

《数字电路与逻辑设计实验》内容与要求

实验一：系列二进制加法器设计

1. 实验名称

系列二进制加法器设计。

2. 实验目的

采用传统电路的设计方法，设计 4 种二进制加法器（含封装），并利用工具软件 Logisim 的仿真功能来检查电路设计是否达到要求。

通过以上实验的设计、仿真和验证等训练过程使学生掌握传统逻辑电路的设计、仿真和调试的方法。

3. 实验所用设备

Logisim2.7.1 软件 1 套，微型计算机 1 台。

4. 课时

课内 4 个课时，课外 4 个课时。

5. 实验内容

设计电路并使用 Logisim 软件进行仿真。除逻辑门、触发器外，不能直接使用 Logisim 软件提供的其它逻辑组件库元件。

具体内容如下：

（1）一位二进制半加器

设计一个一位二进制半加器，电路有两个输入 A、B，两个输出 S 和 C。输入 A、B 分别为被加数、加数，输出 S、C 分别为本位和、向高位的进位信号。

（2）一位二进制全加器

设计一个一位二进制全加器，电路有三个输入 A、B 和 C_i ，两个输出 S 和 C_o 。输入 A、B 和 C_i 分别为被加数、加数和来自低位的进位，输出 S 和 C_o 分别为本位和、向高位的进位信号。

（3）串行进位的四位二进制并行加法器

用四个一位二进制全加器设计一个串行进位的四位二进制并行加法器，电路有九个输入 A_3 、 A_2 、 A_1 、 A_0 、 B_3 、 B_2 、 B_1 、 B_0 和 C_0 ，五个输出 S_3 、 S_2 、 S_1 、 S_0 和 C_4 。输入 $A = A_3A_2A_1A_0$ 、 $B = B_3B_2B_1B_0$ 和 C_0 分别为被加数、加数和来自低位的进位，输出 $S = S_3S_2S_1S_0$ 为本位和， C_4 为向高位的进位。

（4）先行进位的四位二进制并行加法器

利用先行进位的思想设计一个先行进位的四位二进制并行加法器，电路有九个输入 A_3 、 A_2 、 A_1 、 A_0 、 B_3 、 B_2 、 B_1 、 B_0 和 C_0 ，五个输出 S_3 、 S_2 、 S_1 、 S_0 和 C_4 。

输入 $A = A_3A_2A_1A_0$ 、 $B = B_3B_2B_1B_0$ 和 C_0 分别为被加数、加数和来自低位的进位，输出 $S = S_3S_2S_1S_0$ 为本位和， C_4 为向高位的进位。

(5) 将先行进位的四位二进制并行加法器封装成一个组件，并验证它的正确性

将设计好的先行进位的四位二进制并行加法器进行封装，生成一个“私有”元件，并验证它的正确性，以便后续实验使用。封装后的逻辑符号如图 1.1 所示。

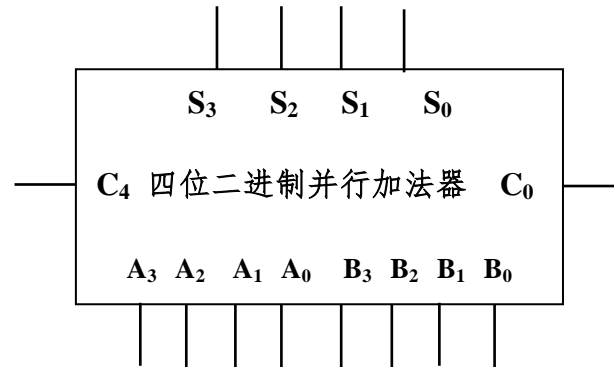


图 1.1 先行进位的四位二进制并行加法器

6. 实验方案设计

要求：(1) 给出函数表达式或逻辑描述；(2) 画出电路图。

实验二：小型实验室门禁系统设计

1. 实验名称

小型实验室门禁系统设计。

2. 实验目的

采用合适的设计方法，对一个“设计场景”进行逻辑电路的设计，并利用工具软件 Logisim 的仿真来验证该设计是否达到要求。

通过以上实验的设计、仿真、验证等训练过程使学生掌握小型电路系统的设计、仿真、调试方法以及电路模块封装的方法。

3. 实验所用设备

Logisim2.7.1 软件 1 套，微型计算机 1 台。

4. 课时

课内 4 个课时，课外 4 个课时。

5. 实验内容

设计场景：某小型保密实验室需要安装一个门禁系统，用于监测、控制和显示该实验室内人数。该实验室只有一个门，最多只能容纳 15 人。假设员工进出实验室都要刷门禁卡，并且保证一次刷卡后有且只有一人能进出。实验室空置时人数显示为 0，刷卡进入时实验室人数加 1，刷卡离开时实验室人数减 1。当实验室满员时，还有员工在门外刷卡进入时，系统报警提示满员，不允许进入，实验室内人数不变。

使用 Logisim 软件对小型电路进行虚拟实验仿真，除逻辑门、触发器、7 段数码显示管外，不能直接使用 Logisim 提供的逻辑组件库。

具体要求如下：

（1）四位二进制可逆计数器

用 D 触发器设计一个四位二进制可逆计数器并**进行封装**，逻辑符号如图 2.1 所示。

该计数器有一个清零端 **CLR**、一个累加计数脉冲输入端 **CP_U**（输入刷卡进入请求）、一个累减计数脉冲输入端 **CP_D**（输入刷卡离开请求）、预置控制端 **\overline{LD}** 、预置初置端 **DCBA**、四个计数器状态输出值 **Q_D Q_C Q_B Q_A**。当预置控制端 **\overline{LD}** 为低电平时，计数器输出 **Q_D Q_C Q_B Q_A** 被预置为 **DCBA** 端输入的值。

（2）二进制转 8421BCD 码电路

用第一次实验所设计的“私有”元件“先行进位的四位二进制并行加法器”和适当元器件，设计一个将二进制数表示的实验室人数转换成 **8421BCD 码** 的电路，并封装。

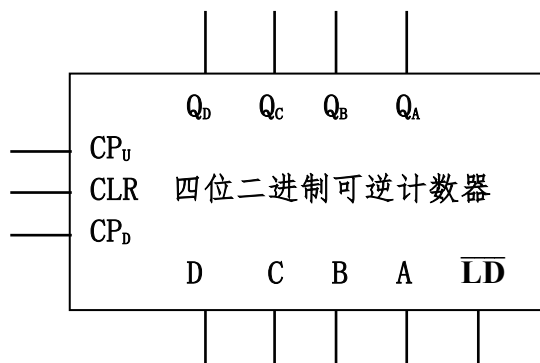


图 2.1 四位二进制可逆计数器

(3) 显示电路

设计一个 7 段译码器（参考教材中的 7448 芯片），将两位十进制数的 8421BCD 码表示的实验室人数用“7 段数码显示管”显示出来，并封装该译码器电路。

该 7 段译码器有四个输入 $A_3A_2A_1A_0$ 和七个输出 $abcdefg$ ， $A_3A_2A_1A_0$ 为 8421BCD 码， $abcdefg$ 为 7 段数码显示管对应的段。

(4) 报警电路

设计报警电路并封装，满足如下要求：

当实验室满员时，在累加计数脉冲输入端 CP_U 刷卡进入请求时，计数器输出端状态值保持不变，系统报警提示满员。当实验室空时，逻辑上不会有实验室内累减计数脉冲输入端 CP_D 刷卡离开请求。为防止信号干扰，在计数输出为 0 时，若 CP_D 有脉冲，计数器状态值保持不变，且不用报警。

(5) 小型实验室门禁系统电路的封装

利用已设计好的“私有”元器件和相应元器件设计一个“实验室门禁系统”电路，并进行封装，封装后的小型实验室门禁系统逻辑符号如图 2.2 所示。

提示：四位二进制可逆计数器的预置控制端 \overline{LD} 和预置初置端 $DCBA$ 在封装时不需要引出引脚。

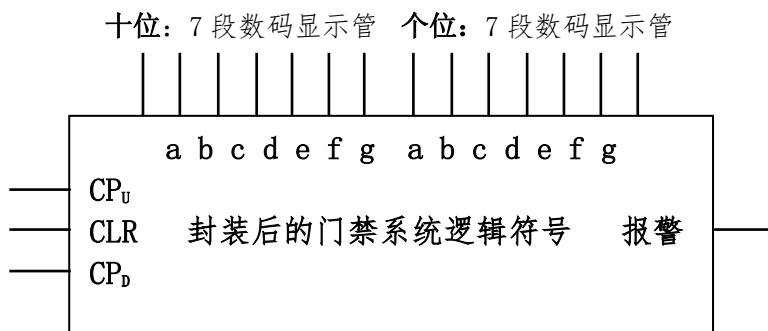


图 2.2 封装后的小型实验室门禁系统

6. 实验方案设计

要求：（1）给出函数表达式或逻辑描述；（2）画出电路图。

实验三：无符号数的乘法器设计

1. 实验名称

无符号数的乘法器的设计。

2. 实验目的

要求使用合适的逻辑电路的设计方法，通过工具软件 Logisim 进行无符号数的乘法器的设计和验证，记录实验结果，验证设计是否达到要求。

通过无符号数的乘法器的设计、仿真、验证等训练过程，使学生掌握数字逻辑电路的设计、仿真、调试的方法。

3. 实验所用设备

Logisim2.7.1 软件 1 套，微型计算机 1 台。

4. 课时

课内 4 个课时，课外 4 个课时。

5. 实验内容

(1) 四位乘法器设计

四位乘法器 $Mul4 \times 4$ 实现两个无符号的 4 位二进制数的乘法运算，其结构框图如图 3.1 所示。设被乘数为 $b(3:0)$ ，乘数为 $a(3:0)$ ，乘积需要 8 位二进制数表示，乘积为 $p(7:0)$ 。

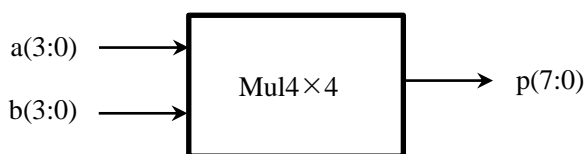


图 3.1 四位乘法器结构框图

四位乘法器运算可以用 4 个相同的模块串接而成，其内部结构如图 3.2 所示。每个模块均包含一个加法器、一个 2 选 1 多路选择器和一个移位器 shl。

图 3.2 中数据通路上的数据位宽都为 8，确保两个 4 位二进制数的乘积不会发生溢出。shl 是左移一位的操作，在这里可以不用逻辑器件来实现，而仅通过数据连线的改变就可实现。

(2) 32×4 乘法器设计

32×4 乘法器 $Mul32 \times 4$ 实现一个无符号的 32 位二进制数和一个无符号的 4 位二进制数的乘法运算，其结构框图如图 3.3 所示。设被乘数为 $b(31:0)$ ，乘数为 $a(3:0)$ ，乘积用 64 位二进制数表示，乘积为 $p(63:0)$ 。

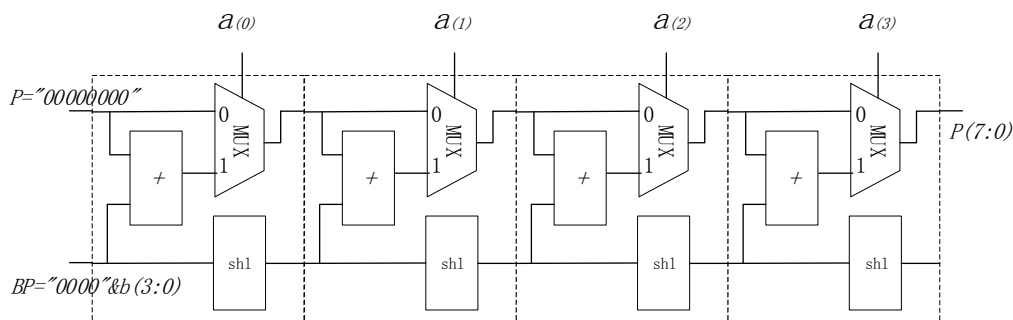


图 3-2 四位乘法器内部结构

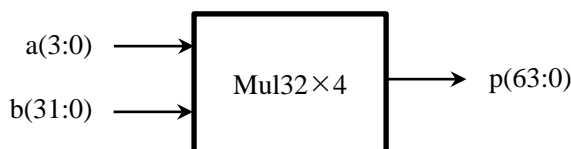


图 3.3 32×4 乘法器结构框图

在四位乘法器 $\text{Mul4} \times 4$ 上进行改进，将数据通路上的数据位宽都改为 64 位，即可实现 $\text{Mul32} \times 4$ 。

(3) 32×32 乘法器设计

32×32 乘法器 $\text{Mul32} \times 32$ 实现两个无符号的 32 位二进制数的乘法运算，其结构框图如图 3.4 所示。设被乘数为 $b(31:0)$ ，乘数为 $a(31:0)$ ，乘积用 64 位二进制数表示，乘积为 $p(63:0)$ 。

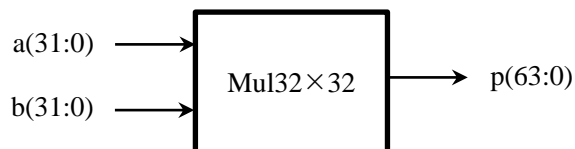


图 3.4 32×32 乘法器结构框图

用 32×4 乘法器 $\text{Mul32} \times 4$ 作为基本部件，实现 32×32 乘法器 $\text{Mul32} \times 32$ 。

设被乘数为 $b(31:0) = (b_{31}b_{30}b_{29}b_{28} \cdots b_{15}b_{14}b_{13}b_{12} \cdots b_4b_3b_2b_1b_0)_2$

乘数为 $a(31:0) = (a_{31}a_{30}a_{29}a_{28} \cdots a_{15}a_{14}a_{13}a_{12} \cdots a_3a_2a_1a_0)_2$

$$= (a_{31}a_{30}a_{29}a_{28})_2 \times 2^{28} + \cdots + (a_{15}a_{14}a_{13}a_{12})_2 \times 2^{12} + \cdots + (a_3a_2a_1a_0)_2 \times 2^0$$

那么，

$$p(63:0) = b(31:0) \times a(31:0)$$

$$= b(31:0) \times ((a_{31}a_{30}a_{29}a_{28})_2 \times 2^{28} + \cdots + (a_{15}a_{14}a_{13}a_{12})_2 \times 2^{12} + \cdots + (a_3a_2a_1a_0)_2 \times 2^0)$$

$$= b(31:0) \times (a_{31}a_{30}a_{29}a_{28})_2 \times 2^{28} + \cdots + b(31:0) \times (a_{15}a_{14}a_{13}a_{12})_2 \times 2^{12} + \cdots + b(31:0) \times (a_3a_2a_1a_0)_2 \times 2^0$$

从上述推导可知， $\text{Mul32} \times 32$ 可以用 8 个 $\text{Mul32} \times 4$ 分组相乘，然后通过 4 的倍数位的左移（相当于乘 2^i ），再将左移结果两两相加得到。

6. 实验设计过程

要求：（1）写出设计思想，画出乘法器的内部逻辑结构框图；（2）给出 logisim 软

件绘制的电路图。

7. 实验结果记录

根据实验方案设计要求，对于相应的乘法器，在给定的输入条件下，填写表 3.1。

表 3.1 无符号数的乘法器实验结果记录表

电路	输入 1 (16 进制)	输入 2 (16 进制)	输出 (16 进制)
Mul4×4	b=0×A	a=0×A	p=
Mul4×4	b=0×E	a=0×9	p=
Mul32×4	b=0×003ABEF1	a=0×A	p=
Mul32×4	b=0×019ABEF1	a=0×7	p=
Mul32×32	b=0×0002BEF1	a=0×00004EF1	p=
Mul32×32	b=0×00003EF1	a=0×0003BEF1	p=

实验四：无符号数的除法器设计

1. 实验名称

无符号数除法器的设计。

2. 实验目的

要求使用合适的逻辑电路的设计方法，通过工具软件 Logisim 进行无符号数的除法器的设计和验证，记录实验结果，验证设计是否达到要求。

通过无符号数的除法器的设计、仿真、验证等训练过程，使学生掌握数字逻辑电路的设计、仿真、调试的方法。

3. 实验所用设备

Logisim2.7.1 软件 1 套，微型计算机 1 台。

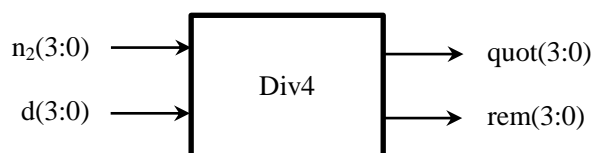
4. 课时

课内 4 个课时，课外 4 个课时。

5. 实验内容

(1) 四位除法器设计

四位除法器实现两个无符号的 4 位二进制数的除法运算，其结构框图如图 4.1 所示。设被除数为 $n_2(3:0)$ ，除数为 $d(3:0)$ ，商为 $quot(3:0)$ ，余数为 $rem(3:0)$ 。



4.1 四位除法器结构框图

四位除法器 Div4 算法步骤如下：

(a) 设 $n_1="0000"$ ，将被除数以 $n_1:n_2$ 的形式拼接，除数为 d ；

(b) 重复 4 次：

将 $n_1:n_2$ 左移 1 位；

if ($n_1 \geq d$) begin $n_1 = n_1 - d$; $n_2(0) = 1$ end

(c) 商和余数的结果为： $quot = n_2$; $rem = n_1$ 。

四位除法器也可以用 4 个相同的模块串接而成。每个模块均包含一个减法器、两个 2 选 1 多路选择器、一个比较器和一个移位器 shl。请参照四位乘法器的设计思路，实现两个无符号的 4 位二进制数的除法器。

(2) 32 位除法器设计

32 位除法器 Div32 实现两个无符号的 32 位二进制数的除法运算，其结构框图如图

4.2 所示。设被除数为 $n(31:0)$ ，除数为 $d(31:0)$ ，商为 $quot(31:0)$ ，余数为 $rem(31:0)$ 。

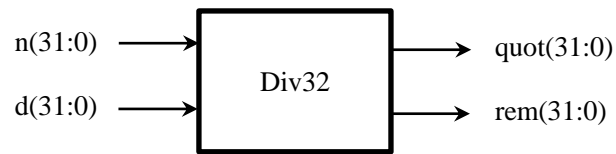


图4.2 32 位除法器结构框图

对四位除法器 Div4 中 4 个相同的模块中的一个进行改进，将数据通路上的数据位宽都扩展为 32 位，得到一个 Div1。将 32 个 Div1 拼接起来即可实现 Div32。

6. 实验设计过程

要求写出设计思想，画出除法器 Div4 的内部逻辑结构框图（仿照 Mul4×4 内部结构）；（2）给出 Logisim 软件绘制的电路图。

7. 实验结果记录

根据实验方案设计要求，对于相应的除法器，在给定的输入条件下，填写表 4.1。

表 4.1 无符号数的除法器实验结果记录表

电路	输入 1 (16 进制)	输入 2 (16 进制)	输出 (16 进制)	
Div4	$n_2=0\times E$	$d=0\times 9$	quot=	rem=
Div4	$n_2=0\times E$	$d=0\times 0$	quot=	rem=
Div32	$n=0\times 019ABEF1$	$d=0\times 00004EF1$	quot=	rem=
Div32	$n=0\times A0504EF1$	$d=0\times 019ABEF1$	quot=	rem=

实验五：多功能电子钟系统设计

1. 实验名称

多功能电子钟系统设计。

2. 实验目的

采用合适的设计方法，对给定的要求进行逻辑电路的设计，并利用工具软件 logisim 的虚拟仿真来验证本设计是否达到要求。

通过以上实验的设计、仿真、验证等训练过程使同学们掌握小型电路系统的设计、仿真、调试方法以及电路模块封装的方法。

3. 实验所用设备

Logisim2.7.1 软件 1 套，微型计算机 1 台。

4. 课时

课内 8 个课时，课外 8 个课时。

5. 实验内容

设计场景：多功能数字钟是一种用数字显示时、分、秒的计时装置，其基本功能如下：

- (1) 显示时、分、秒；
- (2) 可以切换 24 小时制或 12 小时制（上午和下午）；
- (3) 整点报时，整点前 10 秒开始，整点时结束；
- (4) 单独对“时、分”计时校准，分钟值校准时不影响小时值；
- (5) 闹钟，到设定时间提醒 10 秒。

使用 Logisim 软件对所设计电子钟电路进行虚拟仿真验证，具体要求如下：

（采用 Logisim 软件提供的“时钟频率”为 8hz 的信号源。）

(1) 具有校准计数值的六十进制计数器电路

采用实验二所设计的“四位二进制可逆计数器”这个“私有”元件和相应元器件，设计一个具有对计数值进行校准的六十进制计数器，并进行封装，该计数器封装图如图 5.1 所示。

具体要求：

- (a) 封装后的电路输入：一个累加计数脉冲输入端 CP_U 、一个累减计数脉冲输入端 CP_D 、清零输入信号 Clr 、一个计数值校准输入控制信号 Adj ；
- (b) 封装后的电路输出为八个计数器状态输出值 $Q_{1D} Q_{1C} Q_{1B} Q_{1A} Q_{0D} Q_{0C} Q_{0B} Q_{0A}$ （测试时要接两个 16 进制的数码显示管），进位输出信号 $\overline{Q_{cc}}$ ；
- (c) 当 $Adj=1$ 时，可以通过 CP_U 、 CP_D ，对计数值进行加、减调整来设置当前时

间。递减的时候不需要循环累减，回到 0 即可；递增的时候需要循环累加；

(d) 当 $Adj=0$ ，通过输入脉冲 CP_U 计数器累加计数，每当累计满 60 产生一个进位输出信号 $\overline{Q_{cc}}$ ；

(e) Clr 为 1 时，计数器清零；

(f) 计数器的输出为两位 8421BCD 码；

(g) 封装后做出测试电路，测试电路要外接两个 16 进制的数码显示管， CP_U 、 CP_D 接按钮。

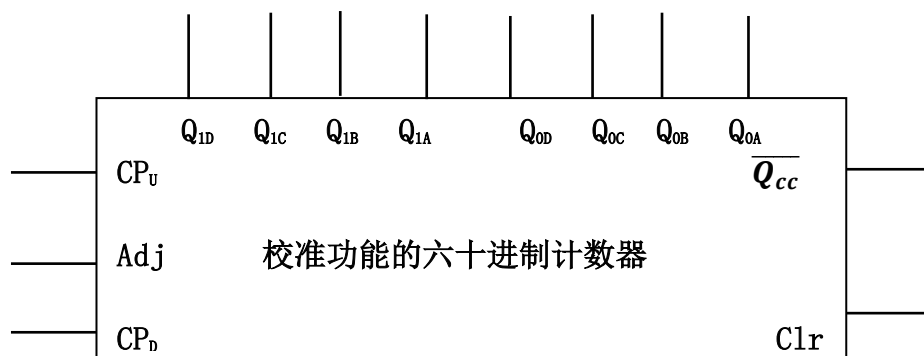


图 5.1 调整计数值的 60 进制计数器

(2) 具有校准计数值的十二进制计数器或二十四进制的计数器电路

采用 (1) 设计的六十进制计数器和相应元器件，设计一个具有对计数值进行校准的十二进制计数器或二十四进制的计数器，并进行封装，该计数器封装图如图 5.2 所示。

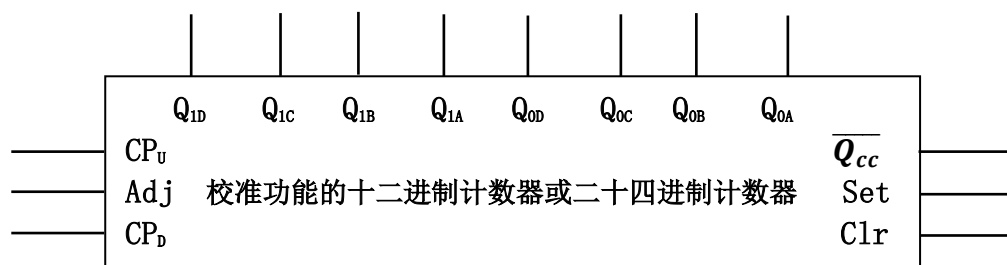


图 5.2 调整计数值的十二进制或二十四进制计数器

具体要求：

(a) 封装后的电路输入：一个累加计数脉冲输入端 CP_U 、一个累减计数脉冲输入端 CP_D 、清零输入信号 Clr 、一个计数值校准输入控制信号 Adj 、12 小时计时或 24 小时计时控制信号 Set ；

(b) 封装后的电路输出为八个计数器状态输出值 $Q_{1D} Q_{1C} Q_{1B} Q_{1A} Q_{0D} Q_{0C} Q_{0B} Q_{0A}$ （测试电路中要两个 16 进制的数码显示管），进位输出信号 $\overline{Q_{cc}}$ ；

(c) 当 $Adj=1$ 时，可以通过 CP_U 、 CP_D ，对计数值进行加、减调整来设置当前时间。递减的时候不需要循环累减，回到 0 即可；递增的时候需要循环累加；

(d) 当 $Adj=0$ ，通过输入脉冲 CP_U 计数器累加计数，每当累计满 12 或 24（根据 Set 确定计数制）产生一个进位输出信号 $\overline{Q_{cc}}$ ；

- (e) Clr 为 1 时，计数器清零；
- (f) 当 Set=0，12 小时计时；当 Set=1 时，24 小时计时；
- (g) 计数器的输出为两位 8421BCD 码；
- (h) 封装后做出测试电路，测试电路要接两个 16 进制的数码显示管，CPU、CP_D 接按钮。

(3) 显示“上午”、“下午”的电路

设计一个采用“Led 点阵”显示器和相应元器件以“上”和“下”的形式表示电子钟的“上午”和“下午”的电路，并封装，文字显示如图 5.3 所示。封装图如图 5.4 所示，测试电路如图 5.5 所示。

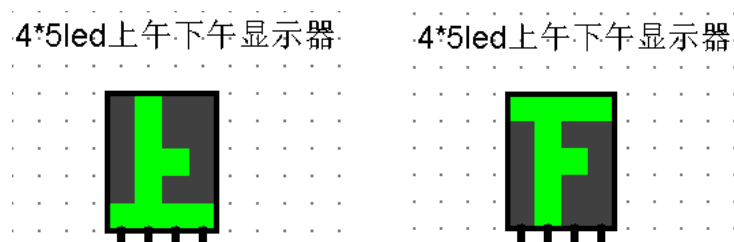


图 5.3 led 点阵显示器

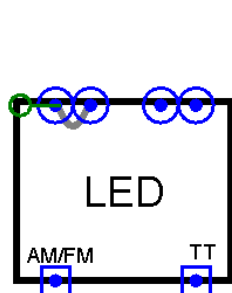


图 5.4 led 点阵封装图

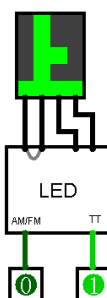


图 5.5 led 点阵测试图

具体要求：

- (a) 封装后的电路输入为：一个上下午显示控制信号 AM/FM、计时控制 TT；
- (b) 封装后的电路输出为 4 个五位的数据，用以接 4*5Led (4 列×5 行) 显示器；
- (c) AM/FM=0，显示“上”； AM/FM=1，显示“下”；
- (d) TT=0 时，24 小时计时，此时“上、下午”显示屏全灭； TT=1 时，12 小时计时，此时根据具体时间显示“上”或“下”；
- (e) 封装时 Led 显示屏不封装在内；
- (f) 封装后做出测试电路，外接 Led 显示屏。

(4) 电子钟整点报时电路

设计一个 10 秒的整点报时电路，并进行封装，该电路在整点前 10 秒 (59 分 50 秒) 被触发，发出报时信息 (用 Led 灯的亮灭来表示)，报时 10 秒结束。

(5) 秒计时脉冲产生电路

按要求以 Logisim 软件的 8hz 信号作为电路震荡源, 设计一个输出为 1hz 的脉冲信号电路, 并封装, 逻辑符号如图 5.6 所示, 它成为秒计数器的计数脉冲信号。

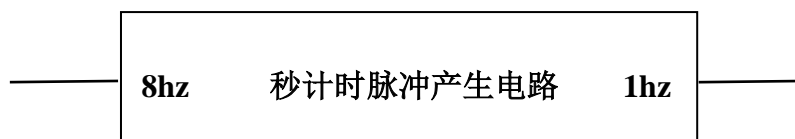


图 5.6 秒计时脉冲产生电路

(6) 闹钟 (选做)

设计定时起闹 (闹钟) 电路, 并封装。

具体要求:

- (a) 可设置闹钟起闹时间, 具体到小时和分钟, 在测试电路中要两个 16 进制的数码显示管;
- (b) 在设定的起闹时间, 闹钟开始响铃, 十秒后结束;
- (c) 闹铃用 Led 灯的亮灭表示;
- (d) 有控制端可以启用或关闭闹钟。

(7) 多功能数字钟电路

充分利用 (1) ~ (6) 设计的 “私” 有元件和相应元器件, 设计满足多功能电子钟 “设计场景” 要求的电路, 并封装, 封装图如图 5.7 所示, 测试图如图 5.8 所示。

(a) 输入信号有 “Set”、“CPU、CPD”、“Adj0、Adj1”、“Clr”、“8hz 信号”; 输出信号为 “时”、“分”、“秒” 对应的 6 个 8421BCD 码、“闹钟” 和 “整点” 输出显示信号以及控制 “上、下午” 显示的信号;

(b) “Set” 为 “小时计数器” 输入信号, 当 Set=1 时, 计数器为二十四进制计数器, Set=0 时为十二进制计数器; 十二进制 (上午: 12, 1-11, 下午: 12, 1-11) 和二十四进制 (0~23) 转换时时间需对应; 注意十二进制时钟没有 0 点, 12 点就是 0 点, 所以十二进制的上午 12,1-11 点对应二十四进制的 0~11 点;

(c) “CPU、CPD” 为计数器计数值进行手动加、减调整的输入脉冲信号;

(d) “Adj0” 为计数器计数值进行校准的输入控制信号, Adj0=0, 表示不调整时钟; Adj0=1, 表示调整时钟。在调整时钟时, 不产生任何进位信息 (秒不向分进位, 分不向小时进位);

(e) “Adj1” 为计数器计数值进行校准的选择输入控制信号, Adj1=0, 表示调整小时; Adj1=1, 表示调整分钟;

(f) “Clr” 为计数器的清除信号, 同时对小时、分、秒清零;

(g) “8hz 信号” 为电子钟脉冲输入信号;

(h) 输出的时间小时、分和秒分别为 6 个 8421BCD 码;

(i) “Led 点阵” 显示器分别对应 “上、下午” 输出信号;

- (j) 两个“发光二极管 (Led 灯)”分别对应“闹钟”，“整点”输出信号。
- (k) 如果选做闹钟，“Alarm”为输入的时间设定提醒值（闹钟值）；
- (l) 封装后做出测试电路，测试电路中小时、分和秒要接 16 进制显示器， CP_U 、 CP_D 接按钮，CP 接时钟源，闹钟和定点报时接 Led 灯，Led 显示接 Led 显示屏，其余接输入引脚。

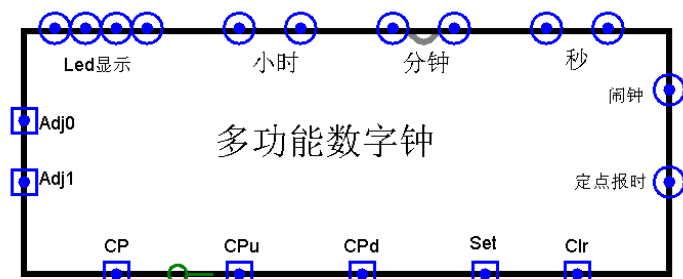


图 5.7 电子钟的“输入、输出检查要求”

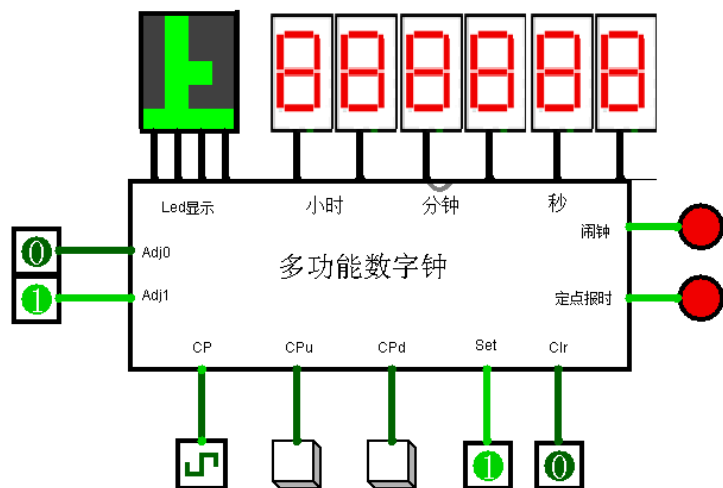


图 5.8 电子钟的测试电路

6. 实验方案设计

要求：(1) 给出设计过程或设计思路；(2) 画出电路图。

7. 实验结果记录

要求：封装后各电路后，截取带有“Logisim”软件仿真调试信息的电路图。

8. 实验后的思考

- (1) 实验的难点在哪些方面？
- (2) 如何解决这些难点？

实验六：斐波那契(Fibonacci)数列计算器设计

1. 实验名称

斐波那契(Fibonacci)数列计算器设计。

2. 实验目的

要求使用合适的逻辑电路的设计方法，通过工具软件 Logisim 进行斐波那契(Fibonacci)数列计算器设计和验证，记录实验结果，验证设计是否达到要求。

通过斐波那契(Fibonacci)数列计算器的设计、仿真、验证等训练过程，掌握数字逻辑电路的设计、仿真、调试的方法。

3. 实验所用设备

Logisim2.7.1 软件 1 套，微型计算机 1 台。

4. 课时

课内 8 个课时，课外 8 个课时。

5. 实验内容

斐波那契(Fibonacci)数列中每项数值都是其两个直接前项的和，其生成规则如下公式 1 所示。

$$F_n = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ F_{n-1} + F_{n-2}, & n > 1 \end{cases} \quad (\text{公式 1})$$

(1) 求 Fibonacci 数的矩阵算法

对于数列的初始条件对应公式 2 的矩阵运算：

$$\begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} \quad (\text{公式 2})$$

更一般化地，有公式 3：

$$\begin{pmatrix} F_{n-1} \\ F_n \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} F_{n-2} \\ F_{n-1} \end{pmatrix} \quad (\text{公式 3})$$

根据递推关系可以得到公式 4：

$$\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} a_n & b_n \\ c_n & d_n \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} b_n \\ d_n \end{pmatrix} \quad (\text{公式 4})$$

由公式 4 可推出， $F_n = b_n$ 。

因此，对求斐波那契数列的第 n 项的问题，可以转化为对一个二维矩阵 $A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$ 求其 n 次幂。采用矩阵的快速幂算法，操作次数可优化为 $O(\log_2 n)$ 。

由于 $F(47)=(2971215073)_{10}<2^{32}$, $F(48)=(4807526976)_{10}>2^{32}$, 电路中采用 32 位二进制数表示一个整数。为了避免整数溢出, 取 $2\leq n\leq 47$, n 用 6 位二进制数表示。

(2) 算法描述

Fibonacci(){

初始化: $X = A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, Start=0;

For (i=5 downto 0)

{

if (Start==0) then

{

if (n[i]==1) then Start=1;

}

Else

{

if (n[i]==1)

then $X=X^2 \cdot A$;

else

$X=X^2$; }

}

return(X);

}

例如: $n = (101100)_2 = (44)_{10}$

step1: $i=5$, Start=0, $n[5]=1$, 此时 Start 置 1;

step2: $i=4$, Start=1, $n[4]=0$, 此时 $X = X^2 = A^2$;

step3: $i=3$, Start=1, $n[3]=1$, 此时 $X = X^2 \cdot A = (A^2)^2 \cdot A$;

step4: $i=2$, Start=1, $n[2]=1$, 此时 $X = X^2 \cdot A = ((A^2)^2 \cdot A)^2 \cdot A$;

step5: $i=1$, Start=1, $n[1]=0$, 此时 $X = X^2 = (((A^2)^2 \cdot A)^2 \cdot A)^2$;

step6: $i=0$, Start=1, $n[0]=0$, 此时 $X = X^2 = (((((A^2)^2 \cdot A)^2 \cdot A)^2 \cdot A)^2)^2$;

循环执行完后, $X = (((((A^2)^2 \cdot A)^2 \cdot A)^2 \cdot A)^2)^2 = A^{44}$

(3) 矩阵计算模块

(a) 计算 X^2 模块 sqrX

$$X^2 = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^2 = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a^2 + bc & ab + bd \\ ac + cd & bc + d^2 \end{pmatrix} \quad (\text{公式 5})$$

其相应的输入/输出如图 6.1 所示。

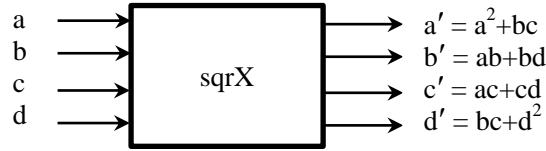


图 6.1 计算 X^2 模块 sqrx 输入/输出示意图

这里， $a, b, c, d, a', b', c', d'$ 都为 32 位无符号二进制整数。

(b) 计算 $X^2 \cdot A$ 模块 $\text{sqrx} \cdot A$

$$X^2 \cdot A = \begin{pmatrix} a^2 + bc & ab + bd \\ ac + cd & bc + d^2 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} ab + bd & a^2 + bc + ab + bd \\ bc + d^2 & ac + cd + bc + d^2 \end{pmatrix} \quad (\text{公式 6})$$

其相应的输入/输出如图 6.2 所示。

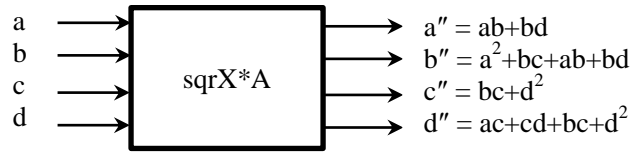


图 6.2 计算 $X^2 \cdot A$ 模块 $\text{sqrx} \cdot A$ 输入/输出示意图

这里， $a, b, c, d, a'', b'', c'', d''$ 都为 32 位无符号二进制整数。

(4) 矩阵快速幂算法迭代模块

该模块 Fibo 输入/输出端如图 6.3 所示。



图 6.3 Fibo 输入/输出示意图

这里， start 为 $\text{Fibonacci}()$ 算法中的 6 位二进制数 n 左移出的第一个（最高位的）1 的标志信号； n_{i-1} 是 $\text{start}=1$ 之后左移出的下一位； clr 为初始化（清零）信号，此时 $X=A$ ； clk 为时钟脉冲信号。 F_i 为 $\text{Fibonacci}()$ 算法迭代的中间结果，根据 n_{i-1} 取 0 或 1 来决定 F_i 是取 sqrx 或者 $\text{sqrx} \cdot A$ 运算后的矩阵元素 b_i ，在第 6 个时钟脉冲时， F_i 即为输入 n 的 Fibonacci 数 F_n 。

其内部逻辑结构图如图 6.4 所示。

(5) Fibonacci 数显示模块

将二进制数转换成十进制数在数码显示管上显示出来。

输入为 32 位二进制的 Fibonacci 数 $F(n)$ 。

由于 32 位二进制 Fibonacci 数表示的最大十进制数的位数是 10 位，该模块的输出为 10 组 8421BCD 码 $D_9, D_8, D_7, D_6, D_5, D_4, D_3, D_2, D_1, D_0$ ，每组 8421BCD 码表示 1 位 10 进制数。

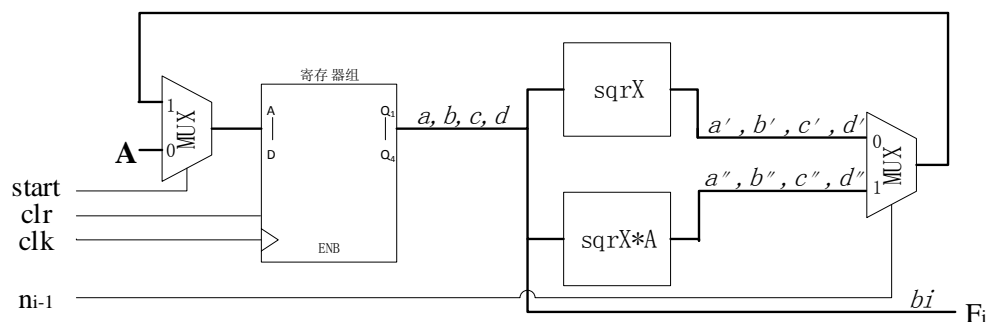


图 6.4 Fibo 内部逻辑结构图

(6) 主模块 main

主模块 main 的逻辑结构图 6.5 所示。

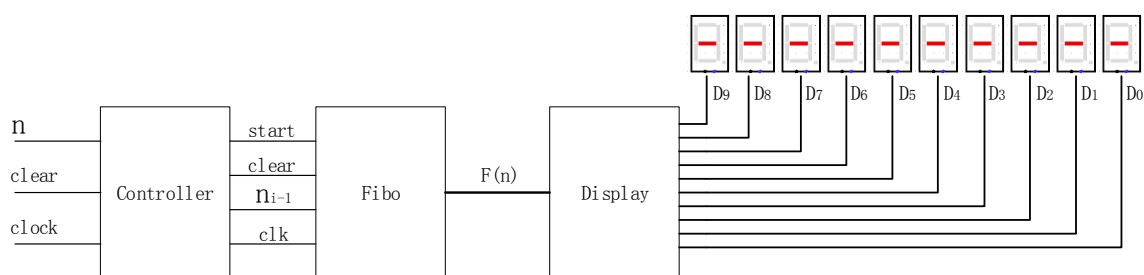


图 6.5 主模块 main 的逻辑结构图

控制器 Controller 中包括三个功能块：6 位二进制数 n 的左移控制电路、6 个时钟脉冲控制电路、start 信号产生电路。

6 位二进制数 n 的左移控制电路,使用一个移位寄存器,在时钟脉冲作用下产生 n_{i-1} 。用 clear 信号装入 n , 进行移位寄存器的初始化。

使用 1 个 8 位计数器、1 个比较器和适当的门电路,可以控制 Fibo 只接收 6 个 clock 时钟脉冲 (产生 clk)。直至下一个 clear 信号初始化后,才准备产生下一组 6 个时钟脉冲。

使用 1 个 D 触发器加适当的门电路构成一个锁存器 Latch, 在接收到 n 的最高位 1 时 start=1, 直至下一个 clear 信号使 start=0。

在 6 个 clock 时钟脉冲信号后,电路就产生了第 n 个 Fibonacci 数 $F(n)$,并经过 Display 电路转换成十进制数在数码管上显示出来。

6. 实验方案设计

具体要求:

(1) 给出 Fibonacci 数列通项公式、Fibonacci 数列的递归算法 (指数时间复杂度) 形式化描述、Fibonacci 数列的多项式时间复杂度算法形式化描述;

(2) 给出矩阵 X^2 计算模块的设计思路、给出 logisim 软件绘制的电路图 (经过仿真验证基本正确)、对矩阵 X^2 模块进行封装,给出封装后的模块图;

(3) 给出矩阵 $X^2 A$ 计算模块的设计思路、给出 Logisim 软件绘制的电路图（经过仿真验证基本正确）、对矩阵 $X^2 A$ 模块进行封装，给出封装后的模块图；

(4) 给出矩阵快速幂算法迭代模块设计思路、给出 Logisim 软件绘制的电路图（经过仿真验证基本正确）；对矩阵 $X^2 A$ 模块进行封装，给出封装后的模块图；

(5) 说明主模块 main 中控制和显示部分的设计思路、给出主模块的 Logisim 软件绘制的电路图（经过仿真验证基本正确）。

7. 实验结果记录

根据下表中所列内容，记录相应信号作用后输出数码管显示数据，并填入表 6.1 中（注：要求 clear、clock 使用按钮输入）。

表 6.1 实验结果记录表

Input n	clear	1 st clock	2 nd clock	3 rd clock	4 th clock	5 th clock	6 th clock	After 6 th clock
2								
5								
10								
17								
25								
32								
44								
45								
46								
47								