Advances in Engineering Research (AER), volume 130
5th International Conference on Frontiers of Manufacturing Science and Measuring Technology (FMSMT 2017)

DSM Modelling for digital design using VerilogHDL

Xing Xue[1,a], YaoChen[2,b], Junchao Wei[3,c]

[1]School of economics and management, ShangLuo University, 726000 ShangLuo, China

[2]Electronic information and electrical engineering college, ShangLuo University, 726000 ShangLuo, China

[3]School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China

[a]18467503@qq.com, [b]16563961@qq.com, [c]weijunchao@mail.nwpu.edu.cn

Keywords: design structure matrix, modeling, Verilog HDL, digital design

Abstract. In the practice of product design, the efficient control of complexity has increasingly gained importance. The Dependency Structure Matrix (DSM) has proved to be a useful tool for analysing system structure and managing structural complexity. In order to provide a deep insight of system structure in designing Verilog HDL source code artefacts for digital system designers and project managers, the paper proposes a DSM modelling method based on the characteristics of the digital system structural modelling with Verilog and the component dependencies relationship. A DSM modelling example is presented. Result shows that with the DSM model, the source code artefacts can be efficiently analyzed.

1. Introduction

Verilog-HDL can be used to model and design a digital system in a form of software programming. How to manage the complexity of source code artefacts is an issue for digital designers. Structural considerations are an established approach to manage complexity [1,2]. A digital system can be decomposed into many sub-systems or modules. Relationships between these modules exist. So system analysis is needed.

The DSM[3,4] is a square matrix to identify the dependencies or relationships between components of a system. Components of a system can be parts in a product, tasks of a project, and departments within an organization to be modelled. An n×n DSM has n elements with identical row and column labels and each entry of the DSM represents a particular kind of relationship, such as information flow, force flow, and so on.

In the example DSM in Fig. 1, elements are represented by the shaded elements along the diagonal. An off-diagonal mark signifies the dependency of one element on another. If element clk sends information to or influences the behaviour of element M2, then the (Row clk, ColumnM2) entry of the DSM contains a mark. Of course, we can also use the binary relations 1 and 0 to represent the relationships between DSM elements. Furthermore, other numerical values can present more detailed information, such as the assessment of importance, rework probabilities, impact of changes, probability of changes, etc.[5]. When a DSM is modelled, it is usually analyzed with clustering algorithms or other algorithms. Through DSMs, valuable insights from the system structure can be derived after examining the interactions within and among the subsystems.

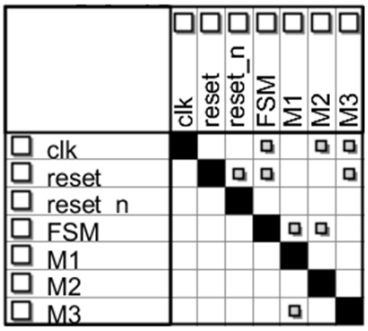

Fig.1 Example DSM

The DSM has been used in many fields, including the software field [5-9]; however, no paper is found to use Dependency Structure Matrices (DSMs) for digital design using Verilog. This paper attempts to introduce the DSM to this field. The focus of the paper is how to model DSMs for Verilog HDL programs from a structural viewpoint without considering technical constraints. A simple analysis of system structure using DSMs is illustrated.

The rest of this paper is structured as follows. Section 2 reviews digital design using Verilog. Section 3 sets up some rules to model the DSM. Section 4 presents an example model. Finally, the paper proposes a conclusion.

2. Structural modelling using Verilog HDL

Digital circuits are composed of interconnected components, such as logic gates and triggers. In most cases of the digital designs, models are used to express the digital circuits and the design work is done using CAD tools. The design using Verilog HDL embodies hierarchical and modular methodology. At every level of a hierarchical design, the efforts focus on the related level, and details of lower hierarchical levels are hidden, thus facilitating the complex design. This methodology helps to reuse the available modules or purchase IP modules. Furthermore, this hierarchical composition makes functional verification easier.

A structural model is composed of 3 kinds of object instances [11]: 1) built-in primitives(and, or, xor, etc); 2) user defined primitives (UDP, such as mux4); and 3) designed modules.

Instantiation refers to the formation of a high-level Verilog module by connecting low-level logic primitives or modules. The instantiated object is called as instance. The structural modelling process is done using instantiation.

A module is a basic unit in Verilog HDL. It describes the function of a logic entity. The function of a module can be described separately, or through other instances of other modules. That is, a structural module can include behaviour statements (such as always), continuous assignment statements (assign), build-in primitives, UDPs and other modules, or a combination of these objects. In this sense, a module describes a structure and behaviour of a hardware block in the form of software. A module includes ports as interfaces to exchange information with external environment. From a systematic and structural viewpoint, ports of a module can stand for this module.

3. DSM modelling for digital designs using Verilog

For a digital system design using Verilog, the source code is a kind of design artefacts. Components of this artefact can be module instances described above or other objects of interest, such as reg variables, wire variables, or always blocks. These components can be considered as elements of a DSM and relationships between these elements are defined by signal interaction or logical dependency.

The DSM modelling procedure consists of three major steps:

Step 1: choose components of digital system according to the hierarchy level.

Step 2: represent these components in a DSM. For a Verilog module instance, its port signal set stands for the module itself as a “macro” DSM element; for a block, a set of variables interacting with the outside of the block represent the block itself as a DSM element; a variable can be directly placed in the DSM.

Step 3: build up relationships between these elements in the DSM. These relationships depend on interface signal interactions or logical dependencies.For example, a variable on the left-hand side of an assignment statement depends on variables on the right-hand side.

4. Example

Fig. 2 shows a portion of FPGA-based functions of a certain control card. There are several instantiated modules:

1) SDAC

M1(.StartDAC(StartSDAC), .DAC_Addr(SDAC_addr), .DAC_Data(SDAC_data), .DAC_Clk(clk_20M), .DAC_SCK(SDAC_sclk), .DAC_SDO(SDAC_sdo),

```
.DAC_CSn(SDAC_cs) );   // serial DAC module
2)    M_PWM  M2(.Clk(clk),.PWM_En(PWM_en),.count1(count1),
.Count2(count2),.Q_PWM(pwm) );   // PWM module
3)    Generator20M M3.clkin(clk),  .clkout(clk_20M),  .reset(reset)   );
```

And there is a finite state machine as a block:

```
always @(negedgeclkor posedgereset )
begin
        ……
End
```

In addition, there is an assignment statement about nets reset_n and reset:

```
assign reset_n = !reset;
```

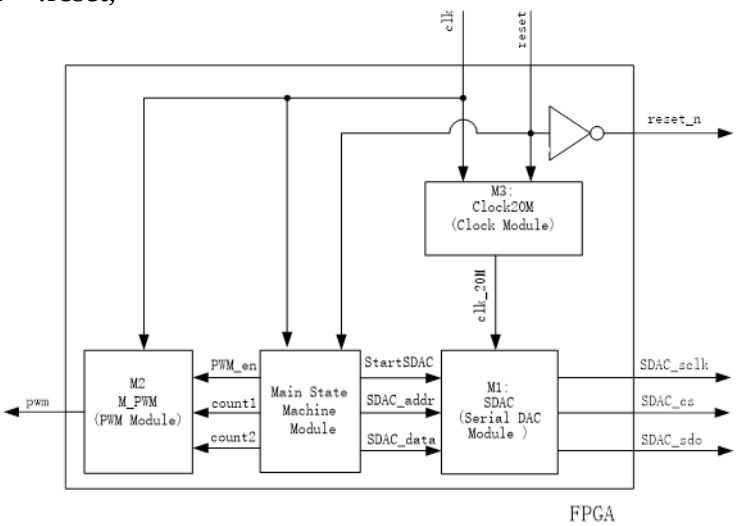

Fig.2 System Composition and Module dependency in the example

These three instantiated modules, the finite state machine block and variables reset_n and reset are chosen as elements for DSM elements. The DSM for these digital system can be obtained, as shown in Fig. 3. From this model, it is shown that the net reset_n depends on the net reset. The behaviour of the finite state machine is dependent on the net clk and reset. All these relationships between elements are obviously signified by marks. If a condensed model is needed, interface signals for modules or the block can be hidden and only element names and relationships between these elements are preserved (see Fig. 1). However, many details are lost in this condensed version.

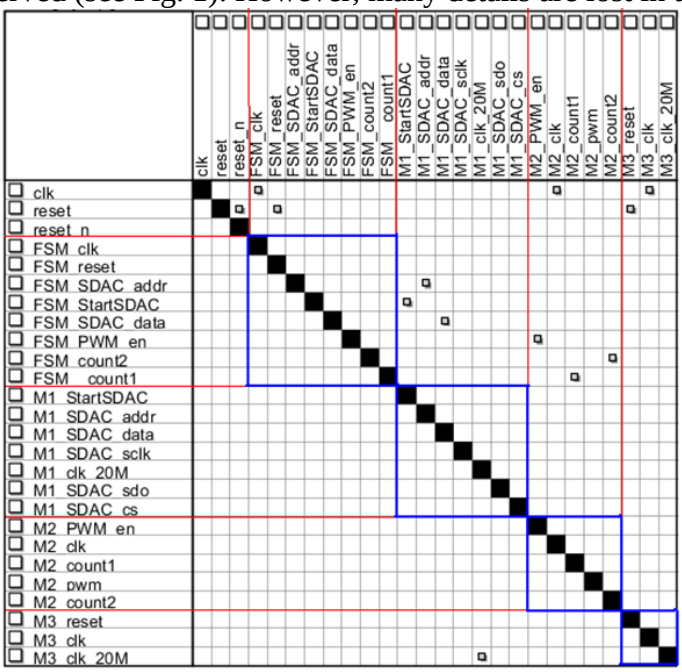

Fig.3 DSM modeling for the system in Fig. 2

A simple analysis of artefact architecture is degree analysis of elements [12]. For example, clk and reset have high out-degrees, so these two elements have more impacts on others while M1 and M2 have no impact on others.

5. Conclusion

DSMs can reveal the relationships between components in a system in a concise way and provide designers with insight into the system. This paper set up some rules about building up DSMs for digital design usingVerilog-HDL. With the DSM model, the source code artefacts can be efficiently analyzed.

Acknowledgements

This work was supported by Natural Science Foundation of Shangluo City of Shaanxi Province of China and ShangLuo University (project no.sk2014-01-21, sk2014-01-16)

References

[1]Lindemann U., Maurer M. & Braun T., Structural Complexity Management - An Approach for the Field of Product Design.Berlin :Springer, 2009.
[2]Biedermann,W.&Lindemann,U., On the applicability of structural criteria in complexity management.18th International Conference on Engineering Design, ICED 11, Copenhagen, Denmark, pp.11-20, 2011.
[3]Steward,D. V., The design structure system: A method for managing the design of complex systems. IEEE Transactions on Engineering Management, 28(3).pp.71–74, 1981.
[4]Browning,T.R., Applyingthedesignstructurematrixtosystemdecompositionandintegrationproblems:areviewandnewdirections. IEEETransactionsonEngineeringManagement,48(3),pp.292–306,2001.
[5]Lee, W.-T.,Hsu, K.-H.&Lee, J., Designing Software Architecture with Use Case Blocks using the design structure matrix,2012 International Symposium on Computer, Consumer and Control,Taichung, Taiwan.pp.654-657, 2012.
[6]Sangal, N., Jordan, E., Sinha, V. &Jackson, D.,Using Dependency Models to Manage Complex Software Architecture. 20th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, San Diego, CA, United states. pp. 167-176, 2005.
[7]Sosa, M. E., Browning, T.&Mihm, J., Studying the dynamics of the architecture of software products.2007 Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Las Vegas, NV, United states.pp. 329-342,2007.
[8]LaMantia, M. J.,Cai, Y.,MacCormack,A.D.&Rusnak, J.,Analyzing the Evolution of Large-Scale Software Systems using Design structure matrices and design rule theory, 7th IEEE/IFIP Working Conference on Software Architecture, WICSA 2008 , Vancouver, BC, Canada .pp. 83-92 ,2008.
[9]Mirson, A.,Skrypnyuk, O.,Elezi, F.&Lindemann, U.,MDM-based software modularization by analysing inter-project dependencies, 13th International Dependency and Structure Modelling Conference, Cambridge, MA, United states.pp.143-157, 2011.
[10]Ashenden,P. J. ,Digital Design: an embedded systems approach using verilog. Morgan Kaufmann Publishers,2007.
[11]Cavanagh, J., Verilog HDL: Digital Design and Modelling.CRC Press,2007.
[12]Sosa, M. E., Eppinger, S. D., and Rowles, C. M., 2007, “A Network Approach to Define Modularity of Components in Complex Products,” ASME J. Mech. Des., 129(11), pp.1118–1129.