Performance characterisation of 8-bit RISC and OISC architectures

Presentation answer

Question: *The power experiment (Fig 4) shows most of the power is auxiliary power. What would you expect if you tried running a more complex problem - would the power consumption increase to dominate over auxiliary-only power, and if so, would you expect a greater difference between RISC and OISC?*

Thanks for a great question!

Firstly, to clarify, auxiliary power includes whole FPGA board, power conversion, and synthesised logic on FPGA required to support a processor (such as PLL, UART, Input/Output control, RAM). For this reason SystemVerilog code is designed to be modular, so a test was performed by just reusing the same support code for a fair comparison. RISC and OISC bars in the graph indicate auxiliary power plus processor power, which means that the processor itself takes relatively small amount comparing to auxiliary power, about 0.5%.

Furthermore, power consumption would increase almost linearly with the clock frequency, meaning that if during this test the clock was set to 1MHz and it would be increased to 100MHz, processor power would increase from about 1.76mW to 176mW which corresponds to 49% of auxiliary power. 100MHz is an extreme case, the synthesis tool has predicted the maximum frequency for OISC to be about 65MHz and 23MHz for RISC, however this still need to be tested on an actual FPGA. Unfortunately labs are closed, but it would be interesting to measure power consumption at different clock frequencies in order to calculate activity factor, which would be an interesting value to compare between OISC and RISC.

Secondly, both processors are not optimised for power, meaning they are always performing almost all tasks. For example, the ALU (arithmetic logic unit) always performs every function on its inputs during every cycle, such as addition, subtraction etc., but only a single output is selected depending on the instruction. Therefore, assuming both processors perform the same complex problem, the difference between power consumption would remain almost identical.

In terms of design and future work, OISC would be much easier and straightforward to optimise for power consumption because of block design — it would only need to lookup a future instruction and enable the required block just before it is used. However, this does not seem as important of a problem due to auxiliary power dominance and an complexity in finding a fair comparison for both architectures.