



Chung-Shu Chen 陳鍾樞 (03)6681193, 0970577923 gamma_chen@yahoo.com.tw

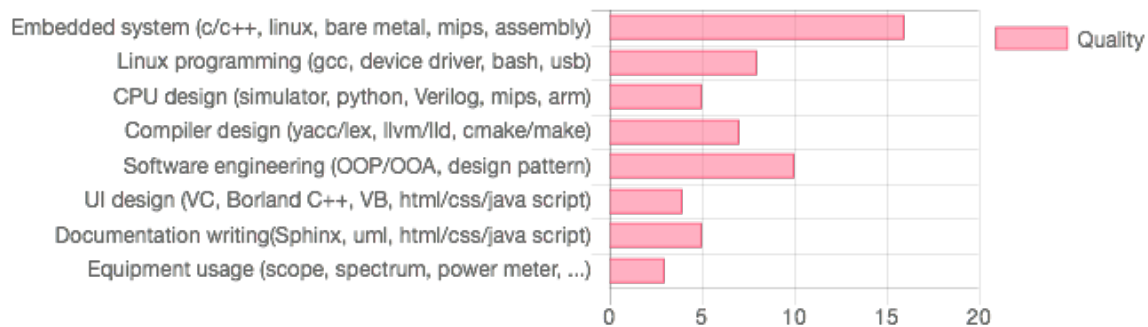
I am a programmer who used different computer languages, tools and computer related knowledge to develop different software and hardware products, such as Set Top Box, CPU, compiler, simulator, ..., in my career. I always have full passion in programming, knowing tools and software engineering method well, loving programming and debugging, and feel happy about writing good document to deliver more readable and maintainable software.

CV (detail resume)

QUALIFICATION

Over 20 years experience in c/c++ embedded system programming, well knowledge in software development and debugging.

SKILLS



MY OPEN SOURCE PROJECT

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

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

EDUCATION

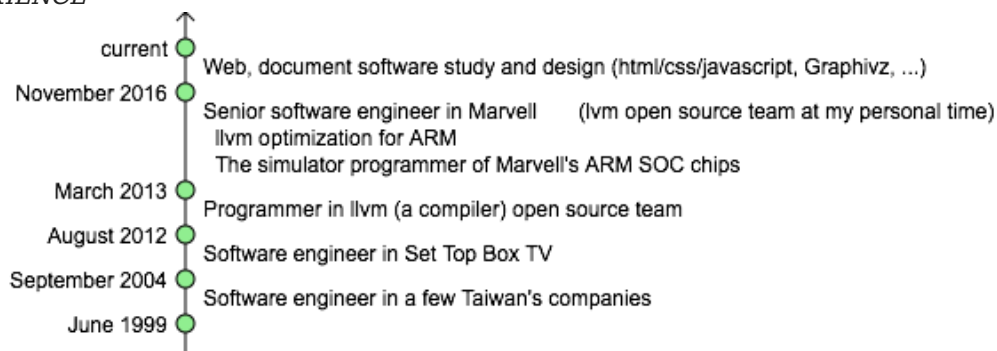
1997-1999 Master, June 1999, National Taiwan Normal University (國立台灣師範大學), Taipei, Major: Information Science.

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](#)

PROPOSAL OF PHD STUDY

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

WORK OF MASTER DEGREE

[Took course "Image processing" and programmed Jpeg decoder](#)

*ACHIEVEMENT**Marvell*

- Introduce polly for Marvell llvm toolchain optimization. Polly is a software for loop optimization.
 - The advantage is that it might be 500% speed up in some numerical application program, such as matrix multiplication, or other kind of matrix operation. Sadly, Marvell's CPUs are not used in numerical applications.
- Apply compiler code generation idea to propose this solution and Csim team adopted it.
- Complete cmake and adjust python programs to build Csim with cmake. It save the effort in maintaining both make and windows project file. Cmake also support iMac user.
 - Present the better option for cmake writting.
 - The version of cmake in Csim server is 2.6.x, however, the cmake 2.8.8 provides shared object to save the unnecessary compilation.
 - Report performance comparing between cmake and ninja.

Mortorola

- Designe QIP7kP1 and P2 board level production line testing code. Over 500 millions units of QIP7k model of Set Top Box are tested with this code on production line, as well as shipping to market. Other major products DCX33, DCX34 and DCX35 (over 1000 millions of box) are ported from QIP7kP1 & QIP7kP2.
- Combine mips cpu and software engineer knowledge to create a frame work for board level testing code as manager request.
 - Use Rational Ross UML tool to design the frame work. In this task. I apply object oriented analysis & program (OOA, OOP, design pattern) knowledge and UML tool skill.
 - Implement the I/O redirection mechanism to make the new version of code porting task easier.
 - Implement Board Level Test with design pattern.

*CPU, COMPILER, OS, AND SOFTWARE ENGINEERING RELATED LEANING & APPLY**CPU architecture study*

- Study books "Computer Organization and Design: The Hardware/Software Interface", "Computer Architecture – A Quantitative Approach", Mips and Broadcom CPU user manual. Verilog language/tool study.
- Apply in work:
 - Find the bug of Code Warrior's compiler in supporting Mips cpu. The bug come from #pack(1) statement code generation.
 - Email Code Warrior the bug along with the example code I write for identify this bug, and they admit that is a bug from their compiler for mips backend support.
 - Implement the "tiny OS" mechanism with mips assembly language to solve the halting problem in some program.
 - After this implementation, the code is running more stable and fast. It is a serious problem since when one test item "halting", the other test items scheduled after that won't be run. In this situation, operator has no idea of the test result since test result is displayed after all test items finish execution. I implement a mechanism to jump out from the test item (by restore the stack and cpu registers status). My USA's coworker call it tiny OS. It's c++ plus few hundred lines of mips assembly with "observe pattern" of "design pattern".
 - Apply knowledge from Verilog and book "Computer Architecture" to change the Verilog code of Cpu0 to become a real cpu design in market for my llvm backend compiler support.

Compiler study:

- Studied llvm front & backend design. Studied book "Compilers – Principles, Techniques, & tools 2nd Aho,...", yacc and lex tool.
- Apply in work:
 - Implement the Cpu0 llvm compiler backend from scratch for tutorial purpose, and writing books as in my publish books. A lot of readers have been asking me questions and I answering them out of working time.
 - Apply the compiler syntax parsing tool, yacc & lex, in test station design (Production line engineer and I can specify test items for each station in script language I defined).

OS study:

- Study book "Linux Device Driver (2nd & 3rd edition)"

- Apply in work:
 - Implement the "tiny OS" mechanism as mentioned in above section "CPU architecture study".

Software Engineering study:

- Study book "Design Patterns" and "OOA Analysis related books and tools (Rational Rose, ...)"
 - Apply in work:
 - In work the Board Level Test Code framework project in Motorola as mentioned above.
 - Utilize five patterns in "design pattern" and Rational Ross tool to create framework, document and generate code.
-

References

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