INFORMATICS AND COMPUTER SCIENCE,
COMPUTER SYSTEMS AND TECHNOLOGIES

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

---

# HDU_ITMO Joint Institute

# Computer Systems Design

# Coursework

## Comparative analysis of computer system implementations

**陈浩东 Chen Haodong**

2019110043

2023.10.28

# Table of Contents

# 1 INTRODUCTION

In this work, in-depth evaluation of different computer systems implementations is given.

The following variant has been implemented:

***Description***: generate ln(1+x) function value using first four terms of Taylor series.

***Input data***: argument value(16-bit, fixed point: 0x00.00)

***Output data***: sine value (16-bit, fixed point: 0x00.00)

The following implementations have been analysed:

- Combinational circuit
- Pipelined circuit
- Software implementations (based on 6x RISC-V processors with increasing pipelining)
- Circuit based on high-Level Synthesis

# 2 COMMENTS

## 2.1 Top frequency:

Table 1: Top frequency of each implementation

| Implementation | Top frequency, MHz |
|---|---|
| Circuit based on HLS | 200 |
| Pipelined circuit | 160 |
| Software-based(6-stage RISC-V) | 150 |
| Software-based(5-stage RISC-V) | 140 |
| Software-based(4-stage RISC-V) | 140 |
| Software-based(3-stage RISC-V) | 80 |
| Software-based(2-stage RISC-V) | 70 |
| Software-based(1-stage RISC-V) | 70 |
| Combinational circuit | 7.81 |

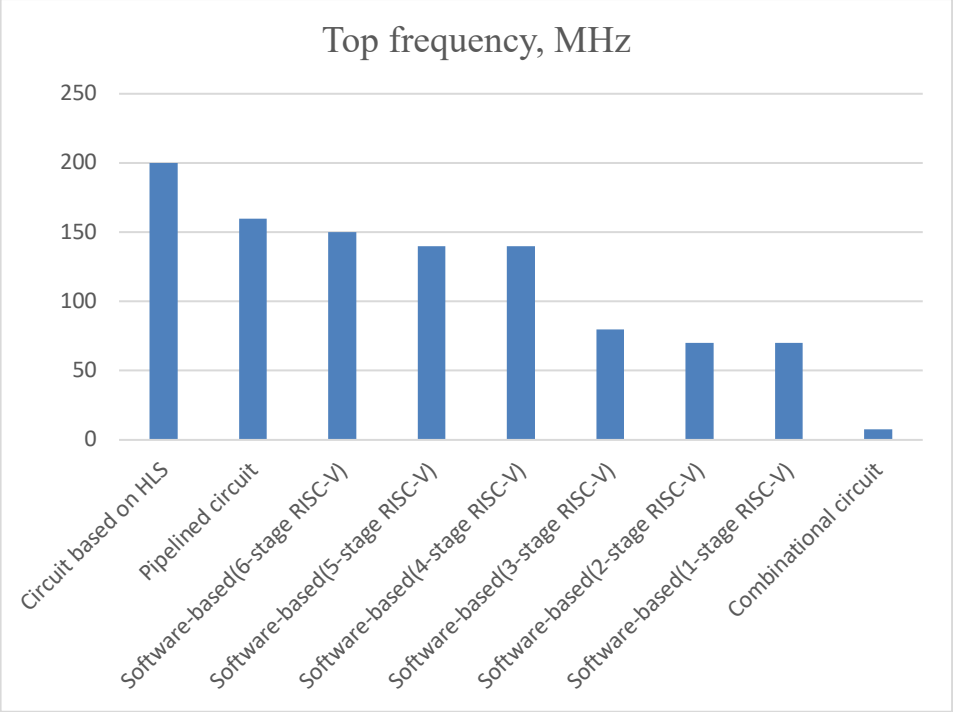


Figure 1: Top frequency of each implementation

Table 1 provides a detailed list of the top frequencies for various implementation methods. It is evident from the table that the circuit based on HLS demonstrates the best frequency performance, reaching up to 200MHz, while the combinational circuit has the lowest frequency at 7.81MHz since all computations should be completed in one clock cycle. Moreover, the frequency of the software-based RISC-V implementations decreases with the increase in stages. This suggests that if one aims to achieve a higher operating frequency, a circuit based on HLS might be a preferable choice.

**2.2 Performance (maximum throughput):**

Table 2: Maximum throughput, Mop/s

| **Implementation** | **Maximum throughput, Mop/s** |
|---|---|
| Circuit based on HLS | 200 |
| Pipelined circuit | 160 |
| Combinational circuit | 7.81 |
| Software-based(6-stage RISC-V) | 6.52 |
| Software-based(5-stage RISC-V) | 6.36 |
| Software-based(4-stage RISC-V) | 5.83 |
| Software-based(2-stage RISC-V) | 2.91 |

| | |
|---|---|
| Software-based(3-stage RISC-V) | 2.42 |
| Software-based(1-stage RISC-V) | 1.67 |

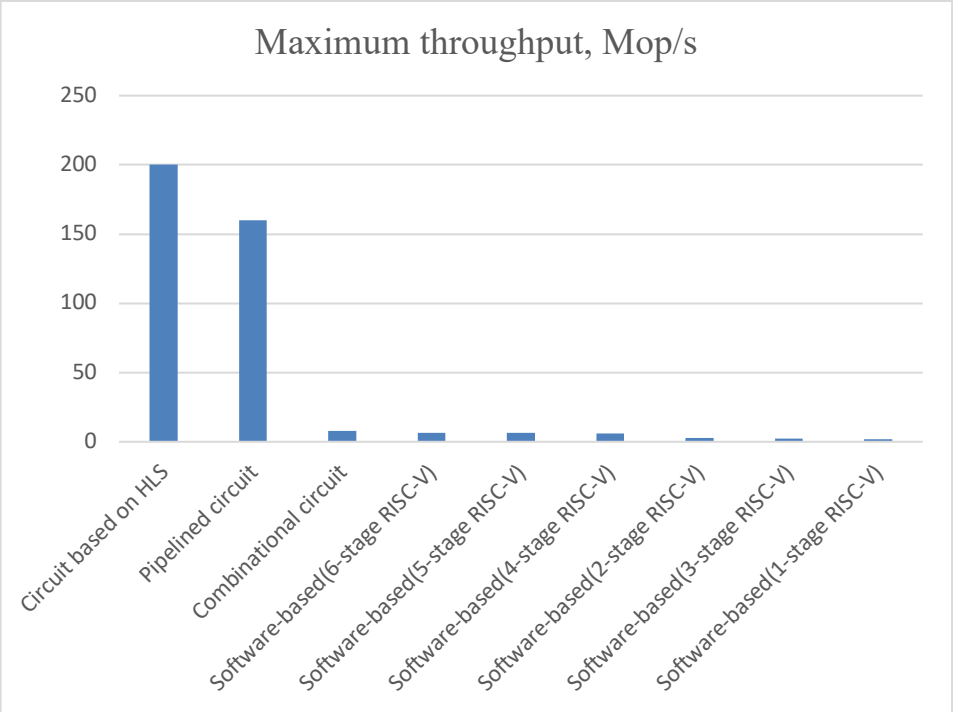


Figure 2: Maximum throughput of each implementation

Table 2 illustrates the performance of different implementations in terms of maximum throughput. The circuit based on HLS boasts the highest throughput, reaching 200 Mop/s, this is because pipeline is implemented in design, while the pipelined circuit has a throughput of 160 Mop/s, since pipeline is also implemented but manually. The combinational circuit has a relatively lower throughput of 7.81 Mop/s. In the software-based RISC-V implementations, a clear trend can be observed: as the number of stages decreases, the throughput also diminishes. Specifically, the 6-stage RISC-V implementation has a throughput of 6.52 Mop/s, whereas the 1-stage RISC-V implementation has only 1.67 Mop/s.

**2.3 Performance (minimum latency):**

Table 4: minimum latency of each implementation

| Implementation | Minimum latency, ns |
|---|---|
| Circuit based on HLS | 25 |
| Pipelined circuit | 50 |
| Combinational circuit | 128 |

| | |
|---|---|
| Software-based(6-stage RISC-V) | 153 |
| Software-based(5-stage RISC-V) | 157 |
| Software-based(4-stage RISC-V) | 171 |
| Software-based(3-stage RISC-V) | 413 |
| Software-based(2-stage RISC-V) | 343 |
| Software-based(1-stage RISC-V) | 600 |

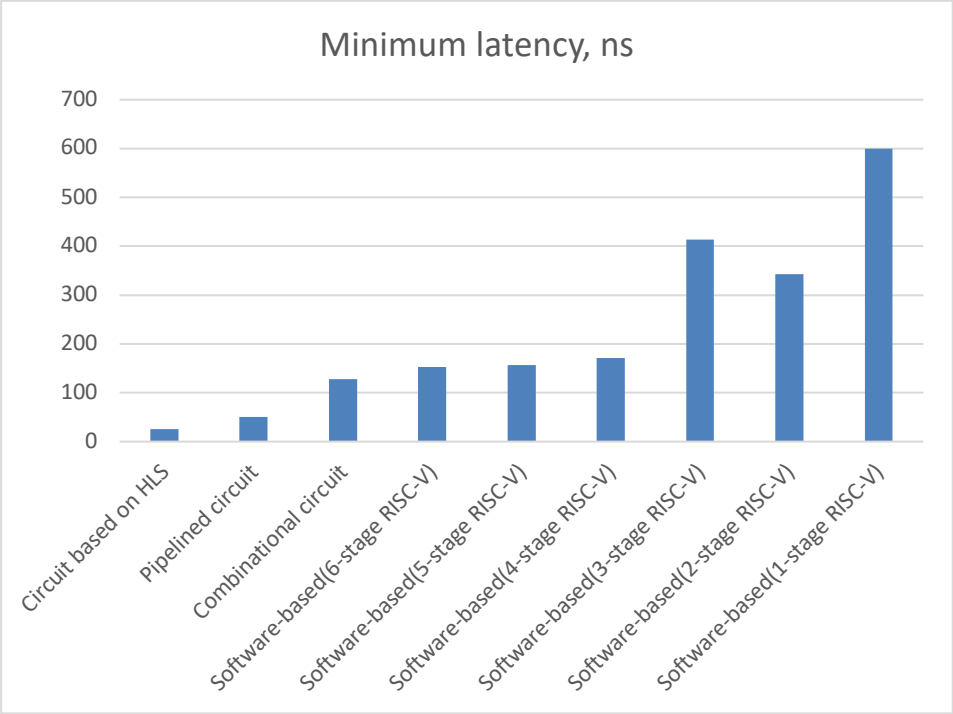


Figure 4: minimum latency of each implementation

Table 4 displays the performance of various implementations in terms of minimum latency. The circuit based on HLS boasts the lowest latency, clocking in at just 25 nanoseconds, followed by the pipelined circuit with a latency of 50 nanoseconds. The combinational circuit has a relatively higher latency of 128 nanoseconds, since combinational logic circuits produce outputs solely based on their inputs, without involving any storage elements (such as flip-flops or registers). All operations occur almost simultaneously, and the entire logic from input to output must complete within a single clock cycle. This means that if certain logic paths are long or contain many logic gates, these paths might become bottlenecks for latency. In the software-based RISC-V implementations, we can observe a trend: as the number of stages decreases, the latency increases. Specifically, the 6-stage RISC-V implementation has a latency of 153 nanoseconds, while the 1-stage RISC-V implementation comes in

at 600 nanoseconds. The reason why software-based designs have higher latency is that they produce more instruction to be executed.

**2.4 Hardware resources:**

Table 5: Hardware resources of each implementation

| Implementation | HW resources (LUT) | HW resources (FFs) |
|---|---|---|
| Circuit based on HLS | 203 | 253 |
| Combinational circuit | 888 | 0 |
| Pipelined circuit | 832 | 140 |
| Software-based(1-stage RISC-V) | 2835 | 1665 |
| Software-based(2-stage RISC-V) | 3193 | 1766 |
| Software-based(3-stage RISC-V) | 3200 | 1920 |
| Software-based(4-stage RISC-V) | 3565 | 2198 |
| Software-based(5-stage RISC-V) | 3739 | 2273 |
| Software-based(6-stage RISC-V) | 3937 | 2406 |

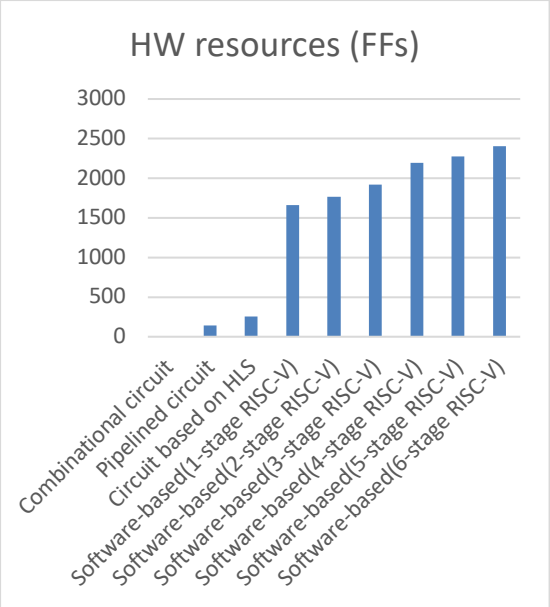


Figure 5: HW resources (FFs)

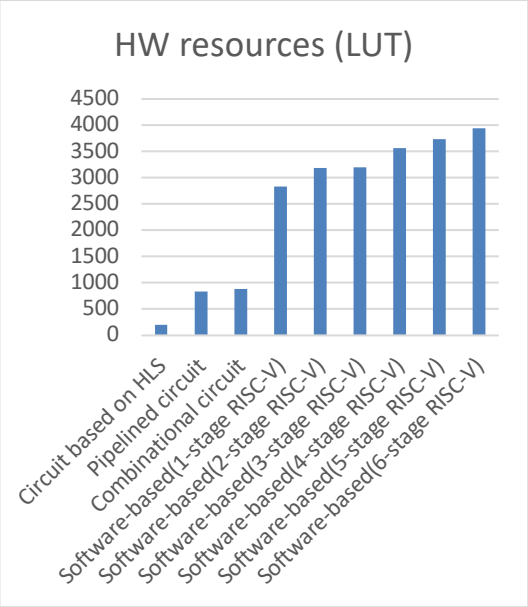


Figure 6: HW resources (LUT)

The HLS-based circuit requires the least hardware resources, needing only 203 Lookup Tables (LUTs) and 253 Flip-Flops (FFs). This is because Vivado HLS automatically build the circuit into a pipeline design with reusable registers and wires. The combinational logic circuit has a higher

requirement in terms of LUTs, reaching 888, but it does not require any flip-flops. The pipelined circuit requires 832 LUTs and 140 FFs. The pipelined circuit requires 832 LUTs and 140 FFs which dues to programmer’s bad design. For the software-based RISC-V implementations, as the number of stages increases, the required LUTs and FFs also gradually increase. For instance, the 1-stage RISC-V requires 2835 LUTs and 1665 FFs, while the 6-stage RISC-V requires 3937 LUTs and 2406 FFs.

## 3 CONCLUSION

When engaging in hardware design, it's essential to compare the pros and cons of different design methods. If we aim for the highest data throughput, a pipelined design approach should be adopted. On the other hand, if the goal is to achieve the lowest latency, then a combinational logic circuit should be designed. When designing such circuits, it's crucial to ensure that data transmission delay within the circuit isn't too long, to avoid increasing the overall circuit latency. A more recommended approach is to design circuits through Vivado HLS. However, once the circuit's HDL is obtained, it requires manual inspection to check for potential optimization areas within the HDL. Software-based designs offer better scalability but consume more resources, so it's advisable to avoid using them when possible.