# 2nd national RISC-V student contest 2021-2022

*Sponsored by Thales, the GDR SOC² and the CNFM*

# Reporting energy results

## Prerequisites

The following steps need to be successful before analyzing the energy 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 MNIST application runs to its end and returns a successful result (the '4' digit is recognized).
- You get power analysis results.

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

- You are using the same tool versions (Vivado, Questa) 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 2021-2022 v1.pdf`** file
- The application runs on your Zybo Z7-20 board and you get a correct result.

## Analyzing the energy

In this year's competition, the goal is to improve the power efficiency of a CV32A6-based processor running the MNIST handwriting recognition application.

This translates into decreasing the total energy (in Joule) needed to recognize the '4' digit with the MNIST application using post-implementation simulations.

As the tools do not directly report the energy, a simple math will be used:

*total_energy*

= *total_power* × *duration_of_the_process*

= *total_power* × *number_of_cycles* × *clock_period*

The *total_power* is obtained from the Vivado power analysis:

| | |
|---|---|
| Total On-Chip Power (W) | 0.306 |
| Design Power Budget (W) | Unspecified* |
| Power Budget Margin (W) | NA |
| Dynamic (W) | 0.192 |
| Device Static (W) | 0.114 |
| Effective TJA (C/W) | 11.5 |
| Max Ambient (C) | 81.5 |
| Junction Temperature (C) | 28.5 |
| Confidence Level | Medium |
| Setting File | --- |
| Simulation Activity File | work-sim/routed.saif |
| Design Nets Matched | 89% (59430/66741) |

The *number_of_cycles* is reported at the end of the simulation:

```
Expected  = 4
Predicted = 4
Result : 1/1
image env0003: 1732800 instructions
image env0003: 2149795 cycles
```

The *clock_period* of your solution is reported in the Vivado power report:

| Clock | Domain | Constraint (ns) |
|---|---|---|
| clk_out1_xlnx_clk_gen | i_xlnx_clk_gen/inst/clk_out1_xlnx_clk_gen | 22.2 |
| clk_sys | clk_sys | 8.0 |
| clkfbout_xlnx_clk_gen | i_xlnx_clk_gen/inst/clkfbout_xlnx_clk_gen | 8.0 |
| tck | tck | 100.0 |

It shall be higher or equal to the maximum frequency of your design as reported after the place & route[1]:

```
Timing Report
Slack (MET) : 0.000ns (required time - arrival time)
Source: issue_stage_i/i_scoreboard/mem_q_reg[5][sbe][valid]/C
(rising edge-triggered cell FDCE clocked by clk_i {rise@0.000ns fall@9.598ns period=19.196ns})
Destination: i_frontend/i_instr_queue/i_fifo_address/read_pointer_q_reg[1]/D
(rising edge-triggered cell FDCE clocked by clk_i {rise@0.000ns fall@9.598ns period=19.196ns})
```

The total power includes the static and the dynamic powers. As part of the contest, you need to improve the total energy, which includes a dynamic and a static part.

The energy estimation will include the top level design implemented in FPGA (module fpga/src/cva6_zybo_z7_20.sv).

However, you can only modify src/ariane.sv module and its submodules as part of the contest. For the sake of clarity, you shall not modify other modules such as peripherals (UART…) or the main memory. Keep in mind that modifications in the core might have an impact on the whole design power, e.g. by changing the activity of the memory bus.

[1] In this example, the organizers have used a smaller period then what you can find in the reference design.

# 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:

- Total power as reported by Vivado: 306 mW (*total_power*)
- Number of cycles as reported at the end of the simulation: 2 149 795 cycles (*number_of_cycles*)
- Actual clock period of the design: 22.2 ns (*clock_period*)
- Reference processing time: 47.72 ms (*number_of_cycles* × *clock_period*)
- Minimum clock period of the design as reported after place & route: 19.196 ns
- FPGA resources as reported in fpga/report_cva6_sim_impl/cva6.utilization.rpt:
    - 14675 LUTs
    - 9291 FFs
    - 36 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. If you still have a minor difference after this, get in touch with the organizers.

# Reporting the results

In both the 6-page report and in your recorded video, you'll clearly report the results in a similar way as:

- Energy of the reference design: *x* Joule for 1 frame
- Energy of your solution: *y* Joule for 1 frame
- Energy gain: - *z* %

In the report, you'll also have to report:

- FPGA resources. You shall not increase each type of resources by more than 10% (LUT, registers, BRAM).
- Number of cycles, period and duration of the process. The application performance shall not decrease w.r.t. the reference implementation (*number_of_cycles* × *clock_period* = 47.72 ms).

The jury reserves the right to verify your results, 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
- Vivado power report (fpga/work-sim/power_routed_mnist.txt)
- Simulation log (fpga/sva_sim.sim/sim_1/impl/func/questa/uart)
- P&R report with the maximum frequency (fpga/report_cva6_sim_impl/cva6.timing.rpt)
- Log of the execution on the FPGA board (copy or screenshot of the hyperterminal output)

Remember that your design shall run on the FPGA board and fulfill all constraints listed in **`Annonce RISC-V contest 2021-2022 v1.pdf`**

During your recorded video, consider showing the execution of your solution on the Zybo board or executions on your machine.

# A few remarks

To speed up the simulation, several optimizations are used: the application runs as baremetal (no OS used); the image to recognize is already stored in the memory at start up; a medium accuracy power estimation is used (post-implementation without timing back-annotation).

The design runs at a frequency about 14% lower than its maximum frequency to leave room for various optimizations. If you increase the frequency up to the maximum frequency, you'll surely improve the static energy in your results[2].

We suggest that you keep your GitHub repository private until the end of the contest (April 22nd, 2022) 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. You can transmit your video to the organizers until April 29th, 2022.

## Version

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

| Version | Date | Comment |
|---|---|---|
| V1 | 2011-11-19 | Initial version |
| | | |
| | | |

[2] This is an acceptable optimization for the contest. Now, all the teams are aware of it.