Original Research Paper | Engineering | Volume : 09 | Issue : 12 | December 2024 | e-ISSN : 2456-5040

HARDWARE IMPLEMENTATION OF ARM CORTEX-M3 DESIGN ON FPGA

Trisha Maddanna[1], Vaishnavi U[2], Gahan A V[3]

[1,2] Student, Department of Electronics and Communication, Bangalore Institute of Technology, Bangalore, India
[3] Assistant Professor, Department of Electronics and Communication, Bangalore Institute of Technology, Bangalore, India

ABSTRACT

The Cortex-M3 processor is designed for high-performance, cost-effective platforms across a wide range of applications, including microcontrollers, automotive body systems, industrial control systems, and sensors. It is specifically chosen for its classification as a microcontroller processor from ARM, making it suitable for embedded IoT applications. The System-on-Chip (SoC) design integrates the ARM Cortex-M3 32-bit processor, with a customized IoT subsystem and peripherals on an FPGA. The design is tailored to meet specific requirements and specifications, ensuring chip functionality validation on the Xilinx FPGA platform, which was previously implemented on an Intel FPGA. The existing design had a limited number of master and slave ports for interfacing peripherals. The customization of peripherals is, to increase the number of master and slave ports by modifying the AHB and APB interconnects while modifying memory mapping of the ports to our specification, using an address decoding logic. FPGA prototyping is done to validate the test cases again for the modified Cortex-M3 Design on the Xilinx platform. The Keil uVision5 software is used to translate C-programs into binary files that is loaded into the SoC's Memory for the processor to run and Xilinx Vivado is used for simulation. The Hardware implementation is done on the Arty100– T FPGA platform to test and verify the functionality of the chip.

KEYWORDS: Cortex-M3, System-on-Chip (SoC), FPGA (Field Programmable Gate Array), Prototyping

INTRODUCTION

System-on-a-Chip (SOC) Design

System-on-a-Chip (SoC) design is the process of integrating many or all of an electronic devices components onto a single chip. SoCs are important for the next generation of computing because they offer advantages like power efficiency and improved production methods. SoC design techniques are based on ASIC (application-specific integrated circuit) design methods. An effective SoC design style should align with existing ASIC design flow.

FPGA Prototyping

FPGA prototyping is a well-established technique for verifying the functionality and performance of application-specific ICs (ASICs), application-specific standard products (ASSPs) and system-on-chips (SoCs) by porting their RTL to a field programmable gate array (FPGA). It is also known as SoC prototyping. FPGA prototyping is done to validate the test cases again for the modified Cortex-M3 design on the Xilinx platform.

ARM Cortex M Series

The ARM Cortex-M is a group of 32-bit RISC ARM processor cores licensed by ARM Limited [1]. These cores are optimized for low-cost and energy-efficient integrated circuits.

The ARM Cortex-M family are ARM microprocessor cores that are designed for use in microcontrollers, ASICs, FPGAs, and SoCs.

ARM Cortex-M3 Design of Processor with IOT subsystem

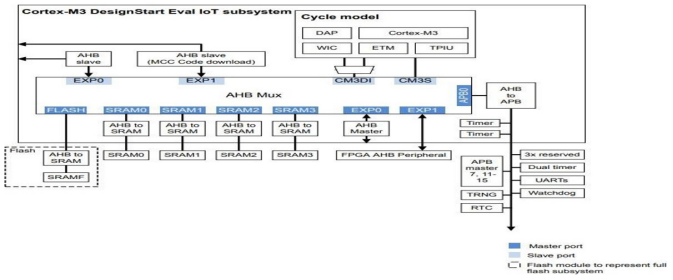

Figure 1: Cortex-M3 Design with IOT subsystem

The Cortex-M family consists of CortexM0, Cortex-M1, Cortex-M3, Cortex-M4, Cortex-M7 and many more. The Cortex M3 processor is chosen as it is a microcontroller class processor from ARM and can be used for embedded IOT applications. The Cortex-M3 Design system is representative of a fully functioning SoC design [1], without any advanced power-saving functionality, includes peripherals of several levels in hierarchy.

Advantages of Cortex-M3 processor:

- Low power consumption
- High performance
- Platform security and ease of use
- Debug and trace capabilities

To design the SoC, the existing designs from ARM [2] are modified according to the requirements and specifications, and to validate functionality of chip on Xilinx FPGA platform.

The limitations of the existing system are that it is implemented on Intel FPGA and has a limited number of master and slave ports for interfacing peripherals. The customization of peripherals is, to increase the number of master and slave ports by modifying the AHB and APB interconnects while modifying memory mapping of the ports to our specification, using an address decoding logic.

FPGA prototyping is done to validate the test cases again for the modified Cortex-M3 design on the Xilinx platform.

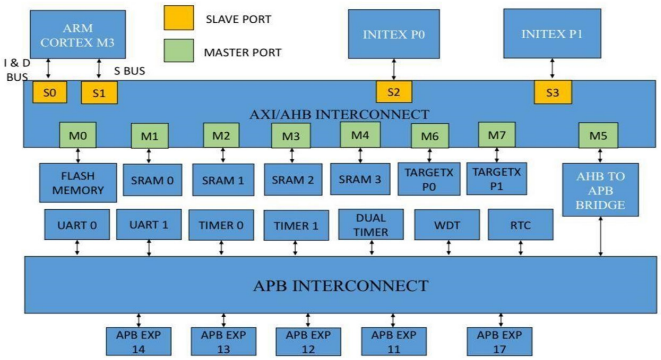

Figure 2: Cortex-M3 Design with IOT subsystem

Features of Cortex-M3 Design with IOT subsystem [2]:
AHB comprises the high speed peripherals whereas APB comprises the low speed peripherals. The AHB interconnect masters initiate data transfers and the slaves respond to the master. The AHB mux is the AHB interconnect interfacing the AHB slaves which are memory mapped peripherals. The Core is connected to AHB interconnect through 2 slave ports: 1. Data and instruction, 2. System ports.

The processor in ARM Cortex-M3 Design is a fixed configuration that enables easy evaluation access to the Cortex-M3 processor technology without the flexibility to configure the design. It is delivered in two forms:

- Obfuscated RTL - Cortex-M3 Design deliver the processors as obfuscated Verilog files. These RTL files are not encrypted, but the internal logic is flattened, and the signal names replaced with random names. You can simulate it with standard Verilog simulators and synthesize it for FPGA testing. The obfuscated RTL ois preconfigured, it is a synthesizable Verilog version of full Cortex-M3 processor.

- Cycle Model – It includes visibility of the internal processor architectural registers, for simulation and debug purposes. The model also generates a Tarmac log, which is a textual trace output file that contains all the instructions executed, and register and memory transactions.

Cortex-M3 Design includes an IoT compute subsystem that extends the processor with the following:
- Bus interconnect and SRAM controller.
- Timer peripherals required by the ARM mbed software.
- Radio product expansion interface (ARM Cordio® radio IP)
- Expansion capability for embedded flash controller.

The features of the IoT subsystem in Cortex-M3 Design include:
- Instantiation of the Cortex-M3 processor with debug and trace.
- Two Advanced Peripheral Bus (APB) timers.
- A multi-layer AMBA AHB-Lite interconnect.

A memory system consisting of:
1. AHB - Lite master expansion and two APB4 master expansion ports dedicated for connection to a flash interface (on chip or external, and optionally with cache).
2. Static memory arranged in four banks of 32KB.

The AHB matrix consists of 3 stages:
1. Input stage - involves buffering and synchronization of AHB signals.
2. Matrix decoder - generates signals, which slave has to be selected based on the input address taken. Here, peripherals are mapped to the address which comes, AHB signals from master go to the AHB signals of slaves.
3. Output stage - mapping AHB output signals from specific master to slave.

The APB matrix consists of 3 stages:
1. AHB decoding - involves selecting of the APB port.
2. APB decoding - The APB decoder determines which APB slave ports are activated based on the address.
3. Address Mapping - Address mapping in the system is achieved by using an

PROPOSED METHODOLOGY
Customization of peripherals to increase the number of master and slave ports by modifying the AHB and APB interconnects [3]:

- Replacing existing peripherals to add design features.
- Digital peripherals like: General purpose Input Output (GPIO) pins; Timers; Pulse Width Modulator (PWM) – usually for motor or power electronic system control; UART for serial communication; SPI (Serial Peripheral Interface) for external hardware modules such as LCDs; I2C / I3C – commonly used for sensors.
- Tightly-coupled timer peripherals – There are two Advanced Peripheral Bus (APB) timer modules that are instantiated within the subsystem hierarchy in Cortex-M3 Design. These APB timers are reserved for the operating system.
- Tightly-Coupled Memory interfaces - There are four AHB to SRAM interface modules instantiated within the subsystem hierarchy in the CortexM3 Design. This multi-bank memory configuration gives the fastest possible access. The IoT subsystem can be configured to provide 1, 2, 3 or 4 banks of SRAM.
- Primary code memory - In the subsystem included with the CortexM3 Design, there is a dedicated AHB interface that provides access to flash storage. The block RAM is preloaded in simulation with the executable code. For the SoC implementation, it will be necessary to replace the

block RAM with a suitable equivalent, either SRAM or a flash interface.

- Closely coupled peripherals - The subsystem in Cortex-M3 Design provides APB master expansion ports used in the example system for UARTs, additional timers, Watchdog, Real Time Clock (RTC), and True Random Number Generator (TRNG). The remaining five APB master expansion ports can be used for communications interfaces, or other peripherals with a low latency requirement. There are six reserved interrupt lines that are suitable for additional peripherals.
- Adding peripherals to memory map - In the IoT subsystem included with the Cortex- M3 Design, both the APB expansion and the FPGA peripheral APB interfaces have unused ports that can be used to add peripherals, without any modifications. If these ports are not sufficient and an existing peripheral cannot be replaced, we are required to modify at least one of the AHB interconnects in the system.
- Memory map definition - The architectures used in the Cortex-M3 processor defines a memory map that allocates address ranges into regions. This allows the built-in peripherals like the interrupt controller and debug components to be accessed by simple memory access instructions, allowing system features to be accessible in the C program code The memory mapping is modified according to our specification. The modification is done using an address decoding logic.

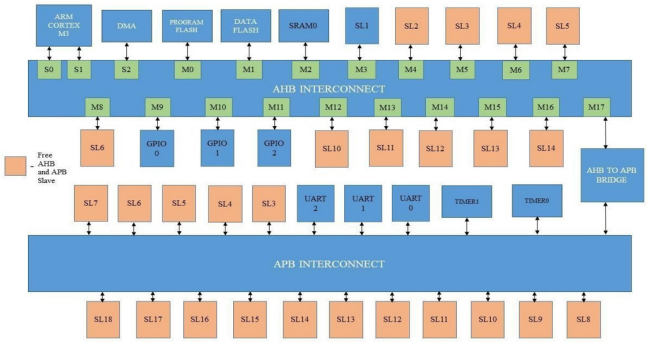

Figure 3: Modified Interconnect Platform

Features of Modified Interconnect Platform in the Cortex M3 Design:

- The AHB and APB Interconnects are modified by adding new slave ports and have to be memory mapped to the system by modifying memory mapping to our specification. The modification is done using an address decoding logic i.e the CPU address comes, based on which it decides which one of the slave gets access.
- Based on the decoder value, the slave is selected (for eg: the APB decoder has a particular set of bits toggling from 1 memory space to another, differentiating for the ports to select APB slave, whereas in AHB decoder, all 32 bits are used for selecting the slave).
- The AHB and the APB Interconnects platforms are extended to 18 ports.
- The input and output ports are modified to more than 15 ports which were 8 ports initially.
- The slaves may be SRAM, UART or GPIO according to the requirements or specifications.
- The SRAM instantiates dedicated block RAMs and 1 bank of SRAM is designed.
- The SRAMs are designed for 128KB for more memory capacity and Flash memory is modified from 256KB to 512KB.

Implementation of Modified ARM Cortex-M3 Design on Xilinx FPGA (Steps):

- In the existing design, the functionalities were verified in Intel FPGA whereas in the proposed design, the functionalities are verified in Xilinx FPGA.
- FPGA prototyping is done to validate the testcases again for the modified Cortex-M3 design on the Xilinx platform.
- During the FPGA flow - the clock is replaced with Mixed-Mode Clock Manager (MMCM) which takes reference clock from board and generates clock frequency required.
- The Keil uVision 5 software is used to translate the C-programs into binary files that is loaded into the SoC's Memory for the processor to run and Xilinx Vivado is used for simulation.
- The Hardware implementation is done on the Arty 100 - T FPGA platform to test and verify the functionality of the chip.
- This involves the process of generating the Bit file, programming FPGA Board, after giving reset to CPU, the application code gets executed from Flash memory.
- The programmed software algorithm is ARM mbed into the programmed FPGA board and it prints the output infinite times on the monitor screen.

RESULTS AND DISCUSSIONS

FPGA prototyping is done to validate the test cases again for the modified Cortex-M3 design on the Xilinx platform. During the FPGA flow – the clock is replaced with Mixed- Mode Clock Manager (MMCM) which acts a reference clock from FPGA board and generates required clock frequency.

The Hardware Implementation Is done on the Arty 100 – T FPGA platform to test and verify the functionality of the chip. This involves the process of generating the Bit file, programming FPGA Board, after giving reset to CPU, the application code gets executed from Flash memory. The programmed software algorithm is ARM mbed into the programmed FPGA board and it prints the output infinite times on the monitor screen as shown in the Fig 4. The Power and LUT reports of the Design are shown in Fig 5 and Fig 6 respectively.

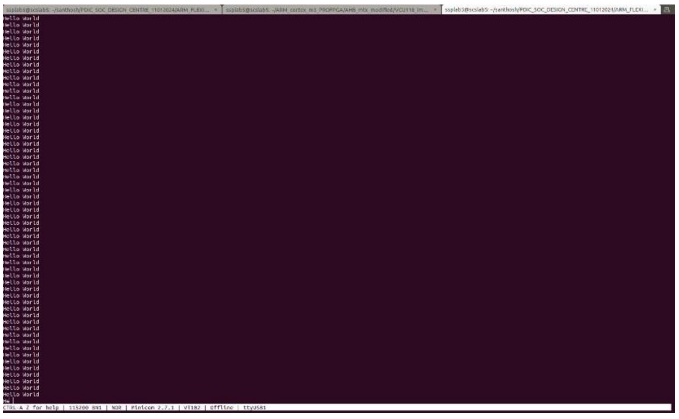

Figure 4: Output printed infinite times on the monitor screen

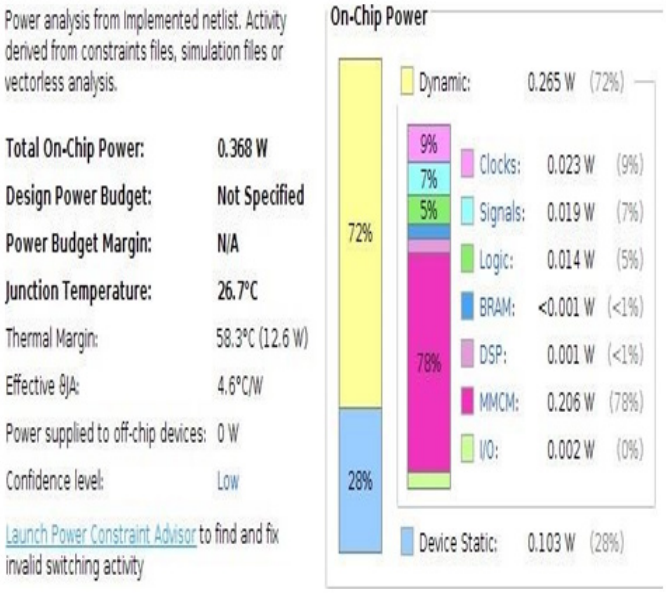

Figure 5: Power Report

Site Type	Used	Fixed	Available	Util%
Slice LUTs	25397	0	63400	40.06
LUT as Logic	24482	0	63400	38.62
LUT as Memory	915	0	19000	4.82
LUT as Distributed RAM	152	0		
LUT as Shift Register	763	0		
Slice Registers	14071	0	126800	11.10
Register as Flip Flop	14071	0	126800	11.10
Register as Latch	0	0	126800	0.00
F7 Muxes	145	0	31700	0.46
F8 Muxes	19	0	15850	0.12

Figure 6: LUT Report

The Total on-chip Power analysis from implemented netlist is 0.368W. The dynamic power consumption is 0.265W and has a static consumption of 0.103W.

The Cortex-M3 includes packages such as RTL, Execution Test bench, and FPGA Evaluation Flow. The three pipeline stages in Cortex M3 are Instruction fetch, Instruction decode and Instruction execution. The Cortex M3 has integrated sleep modes, that can be entered using WFI or WFE instructions. It also has separate clocks for essential blocks.

CONCLUSION\

In the existing design, the functionalities were verified in Intel FPGA whereas in the proposed design, the functionalities are verified in Xilinx FPGA. The AHB and APB Interconnects platforms are extended to 18 ports. The memory mapping of the slaves to the system is done using address decoding logic. The slaves may be SRAM, UART or GPIO according to the requirements or specifications. The SRAMs are designed for 128KB and Flash memory is modified from 256 KB to 512 KB for more memory capacity.

The Keil uVision 5 software is used to translate the C-programs into binary files that will be loaded into the SoC's Memory for the processor to run and Xilinx Vivado is used for simulation. The Hardware implementation is done on the Arty 100 - T FPGA platform to test and verify the functionality of the chip. The programmed software algorithm is ARM mbed into the programmed FPGA board and it prints the output infinite times on the monitor screen.

Future Scope

Prototyping an ARM Cortex-M3 design on an FPGA can offer several applications across a variety of fields as designers and developers can significantly reduce time-to-market, improve design flexibility, and perform robust system-level testing in the real-world environment. Some key applications include:

- Embedded Systems Development and Testing
- IoT and Edge Devices
- Motor Control and Robotics
- Signal Processing
- Communication Systems
- System - on - Chip Design and Validation
- Custom Hardware Interfaces
- Security and Cryptography

REFERENCES

1. Joseph Yiu. (2019): System-on-Chip Design with ARM Cortex-M Processors Reference Book, Version -1.0.3, ARM Education Media, Cambridge.
2. ARM Cortex-M3 DesignStart EvalRTL Testbench and User Guide, ARM Limited, 2017.
3. ARM Cortex-M3 DesignStart Eval Customization Guide, ARM Limited, 2017.