

大中型光立方体及驱动组件建模与仿真

李红波, 李盛, 黄健

引用本文:

李红波, 李盛, 黄健. 大中型光立方体及驱动组件建模与仿真[J]. 液晶与显示, 2020, 35(9): 946–954.

LI Hong-bo, LI Sheng, HUANG Jian. Modeling and simulation of large–medium LED cube and driver components[J]. *Chinese Journal of Liquid Crystals and Displays*, 2020, 35(9): 946–954.

在线阅读 View online: <https://doi.org/10.37188/YJYXS20203509.0946>

您可能感兴趣的其他文章

Articles you may be interested in

固态体积式真三维立体显示器显示体驱动电路设计

Design of circuit driver for display–body used in solid volumetric true 3D display

液晶与显示. 2015(2): 263–268 <https://doi.org/10.3788/YJYXS20153002.0263>

固态体积式真三维立体显示器

Solid–state volumetric true 3D display

液晶与显示. 2015, 30(1): 137–142 <https://doi.org/10.3788/YJYXS20153001.0137>

固态体积式真三维立体显示效果优化

Optimizing effect of solid volumetric true 3D display

液晶与显示. 2016, 31(5): 518–523 <https://doi.org/10.3788/YJYXS20163105.0518>

光栅式自由立体显示器空间视区建模与串扰计算

Modeling and crosstalk calculation methods of spatial viewing zones for the autostereoscopic display device based on parallax barrier or lenticular lens

液晶与显示. 2017, 32(5): 394–401 <https://doi.org/10.3788/YJYXS20173205.0394>

环境光对固态体积式真三维显示闪烁的影响研究

Impact of ambient light on flicker in solid–state 3D volumetric display

液晶与显示. 2017, 32(12): 1006–1012 <https://doi.org/10.3788/YJYXS20173212.1006>

文章编号:1007-2780(2020)09-0946-09

大中型光立方体及驱动组件建模与仿真

李红波^{1*}, 李 盛¹, 黄 健²

(1. 西京学院 理学院, 陕西 西安 710123;

2. 西京学院 信息工程学院, 陕西 西安 710123)

摘要:针对大中型光立方体系统硬件设计的繁杂性和显示控制效果因缺少模型难以验证的问题,基于平行投影原理和 Proteus 软件的建模技术,提出并设计了光立方体虚拟仿真模型及其通用驱动组件模型。以 3D8 光立方体为例进行建模,并对驱动时序和字符图案显示进行了系统仿真验证。同时,根据该建模方法设计了 3D16、3D32 和 3D64 光立方体及驱动组件,仿真验证了汉字、点云模型的显示效果。结果证明:该建模方法能较好地反映光立方体系统的软硬件工作原理,为光立方系统设计验证提供了准三维虚拟可视化方法,为大中型光立方体的多样化显示、扩展以及三维立体显示器的研究和工程应用设计等提供了模型参考。

关键词:光立方体;可视化模型;三维显示;Proteus 软件

中图分类号: TN24; TP391.9 **文献标识码:** A **doi:** 10.37188/YJYXS20203509.0946

Modeling and simulation of large-medium LED cube and driver components

LI Hong-bo^{1*}, LI Sheng¹, HUANG Jian²

(1. School of Science, Xijing University, Xi'an 710123, China;

2. School of Information Engineering, Xijing University, Xi'an 710123, China)

Abstract: According to the complexity of the hardware design and the validation problem of display control without models in large and medium LED cube system, the LED cube virtual simulation model and its universal drive component model are proposed and designed using the principle of parallel projection and the modeling technology of Proteus software. The models are built by the example of 3D8 cube, and the driving timing and character pattern display are also simulated and verified. Based on the modeling method, 3D16, 3D32 and 3D64 LED cubes and driving components are designed, and the display effect of Chinese characters and point cloud models are also verified. The results show that the modeling method can reflect the operating principle of the software and the hardware for the LED cube system, as well as providing a pseudo-three-dimensional virtual visualization method for the LED cube's design and verification. The results can provide a modeling method for the

收稿日期: 2020-03-02; **修订日期:** 2020-03-26.

基金项目: 国家自然科学基金(No.11974289); 西京学院精品课程建设项目(No.JPK201909)

Supported by National Natural Science Foundation of China(No.11974289); Xijing University Excellent Course Construction Project(No.JPK201909)

* 通信联系人, E-mail: newboy311@163.com

diversified show and expansion of the large and medium LED cube, and the proposed method can be used for engineering application design of three-dimensional display.

Key words: LED cube; visualization model; 3D display; Proteus software

1 引言

光立方体显示是三维立体显示^[1-4]的一种基本实现形式,近些年在教学科研^[5-7]和亮化美化工程^[8-9]中有较多应用研究。目前小型光立方体设计^[10-12]比较成熟,由于其显示点数少,电路简单,能快速设计满足一定需要的光立方体。但大中型光立方体设计存在较多困难,由于其显示点数过大,硬件耗材消耗巨大,另外存在焊接工序过多、安装调试困难等问题,使得鲜有研究者尝试设计 $32 \times 32 \times 32$ 点及以上的光立方体,在一定程度上阻碍了大中型光立方体显示器的设计。本文提出基于 Proteus 软件的虚拟光立方体及驱动组件建模与设计,利用软件设计的可重复性、可扩展性、可编程性为大中型光立方体设计带来了全新的设计思路和工程验证模型。本文以 3D8 光立方体为例,详细阐述了其建模原理过程,并通过相同的设计方法实现了 3D16、3D32、3D64 光立方体模型建立和仿真验证,为科研显示、工程设计、三维虚拟仿真提供设计参考。

2 光立方体系统原理

如图 1 所示,光立方体系统由光立方体、显示控制器、层驱动器、显示缓冲器、行列驱动器阵列和 3D 模型库组成。光立方体由 LED 面阵层叠而成,LED 面阵由若干个 LED 灯排列成行列矩阵样式,同一面阵相同行和相同列 LED 灯的阴极和阳极分别连在一起,同时把不同面阵相同行 LED 灯的阴极连在一起并引出,这样共阴引脚的 LED 灯就组成了阴极显示层,由层驱动器驱动,显示层的全部 LED 灯阳极引脚作为数据引脚,连接到行列驱动器阵列,本文采用图 1 这种常见的阴极显示层驱动法。

显示控制器一般为 MCU、DSP 和 FPGA 等控制器,行列驱动器阵列由 74HC573 等锁存驱动芯片组成。提供功率输出的层驱动器由集成功率芯片组成,如 ULN2803 等。3D 模型库存放着要显示的三维点数据。光立方体工作时,由显示控

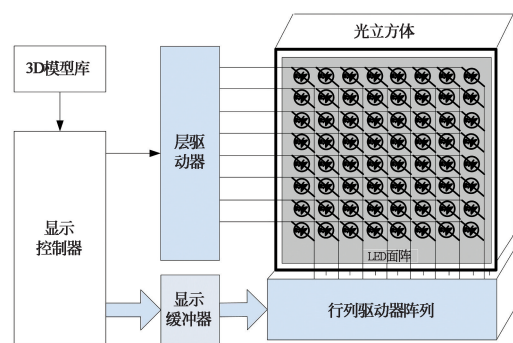


图1 光立方体系统框图

Fig.1 Block diagram of LED cube system

制器通过循环扫描方式依次驱动显示的层,并从模型库中读取该层点数据到显示缓冲器,由行列驱动器阵列根据缓冲器数值具体控制显示层 LED 灯的亮灭。由于人眼的视觉暂留特性,只要刷新频率足够大,就可呈现三维立体的显示。

3 Proteus 仿真模型介绍

Proteus 软件支持 SPICE、Schematic、Active 和 VSM 四种仿真模型。其中 Schematic 模型是由 Primitive 电路元件设计而成的电路模型;VSM 模型存在于 C++ DLL 动态链接库文件中,该模型功能由软件产生,设计灵活且支持高级仿真功能。本文所提及的模型由 VSM 模型和 Schematic 模型方法来设计。Proteus 模型设计包括两步:(1)模型原理图符号设计^[13-14];(2)模型功能设计^[13-14]。模型原理图符号是模型的 Proteus 电气元件符号。模型功能反映模型具有的功能特点,可由电路功能或软件功能实现。

4 光立方体原理图符号设计

Proteus 模型原理图符号是构建 Proteus 原理图的基础要素,是具有图形外观和电气功能的元件符号。

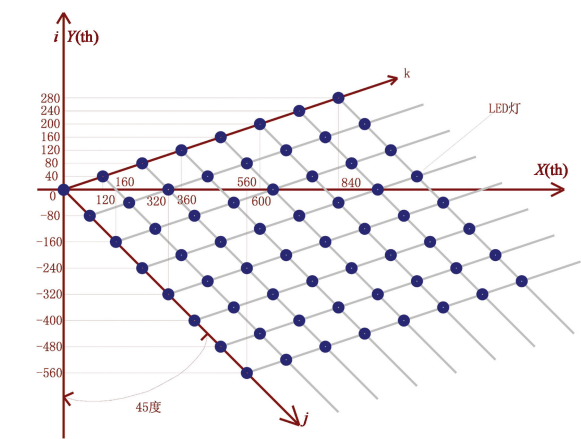
4.1 光立方体图形符号设计

首先,利用 Proteus ISIS 8 软件新建 Schematic

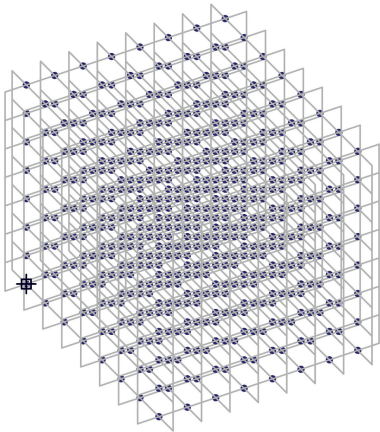
Design,设置视图分辨率栅格大小为 Snap 10 th。然后,利用 2D Graphics 绘图工具,绘制如图 2(a)所示的坐标轴和图形,其中 X 轴和 Y 轴相互垂直且原点和图纸中心点重叠, i 轴、 j 轴、 k 轴由原点引出, j 轴和 Y 轴逆时针呈 135° 夹角, k 轴和 X 轴夹角按照取 k 轴上坐标点的 XY 坐标为 3 : 1 设定, i 轴和 Y 轴重叠;平行于 j 轴和 k 轴分别放置 8 纵 8 横的线条,纵线与 k 轴相交的坐标点水平间距 120 th,横线与 j 轴相交的坐标点垂直间距为 80 th(th 为毫英寸缩写),线宽均为 5 th,线色均为 RGB (192,192,192),呈灰色;实心圆代替 LED 灯,圆半径为 10 th,圆线宽为 18 th,线色 RGB(0,0,128),呈蓝色,放置于纵横线交点上,共 64 点。接着,把图 2(a)中的 LED 点阵图形复制并沿 Y 轴正方向叠放,层间距为 120 th,共放 8 层。然后修边整理,添加 ORIGIN 符号原点_中到最下层的 j 轴和 k 轴交点上,如图 2(b)所示。最后,选择 Make Symbol 符号制作菜单,符号命名为 CUBE3D8_0,保存于用户图形库中。

4.2 光立方体原理图符号制作

在图 2(b)的基础上,添加白背景框和字符名称,重复操作 4.1 节最后两步,符号命名为 CUBE3D8_C。然后,添加 9 个 BUS 总线型输入引脚 L0[0..7]~L7[0..7]和 P[0..7],分别代表行列驱动信号和层驱动信号。选中 CUBE3D8_C 符号和全部总线引脚,通过 Make Device 元件制作菜单,参照表 1 设置元件属性值。最后,制作好的原理图符号如图 3 中的 3D8(LED1)元件所示。



(a)光立方体设计图
(a) Drawing of LED cube



(b)光立方体图形
(b) Figure of LED cube

图 2 光立方体图形符号设计
Fig.2 Drawing of LED cube symbol

表 1 元件属性
Tab.1 Component attribute

属性页	属性名称	属性值	设置说明
Device Properties	Device Name	3D8	元件名
Device Properties	Reference Prefix	LED	名称前缀
Device Properties	Symbol Name stem	CUBE3D8	图符名
Device Properties	NO. of States	2	状态数
Device Properties	Link to DLL	YES	连接库
Component Properties & Definitions	PRIMITIVE/Default Value	DIGITAL,CUBE3D8	基本属性
Component Properties & Definitions	MODDLL/Default Value	CUBE3D8.DLL	DLL 文件

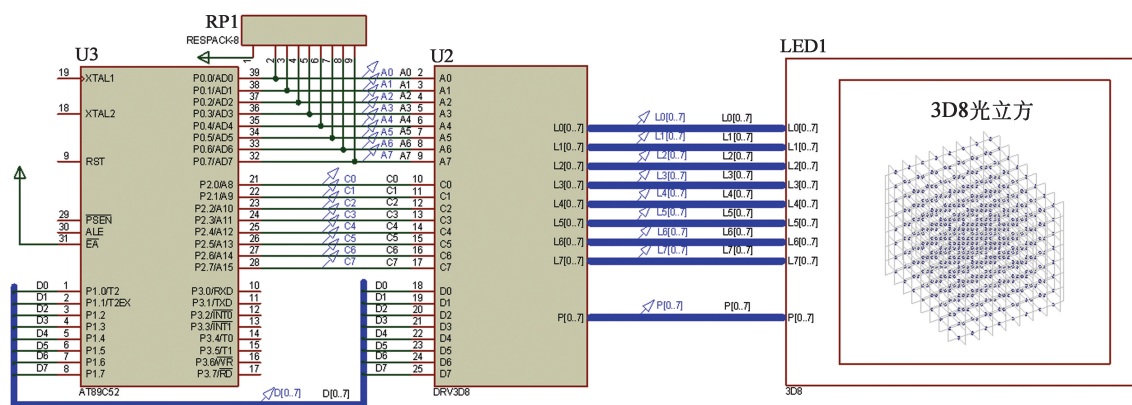


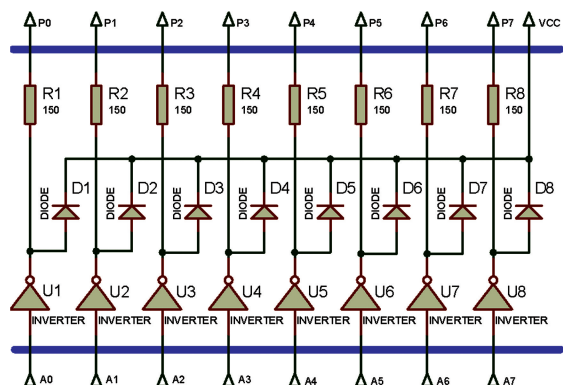
图3 仿真电路图

Fig.3 Diagram of simulation circuit

5 光立方体驱动组件设计

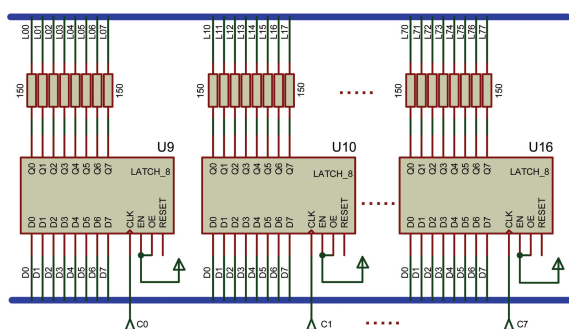
5.1 驱动组件电路设计

根据光立方体驱动电路原理,新建原理设计图,设计图4(a)和图4(b)的电路原理图。



(a) 层驱动器原理图

(a) Schematic diagram of floor drive



(b) 行列驱动阵列原理图

(b) Schematic diagram of ranks drive

图4 驱动原理图

Fig.4 Schematic diagrams of floor and ranks drive

层驱动器电路由 8 路 INVERTER (U1~U8) 缓冲驱动器、DIODE (D1~D8) 二极管和 RESISTOR (R1~R8) 限流电阻构成的子电路组成,模拟 ULN2803 芯片功能,A[0..7]是层驱动输入信号,P[0..7]是层驱动输出信号,二者信号逻辑值互补,且输出低电平信号有效。

行列驱动阵列电路由 8 路 LATCH_8 (U9~U16) 和若干 RESISTOR 限流电阻构成的子电路组成,模拟 74HC573 芯片功能,D[0..7]是输入数据信号,L0[0..7]~L7[0..7]是输出数据信号,C[0..7]是输出锁存信号,上升沿锁存输出。最后,编译模型并命名为 DRV3D8.MDF 文件备用。

5.2 驱动组件原理图符号制作

绘制如图 3 中 DRV3D8 (U2) 所示的原理图符号,A[0..7]、C[0..7]和 D[0..7]均为输入总线引脚,L0[0..7]~L7[0..7]和 P[0..7]均为输出总线引脚,用于连接光立方体 3D8 的输入总线引脚。通过 Make Device 元件制作菜单,参照表 1 设置元件属性值,其中:

“Device Name=DRV3D8”,

“Reference Prefix=U”,

“MODFILE/Default Value = DRV3D8.MDF”。

6 光立方体模型程序设计

Proteus 专为 VSM 模型支持了一套 C++ DLL 接口类虚函数,包括绘图模型实例 createactivemodel、initialize、animate、plot 等函数,数字电气

模型实例 createdsimmodel、setup、simulate、indicate、actuate 等函数。元件建模时需要在 DLL 中实现绘图模型(Graphical Model)接口虚函数和电气模型(Electrical Model)接口虚函数。绘图模型接口类用于实现动态人机接口功能,电气模型接口类用于实现模拟信号和数字逻辑功能。

6.1 软件原理与设计

6.1.1 坐标系

以惯用右手的方式确立三维取模坐标系和二/三维绘图坐标系,包括光立方体模型坐标系 $M(i', j', k')$, 光立方体绘图坐标系 $D(i, j, k)$ 和 $F(x, y)$, 如图 5 所示。模型坐标系由取模软件定义, 三维绘图坐标系是光立方体的数据坐标系, 二维坐标系是 Proteus 的显示坐标系, 三者的参考方向不统一, 需要坐标转换。为了方便分析三者坐标关系, 现将 D 坐标系和 F 坐标系组合在一起, 且坐标系 O 点都设在光立方体的 ORIGIN 符号原点处, 模型坐标系 O 点固定设在模型的最下层且与 ORIGIN 符号原点相对处, 如图 5 所示。以 3D8 光立方体模型为例, 光立方体绘图三维坐标系 $D(i, j, k)$ 和三维模型坐标系 $M(i', j', k')$ 的 O 点水平点距和垂直点距均为 7。于是, $M(i')$ 层数据对应 $D(j=7-i')$ 层数据, $M(j')$ 层数据对应 $D(k=7-j')$ 层数据, $M(k')$ 层数据对应 $D(i'=7-k')$ 层数据, 所以数据坐标点转换可由式(1)进行转换, 例如 $M(1, 0, 2) = D(2, 6, 7)$ 。

$$M(i', j', k') = D(k', 7-i', 7-j') \quad (1)$$

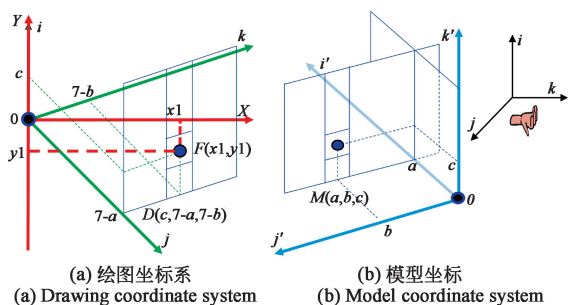


图 5 坐标关系图

Fig.5 Diagram of coordinate relation

同时, 由于 Proteus 的显示环境是二维的, 不能直接显示三维图形。所以, 利用图形变换中的平行投影原理^[15], 把光立方体的各个点平行投影到软件 Proteus 的二维显示平面上, 用二维平面点云显示在视觉上来代替三维图形, 从图 5(a)中

知, 三维绘图坐标系 D 的 O 点投影到二维显示平面 F 的 O 点。在构建 3D8 原理图符号时, 已知行列上下层各点之间的数据关系为常量, 不难推得三维数据点 $D(i, j, k)$ 点映射到二维数据点 $F(x, y)$ 的关系, 见式(2)和式(3)。

$$Fx = 80 \times j + 120 \times k \quad (2)$$

$$Fy = 120 \times i - 80 \times j + 40 \times k \quad (3)$$

其中, 式(2)中的 80 代表 j 行点的 X 方向偏移量, 120 代表 k 列点的 X 方向偏移量; 式(3)中 120 代表 i 层点 Y 方向的偏移量, 40 代表 k 列点的 Y 方向偏移量, 80 代表 j 行点的 Y 方向偏移量。例如 $D(5, 6, 7) \rightarrow F(1\ 320, 400)$ 。

6.1.2 数据结构

以 3D8 光立方体为例, 模型共有 512 个点数据, 需要 64 个字节存储, 于是建立 unsigned char 型模型库数组 CubeData[TYPENUM * 64], TYPENUM 为模型数量; 同时建立 unsigned char 型光立方体显示缓冲区 RAM3D8[8][8], 缓冲区共有 $8 \times 8 \times 8$ 个位, 对应需要点亮的 LED 灯的数量。

以常见的取模方式为例, 如图 5(b)所示。三维模型坐标系 $M(i', j', k')$ 中, 取模时按照坐标正方向依次遍历 $i' \rightarrow$ 遍历 $j' \rightarrow$ 遍历 k' , 得到 $8 \times 8 \times 8$ 个位数据, 灯亮时位数据取二进制数 1, 灯灭时位数据取二进制数 0, 再将遍历 k' 的数据位按照 k' 坐标轴正方向组合成字节数据(先遍历的为低位, 后遍历的为高位), 这样, 每 i' 层就有 8 个字节, 共 8 层, 总计 $8 \times 8 = 64$ 个字节, 按照遍历次序顺序存储在 CubeData 数组中。

光立方体绘图三维坐标系 $D(i, j, k)$ 中, 按照坐标正方向依次遍历 $i \rightarrow$ 遍历 $j \rightarrow$ 遍历 k , 得到 $8 \times 8 \times 8$ 个位数据, 再将遍历 k 的数据位按照 k 坐标轴正方向组合成字节数据(先遍历的为低位, 后遍历的为高位), 这样, 每 i 层就有 8 个字节, 共 8 层, 按照遍历层次序顺序存储在 RAM3D[i][0..7] 显示层 i 的 8 个字节存储单元中, 共 64 个字节。

6.1.3 坐标转换算法

根据数据结构原理和转换公式, 定义了模型数组 CubeData, 绘图数组 RAM3D 和 RAM2D, 得出坐标转换算法如表 2 所示。其中 MatrixChange1 函数算法(Step1~Step8)实现在显示控制器中, MatrixChange2 函数算法(Step9~Step10)实现在 VSM DLL 程序中。

表2 坐标转换算法描述

Tab.2 Description of coordinate transformation algorithm

坐标变换算法	MatrixChange1, MatrixChange2,
Input:	CubeData[64] //模型数组, byte
Output:	RAM3D[8][8], RAM2D[2][512] //目标数组, byte, int
Step1:	For i from 0 to 7
Step2:	For j from 0 to 7
Step3:	Tmp1 ← CubeData[8 * i + j] //取模数据字节赋变量 Tmp1
Step4:	For k from 0 to 7 //遍历每一位
Step5:	Tmp3 ← Tmp1.(k) //取 Tmp1 的第 k 位二进制值给变量 Tmp3
Step6:	Tmp2 ← RAM3D[8 * k][7 - i] //目标字节赋给变量 Tmp2
Step7:	Tmp2.(7 - j) ← Tmp3 //Tmp2 的第 7 - j 位二进制值用 Tmp3 填充
Step8:	RAM3D[8 * k][7 - i] ← Tmp2 //更新目标字节
Step9:	RAM2D[0][64 * i + 8 * j + k] ← 80 * j + 120 * k //计算点的 X 坐标
Step10:	RAM2D[1][64 * i + 8 * j + k] ← 120 * i + 40 * k - 80 * j //计算点的 Y 坐标

6.2 VSM DLL 程序设计

Proteus 模型中,光立方体中的 LED 灯用实心圆代替,所以,点亮控制可通过实心圆的颜色来表示,比如,红色表示点亮,蓝色表示熄灭。Proteus 的 VSM DLL 模型中调用 Draw3Dcube 函数,通过语句 drawcircle(X,Y,20)可以绘制半径为 20 th 的实心点,调用语句 setbrushcolour (RED)选择绘图颜色为红色。只要遍历 RAM3D 数组字节的每一位,根据位值是 1 就绘制红色实心圆,位值是 0 就绘制蓝色实心圆。在 VSM DLL 程序设计中,分为绘图和电气控制两部分,分别由绘图状态机和电气状态机进行描述设计,如图 6 程序状态机所示。

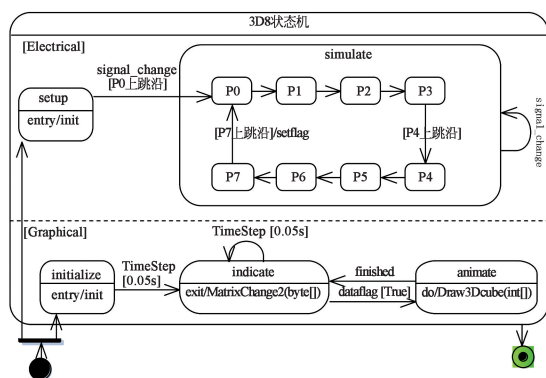


图6 程序状态机

Fig.6 Diagram of program state machine

程序运行后,分别进入 Electrical 初始化 setup 和 Graphical 初始化 initialize,之后电气状态机进入 simulate 组合态,该组合态根据 P0 总线信号的上跳沿,依次循环进入 P0→P1→P2→P3→P4→P5→P6→P7→P0 状态,当 P7→P0 切换时,通过 setflag 函数设置数据有效 dataflag。同时,图形状态机在 indicate 状态每隔 0.05 s 采样一次 dataflag,当条件成立即收到一完整数据帧,然后调用 MatrixChange 函数进行坐标转换,之后转移到 animate 状态调用 Draw3Dcube 函数,进行光立方显示,显示完成后,又进入前一次状态,等待数据有效。

7 光立方体及驱动组件仿真

在 Proteus 软件环境下,搭建的仿真电路如图 3,编制显示控制程序,编译链接后把生成的可执行文件加载进单片机中。同时,在电路中添加电压测试点探针,分别是 A0~A7、C0~C7、D[0..7]、L0[0..7]~L7[0..7]和 P[0..7]。控制器采用通用型单片机 AT89C52 (U3),工作频率为 11.0592 MHz,内部有 256 B 数据存储器和 8 kB 程序存储器,程序存储器固件中有 3D 模型数据。

7.1 驱动时序仿真

在仿真电路中,放置 DIGITAL ANALYSIS

仿真图表^[16],添加探针信号,设置仿真时长 2 s,仿真获取如图 7 所示的完整显示帧驱动波形。从波形分析知:(1) $A[0..7]$ 与 $P[0..7]$ 总线逻辑值互补,说明层驱动器信号正常。(2) $C[0..7]$ 总线中任一锁存信号 C_x (x 取值 0~7 的整数)出现上跳沿时, $D[0..7]$ 总线数据总会输出到 $L_x[0..7]$ 总线,说明行列驱动阵列信号正常。以上两点说明 DRV3D8(U2)驱动器电路模型工作正常。(3)系统工作时,从 $A7$ 信号的下跳沿事件开始,该事件既是上一显示帧的结束,同时也是下一显示帧的开始,如图 7 中的指示箭头 j 。当新的显示帧开始时,置 $A[0..7]$ 总线为 $0x00$,关闭光立方体显示。然后控制器驱动 $D[0..7]$ 数据总线连续输出 8 字节显示层数据,同时每个数据依次在 C_x 信号的上升沿得到锁存并输出到 $L_x[0..7]$ 总线,如图 7 中的指示箭头 k ($C3$ 信号上升沿锁存 $D[0..7]$ 总线数据 $0x08$ 到 $L3[0..7]$ 中)。在 $C7$ 信号的上跳沿完成锁存输出整个显示层数据,如图 7 中的指示框 o 中的数据。接着,置 $A[0..7]$ 总线为 $0x01$,开启光立方体第 1 层显示,如图 7 中的指示箭头 l 。显示一小段时间后,关闭光立方体第 1 层显示,即置 $A[0..7]$ 总线为 $0x00$,如图 7 中的指示箭头 m 。与此同时,开始第 2 层的显示流程,包括数据输出、数据锁存、开启第 2 层显示、显示延时、关闭第 2 层显示等。照此循环扫描显示第 3~8 层,直到第 8 层显示完成,如图 7 中的指示箭头 n 。至此,一个完整的光立方体显示帧就结束了。

从驱动时序分析可知,光立方体的驱动组件工作时序和控制时序相吻合,驱动电路模型工作正常。但需要说明,由于 Proteus 软件图形仿真

在运行速度、刷新频率和动态显示视觉效果不及真实情况,所以,仿真时序中要预留出软件绘制显示模型的时间,大约 50~100 ms,具体数值受计算机运行速度因素影响。

7.2 显示仿真

7.2.1 显示取模

3D8 光立方体显示取模已非常成熟,常用到的软件是 3D8S_alpha 上位机,该软件以三视图的形式平铺展开 8 个二维显示层,每个显示层呈现为 8×8 的二维点阵面,用户只需用鼠标在显示层上点击绘制显示内容,各视图面同步更新数据,同时提供了 8 行 $\times$ 8 列的字节模型数据,供用户建立模型库使用。本文利用该软件获取了“ABC”字符、“箭头”和“笑脸”图案模型数据。

3D16 光立方显示取模用到最多的软件是乐涛工作室开发的 3D16 光立方仿真软件,该软件集成有可视化 3D16 取模、模型素材导入、动画方式编辑、在线取模效果仿真、可执行程序代码自动生成等功能,可极大简化 3D16 光立方显示系统的软件开发,降低程序开发的难度和时间。但是,这款软件与乐涛工作室推出的光立方系统硬件绑定配套,所以,该软件不支持自行设计的光立方系统。文中 3D16 光立方的显示模型数据是利用汉字的二维模型数据和文中显示转换原理自行建立 3D16 字符模型的。

三维点云是刻画三维现实世界最直接和有效的表达方式。目前激光点云是最具代表性的三维数据,主要借助三维激光扫描仪获取原始数据,后经过点云数据处理获得。文中的云点仙人球和马三维点云数据模型来自于斯坦福三维模型数据库,模型经后期处理保存为文本文件,以方便程序读取。

7.2.2 模型显示

通过显示取模,将 3D 模型数据存入数组中,编制坐标变换程序和显示控制程序,编译生成 Hex 文件后,加载到仿真电路中进行仿真。3D8 光立方体显示效果如图 8 所示,光立方显示器在字符、图形图案等显示方面空间感较强,立体效果较好。为了呈现较强的立体表现感,在光立方体模型程序 DLL 中加入了色彩表现功能。

需要说明的是,3D8 光立方体点数少,在显示汉字、点云、三维立体等方面表现差,需要更多点的光立方体做显示器件。根据本文建模设计方法,文中设计了 3D16、3D32、3D64 光立方体显示

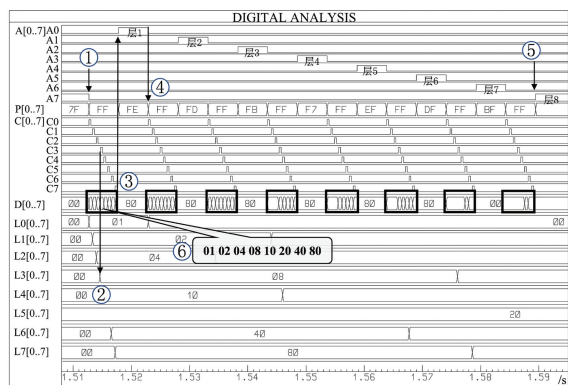


图 7 驱动时序图

Fig.7 Diagram of driving sequence

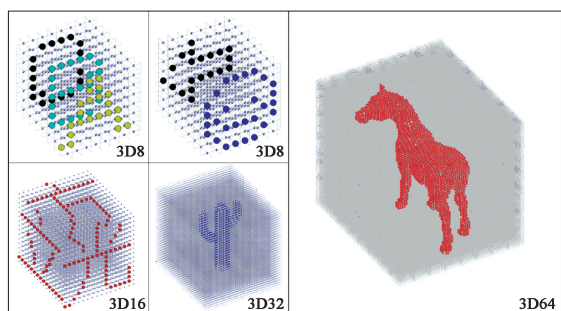


图8 显示效果图

Fig.8 Image of display effect

器件,并在 3D16 光立方体上显示汉字,在 3D32 和 3D64 光立方体上分别显示仙人球点云和马点云模型,效果突出,如图 8 所示。随着点数的增加,考虑光立方体模型设计、驱动配置方法、功率消耗、通讯设计、扫描方式等,使得光立方体显示系统设计越发显得困难。若把 3D8 光立方显示系统看成基本显示单元,大中型光立方体显示系统可由基本显示单元积木构建,如表 3 所示。

表3 光立方体显示系统配置表

Tab.3 Configuration table of LED cube system

光立方系统	每层点数	基本单元数	功耗/W
3D8	8^2	1	1.28
3D16	16^2	8^1	10.24
3D32	32^2	8^2	81.92
3D64	64^2	8^3	655.36
3D128	128^2	8^4	5 242.88
3D256	256^2	8^5	41 943.04

参 考 文 献:

- [1] 王琼华,王爱红.三维立体显示综述[J].计算机应用,2010,30(3):579-581,588.
WANG Q H, WANG A H. Survey on stereoscopic three-dimensional display [J]. *Journal of Computer Applications*, 2010, 30(3): 579-581, 588. (in Chinese)
- [2] 温才,唐丽红,赖永宽,等.晶格结构三维模型“LED光立方”的设计与制作[J].大学物理实验,2014,27(4):52-56.
WEN C, TANG L H, LAI Y K, et al. Design and preparation of LED light cubic model for showing three dimensional crystal lattice [J]. *Physical Experiment of College*, 2014, 27(4): 52-56. (in Chinese)
- [3] 陈晓西,李光勇,王继岷,等.基于液晶多层屏的 3D 显示系统和算法设计[J].液晶与显示,2017,32(4):302-307.
CHEN X X, LI G Y, WANG J M, et al. Liquid crystal multi-layer 3D display system and algorithm design [J]. *Chinese Journal of Liquid Crystals and Displays*, 2017, 32(4): 302-307. (in Chinese)
- [4] LI X, CHEN C P, GAO H Y, et al. Video-rate holographic display using Azo-dye-doped liquid crystal [J]. *Journal of Display Technology*, 2014, 10(6): 438-443.

表 3 中,功耗是按照单个 LED 发光管功耗为 20 mW,并在 1/8 扫描方式下,按层静态全点亮光立方体,并不计其他功耗的情况下计算所得结果。基本单元功耗 $P_{3D8} = N^2 \times p_{LED}$, N 为 3D8 光立方体的每层点数, p_{LED} 为单个 LED 发光管功耗,带入参数,于是 $P_{3D8} = 8^2 \times 20 \times 10^{-3} = 1.28$ W。另以 3D128 光立方系统为例,功耗 $P = P_{3D8} \times n$, P_{3D8} 是基本单元功耗, n 是基本单元数。于是, 3D128 光立方系统的功耗 $P_{3D128} = 8^4 \times 1.28 = 5\ 242.88$ W。

同时,根据表 3 数据可知,光立方系统每增加 1 级,基本单元数增加 8 倍,功耗增加 8 倍,如 3D256 光立方体显示系统需要 8^5 个基本单元构成,功耗超 40 kW,这将会面临巨大工程量和复杂电源设计等问题。所以,光立方体显示系统的设计上限受到了现实条件的制约。

8 结 论

仿真结果表明,通过光立方体及驱动组件模型仿真,模型设计正确,仿真运行立体效果好。该模型方法的提出简化了光立方体及驱动组件的硬件设计,节省了硬件成本、剔除了焊接工序、极大地缩短了设计周期等,在一定程度上避免了硬件设计的困难,不仅为大中型光立方体设计验证提供了一种准三维可视化设计方法,而且为三维立体显示器的设计在软硬件协同验证提供了参考,为光立方系统的工程应用奠定基础。

- [5] 邓立. 基于辅助教学的 $\mu\text{C}/\text{OS-II}$ 光立方三维控制设计[J]. 数字技术与应用, 2014(9): 12.
DENG L. 3D control design of $\mu\text{C}/\text{OS-II}$ LED cube based on assistant teaching [J]. *Digital Technology and Application*, 2014(9): 12. (in Chinese)
- [6] 贾高杰, 邱崧, 蔡茗名, 等. 三维点云重构和体显示在医学辅助诊断中的应用[J]. 中国医学物理学杂志, 2016, 33(6): 593-598, 618.
JIA G J, QIU S, CAI M M, *et al.* Application of three-dimensional point cloud reconstruction and swept volume display in medical assistant diagnosis [J]. *Chinese Journal of Medical Physics*, 2016, 33(6): 593-598, 618. (in Chinese)
- [7] 陈琳, 李洁. 基于虚拟现实技术的三维影像智能显示系统嵌入式设计[J]. 现代电子技术, 2017, 40(8): 100-102, 106.
CHEN L, LI J. Design of 3D image embedded intelligent display system based on virtual reality technology [J]. *Modern Electronics Technique*, 2017, 40(8): 100-102, 106. (in Chinese)
- [8] 谭泽洋. 吸顶灯(光立方): 中国, CN305532511S[P]. 2020-01-03.
TAN Z Y. Lamp affixed to ceiling (light cube): CN, CN305532511S [P]. 2020-01-03. (in Chinese)
- [9] 肖展超. 3D 动画效果展示柜(光立方): 中国, CN305281396S[P]. 2019-07-30.
XIAO Z C. 3D Animation effect show cabinet (light cube): CN, CN305281396S [P]. 2019-07-30. (in Chinese)
- [10] 唐康, 刘耀联, 韩帅, 等. 基于 STM32 的 3D16 光立方设计[J]. 电子技术与软件工程, 2015(23): 119-120.
TANG K, LIU Y L, HAN S, *et al.* Design of 3D16-LED cube based on STM32 [J]. *Electronic Technology & Software Engineering*, 2015(23): 119-120. (in Chinese)
- [11] 康志强, 汪佳, 汤勇明. 基于 FPGA 的 3D 光立方设计[J]. 电子器件, 2012, 35(6): 683-686.
KANG Z Q, WANG J, TANG Y M. Design of LED cube based on FPGA [J]. *Chinese Journal of Electron Devices*, 2012, 35(6): 683-686. (in Chinese)
- [12] ŽÁKOVÁ K. The use of 3D LED cube for basic programming teaching [J]. *IFAC-PapersOnLine*, 2016, 49(6): 203-206.
- [13] 朱清慧, 张凤蕊. 基于 Proteus 的虚拟液晶触摸屏设计与应用[J]. 液晶与显示, 2009, 24(4): 562-566.
ZHU Q H, ZHANG F R. Design and application of virtual liquid touch panel based on Proteus [J]. *Chinese Journal of Liquid Crystals and Displays*, 2009, 24(4): 562-566. (in Chinese)
- [14] 周灵彬, 张靖武. 创建 Proteus 动态器件仿真模型的技术[J]. 现代电子技术, 2009, 32(12): 152-155.
ZHOU L B, ZHANG J W. Technology of dynamic simulation model for Proteus [J]. *Modern Electronics Technique*, 2009, 32(12): 152-155. (in Chinese)
- [15] 韩鲁佳, 李松维. 投影变换的原理及其应用浅析[J]. 科技视界, 2015(26): 171-174.
HAN L J, LI S W. The principle and application elementary analysis of projection transformation [J]. *Science & Technology Vision*, 2015(26): 171-174. (in Chinese)
- [16] 王超, 宋莉莉. Proteus 仿真图表在单片机时序分析中的应用[J]. 实验室研究与探索, 2014, 33(9): 93-97.
WANG C, SONG L L. Application of Proteus simulation diagram in MCU timing analysis [J]. *Research and Exploration in Laboratory*, 2014, 33(9): 93-97. (in Chinese)

作者简介:



李红波(1982—),男,陕西西安人,硕士,讲师,2009 年于西南交通大学获得硕士学位,主要从事智能控制和嵌入式技术方面的研究。E-mail: newboy311@163.com