Modeling and Simulation of Proteus Electrical Simulation Model CT1628
Da-ming PEI
Guangzhou Civil Aviation College Guangzhou Guangdong China

Keywords: Proteus, Electrical simulation model, CT1628; DLL.

Abstract. A Proteus electrical simulation model for CT1628 is implemented based on the study of Proteus C++ programming interface—DLL. The test results show that the CT1628 simulation model can identify the key and can control the LED digital tube, providing a more convenient and intuitive platform for the functional test of the chip, which is beneficial to the software and hardware co-design of the chip development.

Introduction
With the rapid development of computer technology, as a new kind of simulation system, Proteus has been applied to the electronic system design process, not only provides a complete virtual laboratory for single-chip microcomputer teaching and research, and completely change the traditional design scheme: first the test, then improvement and then validation, shorten the research and development process of electronic systems products, save the development cost, promote the improvement of the electronic system design.

However, in the process of Proteus simulation, there will be a problem: practical device and Proteus simulation model mismatch and even some components without the simulation model, there are two ways to solve the problem: one is to use the development board and the emulator, to complete the project development by means of adding hardware circuit; The other is to design corresponding simulation model according to the properties of the device. This paper, by using the second method to create a CT1628 Proteus electrical simulation model based on dynamic link library.

Proteus VSM Simulation Model and CT1628 Profile
In the process of SCM system simulation using Proteus, often can not find the required components, which requires to write your own. A major feature of Proteus VSM is its extensibility through the use of DLL based component models. These models are divided into two categories: Electrical Model and Graphical Model. Electrical model to achieve the electrical characteristics of components, according to the time sequence of the received data and output data; Graphical model to achieve simulation and user interaction, such as LCD display. These models can be purely electrical, or can combine electrical and graphical behavior to allow user interaction with the simulation.

The VSM API \(^{1}\) draws heavily on the concepts underlying COM architecture but does not implement it fully. Specifically, all the major VSM interfaces are implemented as C++ abstract classes. Users create components need to implement the corresponding abstract class in DLL. The Figure 1 provides an overview of how a VSM model communicates with the rest of the Proteus System. The arrows indicate the direction in which function calls are made. electrical part of a model communicates with the PROSPICE simulator kernel, while the graphical part of a model communicates with ISIS\(^{2,3}\).

The Graphical Model Interface consists of two interface classes. Class ICOMPONENT represents an Active Component object within ISIS and provides services which allow a VSM model to draw on the schematic and interact with the user.

Class IACTIVE MODEL represents a base class from which to derive your VSM graphical models. Users to achieve the VSM graphical model need to inherit this class, and to achieve the corresponding graphics and keyboard mouse event processing.
The Electrical Model API consists of the following interface classes.

Class IINSTANCE represents a simulator primitive with PROSPICE and provides services which allow a VSM model to access its properties, analogue nodes and digital pins. It also allows a model to report warnings and errors through the simulation log.

Class ISPICECKT represents the analogue parts of the circuit as held by SPICE. It provides services for accessing, creating and deleting nodes, and for allocating space within the sparse matrices. It also allows a model to force simulation time points to occur at specified times, and to suspend the simulation.

Class IDSIMCKT represents the digital parts of the circuit as held by DSIM. It provides access to DSIM system variables. It also allows a model to create callback events and to suspend the simulation.

Class IDSIMPIN represents a digital component pin as held by DSIM. It provides services for examining the current and previous states of the pin, and for creating new output transition events.

Class ISPICEMODEL provides a base class from which to derive models which exhibit analogue behavior. You are required to implement functions for loading admittance and current values into the sparse matrices, accepting or rejecting a proposed time step, and processing data from completed time points.

Class IDSIMMODEL provides a base class from which to derive models which exhibit digital behavior. You are required to implement functions for determining the effect state changes on the model's pins and for processing callback events.

Class IMIXEDMODEL is a multiple inheritance of ISPICEMODEL and IDSIMMODEL and provides a base class for components which exhibit both analogue and digital behavior.

In order to allow Proteus to access the member functions of a user model, must create an instance of user model. This cannot be achieved through the class interface, can only be derived from the DLL several C functions to achieve, in the user model must be implemented in these C functions, to achieve the results of the construction and analysis of the user model.

(1) Constructor and deconstructor for Graphical Model instance

IACTIVEMODEL *createactivemodel (CHAR *device, ILICENCESERVER *ils)
void deleteactivemodel (IACTIVEMODEL *model)

(2) Constructor and deconstructor for analogue electrical model instance

ISPICEMODEL *createspicemodel (CHAR *device, ILICENCESERVER *ils)
void deletespicemodel (ISPICEMODEL *model)

(3) Constructor and deconstructor for Digital electrical model instance

IDSIMMODEL *createdsimmodel (CHAR *device, ILICENCESERVER *ils)
void deletesimmodel (IDSIMMODEL *model)

(4) Constructor and deconstructor for mixed model instance

IMIXEDMODEL *createmixedmodel (CHAR *device, ILICENCESERVER *ils)
void deletemixedmodel (IDSIMMODEL *model)
CT1628 is a kind of LED (light emitting diode display) driver control circuit with keyboard scanning interface, which is integrated with MCU digital interface, data latch, LED high voltage driver, keyboard scanning circuit and so on. Using SOP28 package form. Mainly used in DVD, induction cooker, set-top boxes, air conditioners, refrigerators and home theater products such as digital display driver.

CT1628 Electrical Simulation Model Development

The development of Proteus VSM simulation model, first drawing element graphics, pins and related symbols, and making elements, setting element attributes, then write components with C++, and realizes the electrical drawing model, the compiler generates DLL, the end is to build the circuit simulation test. The following is an example of CT1628 components to explain in detail the development process. Drawing component graphics, pins and related symbols, set the component attributes can refer to references 4[4]. A CT1628 symbol is shown in Figure 2.

![CT1628 Symbol](image1.png)

![CT1628 Simulation Diagram](image2.png)

Among them, GND, VDD electrical type select PP-Power Pin, DIO electrical type select IO-Bidirectional, CLK, STB electrical type select IP-Input, SG1~SG14, GR1~GR7 electrical type select OP-Output, NC, GND, VCC need to hide, so ‘Draw body’ do not choose. After the configuration is complete, select the components, and then make the components, the components are linked to the CT1628.DLL.

The realization of the electric model, can be implemented by the mean of writing device driver in a high-level language, compiled DLL, complete data exchange by the call of Proteus when simulation. The dynamic link library developed by VC++ platform, from the Proteus installation directory INCLUDE folder copy the virtual system model interface header files VSM.HPP to the current project directory, and then write CT1628 component code.

CT1628 components have only digital electrical characteristics, no the Graphical characteristics, so only to inherit the IDSIMMODEL. Create a class CT1628, public inheritance IDSIMMODEL and implement the member function:

The electrical model member function:
```c++
int isdigital (CHAR *pinname);// Digital circuits always return TRUE
    void setup (IINSTANCE *inst, IDSIMCKT *dsim);// When creating a model instance is called, do initialization
```
void runctrl (RUNMODES mode);// The simulation operation mode control, at the beginning of each frame in the interactive simulation is invoked
void actuate (REALTIME time, ACTIVESTATE newstate);// Interactive simulation when the user changes the buttons and other state is invoked
bool indicate (REALTIME time, ACTIVEDATA *data);
// Interactive simulation at the end of each frame is called, passing ACTIVEDATA communication data and drawing model, thus invoking the animate () for drawing
void simulate (ABSTIME time, DSIMMODES mode);// called when pin state changes, mainly used for processing the data input and output
void callback (ABSTIME time, EVENTID eventid);// Through setcallback () sets the callback function called at any given time

Because the CT1628 has only electrical characteristics, it does not have to interact with the user, so the member function, runctrl(RUNMODES mode), actuate (REALTIME time, ACTIVESTATE newstate), indicate (REALTIME time, ACTIVEDATA *data) three bodies are empty; function isdigital (CHAR *pinname), return 1; In setup (IINSTANCE *inst, IDSIMCKT *dsim) function, associate the digital pin with the member variables and set the pin status; function simulate (ABSTIME time, DSIMMODES mode) mainly used for data input and output; function callback (ABSTIME time, EVENTID eventid) to implement repeating events such as clock generators.

Building Simulation Test Circuit
In Proteus using AT89C51 as the main control chip to build a display circuit, new a project in the Keil platform[5], write test code, load HEX files to the AT89C51, run the simulation program, the results are shown in Figure 3, digital display is the state of second and fifth keys pressed, CT1628 simulation model can identify key and can control the digital display correctly.

Conclusions
Based on the commonly used MCU and its peripheral circuit simulation application in proteus, aim at the problem of the existing simulation models in practical application and the actual device does not match, this paper proposed a design scheme of Proteus VSM simulation model based on high-level languages. By writing the driver, the data of Graphical model completes the simulation of the real time exchange, code written by Keil C software, build a testing system, verify the availability of VSM model. The actual application results show that the method developed by VSM module practicality is strong, good generality, VSM model design is ideal, and largely reduce the development period for hardware dependent.

References