7L－5 Prototyping for Simulation Debugging Environment: An Enhanced Developing Method for Embedded Computer Software

Zhongmin YU and Yoshinao AOKI
Faculty of Engineering, Hokkaido University

1 Introduction

Embedded computers(EC) have been used widely in the world, however, embedded computer software(ECS) are still difficult to develop, because the lack of suitable debugging environment. With the traditional in-circuit debugging method, it is difficult to analze the reason of an encountered error, also it is impossible to debug an ECS before the implementation of the associated hardware. Moreover, the creation of such hardware costs both money and time, so that it can hardly keep up with the ever-changing of a great number of ECs. To tackle this problem, we establish a rapid prototyping system for simulation debugging environment (SDE) to replace the hardware debugging environment. With the system, users can create a SDE at a higher speed and lower cost, the development efficiency for ECS can also be increased greatly.

2 Concept of SDE

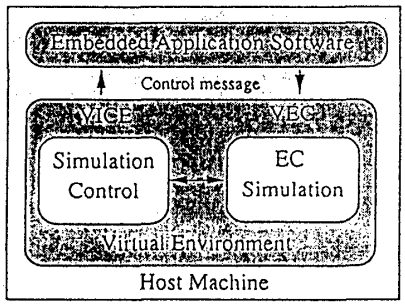

figure.1 Conceptual model for the virtual environment

As shown in figure.1, SDE supports a virtual environment in the host machine, this virtual environment includes a virtual EC(VEC) and a virtual ICE(VICE). By the virtual environment, SDE can provide sufficient functions to support debugging activities in an ordinary computer where no in-circuit hardware is needed. Users can run the program, watch the state change of the EC or trace an output action of one I/O signal. Users may also use copies of SDE to debug different modules or programs at the same time. In addition, 0/1 sequence file can also be used as input signal for execution of a program. All the above functions will enable users to test their application software program for correctness in functionality and timing just as in a true hardware debugging environment, even if there is no in-circuit hardware debugging environment supported, users can evaluate their software design, reduce later debugging schedules and isolate faults before run-time application of the software and finally lead to get a higher qualified computer software.

3 Prototyping for SDE

To get a SDE quickly, we have created a prototyping environment called SPACE(*SDE Prototype Automatic Construction Environment*), which aims at the rapid prototyping for debugging environment of an embedded software instead of hardware circuit.

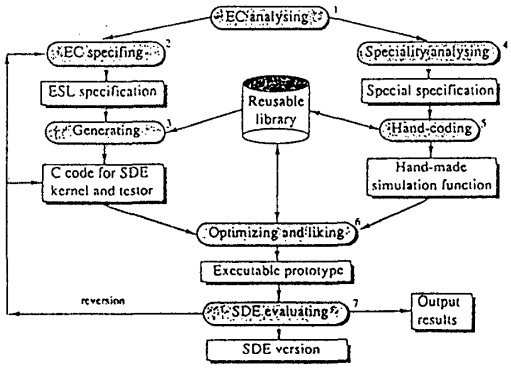

figure.2 Prototyping flow in SPACE

SPACE mainly includes an EC specification language(ESL), a template-oriented ESL editor and a prototype generator, it provides users with a convenient environment for specifing and prototyping. To reduce some tedious programming work, a reusable library is created, it can be accessed both in step of

translating an ESL specification into C code and in step of writing a program, of course, the library can be enriched by user.

4 Description of EC

We have made a small language called ECL for specifing the high level functional behavior of EC, In ESL, we define an EC as several classes, each class has many objects corresponding to the entities in EC, we design ESL to make it fit for specifing the type and attributies of each object to provide SPACE with necessary information for prototyping.

To help user write an errorless specification, we have made a template oriented editor for ESL, in this editor, each class of an EC is assigned with one kind of definition window called *"template"*, the definition of the attributes for an object is done in the associated template. The main role of a template is to check the input data and direct the user in his work, whenever a wrong data which is beyond the limitation of the template is inputted, the template would not confirm it and some warning or guiding messages will be given. In the end, the editor will give out a specification file with extended name of ".ESL".

5 Generator and testor

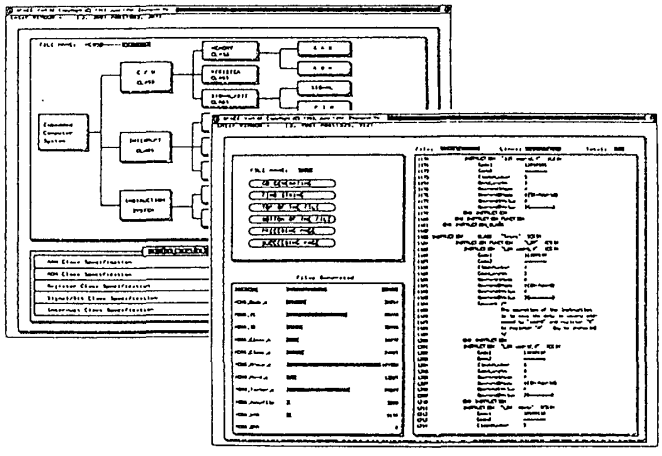

figure.3 The generator window

The generator scans the ".ESL" file for two times to generate the data framework for attributes of objects and the C programs for simulation kernel and testor. As to some particular operations, SPACE generates a function frame with sufficient comment information, users have to refine those frames by adding some handmade code into it, after combining the manual code with the generated program, a prototype of SDE will be established.

Program for prototype testor is also an output of the generator, it can give user a group of test-tun results, from which, the user can easily evaluate the generated prototype, if some errors are found, the user can change their ESL specification or the C file and repeat the previous process.

6 Implementation and conclusions

We have made an experimental system on X11 environment, as experiments, we have used it to create SDE prototypes for 4 and 8 bits microcomputers as NEC-75xx, MC68HC05 and NEC-78k. We are satisfied with the results, because the generation rate is considerably high and the generated C files are clear in structure and easy for reading and refining. We are making efforts toward those problems, a number of future enhancement is planed.

References

[1] Resve A. SALEH et al.: Enhanced Circuit Simulation: Expectations, Problems, Implementation and Integration, *Trans. IEICE*, Vol. J74-A, No.8, 1991

[2] Richard Hartley, et al.: A Rapid-Prototyping Environment for Degital-Signal Processors, *IEEE Design and Test of Computer*, 1991

[3] Luqi: Computer-Aided Prototyping for A Command-and-Control System Using CAPS, *IEEE Software*, 1992

[4] R. Camposano: From Behavior to Structure: High-Level Synthesis, *IEEE Design and Test of Computers*, 1990