

Introduction

Recent advances in High Level Synthesis (HLS) tools promise to bridge the “Ease of Use” gap that prevents FPGA accelerators from becoming mainstream [1]. Our work summarizes how this process is done by many HLS tools in active development, emphasizing what new tools and techniques must be developed in order to bring the gap between user and the FPGA hardware.

Motivation

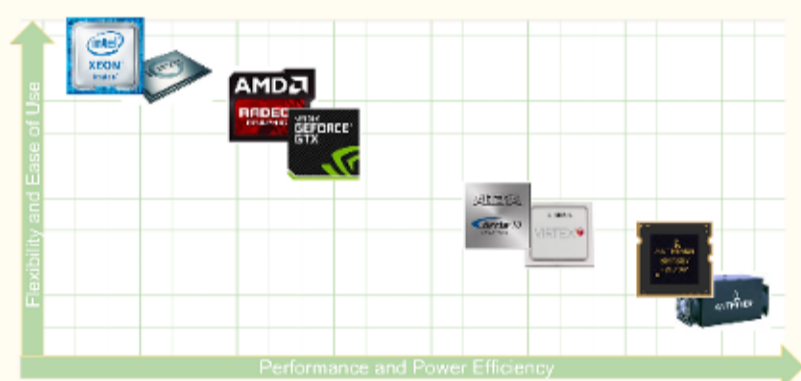
Given that clock frequencies on CPUs have stalled, the High Performance Computing (HPC) community started looking into:

- Multicore CPU architectures
- GPU architectures
- FPGA architectures
- ASIC architectures

These architectures have strengths and weaknesses that should be considered:

- Ease of use
- Performance
- Power consumption
- Time to market

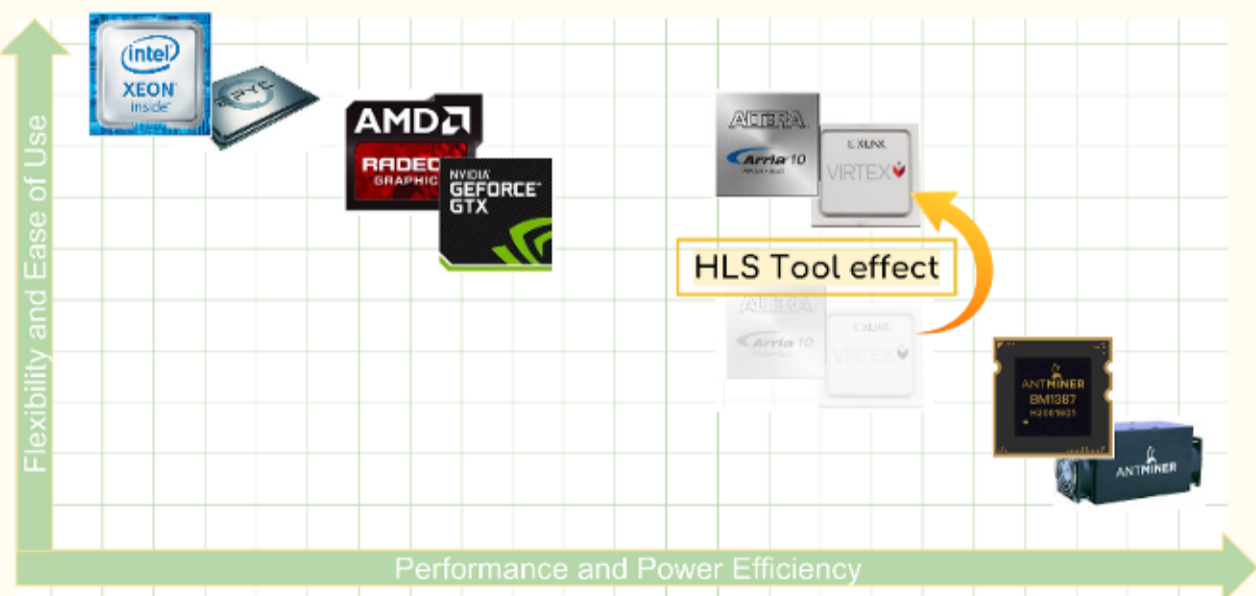
The chart on the right captures where these architectures stand considering the aforementioned factors:



Although performance and power consumption are critical factors, the HPC community has been **focusing its attention** on CPUs and GPUs, due to **Ease of Use** and **Time to Market** [1].

For this reason, recent development tools capable of making FPGA architectures easier to leverage have been on the rise. Some examples of such tools are:

- **BAMBU** [2] interfacing with GNU Compiler Collection.
- **LEGUP** [3] interfacing with the LLVM framework.
- Altera **Quartus OpenCL** with proprietary implementation.
- Xilinx **SDAccel with OpenCL** with proprietary implementation.



How to use it?

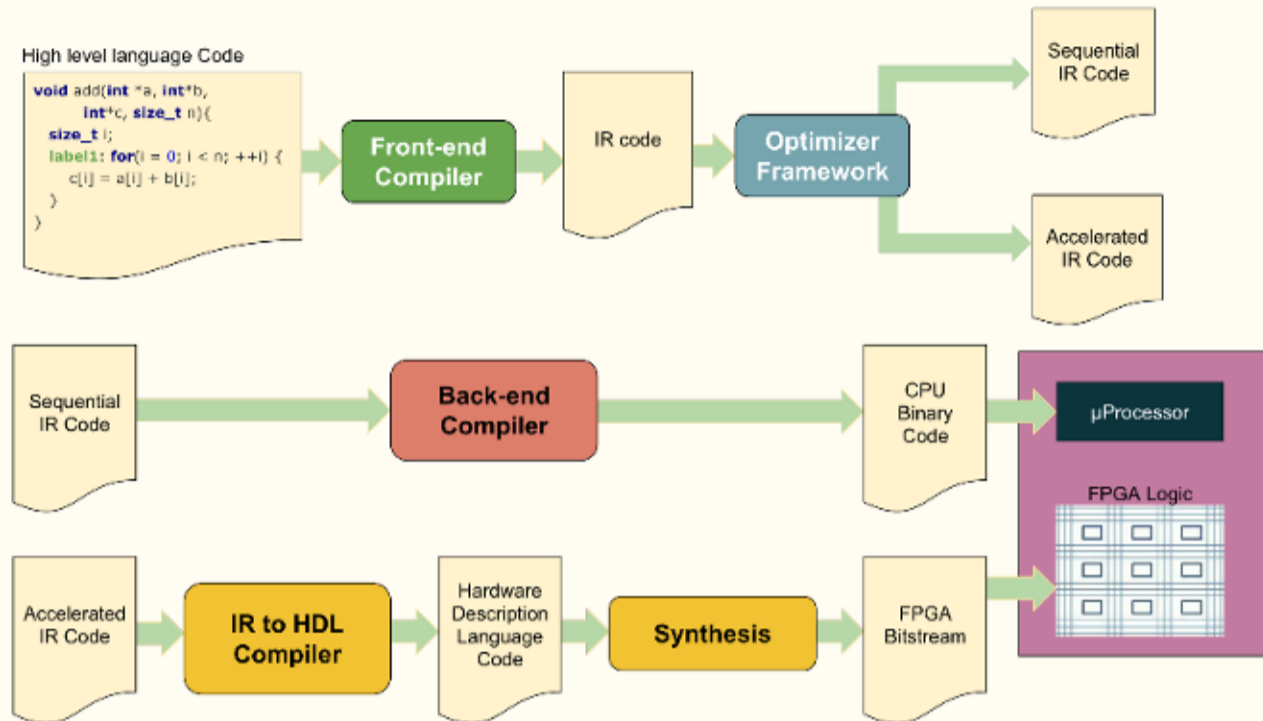
Depending on the tool, the user must highlight the accelerated portion of the code to be executed on the accelerator by:

- Using graphical tools such as **Simulink**
- Using **Labels**
- Using pragmas, such as **OpenMP**
- Using library calls such as **Pthreads**
- Using parallel frameworks such as **OpenCL**

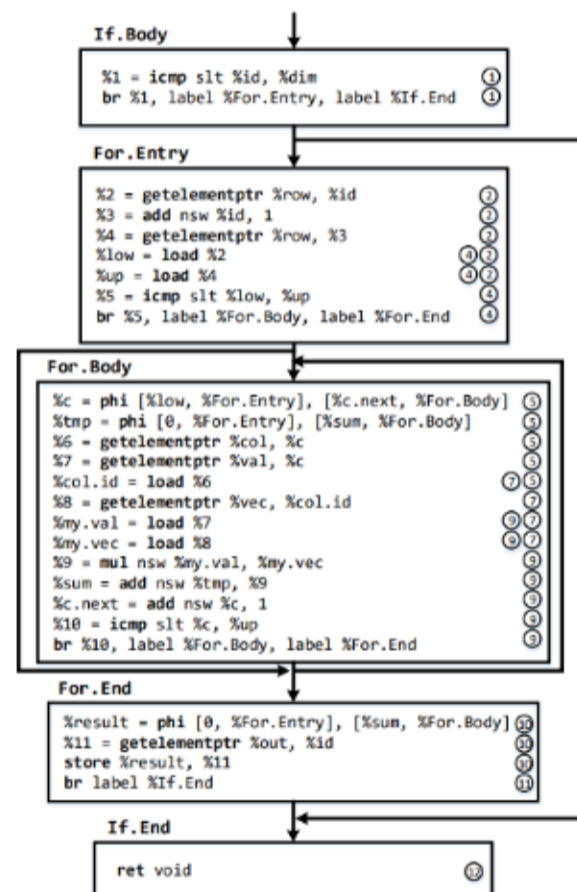
```
pthread_t thread[T]; int tid[T];
for (int i = 0; i < T; i++) {
    tid[i] = i;
    pthread_create(&thread[i], NULL,
                 matrix_add, &tid[i]);
}
for (inti = 0; i < T; i++) {
    pthread_join(thread[i], NULL);
}
```

System Overview

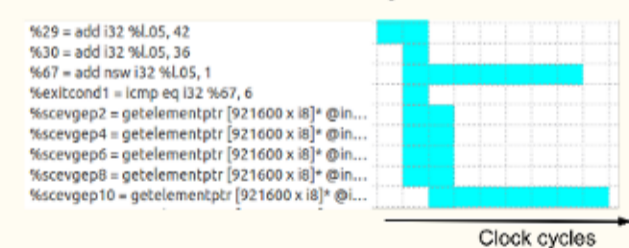
Bellow, the typical flow of HLS tools [2][3] can be found:



Accelerated IR Code Scheduling



- In the “IR to HDL compiler”:
- Instructions are scheduled to different pipeline stages.
 - Instructions that can be executed together are assigned to the same clock cycle.
 - Compiler uses IP cores to implement common tasks such as memory interfaces.



Extracted from: Momeni et al. [4]

Extracted from: Momeni et al. [4]

Literature Results

RESULTS FOR BLACK-SCHOLES ON ARM PROCESSOR-ONLY, ARM HYBRID, AND x86 ARCHITECTURES

Architecture	Time (s)	Power (W)	Energy (J)
ARM SW 1T	50.717	2.128	107.921
ARM SW 2T	25.418	2.327	59.159
ARM SW 3T	25.870	2.337	60.453
ARM Hybrid 1T	1.184	2.301	2.725
ARM Hybrid 2T	0.642	2.529	1.622
ARM Hybrid 3T	0.484	2.683	1.297
Xeon SW 1T @ 3.75 GHz	3.312	55.214	182.878
Xeon SW 2T @ 3.75 GHz	1.668	69.699	116.273
Xeon SW 3T @ 3.65 GHz	1.144	79.234	90.640
17 SW 1T @ 3.9 GHz	2.735	24.096	65.909
17 SW 2T @ 3.9 GHz	1.372	26.504	36.355
17 SW 3T @ 3.8 GHz	0.944	35.459	33.476

Extracted from: Brown et al. [3]

Conclusions

- FPGA platforms present good performance with great power efficiency, as compared to CPU and GPU platform solutions.
- HLS tools are already capable of delivering similar performance as CPU platforms.

- HLS tools are able to simplify their programming flow.
- Increasing the ease of use of FPGAs will increase platform adoption.
- Increasing the platform adoption rates will increase interest by the research community, leading to more breakthroughs.

References

[1] Nane, R., Sima, V.-M., Pilato, C., Choi, J., Member, S., Fort, B., ... Bertels, K. (2016). A Survey and Evaluation of FPGA High-Level Synthesis Tools. *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, 35(10). <http://doi.org/10.1109/TCAD.2015.2513673>
 [2] C. Pilato and F. Ferrandi, "Bambu: A modular framework for the high level synthesis of memory-intensive applications," 2013 23rd International Conference on Field Programmable Logic and Applications, Porto, 2013, pp. 1-4.
 [3] Choi, J., Brown, S. D., & Anderson, J. H. (2017). From Pthreads to Multicore Hardware Systems in LegUp High-Level Synthesis for FPGAs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(10), 2867-2880. <http://doi.org/10.1109/TVLSI.2017.2720623>
 [4] Momeni, Amir, et al. "Hardware thread reordering to boost OpenCL throughput on FPGAs." *Computer Design (ICCD)*, 2016 IEEE 34th International Conference on. IEEE, 2016