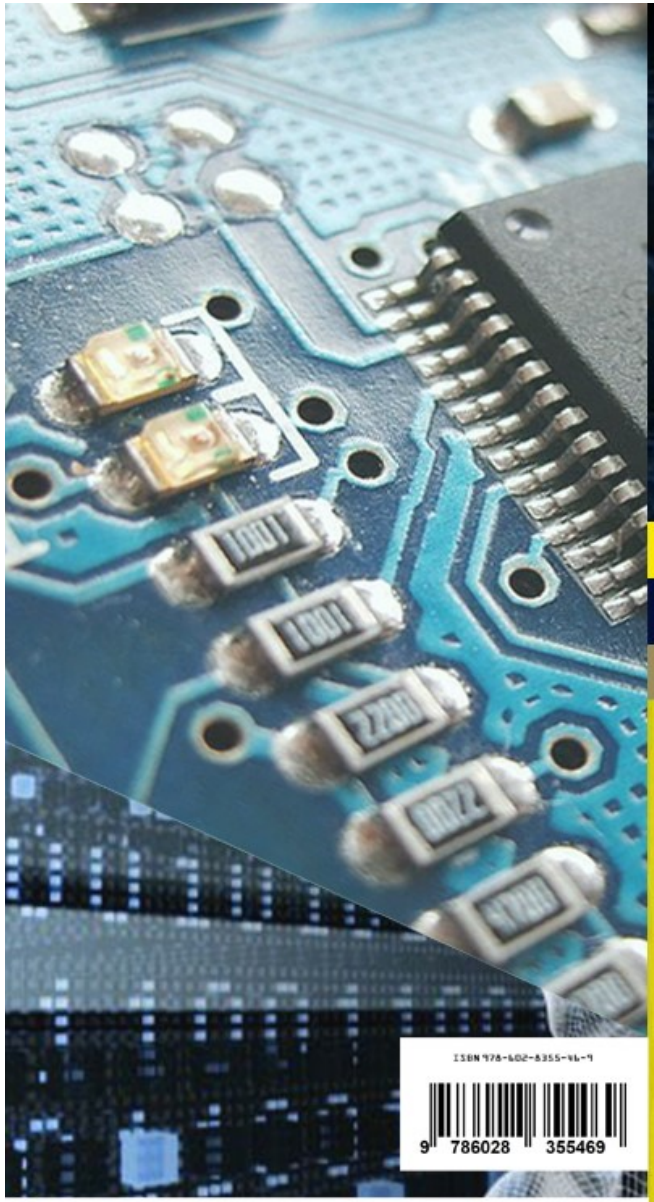

ISBN 978-602-8355-46-9

Untan Press

Badan Penerbit Universitas Tanjungpura (UNTAN PRESS)
Jl. A. Yani Pontianak 78124, INDONESIA
Telp. : (0561) 739630, Fax. : (0561) 739637

ANGGOTA IKAPI No. 004/KLB/03

PROCEEDINGS

INTERNATIONAL CONFERENCE 2016

INTERNATIONAL PROCEEDING ON APPLIED ELECTRICAL, ELECTRONIC, AND INFORMATIC ENGINEERING

infert 2016

ORGANIZER
ELECTRICAL ENGiNEERING DEPARTMENT
Jalan.Ahmad Yani, Pontianak, West Borneo.
Telp:(0561) 736180 / Fax:(0561) 740143
Email:infert2016@gmail.com
Website:http://infert.elektro.polnep.ac.id

Infert 2016
INTERNATIONAL CONFERENCE ON APPLIED ELECTRICAL, ELECTRONICS AND INFORMATICS ENGINEERING

Table Of Contents

The Design of Direct Memory Access Controller Core Based on Advanced Microcontroller Bus Architecture

Wendhi Yuniarto
Dept. of Electrical Engineering
Politeknik Negeri Pontianak
Pontianak 78124, Indonesia

Mariana Syamsudin
Dept. of Electrical Engineering
Politeknik Negeri Pontianak
Pontianak 78124, Indonesia

Irawan Suharto
Dept. of Electrical Engineering
Politeknik Negeri Pontianak
Pontianak 78124, Indonesia

Abstract— Direct Memory Access Controller (DMAC) is a module to support the system's performance in order to access one or more CPU / signal processors and multiple peripherals. The system uses the Advanced Microcontroller Bus Architecture (AMBA), specification defines a communication standard on-chip designed by ARM (Advanced RISC Machines). AMBA AHB is used for high performance systems, high clock frequency, and as well as to support efficient processor connections, on-chip memories and off-chip external memory interfaces. Moreover, it allows the access to high-bandwidth on the memory device chip. This system is integrated with a RISC processor and its memory controller SIEGE32 in System on Chip (SoC). The verification process is using ModelSim simulation that aims to ensure the system's functionality and compatibility and allow data transfer between cores in the SoC. The final stage are the synthesis and layout design by using CAD tools synopsys® with 0:18 µm CMOS technology design analyzer ™ and Astro ™. The implementation of chip layout produces a maximum clock frequency of 26.5 MHz and the chip area of 0.41 mm2.

***Keywords* — Peripheral, Processor, DMAC, AMBA, SoC**

VI. Preface

With the advances in technology, particularly in the field of communication using pictures and sound, real-time video and audio in the local telecommunications services has considered a very important thing. Data transfer on the video and audio requires two characteristics, namely; clear pictures and sound, as well as the need for speed at the time of the transition frame. The video and audio can be displayed in real time.

In order to support the signal processing of video / audio, DMA controller (DMAC) was designed as a module to support the system performance in order to access one or more CPU / signal processors and multiple peripherals. DMA is a hardware mechanism that allows components peripheral to transfer I / O data directly to and from main memory without the involvement of a processor system. The use of this mechanism is needed to be able to work.

Advanced Microcontroller Bus Architecture (AMBA) is a standard bus that was opened and carried by Advanced RISC Machine (ARM) processor. The bus system is used for on-chip communications standard in designing high-performance embedded microcontroller.

One of the most important parts of a system is the processor. Since the processor will be implemented in System on Chip (SoC), the processors must have a small chip area. The processor which is able to meet the above criteria is based on the architectural RISC processor (Reduced Instruction Set Computer). With a RISC architecture, the processor can only support the specific set of instructions that have been defined, and the set of instructions intended to run the program associated with the developed system. With a simple instruction set, the process of decoding the instruction will be more quickly, so that the execution time of the program is faster.

VII. The Method of Design

To guarantee the results validity of the design, it is essential to do stages in the design shown in Figure 1.

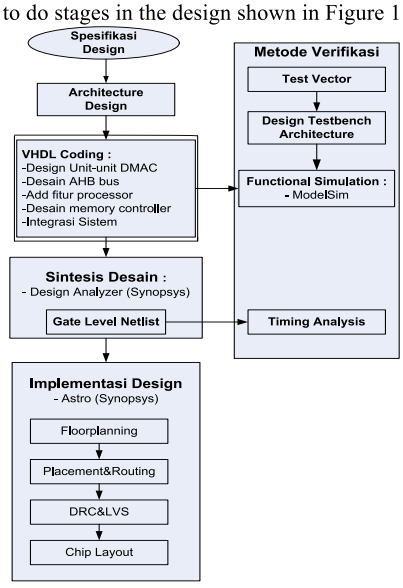

Figure 1. DMA *Controller System Integrasi Design Flow*

The technical stages are as follows:
Stage 1: VHDL Coding
Modeling functional design units in VHDL, then perform functional simulation to see the design's functionality.
Stage 2: Synthesis Design
At this point, the synthesis design is performed to generate the gate-level design descriptions. In the synthesis process, it is tested whether the design have problems in timing loop (combinational loop) or undesirable latch.
Stage 3: Test Coverage
Checking whether the coding that has been made is good enough or not. Good coding complexity will generate low and high test coverage, and vice versa.
Stage 4: Implementation of the chip layout

VIII. The Specification and Architectural Design

Specifications of DMAC as follows:
– DMA meets the AMBA standard bus for system's compatibility.
– DMA has the Compatibility with RISC processor system SIEGE32.
– DMA has the compatibility with the memory control system as an external memory interface models.
– Transfer rate 32 bits of data in RAM ideal memory model internal and external.
– Arbiter with 2 masters (processor SIEGE32 and DMAC).

Architecture on the DMAC is divided into several functional units. A logic buffer, which is separated as shown in Figure 2, is one unit. The unit is the most complex timing and block control, which consists of a large state machine.

DMA block architecture, divided into several main blocks, namely:
1. *Register and Counter* DMAC
2. *Timing* and *Control* DMAC
3. *Address Decode* DMAC
4. *Buffer* DMAC
5. *Arbiter* DMAC

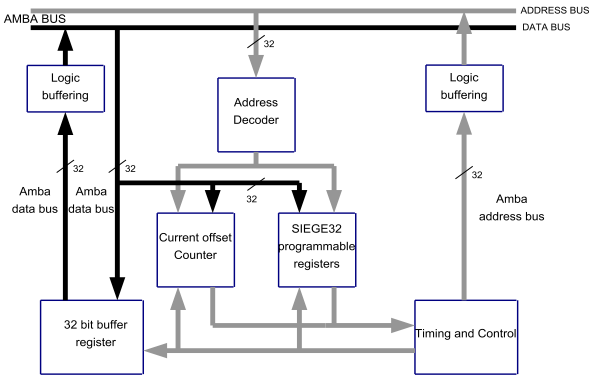

Figure 2. DMAC Architecture

DMAC Registers: Setup Data address from the processors that goes into DMAC, first stored in a register to be forwarded to the timing and control.
Timing and Control DMAC: Incoming and outgoing data are controlled by the Timing and Control. Setup data and the address are stored in AMBA Buffering DMAC.
AMBA DMAC Buffer: Buffer AMBA stores incoming data and remove it if necessary.
Address Decode DMAC: Address Decode generates clock signals enable at 32 bit registers DMA transfer.
Arbiter DMAC: Defines the priority "master" device which will use Bus.

Timing and Control Module is the most substantial on DMAC, since the data traffic is regulated by this module. The module is equipped with a transfer configuration on the AMBA bus master state machine, as well as AMBA bus slave registers.

IX. Verification and Implementation

In the verification process, functional tests is conducted to ascertain whether the design has worked according to the function that has been specified or not. The Functional design of DMAC is to be able to transfer data from the memory to memory without involving many RISC processors SIEGE32.

DMAC flow chart in Figure 3 shows the design verification, which interruptions received by DMAC will be forwarded to the processor and then configured. If the processor provides grants, the DMAC is ready to receive and send data memory.

The configuration process on the DMAC's processor setup will use the assembly program, in which the processor provides instructions for storing data address to register or move data from these registers to address memory.

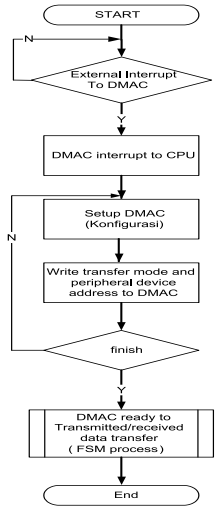

Figure 3. DMAC's Flowchart Process

DMAC will receive the results the configuration from the processor, when the bus access is given. The Instructions are available on the processor as shown in Table 1. On the other hand, Table 2 is the base address DMAC, and the data set-up to process the transfer DMAC.

Table 1: Instruction Memory

Test vector		
Instruction	Instruction Address	Information
E3B01003	00000000	MOV R1,#0x03
E5801000	00000004	STR R1,#0x0000
E3B02009	00000008	MOV R2,#0x09
E5802004	0000000C	STR R2,#0x0004
E3B0300B	00000010	MOV R3,#0x0B
E5803008	00000014	STR R3,#0x0008
E5904000	00000018	LDR R4,[R1,#0x0]
E5905004	0000001C	LDR R5,[R2,#0x4]
E5906008	00000020	LDR R6,[R3,#0x8]

Table 2: Base Address and Data Setup

Base Address	Data Setup	Registry Name	Information
0x00000000	0x00000003	*Incoming address*	Incoming Data, (32 *bit*).
0x00000004	0x00000009	*Outgoing Address*	Outcoming Data, (32 *bit*).
0x00000008	0x0000000B	*Transfer Length*	The amount of transferred data, (32 *bit*).

The configuration result from the processor is sampled to DMAC, hereinafter DMAC is ready to transfer and receive data to and from slave memory, according to the setup data from the processor issued through haddr_out DMAC to address the data on the memory controller.

The transmitted / received transfer process will be in accordance with the process of the state machine of DMAC. This process is done through the memory controller as an interface between the DMAC-memory and external memory.

Once DMAC obtains the access to bus, the DMAC will initiate the transfer from memory 1 to other memories by initiating source address (read) DMAC configuration results to memory controller address. The address will be received by memory 1, then Memory 1 will send data according to the address requested by DMAC. The data is stored in the data buffer DMAC registers. DMAC bus request back to the arbitrator if the arbitrator gave the grant, the DMAC will send the destination address (write) to the destination memory, and the data from DMAC will be transferred to the destination memory in accordance with the desired destination address. The transfer process will be repeated according to the number of words of data to be transferred.

The verification process for the design of DMAC system is integrated with SIEGE32 processor, and a memory controller using ModelSim, whereas the synthesis and layout as the implementation using CAD tools from Synopsys®, namely design_analyzer ™ for the synthesis and Astro ™ for layout.

The synthesis stage includes the mapping process in the form of VHDL design to a gate and optimization. From the extraction of parameters area and the timing of the core design DMAC which is already integrated with the SIEGE32 processor and memory controller after the synthesis process, it appears that the total cell area (without interconnection) is 0.206 mm2 and the maximum path delay of 20:02 ns or equivalent with clock speed of maximum 49.95 MHZ.

From the results extraction of the layout process (post routing), core area and the chip area of 0.29 mm2 and 0.41 mm2 were obtained, as well as the maximum path delay of 37.7173 0.1152 ns ns with slack, or the equivalent of clock speed the maximum of 26.5 MHz, as shown in Figure 4.

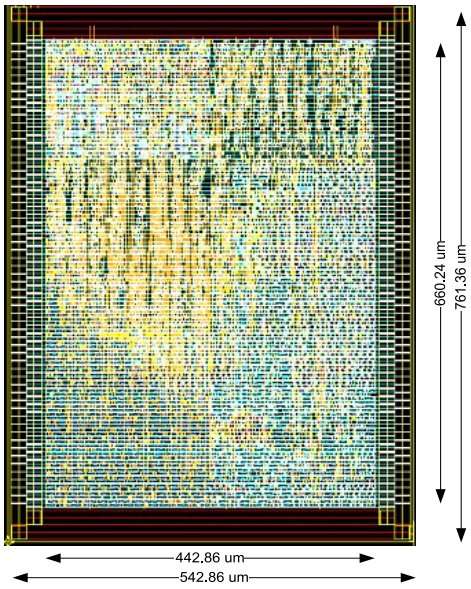

Figure 4. DMAC's Integration Layout

X. The Conclusion

The process of design, integration, verification which covers ModelSim simulation tools, synthesis and layout using CAD tools from synopsys® with 0:18 um CMOS technology on DMAC, the host processor and the memory controller SIEGE32, has been successfully conducted.

This research has also succeeded in adding new features on SIEGE32 processor configuration, which is equipped with interconnecting AMBA-BUS-based as an interface with other dedicated hardware.

The Design of DMAC with the host processor and the memory controller is implemented in layouts using CAD tools from Synopsys® with 0:18 µm CMOS technology. A maximum clock frequency of 26.5 MHz and the chip area of 0.41 mm2 was obtained.

Bibliography

[1]. ARM Corporation, *AMBA™ Specification (rev 2.0)*, ARM Limited, 1999.

[2]. ARM Corporation, *PrimeCell* DMAC *(PL080)*, ARM Limited, 2000.

[3]. Denny Tresna Seswara, *Perancangan dan Implementasi Sistem Host Processor H.263+/ MPEG-4 Codec pada FPGA Altera Stratix® berbasis LEON Processor*. Laporan Tesis Magister ITB, 2006.

[4]. Gareth Morris, *AMBA Bus* DMAC *Specification*, Draft 1.03, 2001.

[5]. Gareth Morris, DMAC *for AMBA Bus IP Core Final Report (version 1.00),* project number: 12, 2001. gwem@preromanbritain.com

[6]. Jiri Gaisler, *LEON-2 Processor User's Manual*, Gaisler Research, Juli 2005. http://www.gaisler.com/.

[7]. Michael Keating, Pierre Bricaud, *Reuse Methodologi Manual – For System on Chip Design*, 2nd ed. Boston : Kluwer Academic Publisher, 1999.

[8]. Nana Sutisna, *Desain dan Implementasi RISC Processor 32-Bit SIEGE32,* Laporan Tugas Akhir ITB, 2005.

[9]. Steve Furber, *ARM-System on Chip Architecture*, Great Britain : Pearson education Limited ,Addison-Wesley, 2000.

[10]. Synopsys, *Astro user Guide*, Synopsys, 2004.

[11]. Synopsys, *Design Synthesis*, Synopsys, 2004.

[12]. Synopsys, *DesignWare DW_AHB_DMAC Databook*, Synopsys, 2006.