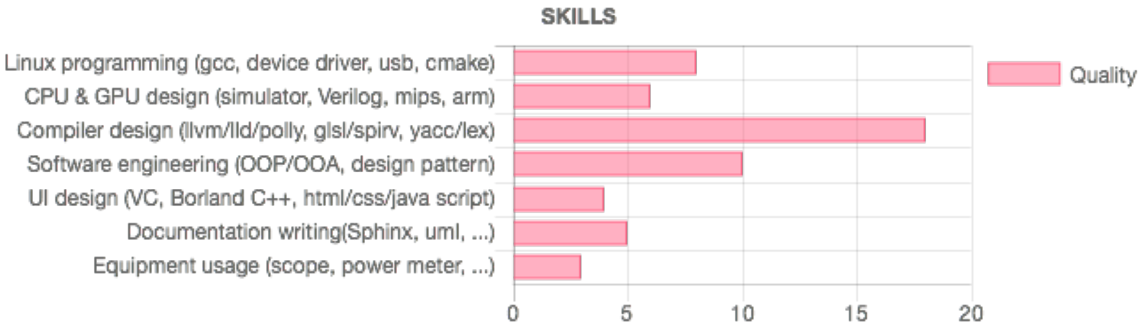
Chung-Shu Chen 陳鍾樞

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.

CV (DETAIL RESUME)

QUALIFICATION

Over 20 years experience in c/c++ programming, 7 years compiler toolchain related experience and research in parallel processing for master degree.



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

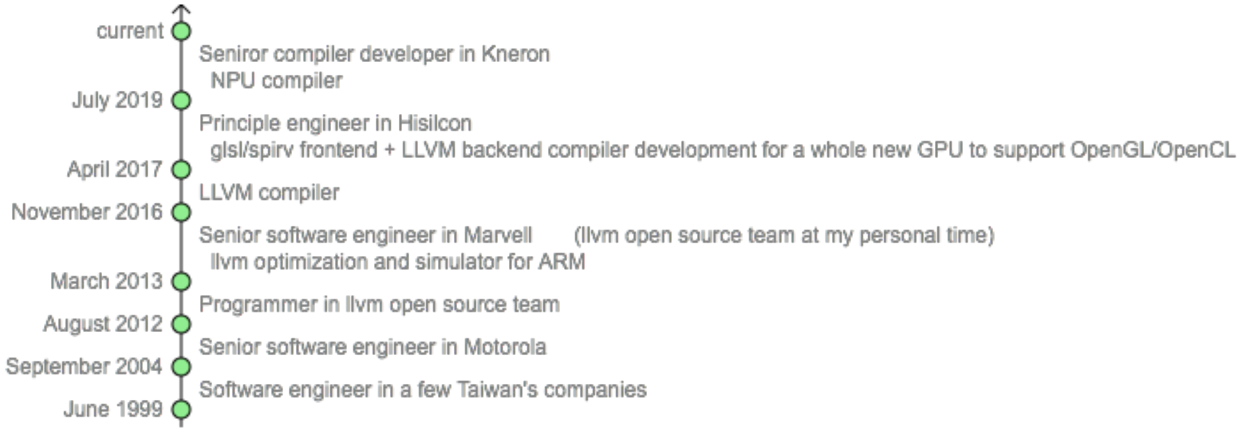
1997-1999 Master, June 1999, National Taiwan Normal University (國立台灣師範大學), Taipei, Major: Information Science.

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



current

Seniror compiler developer in Kneron
NPU compiler

July 2019

Principle engineer in Hisilcon
glsl/spirv frontend + LLVM backend compiler development for a whole new GPU to support OpenGL/OpenCL

April 2017

LLVM compiler

November 2016

Senior software engineer in Marvell (llvm open source team at my personal time)
llvm optimization and simulator for ARM

March 2013

Programmer in llvm open source team

August 2012

Senior software engineer in Motorola

September 2004

Software engineer in a few Taiwan's companies

June 1999

September 2004 - June 1999:
Proton 2014/3 – 2014/9 Manager Digital TV programming, Abocom 2013/6 – 2014/3, Senior Engineer 802.11b programming,
DBTEL 2011/11 – 2013/6 Engineer DECT wireless phone programming, Symmetry 2011/2 – 2011/11 Engineer, SGSN and GGSN for GPRS&3G programming,
Cando 2010/7 – 2011/2 Engineer CAM programming, Spirox 2009/12 – 2010/7, Engineer CAM programming, Intech 2009/6 – 2009/12 Engineer CAM programming

THESIS OF MASTER DEGREE

The Researches of Column Sort and Related Problems

PROPOSAL OF PHD STUDY

The Researches of Sorting Network and Related Algorithm

OTHER WORK

Take course "Image processing" and program: Jpeg decoder

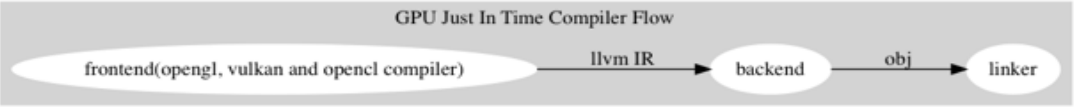
Web and javascript: As my resume and my personal web site

Graphivz: as some graph diagrams used in this CV. Source code: mywork_1.gv and study_and_apply.gv

ACHIEVEMENT

Hisilcon

GPU compiler scope:


GPU Just In Time Compiler Flow
frontend(opengl, vulkan and opencl compiler)
llvm IR
backend
obj
linker

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 and optimization by myself alone and document writing.

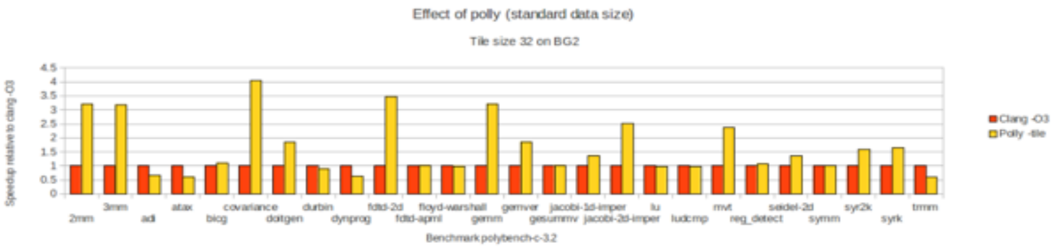
Instruct and help other engineers for the other 20% of texture related API, review their implementation and co-work with the leader of texture part of architect.

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

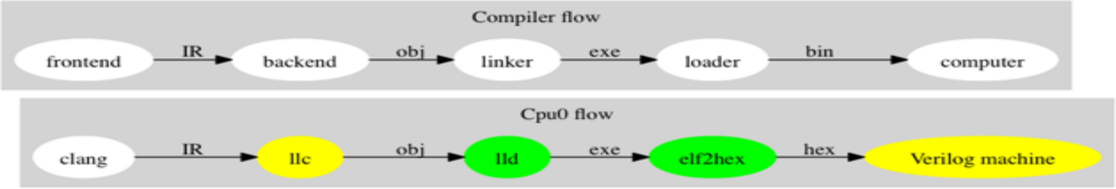
Propose and implement DSL in simulator to save tens of c++ coding.

Complete cmake to replace make form Csim.

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

LLVM open source project

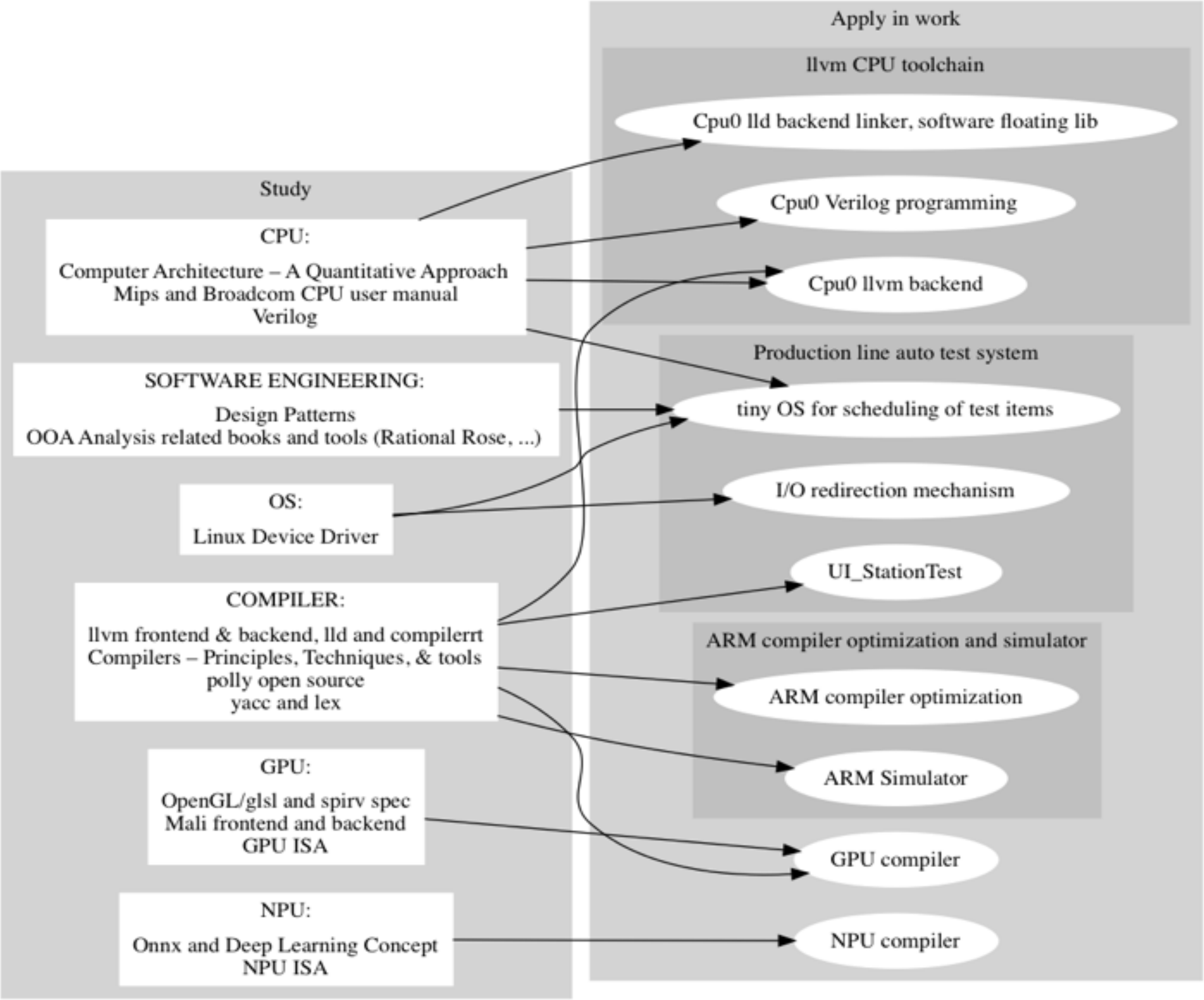
The lower half is the my llvm backend's work flow. Yellow and green parts are my implementation in my books.



Mortorola

Develop Set Top Box's software framework.

Learning after school & applying in work


Apply in work
llvm CPU toolchain
Cpu0 lld backend linker, software floating lib
Cpu0 Verilog programming
Cpu0 llvm backend
Production line auto test system
tiny OS for scheduling of test items
I/O redirection mechanism
UI_StationTest
ARM compiler optimization and simulator
ARM compiler optimization
ARM Simulator
GPU compiler
NPU compiler
Study
CPU:
Computer Architecture – A Quantitative Approach
Mips and Broadcom CPU user manual
Verilog
SOFTWARE ENGINEERING:
Design Patterns
OOA Analysis related books and tools (Rational Rose, ...)
OS:
Linux Device Driver
COMPILER:
llvm frontend & backend, lld and compilerrt
Compilers – Principles, Techniques, & tools
polly open source
yacc and lex
GPU:
OpenGL/glsl and spirv spec
Mali frontend and backend
GPU ISA
NPU:
Onnx and Deep Learning Concept
NPU ISA

References

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