

DESIGN AND PERFORMANCE ANALYSIS OF LOW POWER AHB DMA CONTROLLER

Potnuru Karishma¹, Kommala Kavya²

¹P.G Scholar, Sanketika Vidya Parishad Engineering College, Visakhapatnam, India.

²Assistant Professor, Sanketika Vidya Parishad Engineering College, Visakhapatnam, India.

karishmapotnuru@gmail.com, kommala.123@gmail.com

DOI: <https://www.doi.org/10.58257/IJPREMS36494>

ABSTRACT

DMA (Direct Memory Access) Controller is used to transmission of data from Memory Side to peripheral side and peripheral side to memory side with keeping CPU free during data transmission. AHB (Advance High Performance Bus) is a new generation of AMBA bus which is intended to address the requirements of high performance synthesizable designs. It is a high-performance system bus that supports multiple bus masters and provides high-bandwidth operation. Power consumption and power-related issues have become a major concern for most designs. The primary method used for reducing power has been supply voltage reduction, this technique begins to lose its effectiveness as voltages drop to sub threshold voltage range and further reductions in the supply voltage begin to create more problems than are solved. In this work a new approach to the synthesis problem for finite state machines with the reduction of power dissipation as a design objective. A finite state machine is decomposed into a number of coupled sub machines. Most of the time, only one of the submachine will be activated which, consequently, could lead to substantial savings in reduction power consumption. Designer compute two sub-FSMs that together have the same functionality as the original FSM. To minimize the average switching activity, search for a small cluster of states with high stationary state probability and use it to create the small sub-FSM. This way designer will have a small amount of logic that is active most of the time, during which is disabling a much larger circuit, the other sub-FSM.

Keywords- Direct Memory Access, Direct Memory Access Controller, Advance High Performance Bus, Finite State Mission, Power consumption, propagation delay.

1. INTRODUCTION

The consumer demand for more noteworthy usefulness and higher execution, yet additionally for lower costs includes critical weight on System-on-Chip (SoC) producers. The proceeding with propels in process technology, and capacity to plan exceedingly complex SoCs does not come without a cost. So the next generation of procedures without a doubt achieves the next generation of challenges. With ever increasing System-on-Chip (SoC) unpredictability, energy consumption has turned into the most basic imperative for today's integrated circuit (IC) outline. Thusly, a great deal of exertion is spent in outlining for low-control scattering. Power consumption has turned into an essential limitation in outline, alongside execution, clock recurrence and bite the dust estimate. Low power can be accomplished just by outlining at all levels of reflection: from architectural design to licensed innovation (part determination and physical usage. Designers should utilize segments that convey the most recent improvements in low power innovation. The best power savings can be accomplished by settling on the correct decisions from the get-go amid the framework and design level of deliberation. Notwithstanding utilizing power conscious hardware design methods, it is vital to save power through careful design of the working framework and application programs. In this work, we proposed a technique that is additionally in view of specifically killing bits of a circuit. Our approach is propelled by the perception that, for a FSM, dynamic changes happen just inside a subset of states in a period of time. In a CMOS circuit, by and large, the exchanging action of the gate output contributes most to the total power scattering. For FSM low power design, parcelling method turns out to be viable for diminishing exchanging movement. That is, segment the first FSM into a few littler sub FSMs and just a single of them is dynamic at once. Be that as it may, two issues are frequently presented

2. DMA CONTROLLER ARCHITECTURE

Under AMBA (Advance microcontroller bus architecture) there are 3 protocols defined, which are as follows, AHB (advance high performance bus), APB (advance peripheral bus), ASB (Advance system bus) among which AHB is used for high speed, low latency operation and high frequency operation. AHB is a new generation of AMBA transport which is expected to address the necessities of high-performance synthesizable outlines. It is a high-performance system bus that backings multiple bus experts and gives high bandwidth operation.

1. Features of AHB (Advance high performance Bus):
2. It supports multiple bus masters and provides high band operation.
3. AHB has high clock frequency operation.

4. It can handle burst transfer.

AMBA-AHB has following four components:

1. **AHB Master:** It initiates read write operation. Only single master activates one time for using bus.
2. **AHB Slave:** It responds the read write operation at particular address. It also replies to AHB master that operation has been performed or not.
3. **AHB Arbiter:** It ensures that only one bus master is allowed at single time.
4. **AHB Decoder:** It decodes the address of each transfer and provides a select signal for the slave that is involved in the transfer.

DMA can be programmed using its AHB slave interface by the system processor. The DMA's master controller is at first idle yet comes to dynamic mode after processor programs an arrangement of control registers through its AHB slave interface

2.1 WORKING OF DMA CONTROLLER ARCHITECTURE

The AHB Slave block interfaces the AHB bus to the DMA channel registers. To initiate a transfer, CPU will program all the operational registers of the peripheral IP and all the operational registers of the AHB-DMA. The value programmed into "Block Size Register" of the DMA Controller for a particular channel (Tx or Rx). FIFO (Tx& Rx) inside the DMA, can have a either a fixed or para-meterizable depth. In case of Transmit, When-ever FIFO is not full, Data fetched from the system memory, will be written into the Tx-FIFO. Similarly, in case of receive, when-ever FIFO is not empty, data read from the Rx-FIFO is stored to system memory. Now DMA control signal check for transmitting/reviewing process. Only after the activation of this control signal, data can proceed further. After activation of control signal, receiving side (Ahb side or peripheral interface side) will give the status that it is ready for transition or not. At every transmission, last byte will be matched. When the last bytes are transferred, an "End of Transfer" (Tx or Rx) is generated by the DMA for conforming the completion of operation

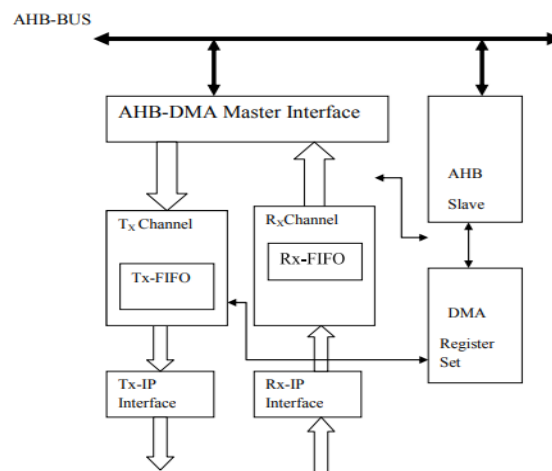


Figure 1: AHB-DMA Block Diagram

3. SIMULATION RESULTS

Internal Architecture of DMA controller is simulated by Xilinx in Verilog HDL.

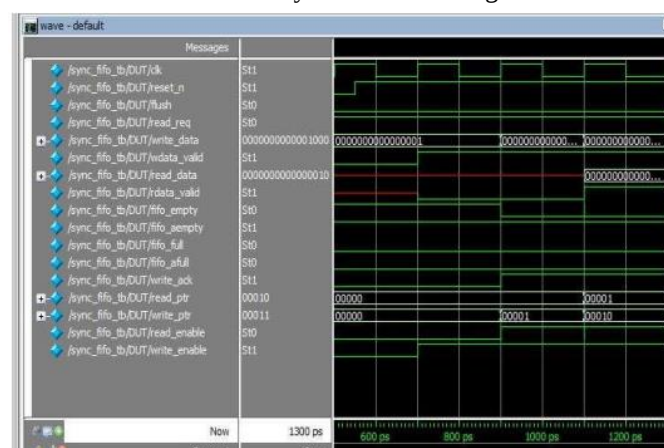


Figure 2: Simulation Result of Verilog coding of read control and write control logic

In this above fig input data is generated by test bench through the 16 bit LFSR for write data operation. The position of write pointer is also declared by test bench. According to the space of FIFO, fifo_full and fifo_afull signal are activate. Write_enable and Read_enable signals are also generated by test bench for starting the transmission process.

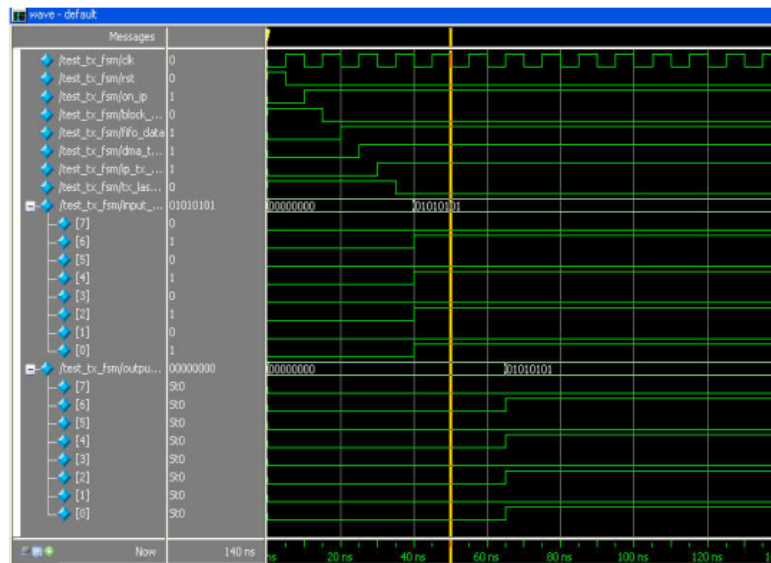


Figure 3. Simulation Result of Verilog coding of DMA Tx FSM

In above figure clk is input which is forced 0 and 1 continuously. Other single bit inputs of Transmitting DMA controller are on_ip, which is forced here 1, block_size bar which is forced 0, fifo_data value is given to 1, dma_tx_process is on. Also 8 bit input is applied at the transmitting end which is received at the receiving end.

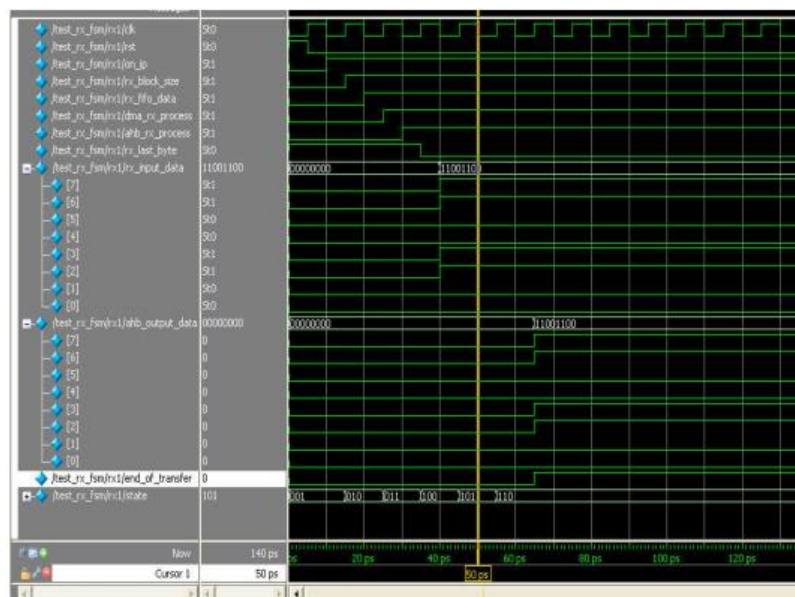


Figure 4. Simulation Result of Verilog coding of DMA Rx FSM

In above figure clk is input which is forced 0 and 1 continuously. Other single bit inputs of Transmitting DMA controller are on_ip, which is forced here 1, rst input signal is used for reset the architecture, block_size which is forced 1, fifo_data value is given to 1, dma_tx_process is on. Also 8 bit input is applied at the peripheral side end which is received at the ahb side. State signal is used for showing states of DMA controller. After receiving last byte, an output signal end_of_transfer will be on that shows the completion of receiving of data.

4. CONCLUSION

The proposed low-power AHB DMA controller leverages finite state machine (FSM) decomposition to achieve significant power savings, addressing critical needs in power-sensitive high-performance systems. By breaking down a single large FSM into two coupled sub-FSMs one small and frequently active, and the other larger but mostly idle power dissipation is minimized. This approach allows the smaller sub-FSM, which handles the high-probability states, to remain active most of the time. This, in turn, significantly reduces the switching activity in the larger sub-FSM, achieving substantial reductions in dynamic power consumption, especially beneficial as voltage scaling reaches its

practical limits. This design balances the high throughput requirements of AHB protocol with low-power operation, making it suitable for power-constrained applications in embedded systems. Through optimizing average switching activity and minimizing idle circuit activity, the FSM decomposition methodology enables the DMA controller to perform data transfers efficiently while keeping the CPU free and preserving the battery life in portable devices. Future work could explore adaptive switching techniques based on real-time data transfer patterns, further enhancing power savings while maintaining system performance.

5. REFERENCE

- [1] J. Liang, Swaminathan, A SOC: a Scalable, Single chip Communications Architecture, Tessier, R. Parallel Architectures and Compilation Techniques, 2000 Proceedings International Conference, pp.37-46.
- [2] Guoliang Ma, Hu He, Design and Implementation of an Advanced DMA Controller on AMBA-Based SoC, IEEE 8th International conference, 2009, pp 419-422.
- [3] Flynn, D., AMBA: Enabling Reuseable On-Chip Designs", IEEE Micro, July-August 1997, 17(4), pp. 20-27.
- [4] AMBA Specification (rev2.0) and Multi Layer AHB Specification, Arm: <http://www.arm.com>, 2001.
- [5] A. Bogliolo, and G. De Micheli, A survey of Design Techniques for System Level Dynamic Power Management, IEEE Trans. on VLSI Systems, June 2000, 8 (3), pp. 299-316.
- [6] A. Macii, E. Macii, L. Benini, and M. Poncino, "Selective Instruction Compression for Memory Energy Reduction in Embedded Systems", ISLPED99: ACM/IEEE 1999 International Symposium on Low Power Electronics and Design, San Diego, California, August 1999, pp. 206-211.
- [7] A. Bogliolo, and G. De Micheli, A survey of Design Techniques for System Level Dynamic Power Management, IEEE Trans.on VLSI Systems, June 2000, 8 (3), pp. 299-316.
- [8] A. Macii, E. Macii, L. Benini, and M.Poncino, "Selective Instruction Compression for Memory Energy Reduction in Embedded Systems", ISLPED99: ACM/IEEE 1999 International Symposium on Low Power Electronics and Design, San Diego, California, August 1999, pp. 206-211.