Advanced Science and Technology Letters
Vol.77 (UNESST 2014), pp.14-17
http://dx.doi.org/10.14257/astl.2014.77.04

A Command Routing Method without the Routing Table for the DIMM Tree Architecture

Young-Kyu Kim[1] and Byungin Moon[1,*]

[1] School of Electronics Engineering
Deagu, Republic of Korea
kyk79@ee.knu.ac.kr, bihmoon@knu.ac.kr

Abstract. The dual inline memory module (DIMM) tree architecture was proposed to overcome the drawbacks of the traditional DIMM interface methods. The command routing based on the routing table, which is used in the traditional DIMM tree architecture, has several problems such as increased hardware cost and routing table initialization. So, this paper proposes a new command routing method based on tree DIMM identifications (T-DIMM IDs) without the routing table, and verified the excellence of the proposed method by HDL modeling, simulations and synthesis reports.

Keywords: DIMM tree architecture, main memory, in-memory computing

1 Introduction

The dual inline memory module (DIMM) tree architecture was proposed to overcome the drawbacks of the traditional DIMM interface methods [1], and to implement the many-DIMM system [2]. This tree architecture uses a special type of DIMMs called tree DIMMs (T-DIMMs) connected by a tree topology, as shown in Figure 1. Level 2 T-DIMMs in Figure 1 receive T-DIMM access commands from the memory controller, and sometimes it must route the received commands to level 3 T-DIMMs in order to access level 3 T-DIMMs, which are not directly connected with the memory controller.

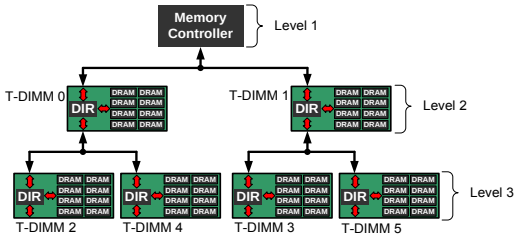

Fig. 1. DIMM tree architecture.

* Corresponding author

ISSN: 2287-1233 ASTL

To implement the command routing function, a T-DIMM must be able to choose between aborting the received command, executing that command or forwarding it to lower level T-DIMMs. According to the previous study [2], each T-DIMM decides where to route a command depending on the value of the routing table entry corresponding to the rank number. This function is performed by the DIMM interface router (DIR) block in the T-DIMM. The Router module in the DIR implements detailed functions related to command routing. This routing method requires high hardware cost because the hardware cost overhead increases geometrically with the growth of the DIMM tree depth. Furthermore, each of T-DIMMs in the DIMM tree architecture needs to initialize the routing table in each T-DIM. The paper proposes a new method that assigns an identification (ID) into each of T-DIMMs and routes the command based on T-DIMM ID without the routing table in order to overcome the drawbacks of using the routing table.

2 Proposed Command Routing Method based on the T-DIMM ID

The proposed method for the command routing assigns an identification (ID) to each of T-DIMMs in the DIMM tree architecture, and routes received commands using this assigned T-DIMM ID instead of the routing table. Since T-DIMM ID is mapped to a part of the physical address, T-DIMM IDs must be sequential and unique in order to provide a contiguous address space, and the T-DIMM ID length is $\lceil log_2^N \rceil bits$, where N is the number of T-DIMMs in the DIMM tree architecture. The proposed T-DIMM ID is divided into level and node fields. A level field value is the number of descendant nodes. The level field of T-DIMM ID can be used to estimate the level of current T-DIMM in the DIMM tree because T-DIMMs in the same level have the same number of descendant nodes. Consequently, the level field are the symbolized value of each level, and therefore the level of the concerned T-DIMM can be identified by checking the level field of its ID. The node field of a T-DIMM is used to indicate its parent T-DIMM and its branch number beneath the parent.

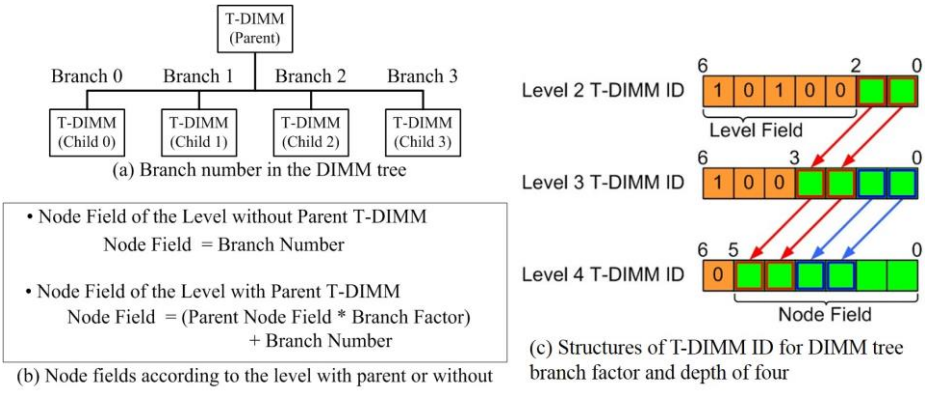

Fig. 2. Structure of the T-DIMM ID for DIMM tree branch factor and depth of four.

The node field of a T-DIMM without any parent such as level 2 T-DIMMs is decided according to its branch number as shown in Figures 2(a) and (b). The node field of a T-DIMM with a parent such as level 3 and 4 T-DIMMs is made by concatenating the node field of parent T-DIMM and its branch number as shown in Figure 2(b). The complete T-DIMM ID consists of level and node fields as shown in Figure 2(c). A T-DIMM must decide what to do as soon as it receives a command. It can choose between aborting the command, forwarding the command to the lower level, or executing the command. A T-DIMM compare its T-DIMM ID and the T-DIMM ID field of the address of the received command. If they are equal, the T-DIMM executes the received command to access its DRAM modules. If not, it has to check whether the target T-DIMM of the command is a child by comparing its node field and that of the target T-DIMM. If the target T-DIMM is its child, it forward the received command to the lower field. Otherwise, it aborts the command.

3 Experiments

The proposed T-DIMM and DIMM tree architecture are modeled with a register transfer level (RTL) using Verilog-HDL. Since DIMM tree with a branch factor and depth of four is modeled, there are 84 T-DIMMs in the modeled DIMM tree architecture. In the experiments, the T-DIMM models of three types are used for comparing the required hardware cost of the proposed method and those of the existing methods. One of the three models is made based on the proposed T-DIMM ID, and the other two are the existing methods based routing tables, which are the indexed table method and the content addressable memory (CAM) table method, respectively. In the indexed table method, Each T-DIMM has a table that stores the rank numbers of all the T-DIMMs in the DIMM tree architecture. The CAM table method has an optimized table for only requiring minimum number of the rank numbers to route the commands, and hardware logics for fast table searching which is generally used in the CAM. The models are implemented using Verilog-HDL, and synthesized into gate-level models with the Synopsys Design Complier [3] using NanGate PDK45 Open Cell library [4]. In order to verify the hardware costs of the models, we analyzed the area reports from synthesize results.

4 Experiment Results

The experimental results are presented separately as for each of the Router and Routing Block. The Router contains all the functions to process received command. The Routing Block, which is a submodule of the Router, provides the function to determine which the Router should choose between aborting, forwarding or executing the command. Figure 3 shows the required cell areas of the proposed method are smaller than those of Indexed Table and CAM Table methods. Since the Indexed Table method has one large table storing all the rank numbers in the DIMM tree architecture, its required cell area for storage is greater than the proposed method.

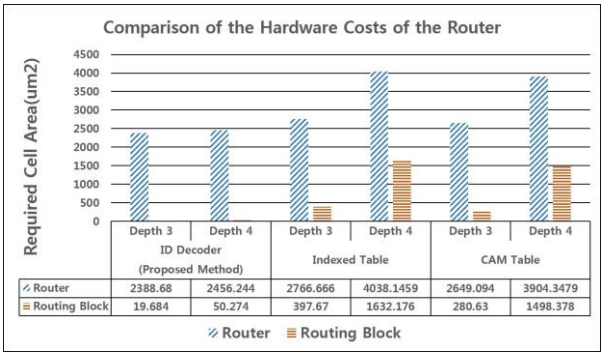

Fig. 3. Comparison of the hardware costs of the Routers and Routing Blocks

On the other hand, since the CAM Table method has an optimized table for the minimum number of rank numbers, it requires smaller cell area for storage than the Indexed Table method, but it needs additional combinational logics for fast table search.

5 Conclusions

We proposed a new command routing method based on the T-DIMM ID without a routing table in order to solve the problem caused by using the rank number based routing table in the T-DIMM architecture. Then, we modeled and synthesized the proposed and existing methods to compare their hardware cost, and the results confirmed that the proposed method demands lower cost compared with the existing methods. Finally, our future work is to study on hardware architecture and organization for the DIMM tree architecture and T-DIMM to implement actually the DIMM tree architecture.

Acknowledgments. This investigation was financially supported by Semiconductor Industry Collaborative Project between Kyungpook National University and Samsung Electronics Co. Ltd.

References

1. Bruce. J., Spencer W. N., David T. W.: Memory System: Cache, DRAM, Disk. Morgan Kaufmann, Burlington (2008)
2. Kainit, T., Gyung-Su, B., Jeremy, I., Glenn, R., Jason, C., Mau-Chung, F.C.: Utiling Radio-Frequency Intrconnect for a Many-DIMM DRAM System. IEEE J. Eme. Sel. Top. Cir. Sys. 2, 2, 210--227 (2012)
3. Synopsys Design Complier, http://www.synopsys.com/Tools/Implementation/ RTLSynthesis/DesignCompiler/Pages/default.aspx
4. NanGate PDK45 Open Cell library, http://www.nangate.com/?page_id=2325

