MB2020 DD4.2– SVE-enabled gem5 with abstract power equations
Version 0.3

Document Information

Contract Number	779877
Project Website	www.montblanc-project.eu
Contractual Deadline	M14
Dissemination Level	Public
Nature	Source Code
Authors	Stephan Diestelhorst (Arm)
Contributors	Stephan Diestelhorst (Arm), Giacomo Gabrielli (Arm)
Reviewers	Adria Armejach Sanosa (BSC), Juha Jaykka (Arm)
Keywords	Scalable Vector Extensions, gem5, power modelling

Notices: *This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N^o 779877.*

Change Log

Version	Description of Change
v0.1	Initial version of the deliverable
v0.2	Ready for internal review
v0.3	Incorporated internal review

Contents

Executive Summary 4

1 Introduction 5

1.1 Power Modelling 5

2 Implementation in gem5 6

3 Conclusion 6

Executive Summary

In addition to performance simulation of SVE applications, we extend the gem5 simulator with support for on-line power modelling. Power is modelled through the notion of *power states* and associated power *equations*.

Generally, power consumption of integrated circuits can be split into static and dynamic power. Static (or leakage) power is the power draw caused by the finite resistance of the chip overall, i.e., by current that flows through the substrate. As such, it is dependant on area (higher area means more static power), temperature (hotter chips cause more static power), substrate characteristics (bulk silicon processes consume more static power than silicon-on-insulator processes), and of course voltage. Dynamic power, on the other hand, is caused by transistors switching, and charging / discharging, and the non-zero resistance of wires. Switching activity (toggles per unit time) and transistor / wire characteristics are the key determining factors for dynamic power consumption. Dynamic power is usually approximated as a sum of activities / events multiplied by specific activation factors.

In this deliverable, we add support for gem5 to obtain SVE-related activities from the core and feed them into power equations. Together with design (and process) specific constants (so called activity factors), the model allows on-line estimation of dynamic power, and (through the use of a thermal model) also static power. The thermal model of gem5 can then be used to model temperature based on the dynamic power and provide temperature input for further static power modelling. Both, static power modelling, and usage of the thermal model are, however, out of scope for this report.

In the literature, these design-specific constants have been obtained for existing designs [WDH$^+$17, WBD$^+$18], and thus allow fine-grained online power modelling. As there is no publicly available SVE-enabled silicon, yet, we use a set of *fictitious* constants to illustrate the model. Over time, we expect project partners to get a better understanding of hardware characteristics and improve the numbers for more detailed analysis.

This deliverable and report do not make available actual activity factors / design-specific constants of Arm IP. These are tightly bundled to Arm product IP and thus cannot be made available here. Instead, the activity factors in this report and the deliverable code are *for illustration, only*!

We make the updated gem5 available at `https://gem5.googlesource.com/arm/gem5/+/mb2020/d4.2`.

1 Introduction

In the Mont-Blanc project, we use gem5 [Bin11] to simulate future CPU cores with the Arm scalable vector extension (SVE) [SBB$^+$17]. Gem5 is a modular simulator at cycle level that can simulate different CPU core microarchitectures (abstract, in-order, out-of-order), and has a flexible, configurable memory hierarchy (cache sizes, memory controllers, etc.). We can use gem5 both as a full system simulator, i.e., executing application binaries and the operating system kernel inside the simulator; or we can chose to only execute application binaries in the simulator with the system-call emulation mode. In MB2020 deliverable D4.1, we have extended gem5 with support for executing SVE instructions, and made the result available at `https://gem5.googlesource.com/arm/gem5/+/mb2020/d4.1`.

1.1 Power Modelling

For this deliverable, we add to the *performance* modelling capabilities also the ability to accurately model the *power consumption* of SVE execution.

Estimating the power consumption of future CPU designs in addition to performance is crucial, as only the relationship between the two will allow us to understand efficiency, performance per watt, and similar metrics. There are various tools available that provide power estimates; McPAT [LAS$^+$09] is well used in the academic community, while several EDA companies provide their own tools, such as Synopsys Platform Architect and PrimeTime PX, Cadence Joules RTL Power Solution, and Mentor PowerPro. The latter tools generally require access to at least RTL, if not a full net list / floor plan of the future design. Therefore, they are unsuitable for power estimations at higher levels of abstraction, and also do not have knowledge of varying usage of the different functional units over time. With this deliverable, we aim at a different goal: we use gem5 simulation to collect activity of various parts of the CPU (the caches, the SVE ALU, etc.) and use that activity to predict power consumption of the overall CPU.

Generally, the total power consumption P_{tot} can be expressed as the sum of two different power draws: P_{stat} as static, or leakage, power, and P_{dyn}, the dynamic power that is caused by transistor and wire activity. Static power is caused by the finite resistance of the chip and substrate causing current to flow (leak) through the substrate independent of transistor switching. Therefore, static power is largely determined by the supply voltage, the substrate characteristics (for example bulk vs SOI), size of the chip, and temperature. Dynamic power, however, is caused by the non-zero resistance of wires and transistors and the switching activity that requires a current flowing through them. Consequently, it depends on wire and transistor characteristics, switching activity, and voltage.

Several of these influences are process and design specific; substrate, wire, and transistor characteristics, and the size of functional units. They can be estimated by either the aforementioned tools, or derived from (and then extrapolated) from existing designs, as we proposed in earlier work [SBB$^+$17] – obtaining them is outside the scope of this deliverable.

Assuming we have these constants (as α_i), we use gem5 to obtain the relevant event counts (as E_i) that are a proxy for the switching behaviour of groups of transistors. We ignore effects of data-dependent power consumption; for example having constant / zero values can reduce switching activities of multipliers. We then also model dynamic voltage and frequency scaling (DVFS), and obtain values for frequency f_{clk} and voltage V_{dd} accordingly. Then, we can express dynamic power as follows:

$$P_{dyn} = V_{dd}^2 * f_{clk} * (\beta + \sum_i \alpha_i * E_i) \tag{1}$$

2 Implementation in gem5

We implemented this activity tracking and power model evaluation functionality on top of gem5 using the existing statistics framework. We add functionality for the user to specify a power model similar to equation 1 in the gem5 configuration file using SVE activities, and activities of other relevant parts of the CPU; a simple abstract example of that can be seen in Figure 1. This power equation is then parsed by the simulator, and evaluated at run-time with the live specified statistics to update the dynamic (and static) power consumption estimate.

In Figure 2 we show comparisons of HPCG with two vector lengths showing different power power consumption and length of SVE-intensive phases. Assuming power consumption of the SVE unit being proportional to vector length, the power consumption of the 512 bit run is not 4x that of the 128 bit version, but closer to 2x. Similarly, the performance of the application (as can be estimated from the length of the high-power execution) roughly doubles with 4x the vector elements. Integrating dynamic power over time suggests that the 128 bit version needs 12.1 J to compute the solution, and the 512 bit version needs 16.1 J. Please note that, again, the actual coefficients used here are fictitious. In actual silicon, the wider vector units may actually reduce energy consumption if other components have higher background dynamic power, and when taking static power into account.

3 Conclusion

In this deliverable, we have extended gem5 with support for accurately tracking activity of the SVE unit and using the obtained activities in on-line power equations. Together with design specific activity factors, these activities can be used to estimate CPU and unit power consumption at very fine granularity.

We have made our extension to gem5 available at `https://gem5.googlesource.com/arm/gem5/+/mb2020/d4.2`, and are also in the process of integrating them into upstream gem5 at `https://gem5.googlesource.com/public/gem5`.

Unfortunately, we cannot make available actual activity factors / design-specific constants for Arm IP as part of this deliverable. Deliverable D4.3 (with limited circulation) will, however, add higher-level guidance on SVE vector unit power consumption. Further details can be obtained commercially as part of Arm IP licensing discussions.

```
# NOTE: These examples are for illustration only
self.dyn = "voltage^2 * (2 * ipc + " \
        "3 * 0.000000001 * dcache.overall_misses / sim_seconds + "\
        "    0.000000001 * %i * "\
        "(40 * (iq.FU_type_0::SimdAdd +"\
        "iq.FU_type_0::SimdAddAcc +"\
        "iq.FU_type_0::SimdAlu +"\
        "iq.FU_type_0::SimdCmp +"\
        "iq.FU_type_0::SimdCvt +"\
        "iq.FU_type_0::SimdMisc +"\
        "iq.FU_type_0::SimdShift +"\
        "iq.FU_type_0::SimdShiftAcc +"\
        "iq.FU_type_0::SimdReduceAdd +"\
        "iq.FU_type_0::SimdReduceAlu +"\
        "iq.FU_type_0::SimdReduceCmp +"\
        "iq.FU_type_0::SimdPredAlu) +"\
        ""\
        "80 * (iq.FU_type_0::SimdMult +"\
        "iq.FU_type_0::SimdMultAcc +"\
        "iq.FU_type_0::SimdDiv +"\
        "iq.FU_type_0::SimdSqrt +"\
        "iq.FU_type_0::SimdFloatAdd +"\
        "iq.FU_type_0::SimdFloatAlu +"\
        "iq.FU_type_0::SimdFloatCmp +"\
        "iq.FU_type_0::SimdFloatCvt +"\
        "iq.FU_type_0::SimdFloatDiv +"\
        "iq.FU_type_0::SimdFloatMisc +"\
        "iq.FU_type_0::SimdFloatMult +"\
        "iq.FU_type_0::SimdFloatMultAcc +"\
        "iq.FU_type_0::SimdFloatSqrt +"\
        "iq.FU_type_0::SimdFloatReduceAdd +"\
        "iq.FU_type_0::SimdFloatReduceCmp))"\
        "/ sim_seconds)" % sve_vl
```

Figure 1: Example of gem5 configuration for dynamic power consumption using *fictitious* constants for illustration, only.

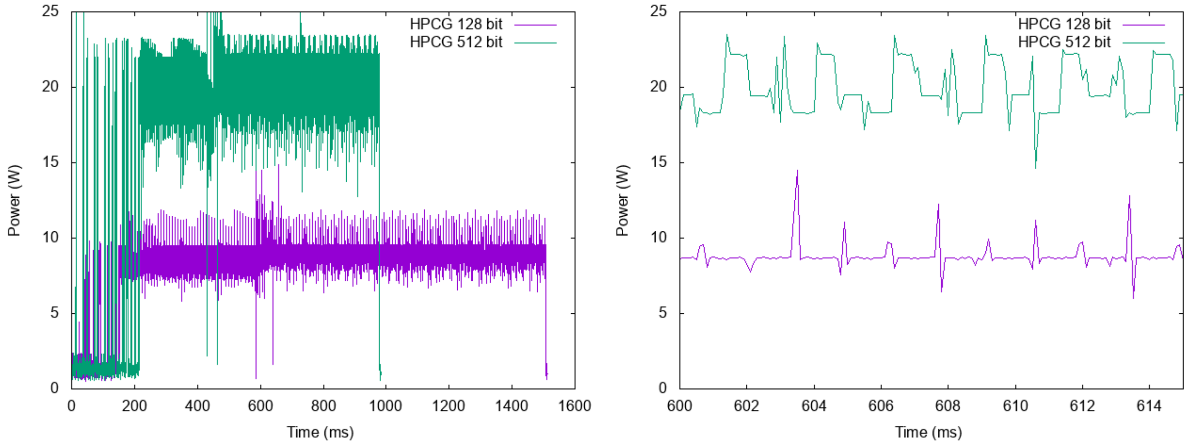

Figure 2: Power trace of running HPCG with different SVE vector lengths and the example configuration above. Left: power trace for entire application; right: detail of the power trace on 0.1 ms granularity.

References

[Bin11] Binkert, Nathan and Sardashti, Somayeh and Sen, Rathijit and Sewell, Korey and Shoaib, Muhammad and Vaish, Nilay and Hill, Mark D. and Wood, David A. and Beckmann, Bradford and Black, Gabriel and Reinhardt, Steven K. and Saidi, Ali and Basu, Arkaprava and Hestness, Joel and Hower, Derek R. and Krishna, Tushar. The gem5 simulator, August 2011.

[LAS$^+$09] Sheng Li, Jung Ho Ahn, Richard D Strong, Jay B Brockman, Dean M Tullsen, and Norman P Jouppi. Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–480. ACM, 2009.

[SBB$^+$17] Nigel Stephens, Stuart Biles, Matthias Boettcher, Jacob Eapen, Mbou Eyole, Giacomo Gabrielli, Matt Horsnell, Grigorios Magklis, Alejandro Martinez, Nathanael Premillieu, et al. The arm scalable vector extension. *IEEE Micro*, 37(2):26–39, 2017.

[WBD$^+$18] Matthew J. Walker, Sascha Bischoff, Stephan Diestelhorst, Geoff V. Merrett, and Bashir M. Al-Hashimi. Hardware-validated CPU performance and energy modelling. In *IEEE International Symposium on Performance Analysis of Systems and Software, ISPASS 2018, Belfast, United Kingdom, April 2-4, 2018*, pages 44–53. IEEE Computer Society, 2018.

[WDH$^+$17] Matthew J. Walker, Stephan Diestelhorst, Andreas Hansson, Anup Das, Sheng Yang, Bashir M. Al-Hashimi, and Geoff V. Merrett. Accurate and stable run-time power modeling for mobile and embedded cpus. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 36(1):106–119, 2017.