

Multi-level Design Methodology using SystemC and VHDL for JPEG Encoder

Duy-Hieu Bui, Xuan-Tu Tran

SIS Laboratory, University of Engineering and Technology, Vietnam National University, Hanoi

144 Xuan Thuy road, Hanoi 10000, Vietnam.

E-mail: {hieubd, tutx}@vnu.edu.vn

Abstract — Nowadays, System-on-Chip (SoC) systems are becoming more and more complex and need more time to model, simulate and verification. To reduce the complexity of the system and to boost development time, a new design methodology is required. Along with SystemC library, multi-level abstraction design methodology is proposed as the key concept in SoC design. In this paper, the authors apply this methodology to model and simulate a JPEG encoder using the combination of SystemC and VHDL to explore the architecture and implement the design into hardware components. Consequently, some parts of the JPEG encoder has successfully synthesized and implemented using FPGA tools. In conclusion, the design methodology gives designers a fast way and step-by-step to explore the hardware architecture, simulate and implement the system.

Keyword: SystemC, VHDL, co-design, co-simulation, JPEG encoder.

1. Introduction

Recently, there is a high demand for a heterogeneous VLSI design environment [2] in which the design can be consistently developed through different design steps, for example, architecture exploration, hardware-software partitioning, RTL coding, synthesis and verification. In current design methodology, the algorithm is often designed using Matlab or C/C++ or some other simulation tools for architecture exploration and evaluation, while the behavior and RTL descriptions of the design use hardware description languages such as VHDL or Verilog. The difference in the development environment in each design step leads to the difficulty to reuse the simulation result and the inconsistent behavior of the system between each step. A heterogeneous environment can help to maintain the consistency of the design between each development cycle and reuse the work more efficiently. This creates a unified process from algorithm level to the testing of the final chip and reduces the time to market. To create a heterogeneous system, it is necessary for a new language providing fast simulation capability. In addition, this language should be able to be used for algorithm development and hardware/software co-design, should be supported by EDA vendor and can do co-simulation with HDL. It is the reason why SystemC has been proposed to response these requirements.

In this paper, a multi-level design methodology is presented through a practical work of a JPEG encoder using SystemC and VHDL. Multi-level design means that the

design is implemented using different views or different levels of abstraction in a heterogeneous environment. SystemC, a C++ class template library for hardware description, is selected because it is easy to learn and has very fast simulation speed in comparison with HDL, which is suitable for architecture exploration and functional description of the system. However, one disadvantage of SystemC is that there are few tools supporting SystemC hardware synthesis. Therefore, VHDL is utilized to translate the SystemC model to hardware components. The SystemC model is the reference model and is used to co-simulate with VHDL to verify the functionality of the model. SystemC and VHDL create a heterogeneous environment that can help maintain the consistency through the whole design steps.

The remaining part of this paper is organized as follows. In the second section, some background about JPEG standard, especially baseline profile of this standard, is briefly presented. Multi-level design methodology will be discussed in section 3. Section 4 is the system architecture of both SystemC model and the VHDL one. After that, the implementation result of VHDL in FPGA tool (Xilinx ISE 10.1) is shown in section 5. Finally, conclusions and remarks will be given in section 6.

2. JPEG Standard

After the announcement by Joint Photographic Expert Group [1], JPEG has become the most popular image compression standard. JPEG is a source coding technique,

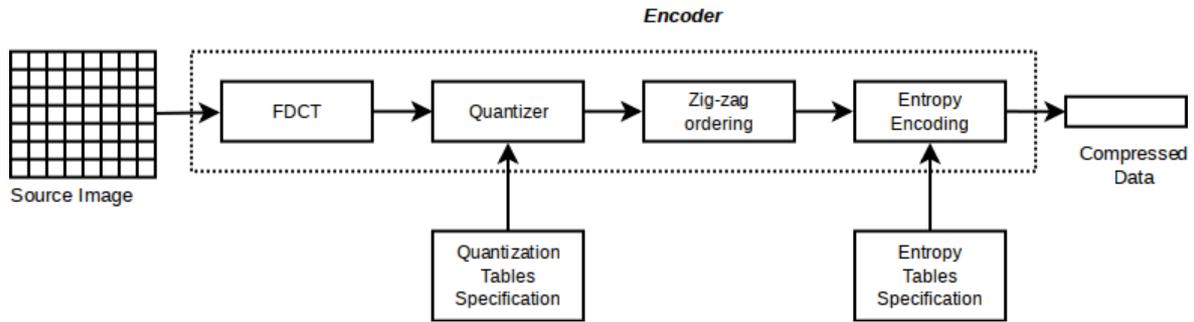


Figure 1: Baseline JPEG encoder.

widely used in digital cameras and the storage of digital images on computers, as well as the transmission of these images in the network. The goal of this standard is to support a variety of applications for the compression of continuous-tone still images with good performance and adjustable compression ratios.

JPEG standard defines four modes of operation: Sequential lossless mode, sequential DCT-based mode, progressive DCT-based mode, hierarchical mode [1]. Among these modes of operation, sequential DCT-based mode with 8-bit input sample and Huffman entropy encoding mechanism, so called baseline profile, is used in most applications. The JPEG system in this paper focuses on baseline profile only because of its popularity and good performance.

Figure 1 presents the block diagram of the baseline profile JPEG encoder system. Firstly, the source image is divided into 8×8 blocks. After that, these blocks are passed through the Forward Discrete Cosine Transform [6] (FDCT) module. The transformed data are then quantized using a quantization table. In the next step, the quantized block is read in a zigzag order to create a data model for entropy encoding. The entropy encoding step includes a Run-Length Coding (RLC) process and a Huffman coding process [5]. RLC will extract the number of consecutive zero coefficients in a zigzag-ordered block followed by a non-zero coefficient. Finally, Huffman encoder will do variable length coding on this information, and the encoded data is obtained at the output of the Huffman encoder.

The decoding process is just the reverse of the encoding processes. At first, the coded data are passed through an entropy decoder. Then, entropy decoded data is reordered by the zigzag reordering process. After that, the data is multiplied with the quantization coefficients in a

quantization table. Finally, the inverse DCT is applied and the approximation of the source image data is obtained at the output of the decoding process.

Baseline profile is a lossy compression mechanism, because of the finite precision of DCT and quantization process, as well as the reverse one, while the zigzag ordering and entropy encoding is lossless.

3. Multi-level Design Methodology

Multi-level design was born to meet the increase in the complexity of the hardware system. Multi-level design can help to model complex hardware/software efficiently in term of both modeling time and simulation speed which will reduce the time to market and the design cost. The hardware and software system can be developed in parallel and then co-simulated in the same design system. In this method, the design will be seen at different levels of abstraction in temporal, data representation, communication and concurrency. Levels of abstraction are different in how detailed they are. The more abstract the model is, the less detailed it is. Take algorithmic level for instance. In this level, the algorithms are focused, while the real hardware implementation or the communication mechanism is not considered.

However, different languages support different levels of abstraction. Grant Martin Brian Bailey *et al.* in [3] show the levels of abstraction in different languages. It is clear that VHDL and Verilog is more suitable for describing low level abstraction which means the detailed hardware description, while pure C and C++ have very high level of abstraction, but these languages are executed sequentially. This makes C/C++ hardly used to describe hardware components. As a result, to have high level of abstraction, we have to use different languages, which lead to the interoperability and consistency problems.

To overcome the disadvantages of pure C/C++ in hardware description and create a heterogeneous system, SystemC, a C++ library to describe hardware, was standardized in 2005 by IEEE [6] and now are supported by EDA vendors. SystemC with the advantages of C and C++ are now added support in temporal, data representation, communication and concurrency [3]. The algorithm can be written in C++ and then the parallel nature of hardware components are described in SystemC in untimed manner. This model and test procedures can be reused in the following design steps.

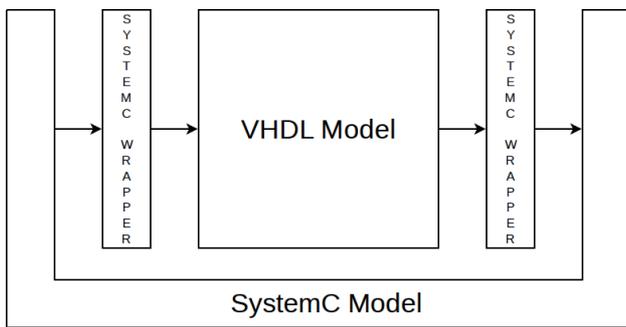


Figure 2: Verification of VHDL model by reusing SystemC model.

In this work, multi-level design method is used with the help of SystemC to fast describe the JPEG system to create an executable specification and architecture exploration with high level abstraction of data and communication. The executable specification means that the description can run as a normal computer program. With this specification, the design can be run and tested without the dependency of tools with fast simulation speed. In the architecture exploration phase, the design is refined by adding more realistic factors into the executable specification consistently. By doing this, the design is explored step by step, and the problems in the design are identified as soon as possible.

In our design, first a highly abstracted model with complicated C++ data structure and high level communication between SystemC modules is created. Next, this model is refined into lower levels of abstraction with timing information and realistic communication interfaces. Finally, the SystemC modules are translated into VHDL and SystemC models is used to verify VHDL model functionality. Because of the difference between the SystemC models and VHDL models, a wrapper is needed to co-simulate SystemC and VHDL as in Figure 2.

4. System Architecture

After considering the baseline profile specification, the design is explored using a multi-level design methodology in a heterogeneous design system. The different levels of abstraction of the baseline profile JPEG encoder are considered and implemented using SystemC and then translate into VHDL.

4.1. SystemC blocks

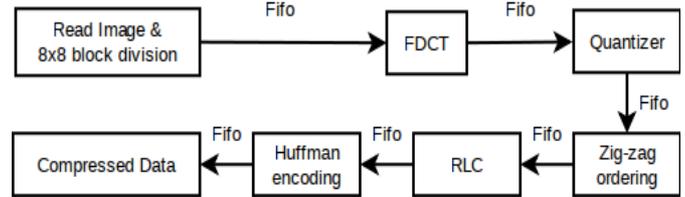


Figure 3: SystemC architecture of JPEG encoder.

In the first phase, the algorithms or the functionality of the system is focused. However, this description is modeled in parallel as the nature of the hardware system. At this level of abstraction, the functionality of each processing block in Figure 3 is described using C++/SystemC to create an executable specification. In this model, the data types and the communication between these blocks are highly abstracted and not close to the real hardware implementation. For examples, the data passing through these blocks are the instances of a data structure of an 8×8 block, while the communication is done by using FIFOs and blocking read and write. Consequently, the advantages of this are short modeling time and fast simulation speed. Data partitioning procedures and data sink are reused for the next steps. This description can be integrated into other description to form a virtual platform for software development.

In the second phase, the SystemC model is refined to the closer behavior of hardware. The communication is turned from blocking reading and writing into serial interface. A clock signal and handshaking signal are added. At each clock, an element of an 8×8 block is read or written. Since the FIFOs are removed, the buffer scheme is inserted (if necessary). The control procedures are changed due to the change in the communication architecture but the behavior of the system must follow the algorithmic model. This model is more realistic and can be used to interface with the model in VHDL for testing purpose.

4.2. VHDL models

Using the method mentioned in section 3, some SystemC module has already translated into VHDL code. The first module is converted into VHDL is FDCT. This module reuses the MDCT model by Michal Krepa on open-cores.org because of its efficiency. The architecture of this module is shown in Figure 4. The 2-D Forward Discrete Cosine transform is done by using two 1-D DCT transformations with the help of a transposed RAM [7].

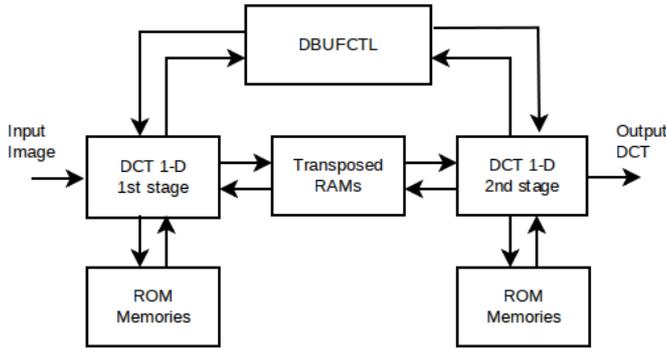


Figure 4: FDCT module.

The architecture of the quantization module is introduced in Figure 5. Two ROMs contain the pre-calculated value of the inversion of quantization coefficients. The quantized value is computed by multiplying this value with the output of DCT transform.

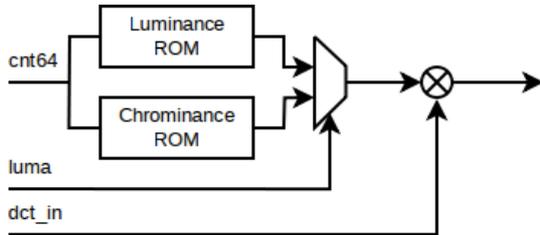


Figure 5: Quantization architecture.

The next module implemented in VHDL is the zigzag ordering module. This module is simple as in Figure 6. It uses a double buffer scheme. When a buffer is written, the other is read. The read operation is in the zigzag order.

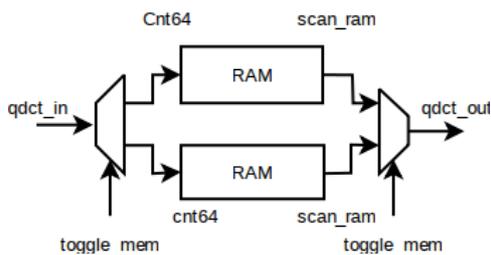


Figure 6: Zigzag architecture.

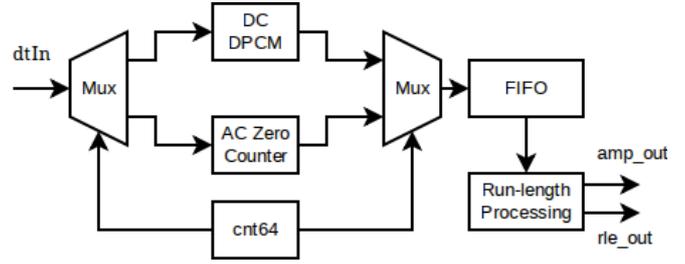


Figure 7: Run-length Encoding block diagram.

Based on the zigzag scanned value, the run-length encoding module use the value of a counter to determine if it has to do Differential Pulse Coded Modulation (DPCM) for DC component or counting the consecutive number of zero of AC component as described in Annex F section F.1.4 of [4]. If the run-length of consecutive zeros is greater than fifteen and the end of block is not reached, further processing is required to write the output in more than one cycle. Therefore, a FIFO is used to buffer the run-length value and to easily processing the output.

The final module is the Huffman encoding (Figure 8). This module contains two parts. The first part bases on the output of RLC module to determine the codeword according to predefined AC and DC tables. The second part is to make the variable length word into a finite codeword to send to the output.

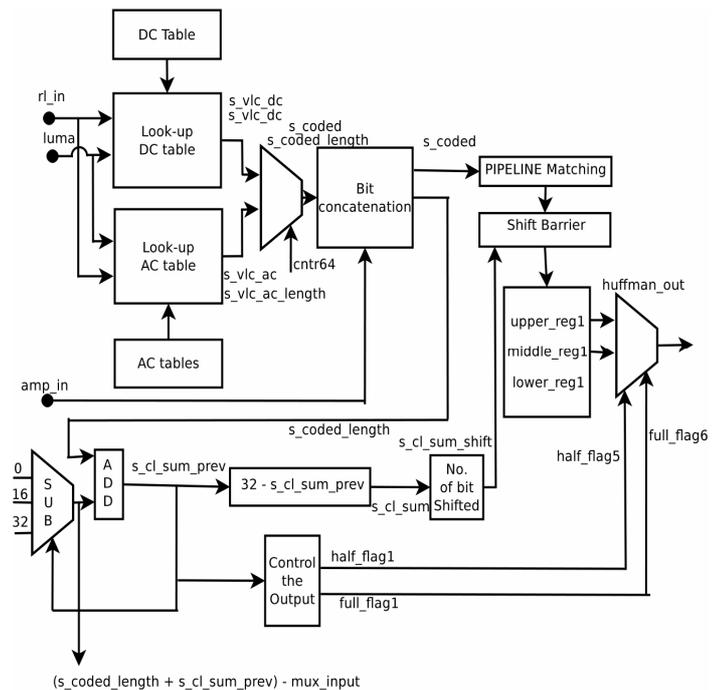


Figure 8: Huffman encoding architecture.

5. Implementation result

The VHDL models mentioned in section 4.2 were simulated using ModelSim 6.5 and then synthesized by Xilinx ISE 10.1 with Virtex-5 XC5VLX30 FPGA device. The synthesis result is shown in Table 1. This design needs 4,524 slice registers among 19,200 of the overall chip, which is 23 percent. The total number of slice LUTs is 3,654 (19 percent in total). The overall occupied slices are 1,643 in quantity (34 percent). The total Memory used in this design is 18Kb which is approximately one percent of the overall available on-chip memory. The synthesized model can run at the maximum speed of 125MHz.

Table 1. Synthesis and implementation summary

Resource	Used	Available	Utilization
Slice Register	4,524	19,200	23%
Slice LUTs	3,654	19,200	19%
Occupied Slices	1,643	4,800	34%
Bonded IOBs	20	220	9%
Block RAM/FIFO	1	32	3%
BUFG/BUFGCTRLs	2	32	6%
Memory used (Kb)	18	1,152	1%
DSP48Es	6	32	18%

After synthesis, the design is co-simulated with SystemC model with the help of some wrappers as in Figure 2. The simulation result is then compared with the pure SystemC model. The obtained result shows that the designed model functions as expected.

6. Conclusion

Multi-level design methodology has shown its benefit in hardware/software co-design. With SystemC, the complex System-on-Chip system now can be explored step-by-step by multi-level design method in a heterogeneous system. With this methodology, the hardware system, as well as its software, can be developed in parallel consistently. Multi-level design methodology has proved its strength in the modeling flow of hardware and/or software system.

To conclude, in this paper, we have presented a multi-level design applied in the practical work of modeling a JPEG encoder. Multi-level design method with the help of SystemC is discussed along with its advantages and disadvantages. The system are successfully implemented in SystemC and then translated into VHDL. The system pre-

sented in section 4 has been successfully implemented in FPGA using Xilinx ISE 10.1.

Acknowledgement

This work is partly supported by Vietnam National University, Hanoi (VNU) through research project No. QGDA.10.02 (VENGME).

References

- [1] Gregory K. Wallace. The JPEG still picture compression standard, IEEE Transaction on Consumer Electronics, vol.38, Issue 1, pp.xviii-xxxiv, February 1992.
- [2] Adnan Shaout, Ali H. El-Mousa., and Khalid Mattar, Specification and Modeling of HW/SW CO-Design for Heterogeneous Embedded Systems, Proc. of the World Congress on Engineering 2009, vol.1, pp.273-278, July 2009, London, U.K.
- [3] Grant Martin Brian Bailey and Andrew Piziali, "Taxonomy and Definitions for The Electronics System Level", in "ESL Design and Verification", pp.21-29, Morgan Kaufmann Series in system on Silicon, Morgan Kaufmann Publishers, 2007.
- [4] The International Telegraph and telephone Consultative Committee, "ITU-T Recommendation T.81", September 1992.
- [5] A. Huffman, "A method for the construction of minimum redundancy codes," Proc of IRE, vol. 40, pp.1098-1101, September 1952.
- [6] Y. Arai, T. Agui, M. Nakajima, "A Fast DCT-SQ Scheme for Images," Transactions of IEICE, vol. E71, no.11, pp.1095-1097, 1988.
- [7] Luciano Volcan Agostini, Ivan Saraiva Silva and Sergio Bampi, "Pipelined Fast 2-D DCT Architecture for JPEG Image Compression", in Proc. Of 14th Symposium on Integrated Circuits and Systems Design, pp.226-231, September 2001.
- [8] IEEE Computer Society, "IEEE Standard SystemC Language Reference Manual," March 2006.