The Optimization of Distributed Cloud Computing Network Routing Model Based on MDA

Yinzhen Zhong[1,a], Wuxue Jiang[2,b], Shanshan Ji[1,c]

[1]Department of Finance, Dongguan Polytechnic, Dongguan 523808, Guangdong, China

[2]Department of Computer Engineering, Dongguan Polytechnic, Dongguan 523808, Guangdong, China

[a]zhongyz@dgpt.edu.cn, [b]jiangwuxue_1@163.com, [c]280934652@qq.com

Keywords: Model, Model Driven Architecture, Distributed system, Cloud computing, Routing

Abstract. The optimization of distributed cloud computing network routing model has become an important way to improve the efficiency of data access. It can realize data information docking through function of C-API, the adopted distributed computer network routing environment can optimize the supply management of routing control, so as to write test bench and implement distributed response for data. The research on the technology of MDA distributed cloud computing network as well as methodology can improve the optimization function of distributed cloud computing network routing.

Introduction

A basic distributed cloud computer network routing platform should be consisted of the following parts: a driver, which is used to apply the different incentive measures to MDA; a monitor, which is used to monitor the output of MDA; a scoreboard, which is used to compare the expected value with the special output of MDA got by the monitor; a reference model, the input of which is exactly the same as MDA, while its output can apply to the scoreboard, which is used to compare the output of MDA.

In the function of the distributed cloud computer network routing process, the outside of tested design (MDA) can set up test-bench. The incentive can be sent to the input of the design through the distributed cloud computing network routing platform. Finally, the output will be compared to see whether the result is correct.

During the distributed cloud computing network routing works, to write test-bench is the major task for the distributed cloud computing network routing. The result and efficiency of the whole distributed cloud computing network routing process is determined by the quality of the test-bench. In the process of writing test-bench, the staff of the distributed cloud computer network routing should pay close attention to these problems which mainly lie in two aspects: firstly, what kind of incentive should be imposed by the input of the design; secondly, it should check the result after the output terminal generates response. In addition, test-bench must be reusable and easy to be used.

The main functions of the distributed cloud computing network routing platform are as follows:

1. Generating incentives.
2. Applying the incentive to the tested design.
3. Checking out whether the result and the distributed cloud computing network routing test is passed, namely, ensuring that the output of the tested design is the same as the expectation.

In the distributed cloud computing network routing platform, MDA should be instantiated firstly, so that it can be connected with each module of the platform. Secondly, when the function of the sub module is conducted with the distributed cloud computer network routing, it

should mainly seize the most important functions and key points, then it can write the corresponding incentive.

After writing incentive according to the key function points of each sub module, it needs to compile, then it can generate the target code, using simulation software to simulate code of the running situation in the actual chip, after the end of the simulation, it can generate the waveform and log file of the corresponding different function module, moreover, it should check and determine whether the system exists an error by checking the waveform, code and schematics.

When the function simulation is carried out, the incentive is compiled by the compiler tool (CodeWarrior), after the compiling, it can simulate through the simulation software (NC-Verilog), after the simulation, it can check the result of the simulation. If the result shows that there is a problem, tracing the root of the problem, so that it can be timely modified.

Design of the Distributed Cloud Computing Network Routing Algorithm of C-API

C-API can quickly realize the interaction between the incentive of design and distributed cloud computing network routing platform, when it is used, it can only call the defined and achieved Macro or function. This way can omit the function of reading, writing or communication when it must use C language to describe for each time, which can greatly improve the efficiency of the work.

In the following, it can use some simple examples to illustrate.

```
INFO(STIMNAME, "Check register reset value");
/* The control register reset value of distributed cloud computing network routing EWM */
EWMxCTRL_reg_check(&ewm_context);
```

The above code is written in testbench, which is the incentive of control register reset value used to test the external watchdog (EWM).

The specific definition and implementation of function EWMxCTRL_reg_check () can be shown as follows:

```
void EWMxCTRL_reg_check(EWM_CONTEXT* ewm_context)
{
  RE8(EWMxCTRL, 0x00);
}
```

The specific analysis of the above codes are as follows:

(1) INFO(STIMNAME, "Check register reset value")

The function of this function is to output string "check register reset value" on the screen, in the incentive EWM module it can insert this function, before checking out whether the register reset value is correct, which can help engineers to position and trace the corresponding code segment during the period of adjusting.

(2) RE8（EWMxCTRL, 0x00）

Among this function, EWMxCTRL is the control register address of EWM module in the chip, the expected value is 0x00, firstly it can read the value from control register of EWM, then it can compare this value with the expected value 0x00, if they are the same, it is true, otherwise it is false. The purpose of the function is to check out whether the value of the control register reset of the distributed cloud computing network EWM is 0.

The Adopted Distributed Cloud Computing Network Routing Environment

(1) The Design of Distributed Cloud Computing Network Routing Platform

In this paper, the distributed cloud computing network routing platform is based on the Incisive distributed cloud computing network routing platform of Cadence Company.

Incisive can provide the fastest and most efficient way, namely, the large complicated chip distributed cloud computing network routing. Incisive can accelerate the complex design of distributed cloud computing network routing, through setting, it can combine the unified tool, language, IP, assertion, debugging and coverage, and so on. Using this platform can greatly save the distributed cloud computing network routing time so that it can improve the distributed cloud computing network routing efficiency.

(2) Distributed Cloud Computing Network Routing Tool

EDA tools company such as Synopsys Company, Cadence Company is the current mainstream company for the distributed cloud computer network routing tool software as well as the development of IC design. The EDA software tools provided by them can cover all designing level and process, this function is very strong, the operation is very convenient and quick.

Here the distributed cloud computing network routing tools used in this paper will be briefly introduced.

a. Simulation Tool NC-Verilog

NC-Verilog is a compiler simulator, which is a custom simulator that can compile the Verilog code into a Verilog program. That is to say it can convert the Verilog code into a C program, then the C program is compiled into a simulator. The working process of NC-Verilog simulator can be shown as follows: firstly, it can compile the source file, then it can describe the design into the form representation form of design by the descriptor, so as to produce intermediate target by the editor and descriptor, then it can put the target into the library and conduct simulation.

It is the world's first design with high speed performance of the distributed cloud computing network routing tool, which is used to complete the simulation of the hardware circuit.

Personally speaking, its shortcomings is that the editor interface of Verilog code is not friendly, which needs to be improved.

b. Debugging Tools Debussy

Debussy tool is also introduced by Cadence Company, which is a very good debugging tool that can help designers quickly understand the complicated design. It can provide waveform, schematic and tracing source code as well as view and other functions, which can be easily position the presence bug of the design and analyze the possible errors, improve the efficiency, shorten the time for products to hit the store shelves.

c. C/C++ Compiler Tool CodeWarrior

Through it, the incentive written by C language can be debugged and compiled, so as to complete the simulation of the distributed cloud computing network routing.

d. Distributed Cloud Computing Network Routing Language Design

In this paper, Verilog and C programming language are adopted to coordinate the distributed cloud computing network routing. System-Verilog is used to compile IP design, distributed cloud computing network routing platform development, monitor, assertion and coverage rate and so on, while Verilog and C language can be used to compile various functional models as well as direct incentive.

MDA Distributed Cloud Computing Network Routing Technology and Optimization of Methodology

(1) The Distributed Cloud Computing Network Routing Technology Based on Simulation

MDA was proposed by OMG the Object Management Group in 2001. MDA can divide the model of software system into: PIM (Platform Independent Model) and PSM(Platform Specific

Model), moreover, they can be linked up through the corresponding transformation rule. Therefore, SOA modeling process based on MDA can be shown in Fig.1.

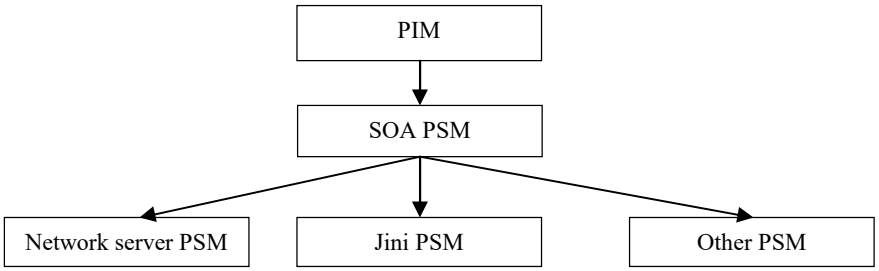

Fig. 1 SOA Modeling Process Based on MDA

Among the simulation tools, each evaluation of the next state is called deltacycle (short time segment). Simulation can contain a consecutive series of deltacycle evaluation calculation, namely, the signal can be scheduled in a time of the assignment operation. When there is no future assignment operation to be scheduled (for example, no design signal will be changed), then it can complete the simulation. Simulation can also be stopped by the program and tool control mechanism to stop.

Distributed cloud computing network routing environment can include distributed cloud computing network routing platform and design. Among the distributed cloud computing network routing based on simulation, the basic operation is through the distributed cloud computing network routing platform to apply the incentive data to the design, so as to calculate and design the value of the next state, checking out whether the next state can meet the expectation of design.

Commonly used simulation techniques are as follows:simulation based on cycle, simulation based on event, soft / hardware coexisted distributed cloud computing network routing, distributed cloud computing network routing based on event, simulation of mixed signal, etc..

In the second layer multi casting, it can not handle Exclude mode source, all Exclude mode sources are equivalent to Exclude(NULL), therefore, in the report of IGMPv3、MLDV2, when Record TYPE is IS_IN、TO_IN（S）、ALLOW type, it means (S, G) is added,while TO_EX、IS_EX means (*, G) is added,thus, TO_IN(NULL) means(*, G) is leaving, BLOCK means(S, G) is leaving.

(2) UVM Distributed Cloud Computing Network Routing Methodology

UVM is the abbreviation of Universal Verification Methodology,namely,the universal distributed cloud computing network routing methodology. It is originated from OVM(Open Verification Methodology), which is a new generation distributed cloud computing network routing methodology launched by Cadence, Mentor and Synopsys jointly. UVM is mainly used for the correctness of digital logic circuit of the distributed cloud computing network routing.

UVM is a base built on the SystemVerilog platform, it provides a series of interfaces, which can carry out the distributed cloud computing network routing more conveniently. The most basic purpose of the vase is to facilitate the use of people. The base can hide these basic and often used details, which can compile the distributed cloud computing network routing platform more conveniently.

In order to understand the UVM distributed cloud computing network routing methodology more clearly, two examples can be cited to illustrate.

As shown in the first example below, when the information is printed it can output time at the same time, in SystemVerilog, it can only call time function in the state of display. But in uvm, as

long as it used uvm_warning, uvm_error, uvm_info or macro etc., UVM will help you to add time automatically.

Example 1:

/ / in SystemVerilog

$display (“DRIVER: @ %0t, Cannot drive the bus”, $ time);

/ / in UVM

`uvm _warning(“DRIVER”, “cannot drive the bus)

In the second example, it often prints some characters when it makes mistakes, when the problem is resolved, these statements do not appear, therefore, people have to delete the display statement manually.But in UVM, by setting the verbosity characteristics of uvm_info macro, it can set that in normal operation, the information needs not to be printed, it can be printed only when it is debugging, therefore, there is no need to delete these statements.

Example 2:

/ / in SystemVerilog

$display (“AAAAA”);

/ / in UVM

`uvm _info (“DRIVER”, “AAAAA”, UVM_DEBUG)

In UVM distributed cloud computing network routing, the driver, monitor, model, scoreboard and other components are all implemented by a class. monitor is used to monitor MDA output; scoreboard is used to compare the expected value with MDA output got by the monitor specifically; reference model, its input is the same as MDA, its output can be given to scoreboard, which is used to compare with MDA output.

Besides class function, it can also have task, through these function and task, it can not only complete the output incentive function of driver, but also can complete the monitoring function of monitor, the calculation function of model, the comparison function of scoreboard as well, class can have member variables, these member variables can control the behavior of class, such as controlling the behavior of driver and so on. Therefore, class is the best choice for the components of the distributed cloud computing network routing platform.

UVM distributed cloud computing network routing platform can be shown in Fig. 2. In this figure, the black line can represent the physical interface of MDA as well as the distributed cloud computing network routing platform, while the blue line can represent the data connection among different component of the distributed cloud computing network routing platform.

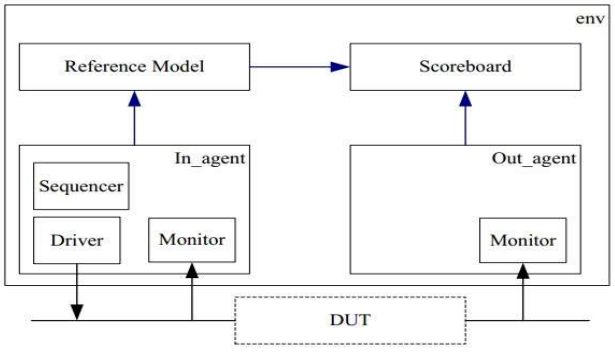

Fig. 2 UVM Distributed Cloud Computing Network Routing Platform

In the above figure, there is a sequencer, which is a unique concept in UVM, driver is responsible for sending data to MDA. Sequencer is just used to generate these data, a sequencer can start a sequence, so as to get data from sequence, then it can send these data to driver. This function can make driver no longer concern about the generation of data, which can only be responsible for the transmission of data, therefore, the function is clearer and easier to be used. In the above figure, the other components that appear are In_agent and Out_agent, which are

agent in UVM, the so-called agent can seal driver, monitor and sequencer together simply. Generally speaking, agent can correspond to the physical interface protocol, different interface protocol can correspond to different agent, interface protocol can define the exchanging format and way of data, agent can realize these contents of the interface protocol through driver and monitor. Usually there are many agents in a distributed cloud computing network routing platform, such as In_agent shown in the above figure, there are driver and monitor which can be used to send data to MDA, while there is only monitor in Out_agent, which is used to monitor the out put of MDA. While env in the above figure is the equivalent of a large container, which contains all uvm_component in its inside and takes them as the variables for its member.

Next each of the concepts in the diagram an be explained as follows:

Driver: it is driving data.

Transaction: one transaction is a package. Data exchange in the physical protocol is based on frame or package as unit, generally speaking, in a frame or a package, it should define the parameters of each item firstly, the size of each package is not the same. Taking Ethernet as an example, the size of each packet is at least 64 Byte. This package should include source address, destination address, the type of package, the crc validation data of the entire package. Little agreement will use bit or byte as the unit to carry out data exchange. Transaction is to simulate this kind of actual situation.

Monitor: it is mainly used to monitor the output of MDA.

Agent: the main function is to seal monitor and driver together in UVM distributed cloud computing network routing platform, because both monitor and driver is dealing with the interface of MDA directly, when they are sealed into one agent, in the whole distributed cloud computing network routing platform, only agent is dealing with the actual physical interface.

Reference model: it is the key and core part of UVM distributed cloud computing network routing platform, because the work of reference model of is consistent with MDA, scoreboard compare it based on the output of the reference model with MDA, if there is something errors in reference model, then the output is not credible, thus the output of scoreboard is not credible.

Scoreboard: it is mainly used to compare whether the reference model and MDA out is consistent or not, which can give the comparison result.

env: it is a large container of the entire UVM distributed cloud computing network routing platform, env is the highest level of the entire UVM tree, which contains the common components of UVM distributed cloud computing network routing platform.

Sequence mechanism: it is the core of UVM. When driver wants to drive data,it can apply one item from sequencer, once it is got it can be sent out. The definition of sequencer is fairly simple,namely, sequencer can generate data by sequence.

Conclusion

One MDA may have many functions, in this paper, the distributed cloud computing network routing environment can not complete all routing only by one time operation, in different case, the distributed cloud computing network routing has different functions. In practical applications, all of these case is based on a basic class: uvm_test. When the simulator is used to simulate, the first module of the system is top(by default, it can also specify other names, such as tb_top).

Acknowledgements

This work was financially supported by the Zheng-Xiao-Hang-Qi project of Dongguan Polytechnic (ZHENG201607), and by the key teaching reform project of Dongguan Polytechnic (No. JGZD1639), and by the teaching research project of Guangdong institute of education (No. GDJY-2015-B-b060).

References

[1]Wu Jian, Furber Steve. A multicast routing scheme for a universal spiking neural network architecture. *Computer Journal*, 53, pp. 280-288, 2010.

[2]Diavastos Andreas,Trancoso Pedro, Luján Mikel, Watson Ian. Integrating Transactions into the Data-Driven Multi-threading Model Using the TFlux Platform. *International Journal of Parallel Programming*, 44, pp. 257-277, 2016.

[3]Liu Genggeng, Guo Wenzhong, Niu Yuzhen, Chen Guolong, Huang Xing. A PSO-based timing-driven Octilinear Steiner tree algorithm for VLSI routing considering bend reduction. *Soft Computing*, 19, pp. 1153-1169, 2015.

[4]Jiang Wu Xue, Hu, Xuan Zi, Wang, Shi, Liang Yan. An optimized algorithm on distributed network node data access path based on behavior drive model. *Applied Mechanics and Materials*, 687-691, pp. 3066-3069, 2014.

[5]Chang Shu-Ping, Huang Hsin-Hsiung, Lin Cheng-Chiang, Hsieh Tsai-Ming. Timing-driven X-architecture routing tree construction among rectangular and non-rectangular obstacles. *WSEAS Transactions on Circuits and Systems*, 8, pp. 433-442, 2009.

[6]Lee Seung-Jae, Lee Kyung-Hoon, Kang Seok-Jin. Study on a pedestrian simulation model of natural movement. *Journal of Asian Architecture and Building Engineering*, 12, pp. 41-48, 2013.

