International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 03 | Mar 2020 www.irjet.net p-ISSN: 2395-0072

Design of RISC-V Bit Manipulation Instruction IP using Bluespec SystemVerilog for Shakti Coprocessors

S. Snehashri[1], Mr. Md. Nooruzzaman Khan[2]

[1]*Department of Electronics and Communication Engineering, Meenakshi Sundararajan Engineering College, Chennai, India*
[2]*Assistant Professor, Department of Electronics and Communication Engineering, Meenakshi Sundararajan Engineering College, Chennai, India*

Abstract - : *Most VLSI designs have two most important parameters: high performance and low power. Bit Manipulation Instructions have helped in the speed up of generations of processors from Intel 8086 to the latest x86 version 8.2. Low power requires area reduction in the form of decreasing the number of logic gates/multiplexers used to implement any design. Clifford Wolf has proposed a Bit Manipulation Extension to the existing RISC-V Instruction Set Architecture (ISA) that on adoption can be used for code density reduction. RISC-V (pronounced "risk-five") is a new instruction-set architecture (ISA) , the latest version of RISC in the Computer Architectures realm , that is a completely open ISA and freely available to academia and industry. This project aims at not only simulating this design of Bit Manipulation IP using the Bluespec SystemVerilog language but also on synthesizing in Xilinx Vivado ise 2018.3. All 106 Bit Manipulation Instructions have been executed in a single clock cycle and have been optimized in terms of Look-up Table (LUT) count.*

1. INTRODUCTION

Even though ARMv8 and x86 have introduced BMI Instructions for better performance and memory reduction, RISC-V has not yet added this B extension into their Instruction Set Architecture (ISA). The IP design aims at synthesized these instructions in the form of Bit Manipulation IP with final results on area optimization by reducing the LUT count as well executing all instructions in a single clock cycle for better performance. This can be used as a hardware reference while confirming the adoption of the B-extension strategy.

The advantages of BMI do not restrict itself to the performance characteristics of any embedded system but also works wonders in security cryptographic and IoT applications. Most relevant fields for advanced BMIs are machine learning and cryptographic applications as there are many algorithms that operate on binary data and often use bit rotations and shuffles.

Even though the applications are so varied and useful, the official ISA supports only two BMI instructions, ie, shift left and shift right. The RISC-V B Extension strategy proposal focuses on the hundred and six instruction for 32 bit as well 64 bit architecture.

The intended conclusion is to design the maximum optimized combinational block for these hundred and six instructions and execute these in a single cycle manner to give enhance speed, more than area or performance.

2. DESIGN OF THE COMBINATION BITMANIP IP BLOCK

The plan of the combinational circuit is as given in Fig 1.1. This was programmed with the help of Bluespec SystemVerilog (BSV), a high-level functional hardware description programming language. It is compiled in Open-BSC compiler and linked to verilator, which is used to convert the BSV automatically into Verilog.

The circuit is fed by a maximum of four inputs including three source operands and the instruction encoding. The source operands vary from 32 to 64 depending on the architecture. The instruction encoding is at a constant 32 bits, with the capacity for immediate operands.

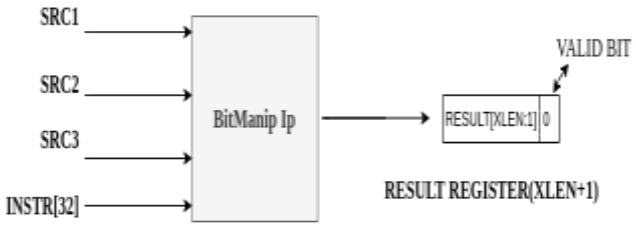

Fig- 1: Design of Bit Manipulation IP Block

The IP can be imagined as a black box that works on the source operands and returns a result register. The instruction can also be any variant in the 32 bits. If the opcode does not match with the opcodes given in BMI, the valid bit in the result register will be reset. The entire result register is (XLEN+1), where XLEN can be 32 or 64 depending on the architecture.

If a BMI operation is done, then the Valid bit will be set, as it occupies the zeroth LSB Bit.

3. PROPOSED ARCHITECTURAL BLOCK

The final standard defines a range of Z-extensions for different bit manipulation instructions, with the "B" extension itself being a mix of instructions from those Z-

extensions. This design is grouped based on the different functionalities, with some instructions overlapping one or more blocks.

1) ZBB- Base Instructions
2) ZBS - Single Bit Instructions
3) ZBR - Cyclic Redundancy Check Instructions
4) ZBM- Matrix operations
5) ZBA- Address Calculation
6) ZBT- Ternary Instructions
7) ZBC-Carry Less Multiplication Instructions
8) ZBF- Field Place Instructions
9) ZBE- Extract Instructions
10) ZBP-Permutations Instructions

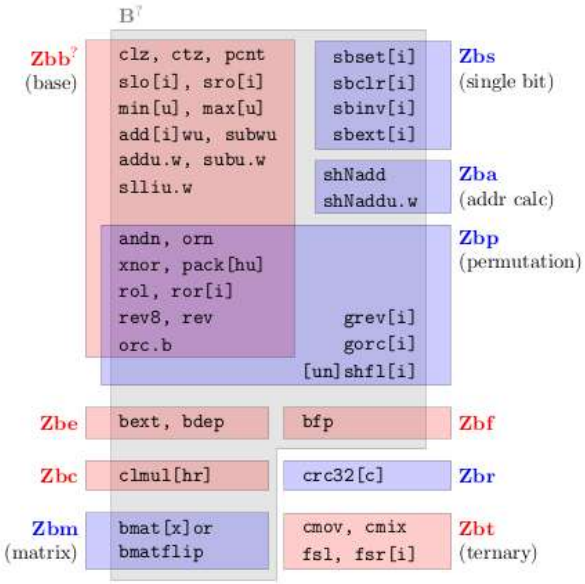

FIG 2: PROPOSED ISA IN ZB EXTENSION

4. BASE INSTRUCTION OPTIMIZATION- ALGORITHM (ZBB)

The functionality can be referred to in the specifications of the bitmanip draft given in references [1]. Let us focus on the hardware specification/algorithm implemented in the Bit Manipulation IP.

4.1 ZBB

This consists of base instructions that include shifters and rotators majorly, along with the pack, counting leading/trailing zeros and combination of logical gates like and, or and xor.

*W instruction variants on RV64 with the semantic of the matching RV32 instruction. Those instructions ignore the upper 32 bit of their input and sign-extend their 32-bit output values.

The logic implemented here uses a barrel rotator along with twiddling of shift amount values and mask values.

A barrel shifter is a specialized digital electronic circuit with the purpose of shifting an entire data word by a specified number of bits by only using combinational logic, with no sequential logic used. The simplest way of achieving this is by using a series of multiplexers where one output is connected to the input of the next multiplexer in the chain, in a specific manner that depends on the amount of shift specified. The design in the IP used for shifting in ones and rotation is a single 2*XLEN barrel right rotator. The full design is as shown in the figure 2 below;

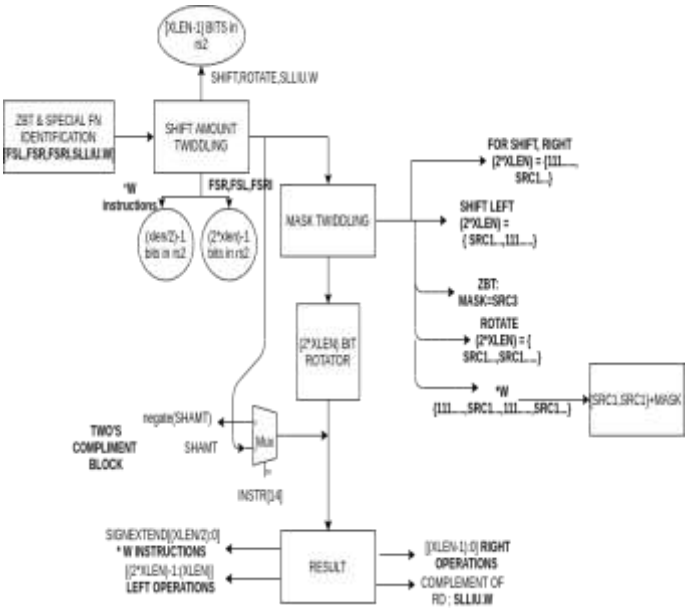

Fig 3: Complete Algorithm of Shifter and Rotator In Block Representation

5. CONCLUSIONS

This project has successfully designed and simulated the RISC-V Bitmanip IP proposed by Clifford Wolf in https://github.com/riscv/riscv-bitmanip using Bluespec SystemVerilog and can be run and tested by anyone using the link:https://gitlab.com/shaktiproject/cores/bbox.

BSV has been converted into Verilog using VERILATOR and the FIGURE 4 shows the outputs for any given set of inputs in the test bench. Charts 1 and 2 shows the lut analysis report for all reduced blocks as well as for ZBB separately.

[110] Sent ANDN to dut

[110] PASSED

[110] inst= 40007033 r rs1 = 1 rs2 = 0 rs3 = 0

[110] Valid Data = 1 rd = 1

[120] Sent ORN to dut

[120] PASSED

[120] inst= 40006033 r rs1 = 0 rs2 = 1 rs3 = 0

[120] Valid Data = 1 rd = fffffffe

[130] Sent XORN to dut

[130] PASSED

[130] inst= 40004033 r rs1 = 0 rs2 = 0 rs3 = 0

[130] Valid Data = 1 rd = ffffffff

[140] Sent SLO to dut

[140] PASSED

[140] inst= 20001033 r rs1 = deadbeef rs2 = 8 rs3 = 0

[140] Valid Data = 1 rd = adbeefff

FIG 4. I/O PROMPT IN BITMANIP IP

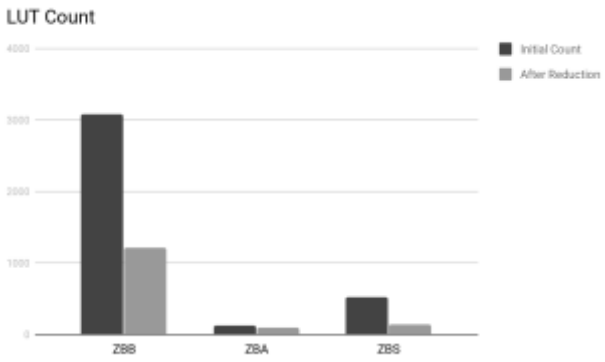

Chart 1 Lut Analysis

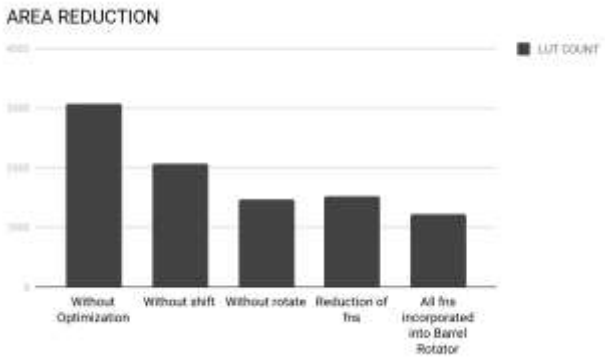

Chart 2 Area Analysis for ZBB

From Moore's law it is understood that the performance of an IC doubles every 18 months. This will increase the number of transistors used and hence increases the area and power consumption of the circuit.

Recently power dissipation is becoming an important constraint in design process. In that Low power design is becoming a new era in VLSI technology, as it impacts many applications.

This reduction was achieved by reducing all common functions like immediate variants and *W variants into one function calls, in decoder block. Each function call corresponds to the execution of one of the 106 instructions. The optimization facilitated the reduction in the number of inputs to the decoder multiplexer and the static elaboration in hardware.

ACKNOWLEDGEMENT

I would like to extend my gratitude to Shakti Labs, Indian Institute of Technology, Madras, which helped me materialize the project into a solid architectural block and helped me understand the intricacies of system design.

REFERENCES

[1] Clifford Wolf,Symbiotic GmbH,**"RISC-V Bitmanip Extension Draft 0.93"**,November 30, 2019

[2] Bastian Koppelmann, Peer Adelt, Wolfgang Mueller, Christoph Scheytt,Paderborn University/Heinz Nixdorf Institute,Paderborn, Germany,**"RISC-V Extensions for Bit Manipulation Instructions",** 2019 29th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)

[3]Michael J. Schulte and E. George Walters III,Computer Architecture and Arithmetic Laboratory,Computer Science and Engineering Department Lehigh University,Bethlehem, PA 18015, USA,**"Design alternatives for barrel shifters"**,Advanced Signal Processing Algorithms, Architectures, and Implementations XII, Franklin T. Luk, Editor,Proceedings of SPIE Vol. 4791 (2002)

[4]Neel Gala, G. S. Madhusudan, Paul George, Anmol Sahoo, Arjun Menon, V. Kamakoti (2018, Sept). **"SHAKTI: An Open-Source Processor Ecosystem"**. In Advanced Computing and Communications.