Verification of System LSIs for Image Processing

● Yoshihiko Hayashi ● Noriyuki Ikuma ● Taku Kawamura ● Takashi Tokue

In recent years, system LSIs have become more complicated in terms of their hardware structure along with their increase in size and greater functions, and this is making it difficult to verify their logic quality. Image-processing LSIs, in particular, have a higher complexity of software processing that they perform with their multi-core structures and increasingly complicated structure of internal buses. This makes it essential to ensure that their functions, performance and power consumption satisfy the requirements by conducting system verification at the pre-silicon stage (LSI design phase). However, product development with a short turnaround time (TAT) is called for because of the recent shorter model change cycles, and efficiently conducting system verification while ensuring quality is very important. Fujitsu Semiconductor is making positive use of a hardware emulator (a device capable of mapping the circuit to be verified to dedicated hardware for running it at high speed) to establish verification technology that ensures quality and improves verification efficiency at the same time, and applying it to product models. This paper presents hardware/software co-verification, performance verification and power consumption estimation by using a hardware emulator, which we believe is very effective for verifying image-processing LSIs.

1. Introduction

Recently in the industry of digital consumer electronics including digital cameras, there has been a growing demand to bring products to market in a short time along with rapid changes in the market and diversification of user needs. In addition, recent system LSIs have complex hardware structures with multi-core processors, many functions, many interfaces and mass memory all loaded on a single chip, and verifying these structures to ensure logic quality is becoming difficult. The occurrence of a failure after the actual chips have been completed is likely to lead to a cost increase because of the need to remake them and lost product opportunities due to extended development periods. For that reason, Fujitsu Semiconductor believes it is very important to verify the system of an entire LSI in the pre-silicon stage (LSI design phase) so that the verification results can be fed back to the logic design stage. Important items of system verification can be roughly classified into three categories:

1) Functional verification (hardware/software co-verification)
2) Performance verification
3) Power consumption estimation

In functional verification, processes including IP-level verification and block-level verification should be built up to eventually conduct hardware/software co-verification, in which it is important to make sure that the actual application runs normally at the system level with the major IPs combined (entire chip). Another point is how quickly problems, if any, can be investigated. We have realized an in-circuit emulator (ICE) debugging environment that makes use of a Joint Test Action Group (JTAG) port to allow verification of application debugging on an entire chip.

In addition, recent system LSIs have increasingly complicated multi-core and bus structures and factors that may influence performance including bus bandwidth, memory controller performance and image processing speed are intricately linked together, which gave rise to the need for realizing "visualization of

performance." By building a performance verification platform, we have improved the efficiency of both performance verification and functional verification.

Furthermore, with an increasing number of products premised on outdoor use including mobile phones and other devices, battery life is one of the factors indicating product value and low-power designs are attracting attention. Estimating power consumption for this purpose has been successfully carried out in the pre-silicon stage.

This paper presents hardware/software co-verification, performance verification and power consumption estimation by using a hardware emulator (hereafter "emulator"), which is believed to be the most effective as system verification of image-processing LSIs.

2. Improvement of verification efficiency by emulator

Simulators are commonly used as effective tools for conducting verification. To verify the systems of image-processing LSIs, however, massive data processing makes it impossible to complete verification within a practical amount of time by simulation. That is why Fujitsu Semiconductor uses an emulator. An emulator is a device capable of mapping the circuit to be verified to dedicated hardware and running it at a high speed, and its execution speed may be 500 to 1000 times faster than a simulator. In addition, it can accommodate the verification circuit for an entire chip and allow the internal waveform of the verification circuit to be monitored at any point, which provides excellent analyticity when any problem occurs. An emulator is a tool essential for verifying image-processing LSIs and is effective as a means of accelerating system verification.

3. Hardware/software co-verification

Hardware/software co-verification is an effective method for verifying the functions of system LSIs. This method runs software at the application level on an entire chip in the pre-silicon stage following IP verification and block verification. The benefits of this verification include:

- Verification of hardware functions at the system level
- Advanced development of software

The following describes hardware/software co-verification.

3.1 Verification using emulator

As for the verification execution environment, execution time can be fairly closely approximated to the actual device by using an evaluation board with a field-programmable gate array (FPGA). However, it is impossible to implement an entire chip with a large circuit scale and verification is unavoidably confined to hardware/software with a configuration very different from that of the actual device. In addition, the state of the inside of the hardware cannot be readily traced by using an evaluation board with an FPGA, which makes it very difficult to debug any problem generated in the system verification phase. To solve these problems, we have realized an emulator ICE verification environment with an ICE debugger connected to the JTAG interface of the emulator environment mentioned above (**Figure 1**).

The entire system has been efficiently built into an emulation environment by making use of the proven platforms and verification IPs we have accumulated up to now. During system verification, a detailed waveform of hardware operation corresponding to the behavior of the software can be obtained. In addition, any error or freeze on a bus can now be detected by inserting an assertion checker. Furthermore, by using a dedicated cable to connect the JTAG interface with the ICE debugger, a software debugger similar to the actual evaluation board can be used in an emulator verification environment, which allows the developer to monitor register values, set break points, and execute in a stepwise manner software and memory read/write. This has made it possible to debug the software in an environment similar to the actual hardware even before the actual device comes out.

As an effect from the perspective of hardware verification, connection of a debugging circuit by a JTAG interface, which has a multi-core processor and complicated connection, can be verified by connecting the actual ICE. This verification, which is definitely impossible by simulation, is very effective in verification in the pre-silicon stage.

3.2 Example of application

We have conducted hardware/software co-verification on a graphics LSI for in-vehicle use. We have run a car navigation application in an emulator verification environment to successively output a few tens of

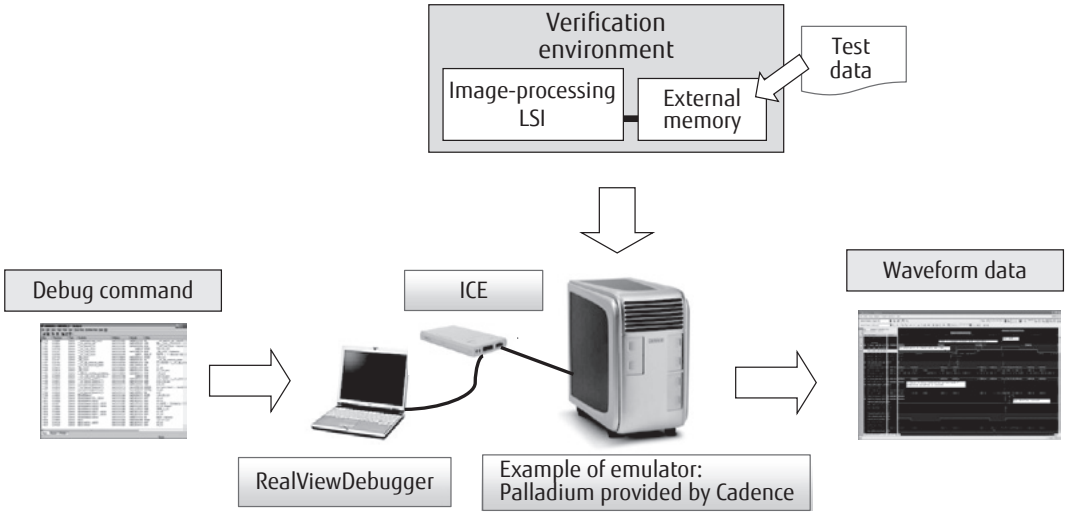

Figure 1
ICE verification environment with emulator.

frames of images, which is practically impossible by simulation, and successfully confirmed that the images had no problems.

In debugging the OS integrated in the system, we have made use of a function to reference internal registers and memory using the ICE environment to solve problems, thereby achieving system-level debugging at the pre-silicon level.

4. Imaging LSI performance verification

In verifying LSIs, it is important to have more efficient logic verification (functional verification) and, at the same time, there is also an issue of how to ensure the performance requirements (performance verification) before the actual device is completed. We have made performance verification more efficient by building a performance verification platform and applying it to image-processing LSIs for digital cameras.

The following describes how to verify the performance of image-processing LSIs.

4.1 Conventional performance verification and its issues

Conventionally, performance was verified by obtaining a waveform in a simulation or emulation verification environment and analyzing it. However, obtaining and analyzing a waveform take time. To deal with this issue, we have used a dedicated module to directly monitor the states of hard macros and made this information visible, thereby improving the efficiency of performance verification.

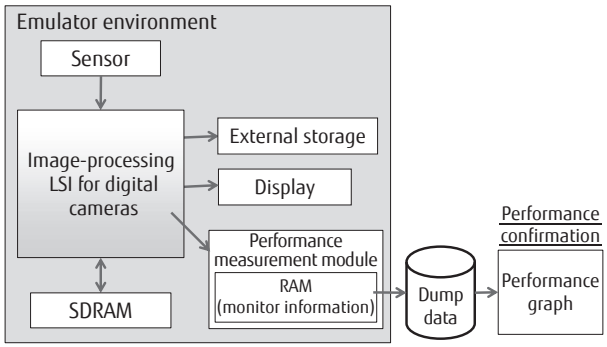

Figure 2
Configuration of performance verification platform.

4.2 Configuration of performance verification platform

External, peripheral pseudo circuits including a sensor, display, SDRAM and external storage and performance measurement module for monitoring hard macros are connected to the target image-processing LSI to build a verification environment in an emulator (**Figure 2**). When a test program such as one to continuously take still images is run, the information on the hardware monitored during the execution is stored in the internal RAM. After the emulation is completed, this information stored in the RAM is visualized as a

performance graph.

The states of hardware to be monitored with the performance verification platform are:

- Start and end times of the respective hardware macros
- State of FIFO of the respective hardware macros
- State of bandwidth of bus and SDRAM controller (**Figure 3**).

4.3 Benefits of performance verification platform

The following has been realized by the performance verification platform.

1) Optimization of bus arbitration setting

The optimum bus arbitration setting can be identified. For example, of the hardware macros connected with a bus, higher priorities are given to those that require more bandwidth, and lower priorities are given to those that do not need much bandwidth.

2) Optimization of frame buffer allocation

The hardware macro that causes a bottleneck in processing performance can be identified to determine the optimum number of frame buffers for the respective hardware macros. In addition, in a system with multiple SDRAM controllers, the frame buffer allocation in view of effective use of bandwidth can be determined. As a result, a transaction conflict between hardware macros requiring a large amount of bandwidth can be avoided.

3) Estimation of maximum number of consecutive frames of continuous still image shooting

The maximum number of consecutive frames can be estimated by using the processing time per frame of the respective hardware macros measured on the performance verification platform, the number of frame buffers and the SDRAM size.

We have used an image-processing LSI for digital cameras to actually apply 1) and 2) above to deal with bandwidth failure (performance failure) found in system verification. This has enabled us to make appropriate hardware modifications such as increasing the number of FIFO stages and revising the arbitration logic. That is, detecting any performance failure before the actual device is complete has allowed us to feed back the results to the hardware logic in an efficient and appropriate manner.

5. Power consumption estimation by emulator

This section outlines how to estimate power

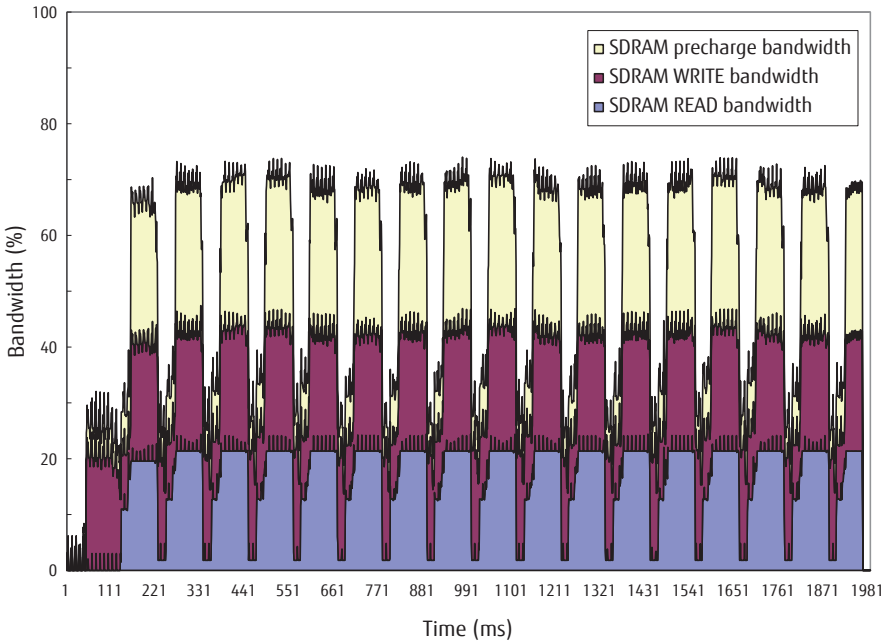

Figure 3
State of bandwidth of bus and SDRAM controller.

consumption with an emulator and presents an example of its application.

5.1 Need for power consumption estimation and its issues

For system LSIs that are growing in complexity and scale, having a power-saving design is a decisive factor in their commercial value and an essential item of design. Verification to make sure that power-saving design is correctly implemented is equally important but measuring power after the actual LSI is completed is of no benefit as feedback to the design. Accordingly, how accurately power consumption can be estimated in the pre-silicon stage is an important point.

However, recent LSIs have complicated power control and many LSI manufacturers now adopt measures for power consumption reduction such as clock gating. This means that accurately estimating power is impossible by desk calculations. In other words, calculating a value close to the actual power is difficult unless the actual application and scenario are used to accurately measure the entire LSI.

In addition, it is meaningless to run the actual application for measuring a transient value and a certain period of execution time is required. For example, with an image-processing LSI, processing one frame of video takes 33 ms when the frame rate is 30 fps, which means that estimating the power consumption of system operation requires measurement of an order of a few seconds for multiple frames. If this measurement is run with a simulator, a few months are required and the verification period becomes enormous. Accordingly, it is practically impossible to use a simulator for this.

5.2 Power consumption estimation by using emulator

The power consumption value is calculated by extracting the operating rate of an LSI and inputting it in an existing tool that analyzes power consumption. The point is this operating rate of LSI and a higher operating rate provides a more accurate power consumption value.

To accurately extract the operating rate, we have developed a technique to estimate power consumption by using an emulator.

The advantage of this technique is its processing speed and output format of the operating rate. For the processing speed, as described in functional and performance verification, a processing efficiency 500 to 1000 times higher than HDL simulation has been achieved by using an emulator, which has made it possible to extract the operating rate over a long time.

Regarding the operating rate output format, with the conventional waveform format, the volume of information may grow to be massive depending on the gate scale and time, and such massive information is practically impossible to handle. Accordingly, we have introduced an output format that only represents the operating rate suited to the power consumption to reduce the amount of data. This has made it possible to extract the operating rate over a long time. As a result, an operating rate over a time in the order of a few seconds can now be handled and it has become possible to calculate power consumption with high accuracy and analyze changes in power consumption over time (**Figure 4**).

5.3 Example of application

Power estimation by means of an emulator has been applied to about 10 ASSPs, mainly image-processing LSIs. This section presents an example of application to Milbeaut, an image processing LSI for digital cameras.

For Milbeaut, an emulator was used to calculate the average and maximum power consumption, which resulted in estimation of power consumption with higher accuracy than desk estimation. The result of estimation was below the target power consumption value in the customer usage conditions, and this was successfully verified before obtaining an engineering sample (ES). Furthermore, our power consumption estimation allows the power consumed by the respective IPs to be estimated. By visualizing the correlation between operation and power of an arbitrary IP (**Figure 5**), we verified whether power consumption was restrained when IP operation was not necessary. In fact, we detected an inadequate low-power control with some IPs, and this finding was fed back to the design stage before the actual chip was manufactured.

6. Conclusion

This paper has described technology for verifying the systems of image-processing LSIs.

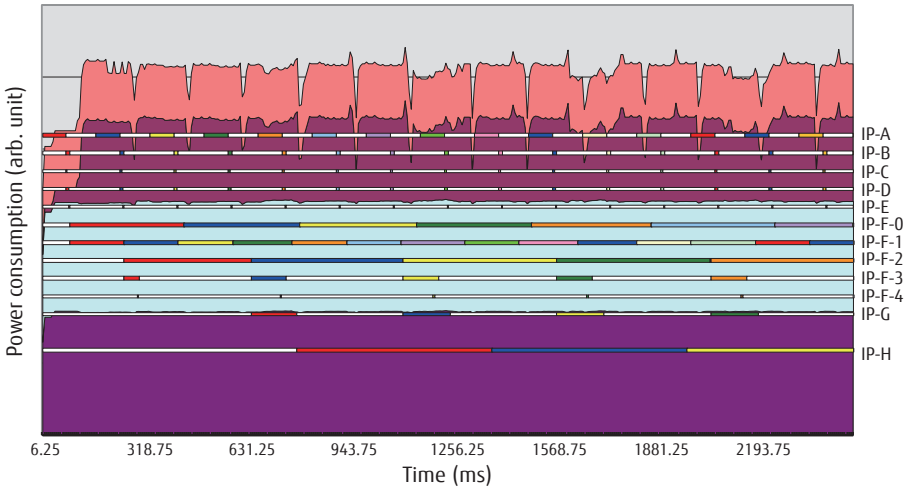

Figure 4
Analysis of power consumption over time.

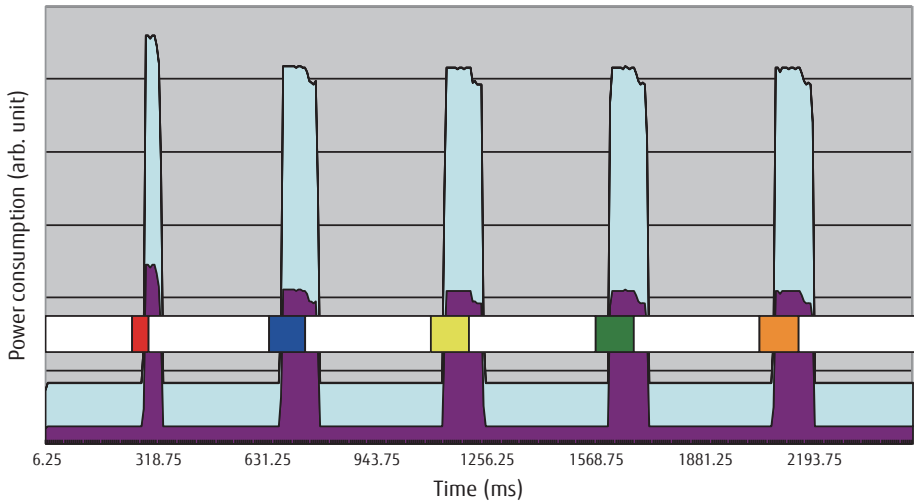

Figure 5
Analysis of operation and power of an arbitrary hardware IP over time.

By effectively using the functions of emulation for functional verification, performance verification and power consumption estimation using the actual applications, we were able to conduct verification in the pre-silicon stage before the actual chip was completed and give feedback to hardware and software. We intend to develop and apply our efficient system verification techniques to realize high-quality system LSIs, which will continue to grow in functionality and complexity in the future.

Yoshihiko Hayashi
Fujitsu Semiconductor Ltd.
Mr. Hayashi is currently engaged in development of technology to verify system LSIs.

Taku Kawamura
Fujitsu Semiconductor Ltd.
Mr. Kawamura is currently engaged in development of technology to verify system LSIs.

Noriyuki Ikuma
Fujitsu Semiconductor Ltd.
Mr. Ikuma is currently engaged in development of technology to verify system LSIs.

Takashi Tokue
Fujitsu Semiconductor Ltd.
Mr. Tokue is currently engaged in development of technology to verify system LSIs.