



陳鍾樞

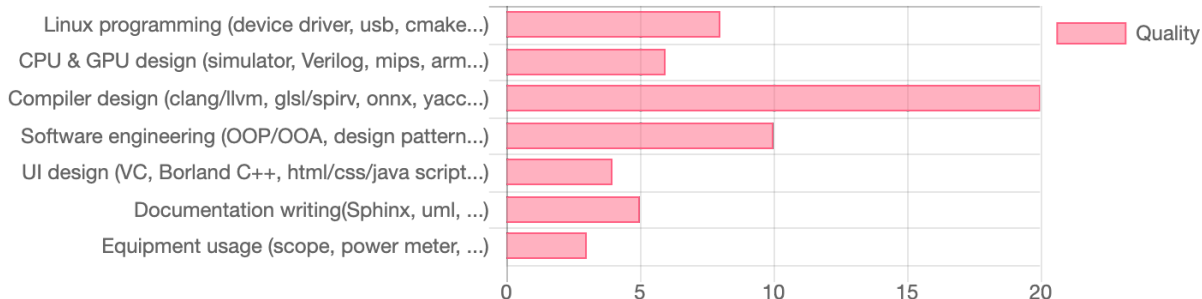
我是一位編譯器開發者，擁有在 LLVM CPU 和 GPU 後端、LLD 連結器、NPU/ONNX、C++、OpenGL/GLSL、模擬器等方面的紮實經驗。我熱衷於編譯器及相關技術的開發工作。

履歷

資格

二十年c/c++軟體開發經驗，十三年編譯器相關工具開發經驗，碩士時研究平行處理。

SKILLS



我的開法原始碼專案

很高興我的作品已被LLVM接受，出現在 <http://llvm.org/docs/tutorial/#external-tutorials>

如何建立LLVM後端編譯器 <http://jonathan2251.github.io/lbd/index.html>

如何建立LLVM後端系統工具 <http://jonathan2251.github.io/lbt/index.html>

GPU編譯器概念 <http://jonathan2251.github.io/lbd/gpu.html>

學歷

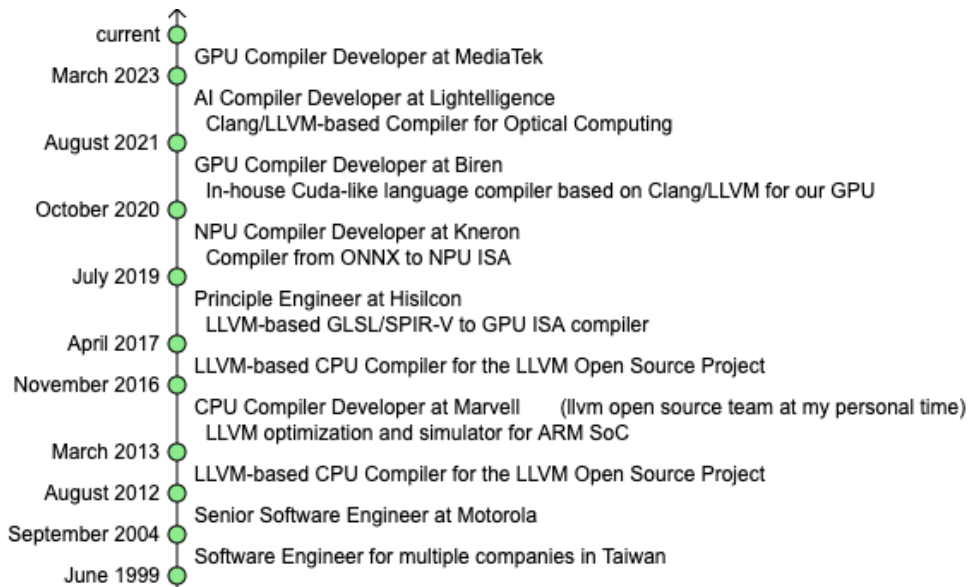
1997-1999 碩士班，國立台灣師範大學，台北，主修：資訊科學。

1991-1994 學士班，國立台灣科技大學，台北，主修：工業工程。

證照

1995年 高資資訊技師及格。

經驗



碩士論文



[The Researches of Column Sort and Related Problems](#)

論文期刊：上述鏈結網頁搜尋"行排列法簡化步驟之研究"

博士班研究計畫



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

其餘作品

修影像處理課程與撰寫：[Jpeg decoder程式](#)

網頁與javascript：[html簡歷](#) 與 [我個人網頁](#)

[Graphviz](#)：如此詳細履歷裡的一些圖學的圖。原始碼：[mywork_1.gv](#) and [study_and_apply_ch1.gv](#)

工作貢獻

Lightelligence

實作TaskGraph並開發與TVM-compiler以及Runtime合作介面使得我們的平台可以支持深度學習的圖編譯。

為Lightelligence的基於RISC-V的光學NPU開發後端編譯器，內容包括：

1. 基於GCC、LLVM和QEMU/Gem5等開源專案，建構完整的RISC-V編譯器工具鏈。評估RISC-V供應商並進行價格協商，利用我們內部從開源自行建構工具鏈的能力作為談判優勢。
2. 主導Aurora硬體產品的軟體開發，並親自負責編譯器後端的程式設計。
3. 在C++編譯器中開發TaskGraph元件及其與Runtime模組的介面，實現我們平台上的深度學習計算圖功能。

Biren

GPU tensor指令與usharpid處理。

GPU編譯器最佳化與bug fix。

我們的Cude-like語言[async{...}](#)平行處理解法白皮書。

Kneron

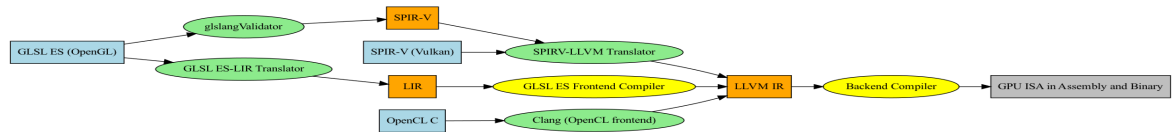
改寫我們的NPU編譯器上兩層的IR中間碼轉換程式以提供共同的hardware independent圖形資料結構，以利多種NPU的支持。

支持加密格式的ONNX與config檔輸入。

確認如何支持MLIR。

Hisilcon

GPU編譯器範圍：



為支援自行設計的手機GPU，移植ARM的code。如上圖黃色部份：20%前端需修改，50%後端需修改(以行數計算)。

貢獻：

獨立完成80% texture相關的API與optimization (frontend + LLVM backend)與document撰寫。

指導別的工程師完成其餘20% texture相關的API, [80 APIs totally here](#), 核對並與texture的架構leader一起合作。

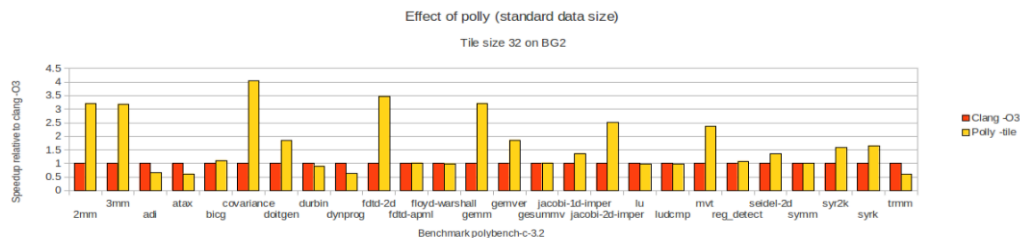
完成Prefetch-Sample optimization, 讓driver在載入GLSL bin與執行sampling指令前就可驅動2D sampling指令。

獨立完成GPU對Vulkan load/store RGBA 固定浮點格式(32, 16, 11, 10 and 2 bits; NaN Infinity)支援的指令生成與document撰寫。

Marvell

設計半自動的軟體系統，自動執行用GCC編譯器編譯benchmark程式，並產生Excel比較圖表。

為提升Marvell公司gcc與llvm編譯器軟體工具效能，介紹Polly軟體系統。Polly是針對loop最佳化的開放原始碼專案。



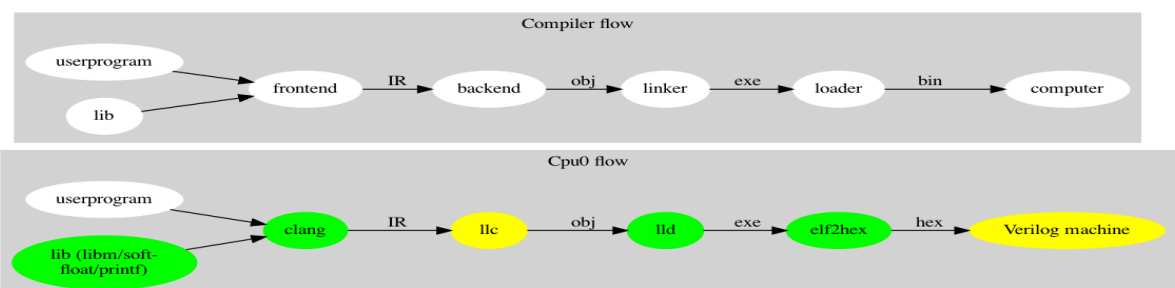
提出與實作DSL領域語言解決方案應用在ARM 64位元Csim上。

用cmake替代make於CSim上。

優點: 比make簡潔與跨平台。

我的llvm開放原始碼專案

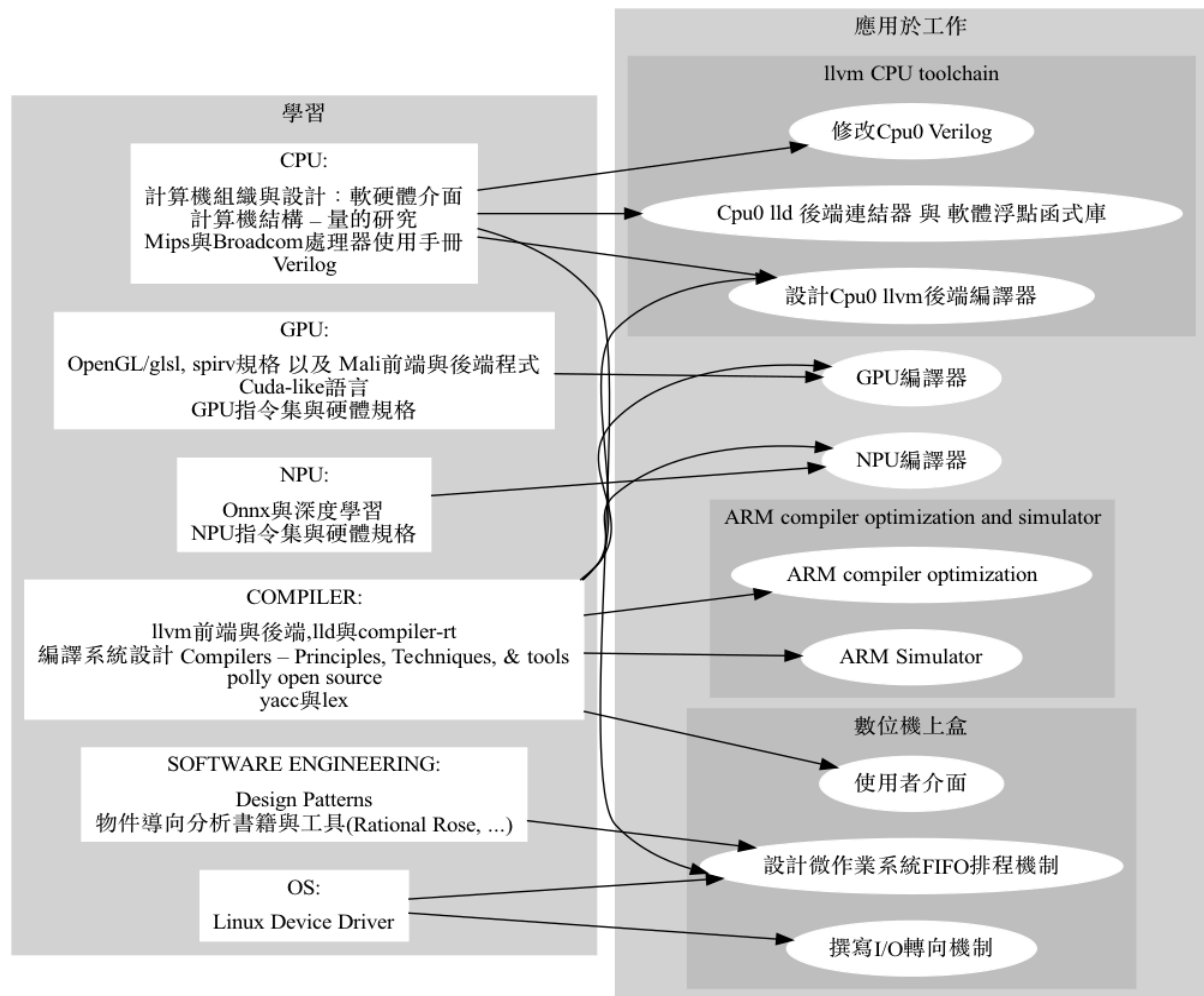
下半部是llvm的流程圖。黃色與綠色分別是我書中（如上，我的開法原始碼專案）。



Mortorola

開發數位機上盒

出社會後的學習並運用於工作



推薦函

前主管推薦函: https://jonathan2251.github.io/ws/ch1/RL_Marvell.pdf