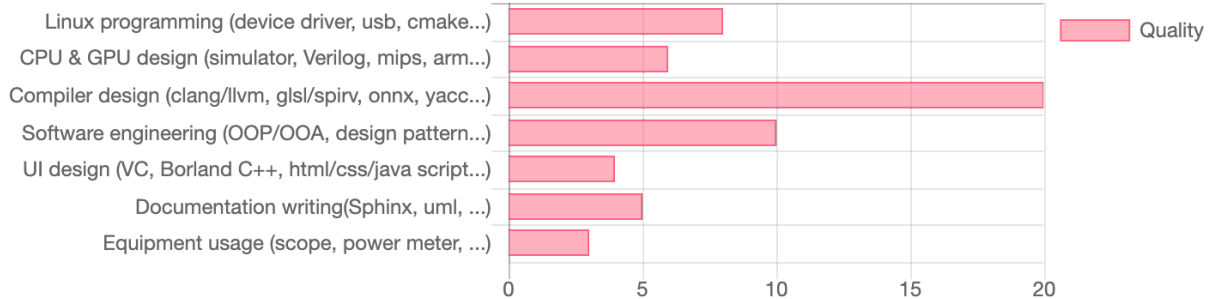*I am a compiler developer with good experience in llvm cpu and gpu backend, lld linker, npu/onnx, c++, OpenGl/glsl and simulator, ..., and enjoy with compiler.*

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

Skill	Quality
Linux programming (device driver, usb, cmake...)	8
CPU & GPU design (simulator, Verilog, mips, arm...)	6
Compiler design (clang/llvm, glsl/spirv, onnx, yacc...)	20
Software engineering (OOP/OOA, design pattern...)	10
UI design (VC, Borland C++, html/css/java script...)	4
Documentation writing(Sphinx, uml, ...)	5
Equipment usage (scope, power meter, ...)	3

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

Date	Role
current	GPU compiler developer in Biren — Our designed Cuda-like language compiler based on clang/llvm for our GPU
October 2020	Senior compiler developer in Kneron — NPU compiler
July 2019	Principle engineer in Hisilcon — glsl/spirv frontend + LLVM backend compiler development for a whole new GPU ISA
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	

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

Graphivz: 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 optimatization 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 encription-onnx and config file format.

Confirm solution for MLIR supporting.

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, 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.
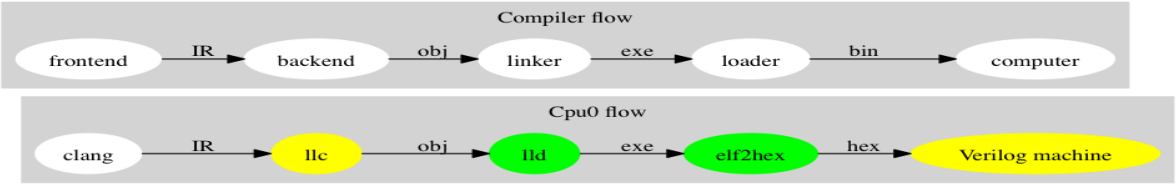
Marvell

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.

Effect of polly (standard data size)

Tile size 32 on BG2

Benchmark polybench-c-3.2
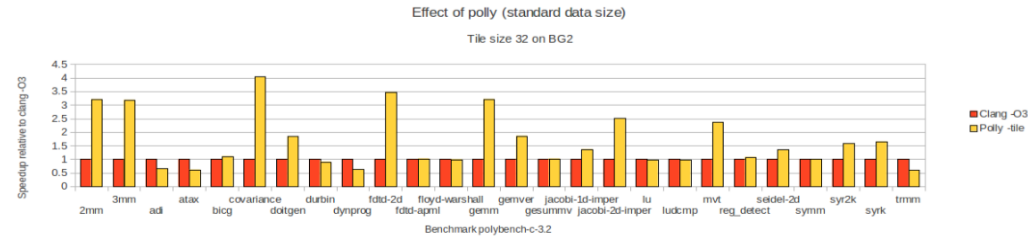
Implement co-simulator for a few Marvell's ARM based 64-bit cpu.

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.

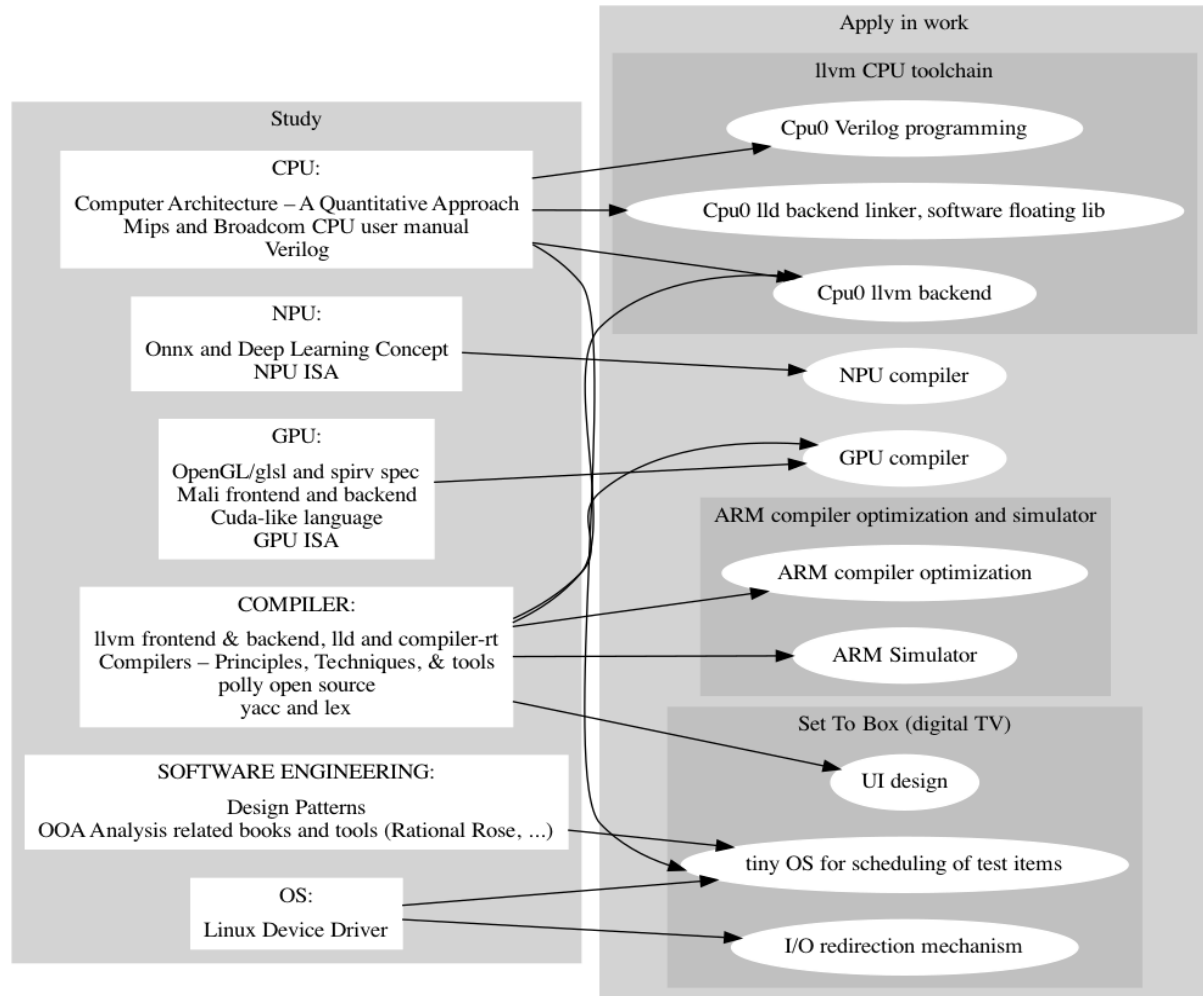### LLVM open source project

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

Compiler flow

frontend → IR → backend → obj → linker → exe → loader → bin → computer

Cpu0 flow

clang → IR → llc → obj → lld → exe → elf2hex → hex → Verilog machine

Mortorola

Develop Set Top Box's software framework.

Learning after school & applying in work

Apply in work

llvm CPU toolchain

Study

CPU:

Computer Architecture – A Quantitative Approach
Mips and Broadcom CPU user manual
Verilog

Cpu0 Verilog programming

Cpu0 lld backend linker, software floating lib

Cpu0 llvm backend

NPU:

Onnx and Deep Learning Concept
NPU ISA

NPU compiler

GPU:

OpenGL/glsl and spirv spec
Mali frontend and backend
Cuda-like language
GPU ISA

GPU compiler

ARM compiler optimization and simulator

ARM compiler optimization

COMPILER:

llvm frontend & backend, lld and compiler-rt
Compilers – Principles, Techniques, & tools
polly open source
yacc and lex

ARM Simulator

Set To Box (digital TV)

UI design

SOFTWARE ENGINEERING:

Design Patterns
OOA Analysis related books and tools (Rational Rose, ...)

tiny OS for scheduling of test items

OS:

Linux Device Driver

I/O redirection mechanism

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