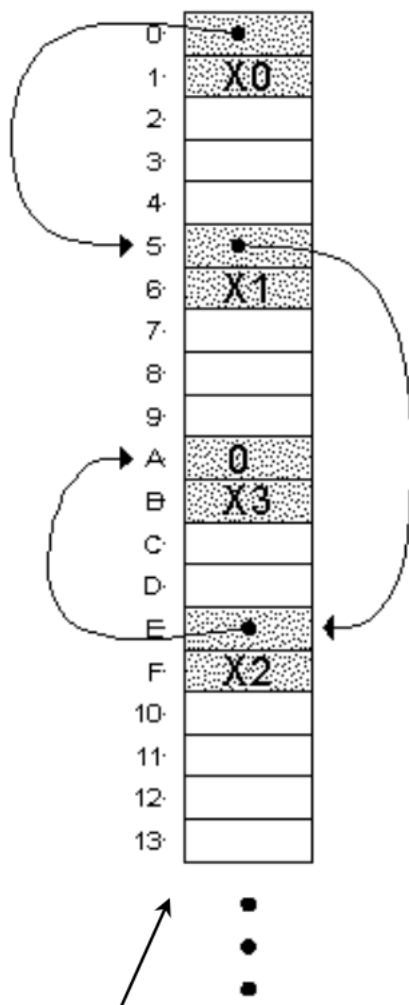


图 3 数据链表及其在存储器中的存放格式

利用上面设计的数据通路、有限状态机，将它们集成起来，设计并实现一个能够进行上述图3所示链表数据的自动求和运算，该电路的总体框架如图4所示，数据通路中存储器所存放的链表可以参考下面所示的文本文件来进行初始化。具体要求如下：

- 完成自动运算求和电路的设计，能够正确综合；
- 编写仿真程序，进行功能仿真，仿真结果正确；
- 功能仿真正确后修改约束文件、生成比特流文件；
- 进行最后的时序仿真，时序仿真结果正确。

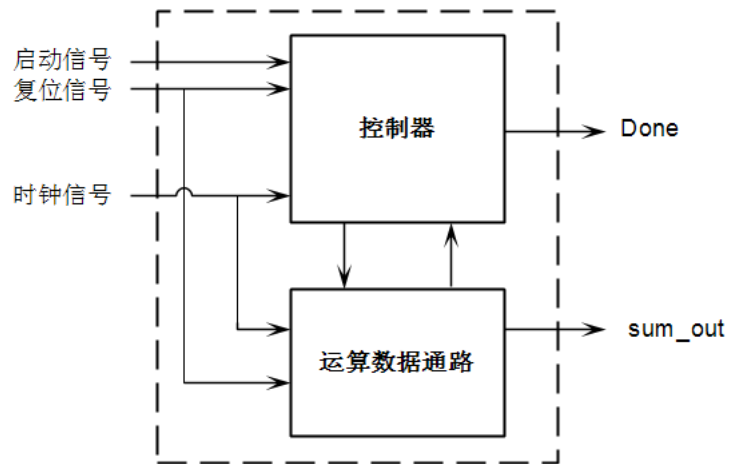


图 4 自动运算电路模块构成图

存储器初始化文件（存储器每个存储单元 32 位，共有 16 个存储单元，最后的求和运算结果 $= 2+4+6+8 = 20$ ）：

00000003

00000002

00000000

00000007

00000004

00000000

00000000

0000000b

00000006

00000000

00000000

00000000

00000008

00000000

00000000

00000000

4. 带数码管显示的自动运算电路（选做）

将上面设计实现的自动运算电路配上实验 2、3 设计的数码管动态显示电路，设计一个能够通过数码管显示求和结果的电路，为了能够看清每步求和的中间结果，需要调慢数据通路和控制器的时钟让运算速度变慢（建议分频器的输出时钟为 1Hz），该电路的总体框架如图 5 所示。具体要求如下：

- 完成带数码管显示的自动运算求和电路的设计，能够正确综合；
- 修改约束文件、生成比特流文件；
- 下载到 Nexys4 DDR 开发板上运行，下载后拨动一个开关开始求和运算，每步的求和结果在数码管上正确显示，整个求和运算结束后一个 LED 指示灯亮，表明运算完成，数码管上显示最终的和，整个求和运算步数不能少于 3 次。
- 如果能够用 10 进制显示求和结果更好。

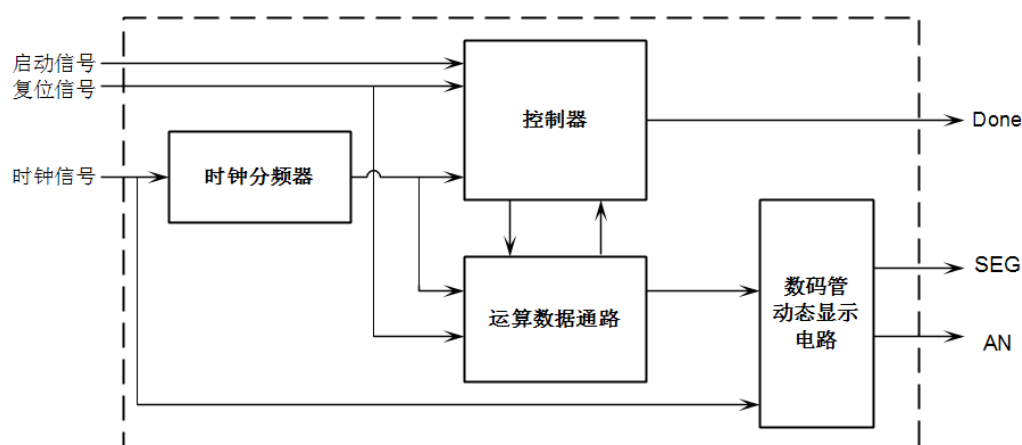


图 5 带数码管显示的自动运算电路框架图