SoC Platform Design and Verification for Multimedia Application

Hongkyun Jung, Xianzhe Jin, Younjin Jung, Ok Kim, Byoungyup Lee and Kwangki Ryoo
Graduate School of Information and Communication, Hanbat National University,
San16-1, Duckmyoung-Dong, Yuseong-Gu, Daejeon 305-719, Korea
E-mail: hkjung@hanbat.ac.kr, suntreekim@hanmail.net, yjjung39@gmail.com, kusl2510@gmail.com,
roll83@hotmail.com, kkryoo@hanbat.ac.kr

Abstract: This paper proposes an SoC platform for the development of multimedia applications. The SoC platform uses a 32-bit RISC processor with 4-way set-associative cache for improvement of performance, a VGA controller and an AC97 controller for multimedia, an SoC debug interface for debugging and supports WISHBONE compatible peripheral IPs. The multimedia SoC platform is implemented on Xilinx VIRTEX-4 XC4VLX80 FPGA device and the FPGA executes at the maximum frequency of 64.574MHz. As a result of system-level verification using test programs in development board including the FPGA device, the proposed SoC platform was satisfactory for the desired functions.

1. Introduction

System-on-Chip (SoC) design for multimedia applications becomes more and more important these years. Strong consumer demand together with advances in technology for portable multimedia devices are challenging device manufacturers to integrate more features and services into smaller, cheaper and more versatile products. As billions-of-transistor SoC becomes commonplace, the design complexity continues to increase. Because of this, designers are faced with the daunting task of meeting escalating design requirements in shrinking time-to-market windows. Before a chip is taped out, verification on a FPGA based system is usually required. Most people tend to build a dedicated FPGA based evaluation board for a particular application [1]. To solve this problem, a platform based SoC system design method is introduced recent years. It extends the concept of design reuse and put more stress on system level reuse [2]. Platform based design method shortens SoC development cycle and improves reusing ratio design quality. The platforms provide users with ample room for product differentiation. Derivative design can be accomplished quickly by adding just a few IP components. Moreover, integrated architecture minimizes verification uncertainties that greatly reduce design effort and risk [3].

This paper presents a synthesizable SoC platform, which is implemented on an FPGA and used for multimedia application. For this purpose, a VGA controller, AC97 controller and debug interface are designed and verified to implement the multimedia SoC platform. To improve performance of SoC platform, direct-mapped cache of 32-bit RISC processor is modified to 4-way set-associative cache. The proposed SoC platform is implemented on Xilinx's XC4VLX80 FPGA device and verified with HW/SW co-verification using GDB/HDL simulator and HW/SW co-emulation using GDB/FPGA development board.

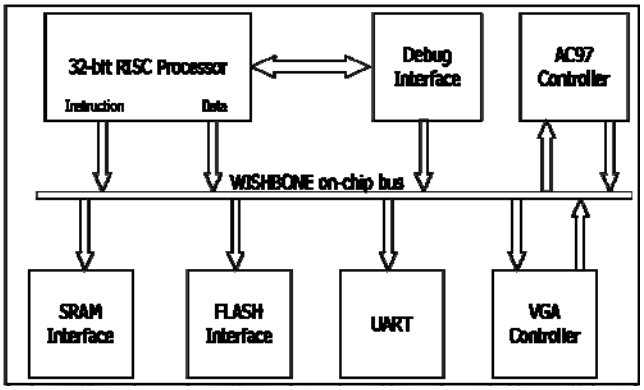

Figure 1. Architecuture of proposed SoC platform

As a result of implementation of the SoC platform, the maximum clock frequency currently attained is 64.574MHz and the number of the used gates is 1,198,354.

In section 2, we will present the architecture of the multimedia SoC platform and its functionalities. Section 3 exposes the implementation of the WISHBONE compatible IP's. Section 4 shows the experimental results on the implementation of the SoC platform. Finally, the conclusion is presented.

2. Proposed SoC platform

The proposed SoC platform uses 32-bit RISC processor, WISHBONE interconnect, a VGA controller, an AC97 controller, an SoC debug interface, memory interfaces and a UART. The UART (Universal Asynchronous Receiver Transmitter) controller provides serial communication capabilities to the platform using an RS232 protocol[4]. WISHBONE interconnect is the basic communication channel that has synchronous data and address buses with multiple masters and slaves.

2. 1 32-bit RISC processor

The 32-bit RISC processor for the SoC platform is OpenRISC 1200, a publicly available processor. OpenRISC 1200, synthesizable core, is designed with Verilog HDL, and has high flexibility because all configuration options for the processor are gathered together into a single file containing numerous define statements. OR1200 is a 32-bit scalar RISC with Harvard micro architecture which has separated instruction/data bus, 5 stage integer pipeline and basic DSP capabilities. OR1200 is intended for embedded, portable and networking applications. Instruction and data caches are present in the microprocessor, and both default to 1-way set-associative 8KB caches[5].

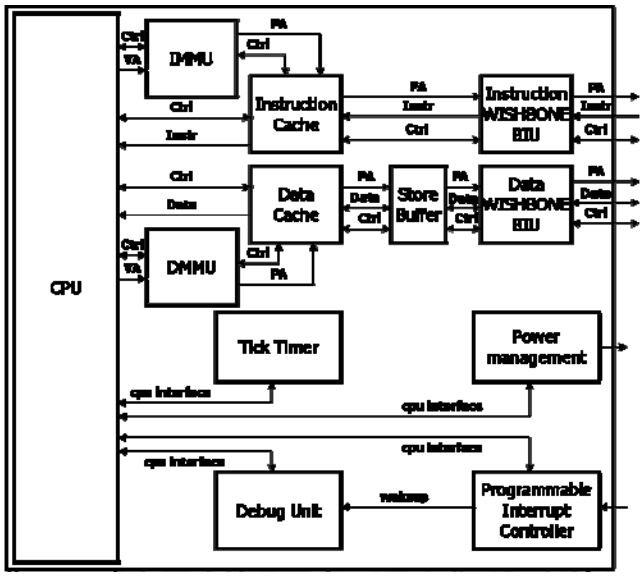

Figure 2. Block diagram of 32-bit RISC processor

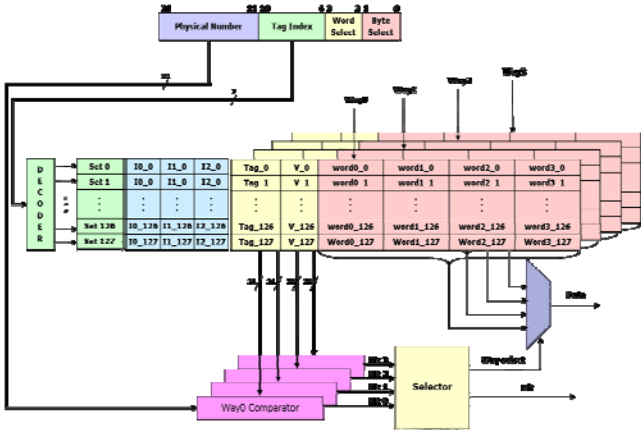

Figure 3. Architecture of 4-way set-associative cache

The memory management units are implemented for data and instruction MMU as 64-entry, 1-way direct-mapped translation look-aside buffer[6].OR1200 communicates with the WISHBONE bus interconnect. A block diagram of the OR1200 architecture is depicted in figure 2.

Direct-mapped cache of OpenRISC 1200 is modified to 4-way set-associative cache to reduce access times between 32-bit RISC processor and main memory in SoC platform. Figure 3 shows the architecture of 4-way-set-associative cache. The structure of 4-way set-associative cache is that 4 main memory blocks are mapped to only one cache blocks, so the access miss rate is lower than direct-mapped cache. Least Recently Used algorithm of 4-way set-associative cache is pseudo LRU algorithm that requires an additional field in the cache line composed of three bits.

2. 2 AC97 controller

The AC97 controller supports variable and fixed sample rate, variable sample size, 6 output channel surround sound, 3 input channels and external DMA engine. The AC97 controller is composed of WISHBONE Interface, Serial In/Out Register, Serial IO Controller, In/Out FIFO, PCM

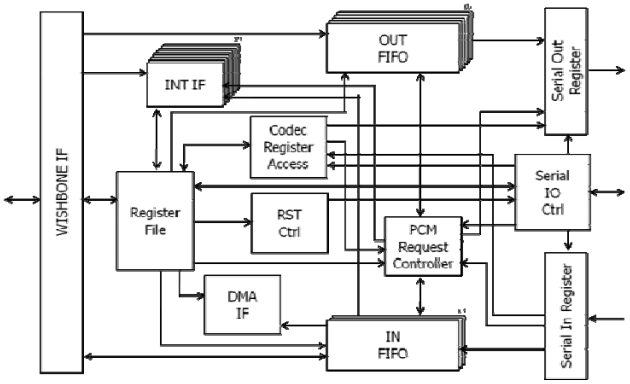

Figure 4. Architecture of AC97 controller

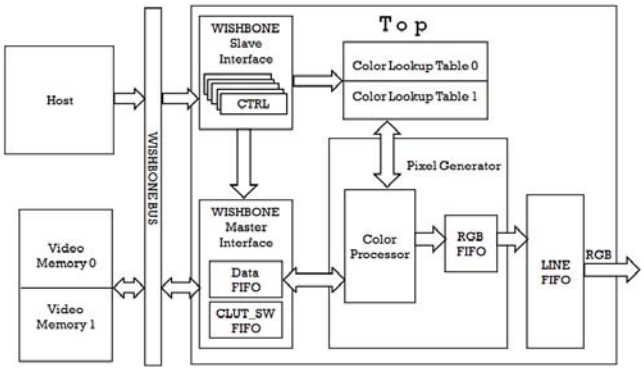

Figure 5. Block diagram of VGA controller

Request Controller, Codec Register Access and other controller logics. The WISHBONE Interface implements 32 bit bus widths but not support other bus widths. The Serial In/Out Register converts the parallel input from the FIFOs and control logic to a serial bit stream and vise versa. The serial bit stream is synchronized to the Sync signal from the Serial IO Controller. The In/Out FIFOs hold the data to be transmitted/has been received. The FIFOs are 4 entries deep. Each entry is 32 bits wide. The PCM Request Controller monitors the requests from the Codec and controls when data is being sent out or latched. The PCM Request Controller supports variable and fixed sample rate operation. The Codec Register Access module sends requests to the Codec whenever the host wants to read or write a Codec register. It also provides feedback when a Codec register read has completed and the data is available for the host to retrieve. Figure 4 shows a block diagram of the AC97 controller.

2. 3 VGA controller

The VGA controller is compatible to WISHBONE, system on a chip bus, and consists of master interface, slave interface, pixel generator and CLUT. Taking a processor generated picture from memory space, the controller provides digital RGB values for each pixel as well as horizontal and vertical synchronization signal. Figure 5 shows a block diagram of the VGA controller. The VGA controller receives the timing signals and variable control

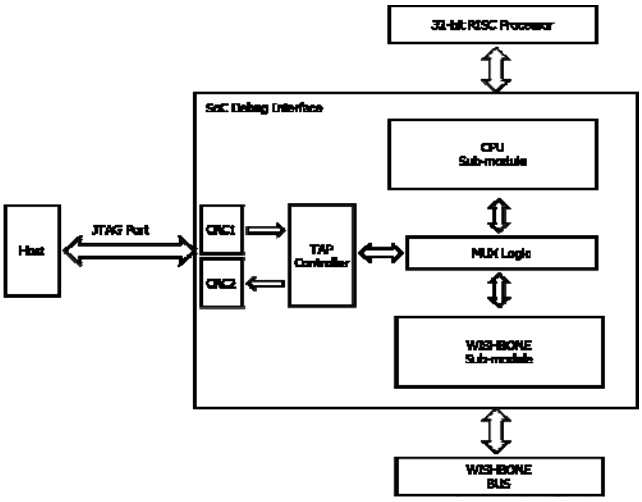

Figure 6. Architecture of SoC debug interface

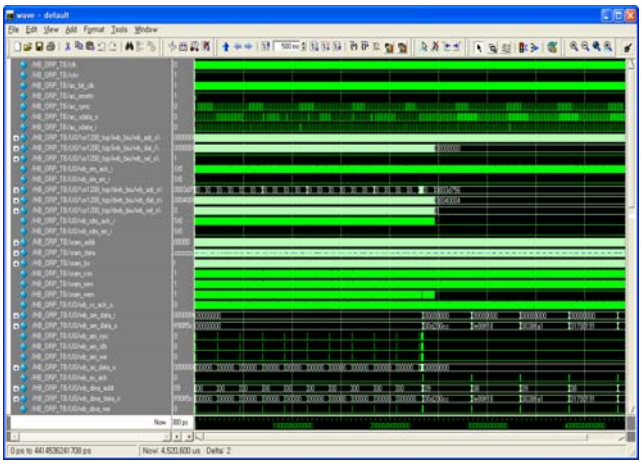

Figure 7. Result of simulation SoC platform using wave-play program

signals from host processer for display operation. Once this signal initializes internal registers in slave interface, the VGA controller can operate with different color mode by the initialized value. Master interface accesses the external memory through the WISHBONE bus and brings image data from initialized address area of memory. After transaction of pixel data, the VGA controller transfers the received RGB data and the generated timing signals both according to the initialized resolution to DAC and analog monitor respectively. The VGA controller has flexible features such as programmable resolution and video timing, and a variety of color mode in 32bpp, 24bpp, 16bpp, 8bpp grayscale and 8bpp pseudo color.

2. 4 SoC debug interface

The SoC debug Interface for debugging on this platform is an interface between the TAP controller and the sub-modules that are target specific (CPU, WISHBONE). It supports three sub-modules now but can be expanded up to sixteen sub-modules. The sub-modules used on this platform are CPU Debug Module, WISHBONE Debug Module and CRC Module. Figure 6 shows architecture of SoC debug interface.

The first step of an operation is selecting a sub-module which is either the CPU Debug Module or the WISHBONE Debug Module. This is done with the module select instruction. The CRC Module where 32-bit CRC calculation is performed does not need to be selected because the incoming and outgoing data must go through it before and after the operation performed in either the CPU Debug Module or the WISHBONE Debug Module.

The CPU Debug Module is an interface to the CPU debug facilities. It consists of the Command Register, Control Register and the CPU Interface that is connected to the Debug Unit of the OpenRISC core on this platform to read or write the values of the registers in the core. The CPU Debug Module can only perform 32-bit read and write accesses. There are two steps needed to write/read the data to/from the OpenRISC core. In the first step, address, data size and the type of operation need to be written to the Command Register. In the second step, CPU read or write operation is executed. The address written in the Command Register is incremented automatically after each successful access. At the end of the operation it points to the first data after the last accessed. The WISHBONE debug Module is used for the WISHBONE devices' debugging. It connects to the WISHBONE bus through the WISHBONE interface and can perform 8, 16 and 32-bit read and write accesses. There is only one register in this sub-module, which is called WISHBONE Command Register[8].

There are also two steps needed to write/read the data to/from the WISHBONE device. The operation is similar to the operation performed in CPU Debug Module. The only difference is the WISHBONE Debug Module can perform 8, 16 and 32-bit read and write accesses while the CPU Debug Module can only perform 32-bit read and write accesses.

3. verification

To verify the SoC Platform, the test programs, which are used for functional, timing simulation and FPGA test are wave-play program and image program. The image program sets Control, Timing and Video memory Base Address Register. Because the VGA controller contains WISHBONE bus master Interface, the operation just executes the simple task and the VGA controller, which reads video data from sram memory automatically. The Wave-play program is DMA, AC97 controller register set code. The AC97 controller receives audio data through DMA operation.

Figure 7 shows the result of timing simulation for SoC platform using wave-play program. After reset, 32-bit RISC processor send AC97 Codec Cold Reset signal to the AC97 controller. Cold Reset to the AC97 Codec by writing a 1 to the CRST bit in the CSR register. Writing a 1 to the CSR register CRST bit will assert reset to the Codec for at least 1us. After the AC97 Codec reset, the AC97 Codec and the AC97 controller are set by 32-bit RISC processor. As time goes on, the AC97 controller reads the sound data from SRAM memory.

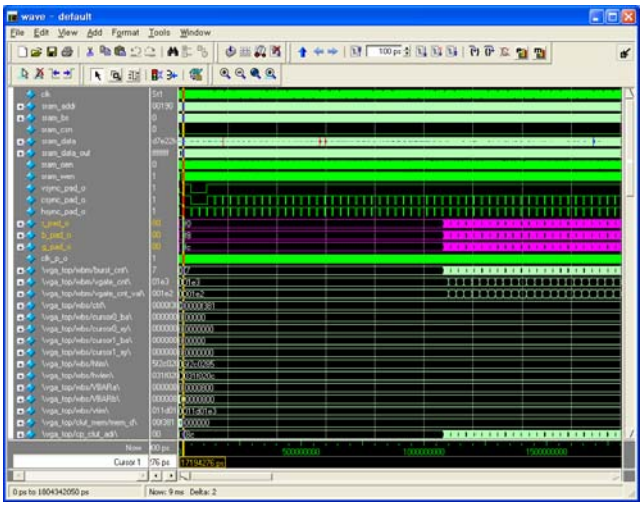

Figure 8. Result of simulation SoC platform using image program

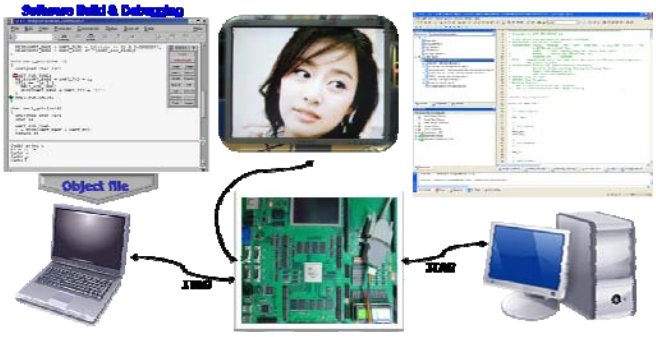

Figure 9. System-level verification using FPGA prototyping

The result of timing simulation for SoC platform using image program is shown in figure 8. After 17usec from the point of Starting simulation, the VGA controller register setting will be done by 32-bit RISC processor. As time goes on, the VGA controller reads the video data from SRAM memory (the VGA controller Base address register has already set 0x00000800) through the VGA controller master Interface.

For system-level verification, the image program including binary image files is generated by using cross-compiler that includes gcc, binutils, gdb and jtag utility which have configured appropriately for the OpenRISC architecture. The machine code is downloaded to an FPGA board through parallel cable, using jtag utility software and gdb. As a result of system-level verification using the image program, still image is displayed on a monitor that connected to an FPGA board using D-sub connector. Figure 9 shows the result of system level verification using FPGA.

4. Implementation

The Multimedia SoC platform is implemented on Xilinx's XC4VLX80 FPGA device and the SoC platform in an FPGA executes at the maximum frequency of 64.574MHz

Table 1. Device utilization

Item	Amount	Percentage
Flip Flops	2,735	3%
Input LUTs	7,808	10%
IOBs	107	13%
RAMB16	17	7%
Total equivalent gate count	1,198,354	
Maximum frequency	64.574MHz	

and the maximum net delay of 13.119ns. The Multimedia SoC platform uses 3% of slice Flip Flops and 10% of 4 input LUTs of the Xilinx XC4VLX80. Table 1 details device utilization.

5. Conclusion

In this paper, the SoC platform is proposed for multimedia applications. The SoC platform uses 32-bit RISC processor with 4-way set-associative cache, a VGA controller for image and video, AC97 controller for sound and Debug interface for downloading image file and debugging. The platform has been implemented on FPGA development board with VIRTEX 4 XC3S1000 FPGA. The SoC platform in an FPGA device executes at the maximum frequency of 64.574MHz. As a result of system-level verification using test programs in development board including the FPGA device, the proposed SoC platform was satisfactory for the desired functions.

References

[1] Yi-Li Lin, Chung-Ping Young, Yuan-Jui Chang, Yi-Hui Chung, and Alvin W.Y. Su, "Versatile PC/FPGA Based Verification/Fast Prototyping Platform with Multimedia Applications," Intrumentation and Measurement Technology Conference, vol. 2, pp.1490-1495, May 2004

[2] Younghoon Bin, Kwangmyong Kang, Hyungjun Kim, Hongkyun Jung and Kwangki Ryoo, "The Development of SoC Platform for Embedded System Applications," International Conference on Convergence Information Technology, pp.2286-2291, Nov 2007

[3] Diana Wu, Platfrom-based SoC design comes of age, Faraday Technology Corp, April 20, 2003

[4] Jacob Gorban, UART IP Core Specification, Rev. 0.6 August 11, 2002

[5] Damjan Lampret, OpenRISC1000 Architecture Manual, January 28, 2003

[6] Damjan Lampret, OpenRISC1200 IP Core Specification Rev. 0.7, September 6, 2001

[7] Rudolf Usselmann, AC97 Controller IP Core Specification Revision 1.2, September 19, 2002

[8] Richard Herveille, VGA/LCD core specification, Rev 2.0, March 20, 2003

[9] Igor Mohor, SoC Debug Interface, Rev. 3.0 April 14, 2004