FlexSoC - Past, Present & Future

Magnus Själander, Martin Thuresson, Per Larsson-Edefors, and Per Stenström

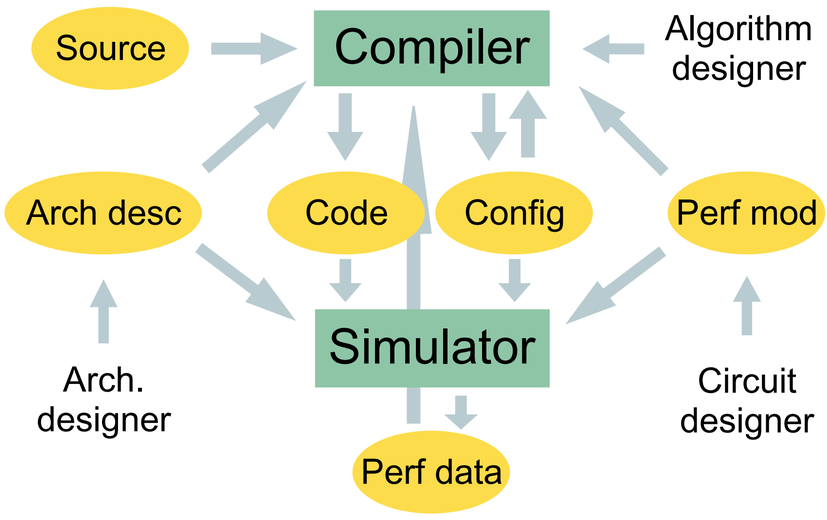
Computer Science and Engineering

Chalmers University of Technology

Background

The FlexSoC program, launched in 2003, aims to develop new architectural techniques and programming models for high performance System on Chip. The target platforms are embedded systems where long battery life, high performance, and the flexibility to adapt to new protocols and standards are important. ASIC design makes it possible to tailor the hardware for a specific application thus achieving an efficient solution with the lowest power dissipation but the total lack of flexibility makes it costly to adapt to new standards. GPPs on the other hand offers flexibility at the cost of high power dissipation and lower performance.

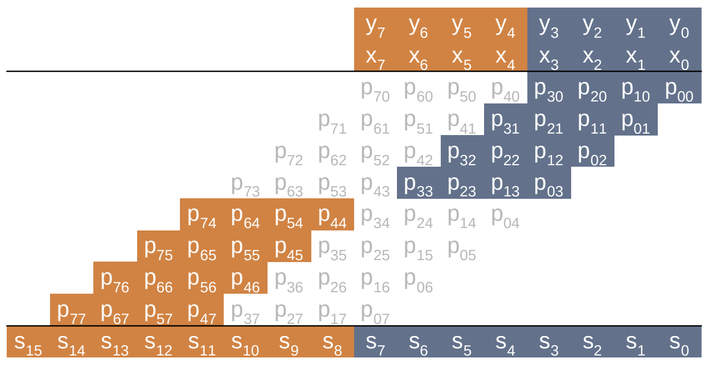
FlexSoC exposes a more fine-grained control and a richer native ISA than a GPP. The proposed idea is to generate an application specific ISA as a subset of the native ISA. That way we hope to gain in performance and power efficiency and also to store programs in a more compact way.

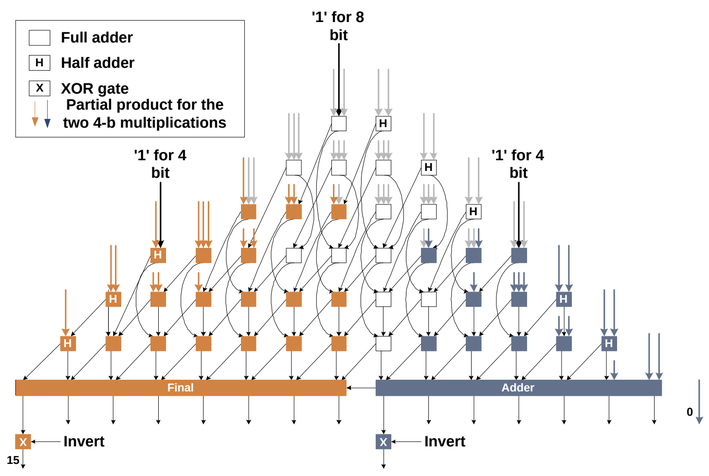


Simulator and evaluation framework

Twin-Precision Multiplier

When performing an *N*/2-bit multiplication only one quarter of the partial products of an *N*-bit multiplication is used (blue).
This opens up the possibility of performing a second *N*/2-bit multiplication in parallel.

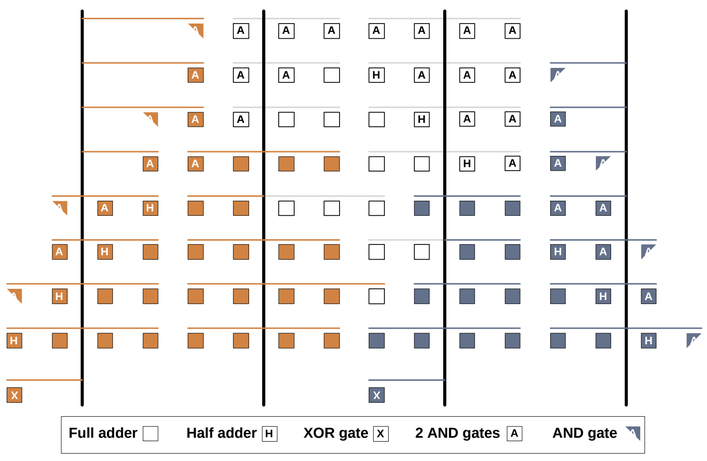


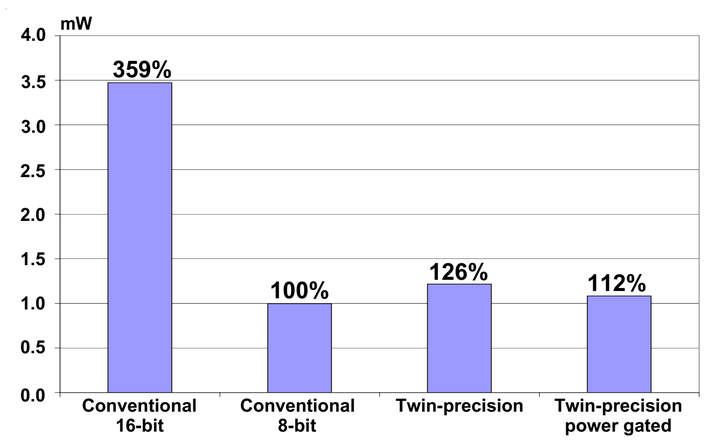


Forcing inactive partial products to zero (gray) enables the possibility of computing a second *N*/2-bit multiplication in parallel (red). To improve the delay of the *N*/2-bit multiplicaions the active partial products (red&blue) are placed in the reduction tree close to the final adder.

To further reduce the power dissipation, power gating of incative adder cells can be applied. This eliminates the dynamic power as well as reduces the static power dissipation.

A flexible power grid for turning off inactive logic has been presented.





Simulations done with a 16-bit twin-precision multiplier shows a power reduction of more then three times, when performing signed 8-bit multiplications.

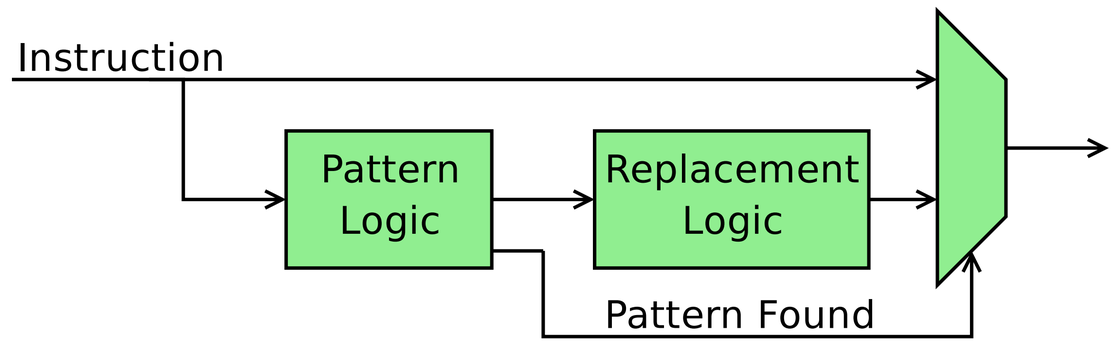
- ✓ 0.13µm technology
- ✓ V_{DD} = 1.2 V
- ✓ Temperature = 80° C

Static Code Compression

Memory is an important component in embedded devices and by reducing the static code size of the applications we can: dissipate less energy, reduce cost of system, and get higher performance.

Traditional dictionary based compression identifies identical sequences of instructions in the code and stores them only once, in the dictionary. In the extended schemes, the dictionary entries may also be generated from similary sequences of instructions. Bitmask Echo and DICE are two previously proposed schemes that allows mismatch in complete instructions and operands.

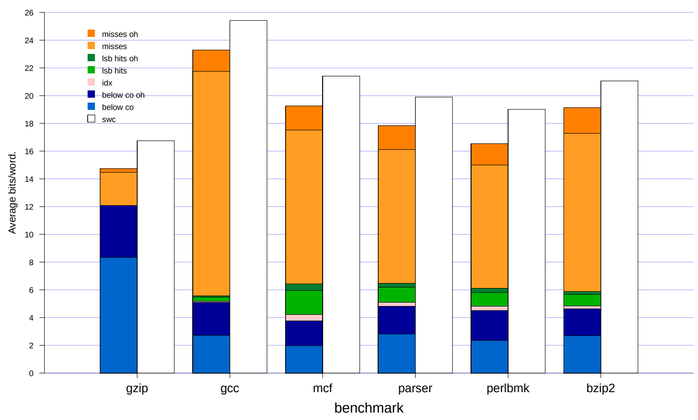
We have evaluated dictionary-based compression and proposed a flexible scheme that allows for a more efficiant execution of a compressed program. Our evaluation shows that the operand flexibility is more efficiant than complete instructions, if used alone. We also show that the combination of the two allows for a more efficient format to be executed.



Decompression engine inside the instruction fetch pipeline

Memory Compression

Continuing the work on efficient storage and execution, we are looking into ways to efficiently transfer not only instructions, but also data. Our current studies look at the data transfered between CPU and memory, but a long term goal is to use the same efficient representation at other levels in the memory hierarchy as well. Our initial results shows that significance based compression is a well suited tool for this.



Average number of bits/transfered word on a 64-bit sparch Architecture using on significance width compression and a 16-entry cache with frequent values.

Acknowledgement

This research has been sponsored by the Swedish Foundation for Strategic Research (SSF) under the FlexSoC program. The FlexSoC project consists of the following members: Per Stenström, Per Larsson-Edefors, Kjell Jeppson, Mary Sheeran, Lars Svensson, John Hughes, Magnus Själander, and Martin Thuresson.

http://www.flexsoc.org