# IOb-SoC-SHA

## Acceleration Proposal


April 26, 2022

# Contents



# List of Tables



# List of Figures

# 1 Introduction

This document outlines an acceleration plan for the SHA256 application execution in the IOb-SoC-SHA system.

The document is divided in two parts:

- the first part presents profiling data for the SHA256 application running exclusively on the riscv CPU.
- the second part elaborates on the profiling conclusions and establishes functional unit architectures to accelerate the application.

# 2 Profiling Assessment

The execution time for the SHA256 application is presented in Table 1. The program runs exclusively on software with instructions stored in internal memory and data stored in external memory (DDR). The system uses the VexRiscv CPU at a clock frequency of 100 MHz.

| Function | Time ($\mu$s) | Time (%) |
|---|---|---|
| Global | 28381 | 100 |
| sha256 | 25360 | 89 |
| sha_init | 886 | 3 |
| sha_finalize | 22816 | 80 |
| crypto_hashblocks | 19435 | 68 |
| ld_big_endian | 2262 | 7 |
| st_big_endian | 1282 | 4 |
| F_32 | 4686 | 16 |
| Expand32 | 2755 | 9 |
| sha_ctxrelease | 571 | 2 |
| mem | 1351 | 4 |

Table 1: Baseline application profile data.

The profile analysis tracks the time in clock cycles since the input data is in the external memory, until the output data in stored in the external memory.

The results from Table 1 demonstrate that about 80% of the execution time is used to run the `sha_finalize()` function, in particular, the `crypto_hashblocks()` function. The functions and macro calls inside the `crypto_hashblocks()` function have the same order of magnitude with regards to duration.

The acceleration efforts should be focussed on the `crypto_hashblocks()` function and respective subfunction and macro calls.

# 3 Acceleration Proposal

Figure 1 presents the flowchart of the `crypto_hashblocks()` function.

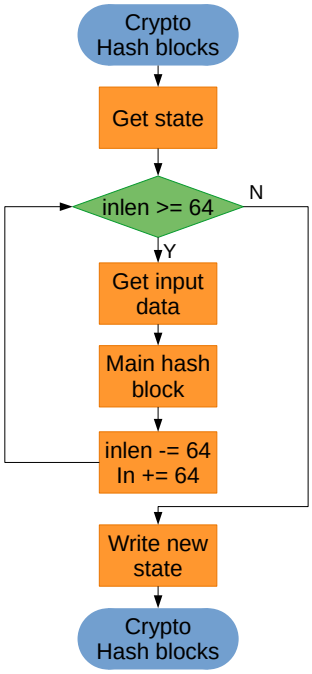


Figure 1: `crypto_hashblocks()` function flowchart.