5th national RISC-V student contest 2024-2025

Sponsored by Thales, the GDR SOC² and the CNFM

Guidelines to report results

This document is a complement to the **`Annonce RISC-V contest 2024-2025.pdf`** file (and its subsequent versions) to prepare the submission of results.

Prerequisites

The following steps need to be successful before analyzing the performance and starting your optimizations:

- The kit available at https://github.com/thalesgroup/cva6-softcore-contest is running in your Linux environment.
- You can launch all targets of the Makefile as described in the **`README.md`** file.
- The simulation of the CoreMark application runs to its end and returns a performance result.
- The CoreMark application executes on the ZYBO Z7-20[1] board.
- The simulation of the MNIST[2] application runs to its end and returns a successful result: the '4' digit is recognized and the credence (probability) is 82.
- The MNIST application executes on the ZYBO Z7-20 board and reports the same result.

In addition, before working on your optimizations, you need to ensure that:

- You are using the same OS, tools and versions[3] as in the **`README.md`** file.
- You get the same metrics with the reference design as the organization team (see below).

And finally, before reporting results for the contest, you need to ensure that:

- You fulfill all constraints described in the **`Annonce RISC-V contest 2024-2025.pdf`** file (or its subsequent versions).

[1] No other FPGA boards will be accepted as the jury will replay results on its board.
[2] The MNIST application is run for non-regression purposes. Its performance is not analyzed in this year's contest.
[3] This is a recommendation so that the organizers can provide support. However, you can use other tools, e.g. Verilator, but you won't receive support on these tools.

Analyzing the results

In this year's competition, the goal is to increase the frequency of a CV32A6-based processor as reported by Vivado static timing analysis. There is a tolerance to decrease the CoreMark/MHz performance up to 10%.

To run the design at a higher frequency, it is necessary to change the parameter `FPGA_UART_0_FREQUENCY` in file `sw/bsp/config/fpga_platform_config.h` and recompile to get a correct console display. However, the CoreMark/MHz performance shall be measured without recompiling (and staying at the baseline frequency), as the recompilation can slightly modify the CoreMark/MHz figure.

Maximal frequency

You can determine the maximal clock frequency of your design with the static timing analysis report after the place & route:

```
Timing Report
Slack (MET) :  0.287ns  (required time - arrival time)
Source: i_ariane/i_cva6/csr_regfile_i/pmpaddr_q_reg[3][1]/C
(rising edge-triggered cell FDCE clocked by clk_out1_xlnx_clk_gen  {rise@0.000ns fall@12.500ns period=25.000ns})
Destination: i_ariane/i_cva6/gen_cache_wt.i_cache_subsystem/i_wt_dcache/i_wt_dcache_mem/gen_data_banks[3].i_data_sram/gen_cut[
0].i_tc_sram_wrapper/i_ram/Mem_DP_reg/ADDRARDADDR[9]
(rising edge-triggered cell RAMB36E1 clocked by clk_out1_xlnx_clk_gen  {rise@0.000ns fall@12.500ns period=25.000ns})
```

The maximal clock period is 1/(clock_period-slack), i.e. $1/(25.000*10^{-9}-0.287*10^{-9}) = 40.46$ MHz in this illustration. Your actual critical might be different from this one.

The synthesizer might cease optimizing the frequency when the slack is met. To overcome this, you can reduce the period until you get a "violated" slack[4]. The formula above still applies with a negative slack.

The simulation testbench and the FPGA top design are a bit different, mostly with respect to main memory access time. Therefore you can experience slightly different number of cycles between both environments. The jury will consider the **number of cycles in simulation** board for the CoreMark performance.

CoreMark performance

The CoreMark[5] shall run successfully and return a result in simulation and on the board. The figures are reported through on the console through the UART.

If the figures differ between the board and the simulation, the simulation result will prevail.

MNIST execution

The MNIST shall run successfully and return the predicted digit 4, with the credence 82, both in simulation and on the FPGA board. For this contest, we do not care about the MNIST performance.

```
Expected  = 4
Predicted = 4
Result : 1/1
credence: 82
image env0003: 1760387 instructions
image env0003: 2790524 cycles
```

[4] Be careful to ensure the violation is on a critical path, typically on a setup timing. Other violations, like hold time violations, need to be fixed.

[5] Note that the goal of the CVA6 configuration used in the contest is not optimize the CoreMark score. Other configurations can get higher CoreMark, up to 3.1 CoreMark/MHz in single issue and more than 4 with dual issue on more recent CVA6 versions.

FPGA resources

If there are ties between teams, i.e. teams with a close frequency (<1%), the results will be separated according to the FPGA resources (number of LUTs + number of flip-flops) used by the solution.

You will report the number of LUTs and flip-flops of your top-level design (not only the CVA6 core). Other resources (BRAM, DSP…) will also be reported for information.

Reference

The reference project is the one you can find in https://github.com/thalesgroup/cva6-softcore-contest.

Before getting further in the contest, you have to check that it provides the same results on your Linux machine than on the organizers' machine:

- CoreMark score: 2,604 CM/MHz
- Number of MNIST cycles as reported at the end of the simulation: 2790524 cycles
- Actual clock period of the design: 25 ns
- Minimum clock period of the design as reported after place & route: 24,713ns
- FPGA resources of your top-level design as reported in corev_apu/fpga/report_cva6_fpga_impl/cva6_fpga.utilization.rpt:
    - 13093 LUTs
    - 5461 FFs
    - 16 RAM36

If the results with the reference project are not the same as above, you first need to check that you are using the same versions of the tools[6]. If you still have a difference of more than 1% after this, get in touch with the organizers.

For MNIST, you shall also get the '4' digit recognized with a credence of 82.

Reporting the results

In both the 6-page report and in your recorded video, you'll clearly report these results:

- Maximal frequency of your design before and after your optimizations (and the corresponding change in %)
- CoreMark/MHz score before and after your optimizations (and the corresponding change in %) in simulation and on the FPGA board.
- Recognized digit(s) and their credence(s) by MNIST, before and after your optimizations
- Resources used in your design (LUTs, flip-flops, BRAM, DSP...) of your top-level design, before and after your optimizations

The report shall be written as a scientific paper, presenting your approach, your solution and commenting your results. It should help the jury to assess your solution.

The jury will double check in their environment the results of the teams who claim the best results in their reports. This also ensures that results are compared based on the same tool versions. So you'll publish a modified GitHub repository, as a fork of https://github.com/thalesgroup/cva6-softcore-contest. You'll create a **`report`** folder where you will upload:

- Your 6-page report written as a scientific paper
- Simulation log (uart)
- P&R report with the maximum frequency (corev_apu/fpga/report_cva6_fpga_impl/cva6_fpga.timing.rpt)
- Report of resources (corev_apu/fpga/report_cva6_fpga_impl/cva6_fpga.utilization.rpt)

[6] The organizers know this rule cannot be strictly enforced because of IT constraints.

- Log of the execution on the FPGA board (copy or screenshot of the hyperterminal output) for CoreMark and MNIST.

Remember that your design shall run on the FPGA board and fulfill all constraints listed in **`Annonce RISC-V contest 2024-2025.pdf`**.

During your recorded video, consider showing the execution of your solution on the Zybo board or executions on your machine. Because of the file size, do not upload the video in GitHub; you'll later get instructions for the file transfer.

A few remarks

We suggest that you keep your GitHub repository private until the end of the contest (May 5th, 2025) and make it public just after. We'll look at the file timestamps to ensure you completed the contest on time.

Your presentation video shall last at most 10 minutes. All team members are encouraged to participate. The deadline to submit your video, about one week after, will be announced by the organizers. There is no special prize for the video, but it should be good enough to be presented to a large audience if your time is a winner. Make sure that you do not include others' copyrighted material.

A final word

If you find the contest difficult, remember that the other teams face the same level of difficulty. Even if you think your results are limited, submit them!

Version

These guidelines can undergo some evolutions based upon the teams' feedback. Please check their final version before submitting your results.

Date	Comment
2024-11-08	Initial version
2025-01-08	Update of performance figures