



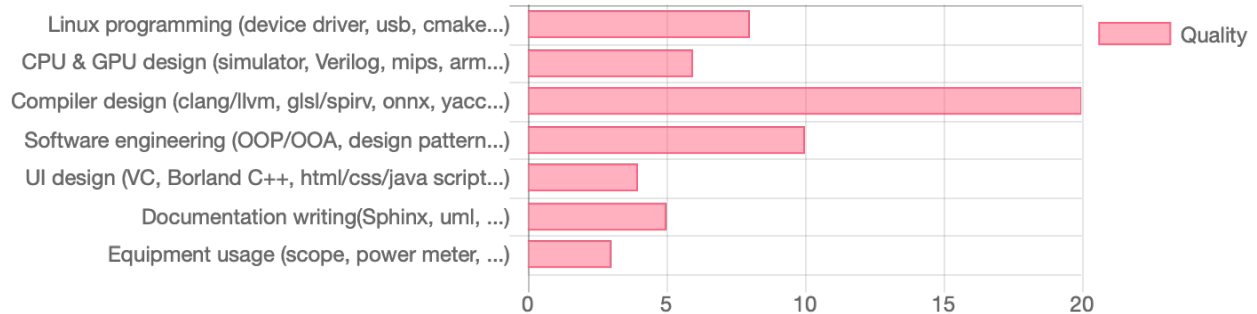
I am a compiler developer with good experience in llvm cpu and gpu backend, lld linker, npu/onnx, c++, OpenGL/glsl and simulator, ..., and enjoy with compiler.

RESUME

QUALIFICATION

Over 20 years experience in c/c++ programming, 9 years compiler toolchain related experience and research in parallel processing for master degree.

SKILLS



MY OPEN SOURCE PROJECT

I am proud of my work is accepted by LLVM documentation, appears at <http://llvm.org/docs/tutorial/#external-tutorials>

Tutorial: Create an LLVM Backend compiler <http://jonathan2251.github.io/lbd/index.html>

Tutorial: Create an LLVM Backend Toolchain <http://jonathan2251.github.io/lbt/index.html>

The concept of GPU compiler <http://jonathan2251.github.io/lbd/gpu.html>

EDUCATION

1997-1999 Master, June 1999, National Taiwan Normal University (國立台灣師範大學), Taipei, Major: Information Science.

1991-1994 B.S., June 1994, National Taiwan Technology University of Science and Technology (國立台灣科技大學), Taipei, Major: Industry Engineer.

LICENSE

Taiwan National Computer Engineer license, 1995 高考資訊技師及格.

EXPERIENCE



THESIS OF MASTER DEGREE

[The Researches of Column Sort and Related Problems](#)
Conference : Search "行排列法簡化步驟之研究" on above link

PROPOSAL OF PHD STUDY

[The Researches of Sorting Network and Related Algorithm](#)

OTHER WORK

Take course "Image processing" and program: [Jpeg decoder](#)

Web and javascript: [As my resume](#) and [my personal web site](#)

[Graphviz](#): as some graph diagrams used in this CV. Source code: [mywork_1.gv](#) and [study_and_apply.gv](#)

ACHIEVEMENT**Biren**

Gpu CodGen for tensor instructions and usharpid handling.

Gpu optimization and bug fix.

Propose solution for paralel processiong of our Cude-like language async{...}.

Kneron

Re-implement top 2 layers of our npu compiler for our common graph data structure to handle onnx.

Implement compiler input interface to support encryption-onnx and config file format.

Confirm solution for MLIR supporting.

Hisilcon

GPU compiler scope:



To support an our new designed GPU for cell phone, ported from ARM. 20% of frontend is changed, 50% of backend is changed in aspect of number of code lines.

My work:

Implement compiler (fontend + llvm backend) for 80% of texture related API, [80 APIs totally here](#), by myself alone and document writing.

Instruct and help other engineers for the other 20% of texture related API, review their implementation and co-work with the leader of texture part of architect.

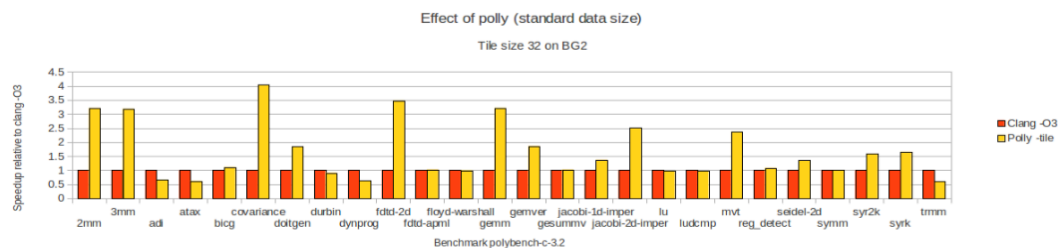
Implement Prefetch-Sample optimization for running 2D-sampling instructions by driver before load/run glsl shader.

Implement compiler supporting our GPU's load/store for RGBA fixed floating point format of vulkan (32, 16, 11, 10 and 2 bits; NaN Infinity) alone and document writing.

Marvell

Implement semi-auto software system of running benchmark and generating report for gcc toolchain.

Demonstrate polly and the concept of polyhedral optimization model for Marvell llvm and gcc toolchain optimization. Polly is a software for loop optimization.



Implement co-simulator for a few Marvell's ARM based 64-bit cpu.

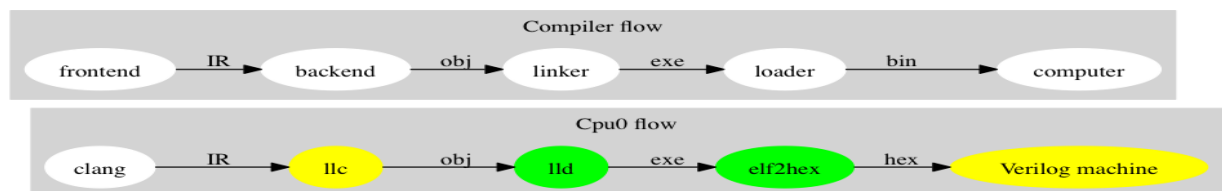
Propose and implement DSL on simulator to save tens of system verification in c++ coding.

Complete cmake to replace make for Csim.

Advantage: simpler and cross-os-platform than make.

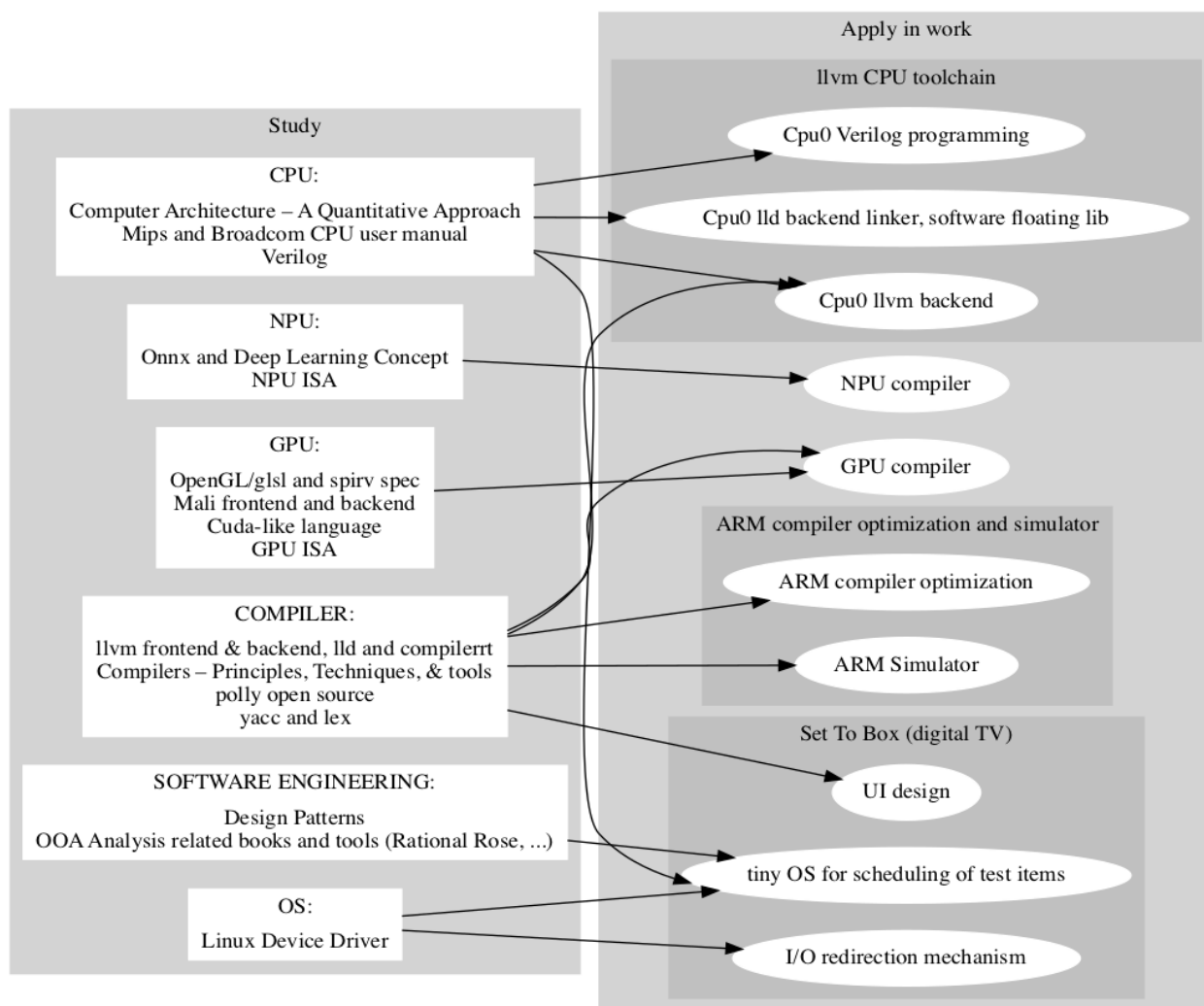
LLVM open source project

The lower half is the my llvm backend's work flow. Yellow and green parts are my implementation in my books.

**Mortorola**

Develop Set Top Box's software framework.

Learning after school & applying in work



References

[My former manager's recommendation letter: https://jonathan2251.github.io/ws/en/RL_Marvell.pdf](https://jonathan2251.github.io/ws/en/RL_Marvell.pdf)