
RISC-Vの量子計算拡張の 参照実装とqlang言語開発

新里 祐教

自己紹介

■名前：新里 祐教(hirotakaster)

■所属：SIProp Project

GM0 Internet, Inc 代表室
programmer

■得意：IT全般

ものづくり

特集

ビデオ会議のネットワーク技術

主要なサービスで徹底調査！

+ 特集をフォロー

新里 祐教 GMOインターネット 特命担当技術分析官

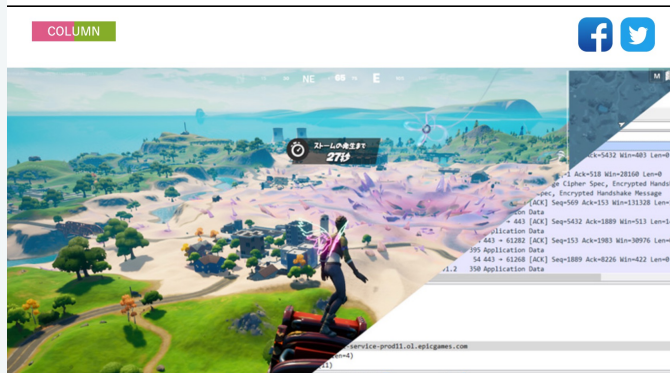
2020.11.10
「Maker Faire Tokyo 2020」レポート
#5 — 量子ビットの操作を可視化した
「8bit Quantum Computer」
Text by Takako Ouchi

tags Blog Computers Maker Faire

いいね！ 138 ツイート 17



『フォートナイト』のバトルはどうや
ネット対戦ゲームから見える通信のしくみ



Cypress Semiconductor – An Infineon Technologies Company

News Letter



国内でも拡大するIoT — IoT エッジ デバイス開発における
サイプレス IoT-AdvantEdge ソリューションの役割
～「IoT-AdvantEdge を使ってスマートケトルを作ってみた」も解説～

インフィニオン テクノロジーズ グループであるサイプレスは、IoT エッジ デバイス開発の課題を解決し迅速な製品市場投入を支援する目的で、コネクティビティ デバイスやマイクロコントローラー、ソフトウェア、ツール&サポート、そしてエコシステム パートナー 各社の機能をすべて統合した IoT-AdvantEdge™ソリューションを市場展開しています。世界的にビジネス規模が拡大を続けている IoT ですが、国内では実際のどのくらい浸透しているのでしょうか。また、サイプレスの IoT-AdvantEdge はどのような役割を果たすのでしょうか。GMO インターネット株式会社で特命担当 技術分析官を務める新里 祐教氏に、サイプレスが独自で実施した市場調査の結果分析と、IoT-AdvantEdge を使用したスマートケトルを設計いただいた内容も解説いただきました。

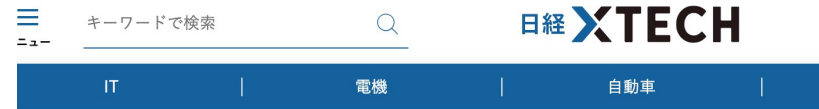
セブンイレブン向けネットプリントアプリに脆弱性

セブンイレブンの店頭複合機で印刷が行える富士ゼロックス製のスマートフォン向けアプリに中間者攻撃を受けるおそれがあることがわかった。

iOS向けに提供されている「netprint」、iOSやAndroid向けに提供されている「かんたんnetprint (旧写真かんたんプリント)」にSSL/TLSサーバ証明書 を正しく検証しない脆弱性が明らかとなったもの。中間者攻撃により、盗聴などが行われるおそれがある

CVE番号はそれぞれ「CVE-2020-5520」「CVE-2020-5521」「CVE-2020-5522」が割り当てられている。Android向けに提供されている「netprint」は影響を受けないという。

同脆弱性は、新里祐教氏が情報処理推進機構 (IPA) へ報告したもので、JPCERTコーディネーションセンターが調整を実施。脆弱性を修正した新版がリリースされている。



本質はすごい新楽器、誰でも音楽を遊べる
toio新作「ピコトonz」

新里 祐教 GMOインターネット 特命担当技術分析官

2020.09.10

未踏量子、イベント展示、作品制作、
IoT的な物、ゲーム、IPSJ向けネタ、
セキュリティ、ネットワークもの、受賞物

作品など

8bit 2020

- Overview
Quantum computer qubit gate control
microwave pulse design sound and
chladni figure.
- Exhibit
Maker Faire Tokyo 2020
- Others
<https://www.youtube.com/watch?v=xUeF0tJOaYw>



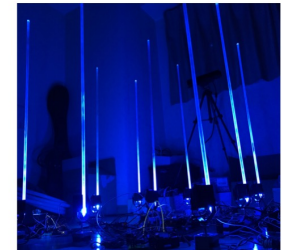
chaotic fidget spinner 2018-19

- Overview
magnetic correlation and chaotic butterfly effect
movement with fidget spinners.
- Exhibit
Maker Faire Tokyo 2018
Maker Faire Bay Area 2019
- Others
<https://www.youtube.com/watch?v=z3SG6RvWbY>
<https://makezine.com/2018/10/11/introduce-some-chaos-into-your-life-with-those-old-fidget-spinners/>



arlms 2017

- Overview
LED robot arm tracking human
movements interaction.
- Exhibit
Santomyuze(Ueda city Culture and Art
Center)
Maker Faire Bay Area 2018
- Others
<https://www.youtube.com/watch?v=JGYQzyZl6jU>



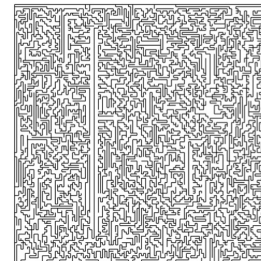
Typing Board 2015

- Overview
keyboard auto typing machine.
- Exhibit
Maker Faire Tokyo 2015
Maker Faire Bay Area 2016
- Others
<https://www.youtube.com/watch?v=FJJPddTjKw>



Brain Maze 2013-14

- Overview
create a maze automatically apps with brain
wave sensor.
- Exhibit
Internet Yami-ichi 2014, 2016
Maker Faire Tokyo 2014
SHUTTER magazine Art Lounge Vol.1
- Others
<https://www.youtube.com/watch?v=w-6oEhJzhnM>



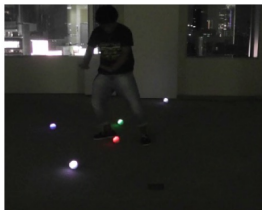
slime lamp 2013

- Overview
combines, bends, and shreds the light by
touching.
- Exhibit
Exploratorium after dark 2018
Maker Faire Bay Area 2013/Taipei
2014(workshop)
- Others
https://www.youtube.com/watch?v=PbZG_UvPccr



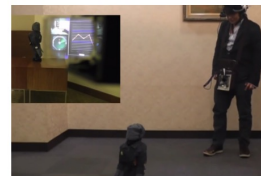
nebula 2012

- Overview
hand gesture controlled picover balls
interaction.
- Exhibit
Fukuoka Asian Art Museum 2013
National Science Museum Thailand 2013
- Others
<https://www.youtube.com/watch?v=x9PizzCYBxs>
Asia Digital Art Award Interactive Art
category prized 2012



Treasure Hunting Robot 2012

- Overview
AR treasure hunting game control robotics
with transparent glass and brain wave sensor.
- Exhibit
Embedded Linux Conference Europe 2012
Maker Faire Bay Area/Taipei/Hong Kong/
ShenZhen 2012
Android Builders Summit 2012
- Others
<https://www.youtube.com/watch?v=GcfXKjuEb4I>



Auto Chasing Turtle 2011

- Overview
By autonomous control, robots
recognizes people's face and
approaches to the detected human.
- Exhibit
Campus Party 2011 Columbia
Google Developer Day 2011 Tokyo
- Others
<https://www.youtube.com/watch?v=8EgfAk5RBVo>



目次

- なにこれ？
 - 目的・開発するプログラムについて
 - RISC-V量子拡張命令セット
 - デモ
-

未踏プロジェクト

2019未踏ターゲット

**RISC-V量子拡張の参照実装と
マイクロ波制御
量子ファームウェアの開発**

2020未踏ターゲット

**量子コンパイラ基盤と
量子計算を体感する
プレゼンテーション**

未踏ターゲット

未踏ターゲット事業

最終更新日：2021年3月22日
IT人材育成センター
イノベーション人材部
未踏実施グループ

未踏事業関連ページへのリンク

[未踏事業ポータル](#) | [未踏IT人材発掘・育成事業](#) | [未踏アドバンスト事業](#) | [未踏ターゲット事業](#) |

未踏ターゲット事業について

「未踏ターゲット事業」は、革新的な次世代ITを活用して世の中を抜本的に変えていけるような先進分野について、基礎技術や領域横断的技術革新に取り組むIT人材に対し、支援を行う事業です。研究開発が進む新たな技術プラットフォームを用いる分野、数学等の専門性を必要とする分野等のなかから事前にテーマを定め、そのテーマに取り組む人材を募集し、プロジェクトマネージャーや関連する企業・大学等と連携して育成、支援を行います。採択者は、自らが提案するテーマの実現を目指すプロジェクトを通して、自身の能力の向上を図ります。

2021年度は「量子コンピューティング技術」をターゲット分野に取り上げます。

公募情報

2021年度未踏ターゲット事業の公募を行っています。

未踏ターゲット事業は、中長期的視点で革新的技術を活用しイノベーションを起こしうる、先進分野のIT人材の発掘・育成を目的として、世界で研究開発が進む先端分野、数学・理学等の専門性を必要とする分野等のなかから事前にターゲット分野を定め、その分野に取り組む人材からのプロジェクト提案を募集します。

2021年度は、2020度に引き続き「量子コンピューティング技術を活用したソフトウェア開発」をターゲット分野として取り上げます。「区分1：アニーリングマシン向けソフトウェア開発」「区分2：ゲート式量子コンピュータ向けソフトウェア開発」の2区分でプロジェクト提案を募集します。

【区分1】 【区分2】 のいずれかを決めかねるプロジェクトの場合は【区分3】として応募してください。

未踏／セキュリティ・キャンプ

＞ 未踏事業

＞ セキュリティ・キャンプ事業



なにこれ？

RISC-V量子拡張の参照実装と マイクロ波制御 量子ファームウェアの開発

量子コンピュータ

量子力学を計算過程に使うことで、古典コンピュータに比べて圧倒的な処理能力を出すことを目的としたもの。



量子コンピュータ凄いの？

多分、特定用途で凄くなるはず…

量子コンピューター



200秒♪

👑 量子超越性の実証

- 同じ計算で競争
- 量子コンピューターが圧倒的に高速
- スパコンで高速計算する方法はなさそう

スーパーコンピューター

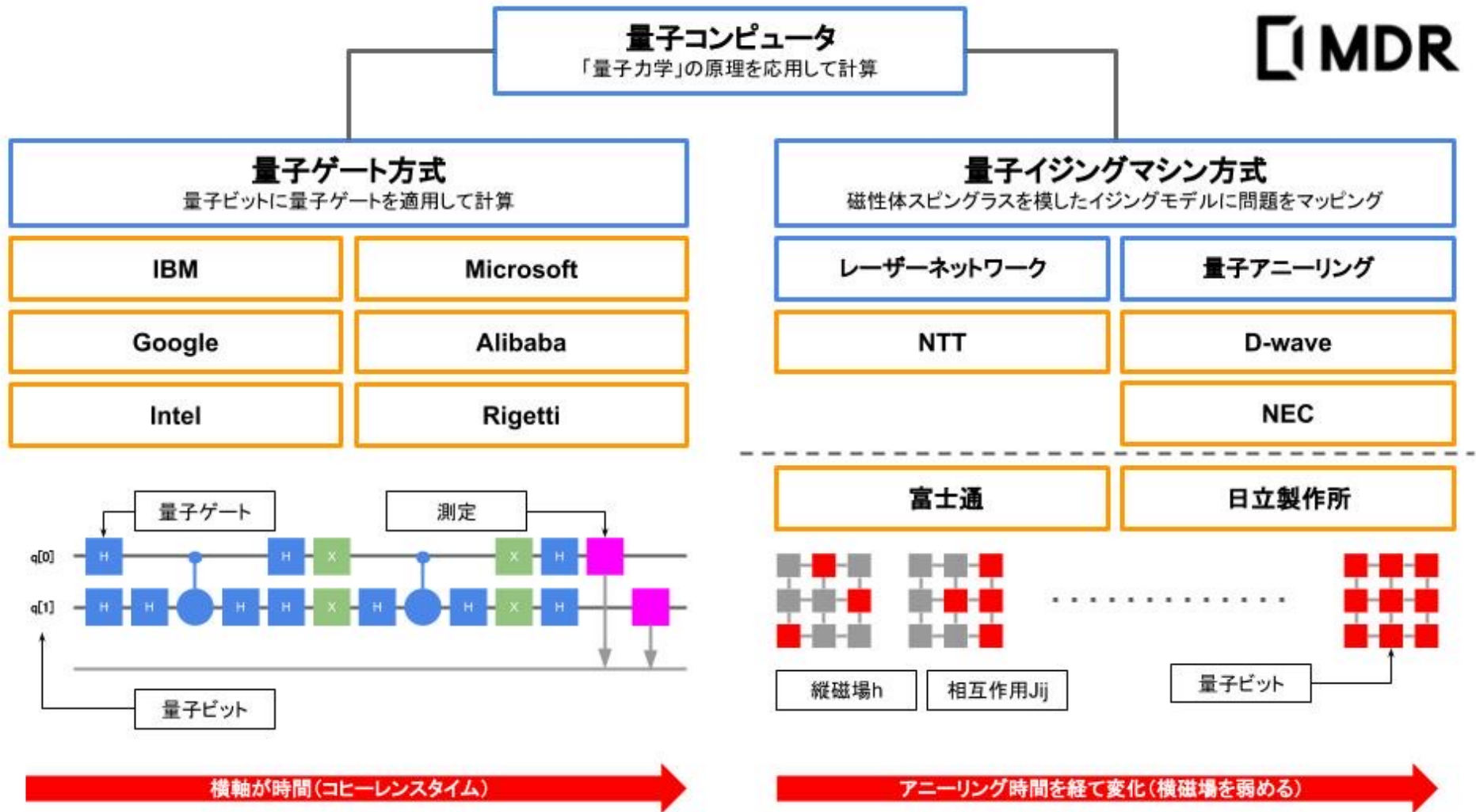
動作をシミュレーション



1万年…

<http://www.tsukuba-sci.com/>
Googleの「量子超越性」実証とは何なのか？

量子ゲート・アニーリングの違い

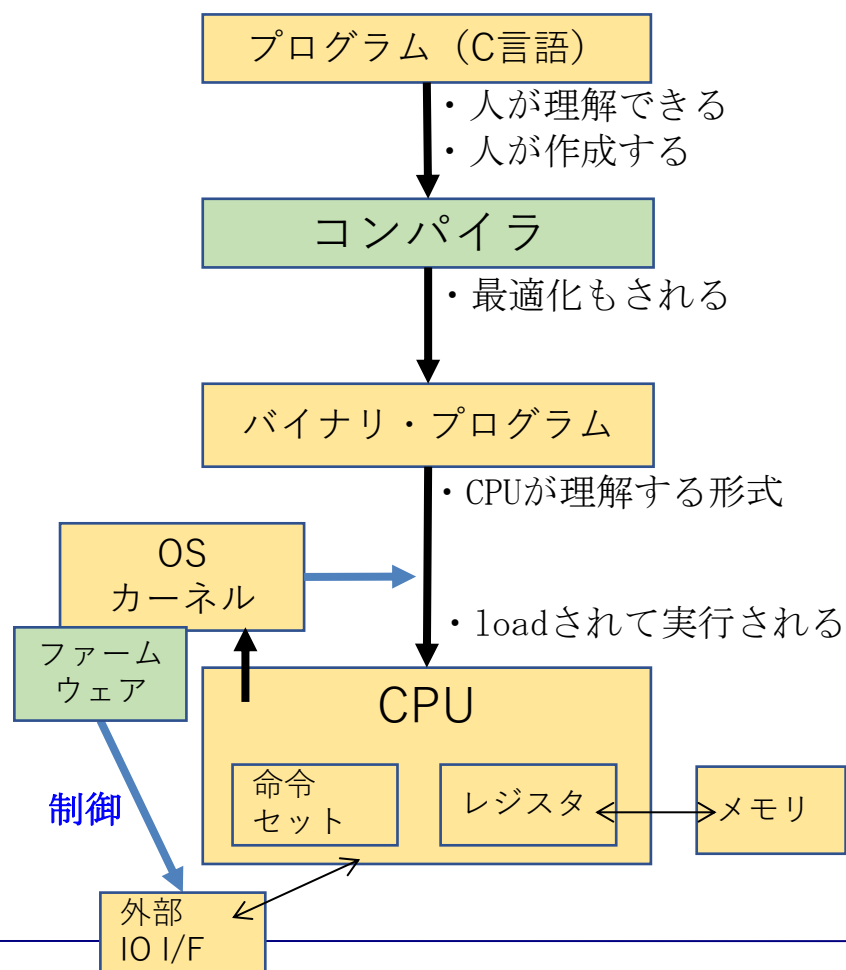


<https://qiita.com/YuichiroMinato/items/2d5ceaa31f819cdc2601> より

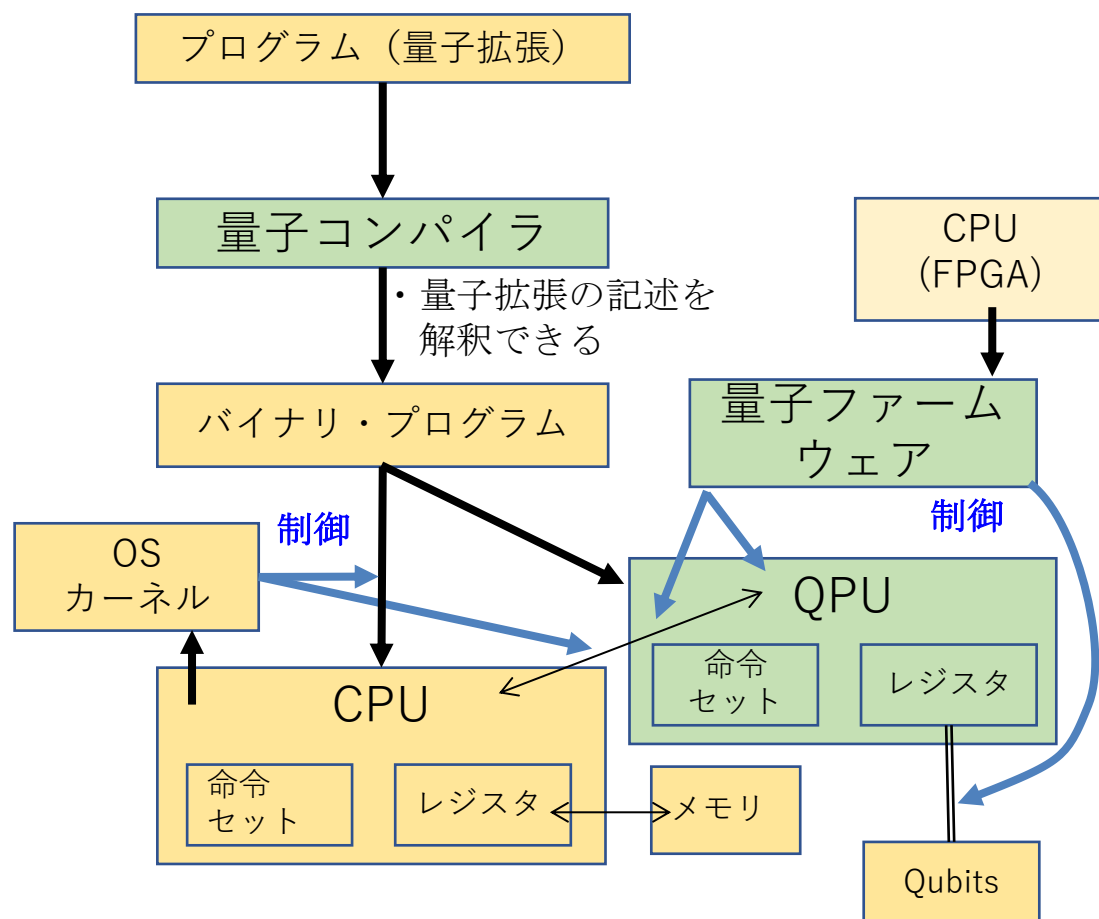
コンピュータ・アーキテクチャの量子拡張

量子計算のアーキテクチャは、古典コンピュータ・アーキテクチャ（左図）にQPUをコプロセッサとして付加した構造（右図）が1つの候補として想定されます。

古典コンピュータ・アーキテクチャ



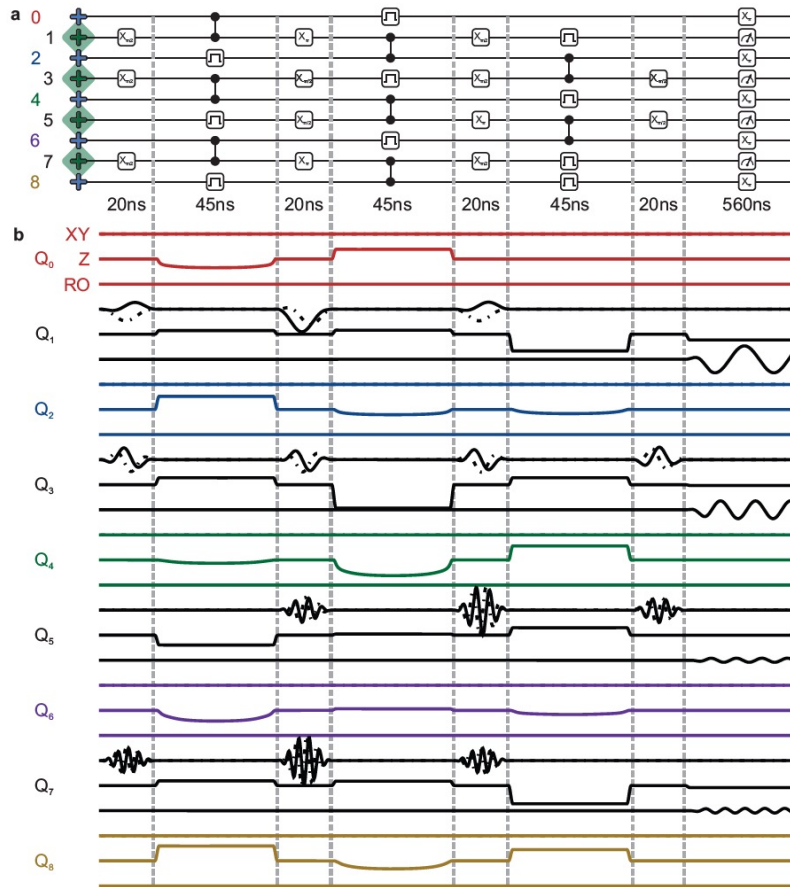
量子コンピュータ・アーキテクチャ



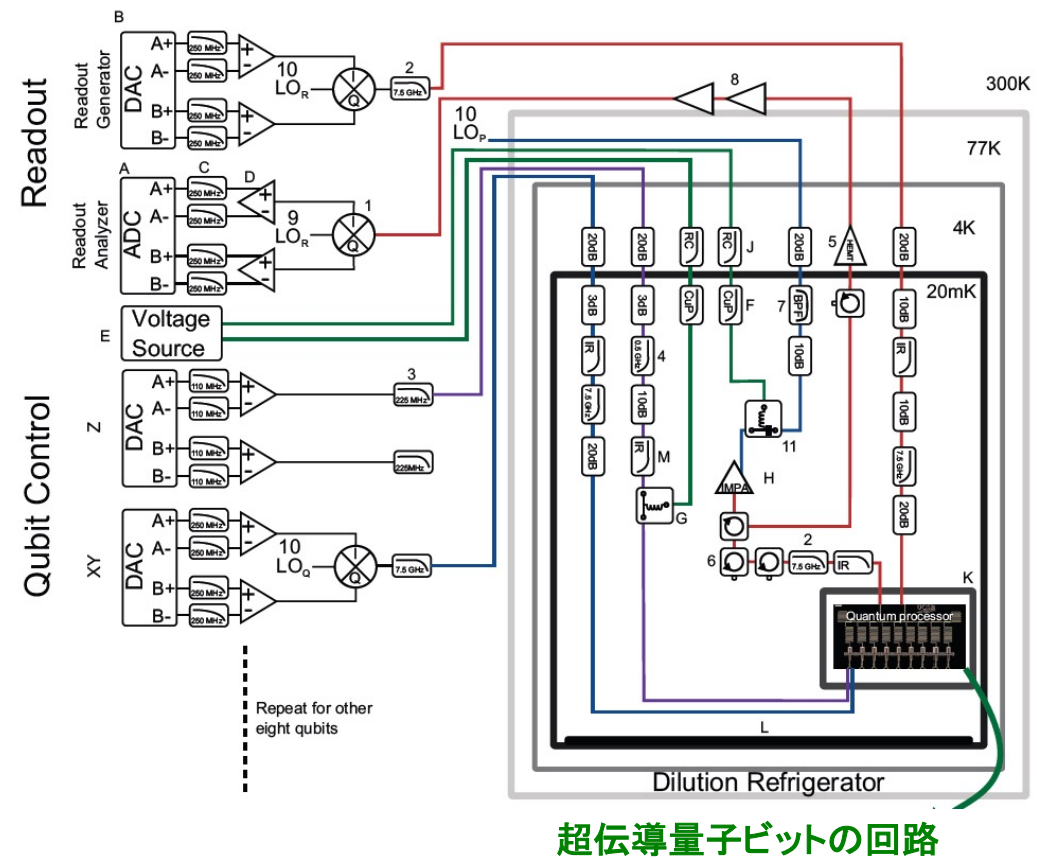
マイクロ波制御

量子ビットとして超伝導量子ビットを仮定すると
「超伝導量子ビット」を操作するには、マイクロ波を使って操作します。

【入力するマイクロ波の例】



【回路の例】

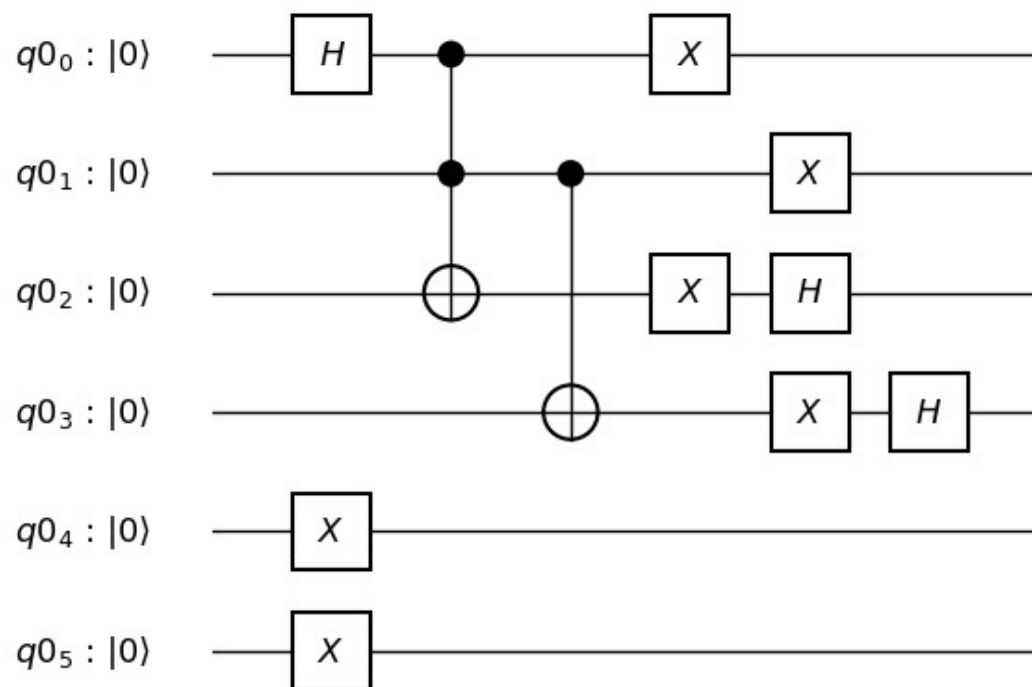


(arXiv:1411.7403 State preservation by repetitive error detection in a superconducting quantum circuit より)

量子プログラミング

量子プログラミング(量子回路 : Qiskit)

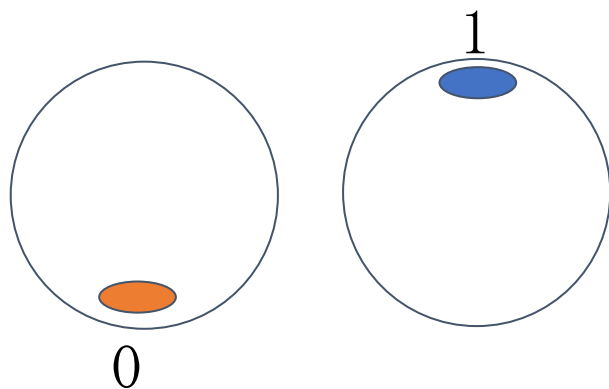
qubitにゲート演算をしていく



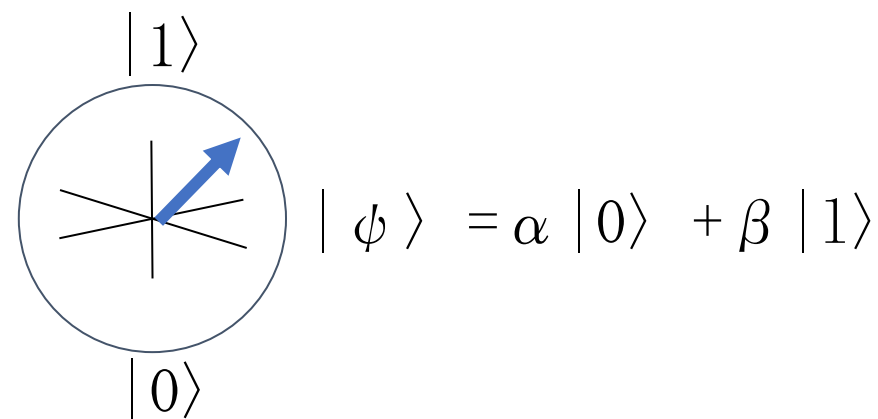
回路？プログラミング？

量子ビットってなに？

古典コンピュータ
ビット



量子コンピュータ
Qubit



ゲートってなに？

古典コンピュータ

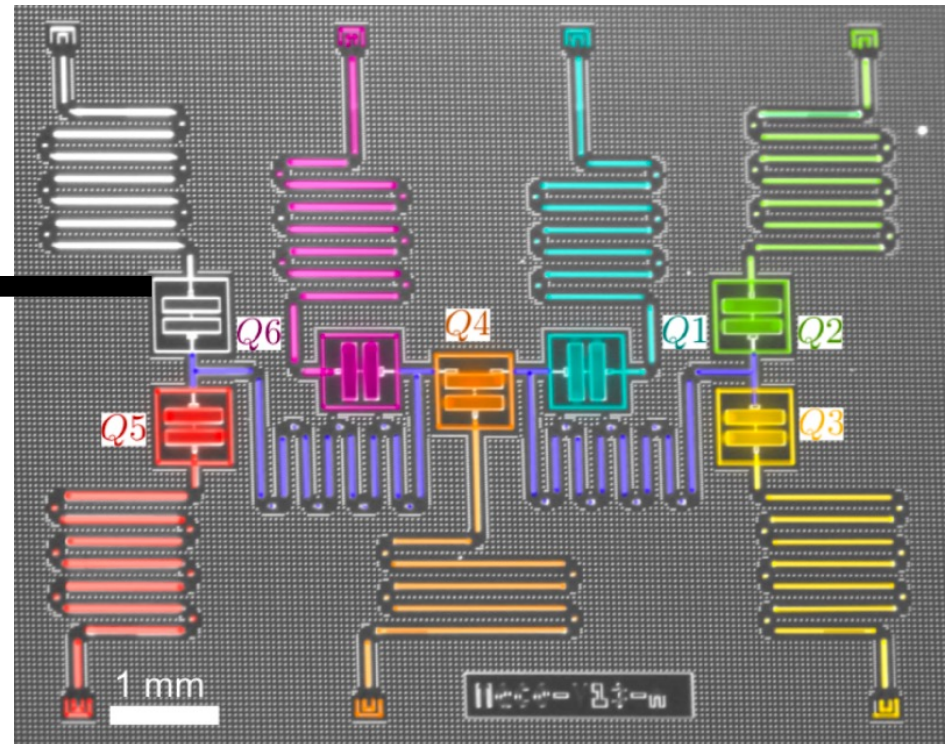
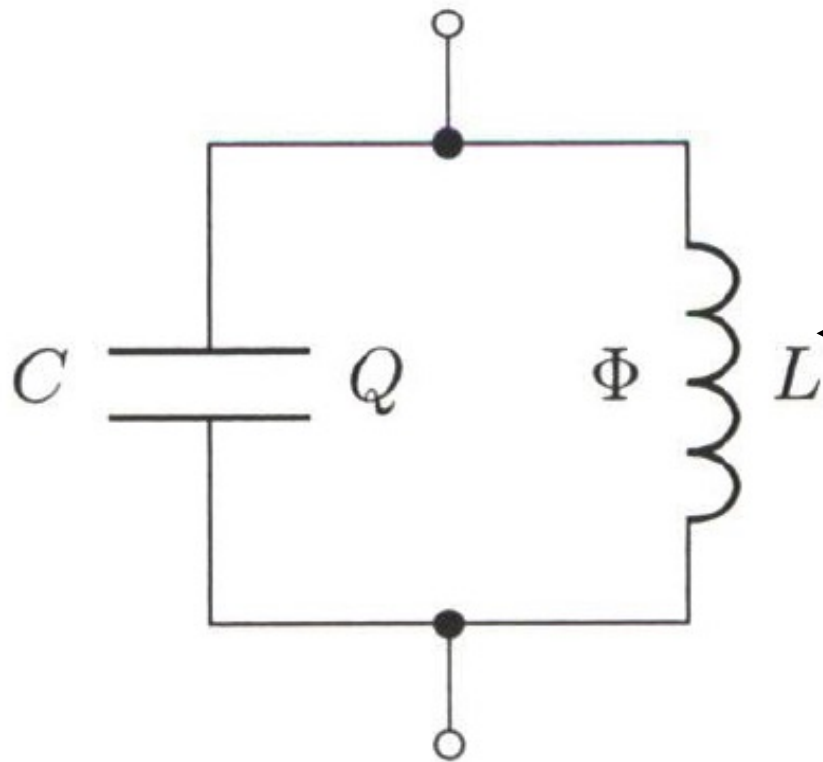
GATE	CIRCUIT SYMBOL	TRUTH TABLE	
NOT The output is 1 when the input is 0 and 0 when the input is 1.		Input	Output
		0	1
		1	0
AND The output is 1 only when both inputs are 1, otherwise the output is 0.		Input	Output
		0 0	0
		0 1	0
		1 0	0
		1 1	1
OR The output is 0 only when both inputs are 0, otherwise the output is 1.		Input	Output
		0 0	0
		0 1	1
		1 0	1
		1 1	1
NAND The output is 0 only when both inputs are 1, otherwise the output is 1.		Input	Output
		0 0	1
		0 1	1
		1 0	1
		1 1	0
NOR The output is 1 only when both inputs are 0, otherwise the output is 0.		Input	Output
		0 0	1
		0 1	0
		1 0	0
		1 1	0
XOR The output is 1 only when the two inputs have different value, otherwise the output is 0.		Input	Output
		0 0	0
		0 1	1
		1 0	1
		1 1	0
XNOR The output is 1 only when the two inputs have the same value, otherwise the output is 0.		Input	Output
		0 0	1
		0 1	0
		1 0	0
		1 1	1

量子コンピュータ

GATE	CIRCUIT REPRESENTATION	MATRIX REPRESENTATION	TRUTH TABLE	BLOCH SPHERE						
I Identity-gate: no rotation is performed.		$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	<table><tr><th>Input</th><th>Output</th></tr><tr><td>$0\rangle$</td><td>$0\rangle$</td></tr><tr><td>$1\rangle$</td><td>$1\rangle$</td></tr></table>	Input	Output	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	
Input	Output									
$ 0\rangle$	$ 0\rangle$									
$ 1\rangle$	$ 1\rangle$									
X gate: rotates the qubit state by π radians (180°) about the x-axis.		$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	<table><tr><th>Input</th><th>Output</th></tr><tr><td>$0\rangle$</td><td>$1\rangle$</td></tr><tr><td>$1\rangle$</td><td>$0\rangle$</td></tr></table>	Input	Output	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$	
Input	Output									
$ 0\rangle$	$ 1\rangle$									
$ 1\rangle$	$ 0\rangle$									
Y gate: rotates the qubit state by π radians (180°) about the y-axis.		$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	<table><tr><th>Input</th><th>Output</th></tr><tr><td>$0\rangle$</td><td>$i 1\rangle$</td></tr><tr><td>$1\rangle$</td><td>$-i 0\rangle$</td></tr></table>	Input	Output	$ 0\rangle$	$i 1\rangle$	$ 1\rangle$	$-i 0\rangle$	
Input	Output									
$ 0\rangle$	$i 1\rangle$									
$ 1\rangle$	$-i 0\rangle$									
Z gate: rotates the qubit state by π radians (180°) about the z-axis.		$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	<table><tr><th>Input</th><th>Output</th></tr><tr><td>$0\rangle$</td><td>$0\rangle$</td></tr><tr><td>$1\rangle$</td><td>$- 1\rangle$</td></tr></table>	Input	Output	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$- 1\rangle$	
Input	Output									
$ 0\rangle$	$ 0\rangle$									
$ 1\rangle$	$- 1\rangle$									
S gate: rotates the qubit state by $\frac{\pi}{2}$ radians (90°) about the z-axis.		$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$	<table><tr><th>Input</th><th>Output</th></tr><tr><td>$0\rangle$</td><td>$0\rangle$</td></tr><tr><td>$1\rangle$</td><td>$e^{i\frac{\pi}{2}} 1\rangle$</td></tr></table>	Input	Output	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$e^{i\frac{\pi}{2}} 1\rangle$	
Input	Output									
$ 0\rangle$	$ 0\rangle$									
$ 1\rangle$	$e^{i\frac{\pi}{2}} 1\rangle$									
T gate: rotates the qubit state by $\frac{\pi}{4}$ radians (45°) about the z-axis.		$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$	<table><tr><th>Input</th><th>Output</th></tr><tr><td>$0\rangle$</td><td>$0\rangle$</td></tr><tr><td>$1\rangle$</td><td>$e^{i\frac{\pi}{4}} 1\rangle$</td></tr></table>	Input	Output	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$e^{i\frac{\pi}{4}} 1\rangle$	
Input	Output									
$ 0\rangle$	$ 0\rangle$									
$ 1\rangle$	$e^{i\frac{\pi}{4}} 1\rangle$									
H gate: rotates the qubit state by π radians (180°) about an axis diagonal in the x-z plane. This is equivalent to an X-gate followed by a $\frac{\pi}{2}$ rotation about the y-axis.		$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	<table><tr><th>Input</th><th>Output</th></tr><tr><td>$0\rangle$</td><td>$\frac{ 0\rangle + 1\rangle}{\sqrt{2}}$</td></tr><tr><td>$1\rangle$</td><td>$\frac{ 0\rangle - 1\rangle}{\sqrt{2}}$</td></tr></table>	Input	Output	$ 0\rangle$	$\frac{ 0\rangle + 1\rangle}{\sqrt{2}}$	$ 1\rangle$	$\frac{ 0\rangle - 1\rangle}{\sqrt{2}}$	
Input	Output									
$ 0\rangle$	$\frac{ 0\rangle + 1\rangle}{\sqrt{2}}$									
$ 1\rangle$	$\frac{ 0\rangle - 1\rangle}{\sqrt{2}}$									

ゲート式超電導量子ビット

(非線形) 並列 LC共振回路 $\equiv$ 量子ビット



A. Kandala et al., Nature 549, 242 (2017)

背景にある課題と目的

■「量子プログラミング」における課題

- 量子コンピュータ上での整数の足し算など、古典計算には普通にあるような操作（量子計算でより抽象的な操作）を記述する方法は、ほとんど確立していない。量子回路でビットごとの操作を記述させられており、将来の複雑なプログラムの記述には耐えられない。

- Q#、QISKit 等ではこの記述を 1 から構築する動きはあるが、長年培ってきた既存のコンパイラと同等の動作を 1 から構築して実現するのは、困難で現実的ではない。

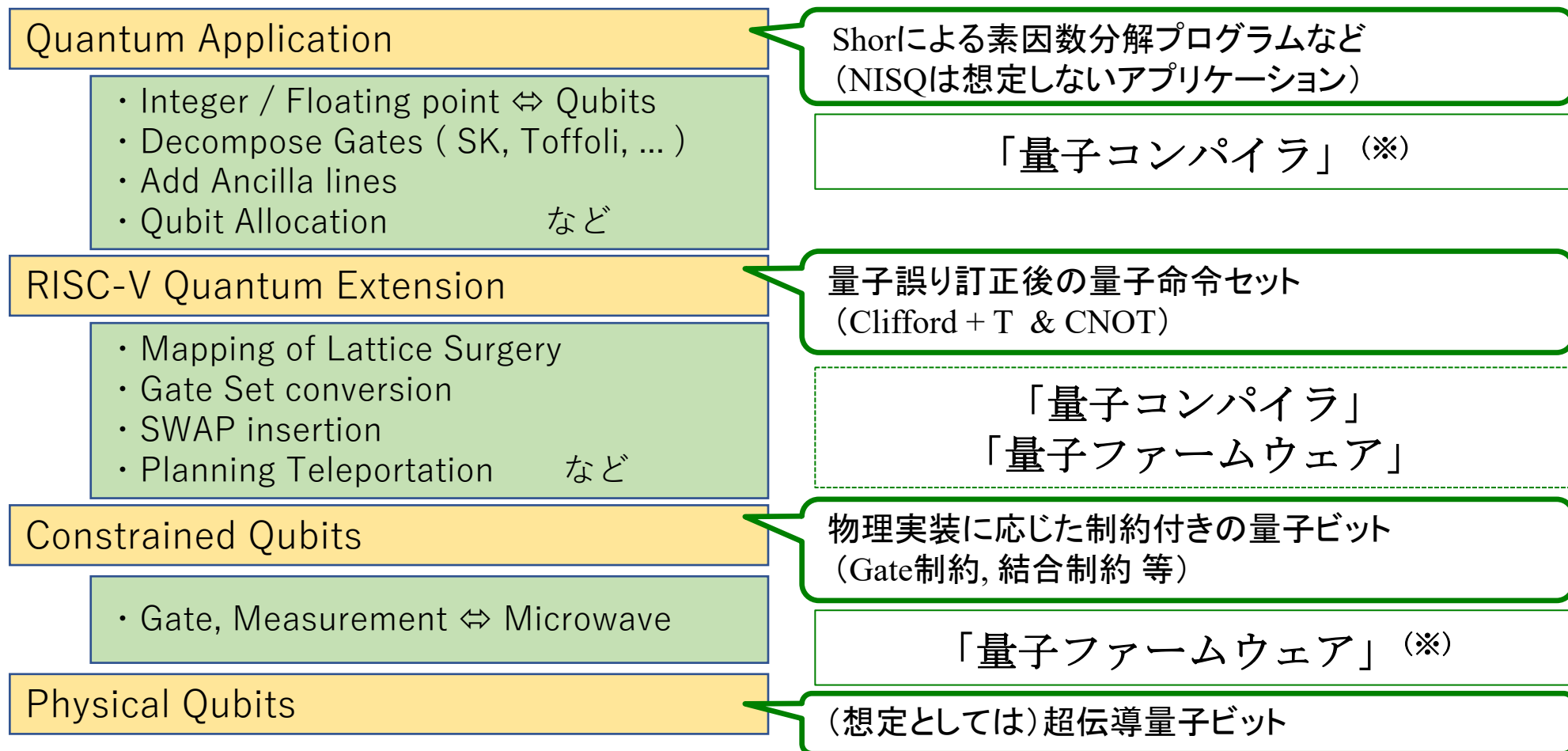
⇒ **既存のアーキテクチャを拡張し、既存の資産を生かすとともに
古典コンパイラの技術者とのつながりを構築する**

■「量子ファームウェア」における課題

- 量子コンピュータの実用化開発において、ハードウェアにより近い階層の開発では、日本は世界から大幅に出遅れているのが現状。

量子コンピュータの階層構造

量子コンピュータの階層構造は研究が進められてる分野で、確定的なアーキテクチャはありません。本PJでは、下図のような構造化を想定したフレームワークを提案します。



※抽象的なコードの最適化は長期的な応用で有効なため、
超伝導量子ビットを用いた量子誤り耐性計算用のコンパイラに焦点を絞って検討します。

新規性と期待される効果

「量子コンパイラ」「量子プロセッサ・シミュレーター」「量子ファームウェア」の開発は世界的には類似コンセプトの研究開発もあります。本PJが、それらと異なる独自性を主張できる点は次のポイントになります。

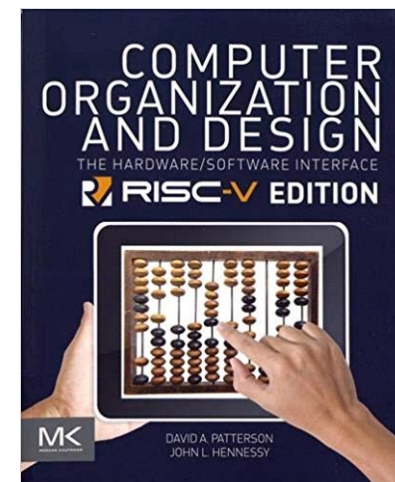
全く新しい仕組みを構築するのではなく、**今まで莫大な蓄積がある古典計算機のコンパイラやマイクロ波制御の枠組みを、量子へ拡張することによって量子計算機の開発環境を作る。**

- ・既存の計算機のコンパイラ、マイクロ波制御の技術者にとって理解しやすく、量子分野への参入や開発がしやすくなることが期待できる。
- ・コンパイラ最適化などの、古典における既存の効率化の技術をそのまま量子へ活用できる可能性がある。
- ・計算機をスケールした際に高い性能を発揮することは歴史的に見て非常に困難な課題だが、現代のアーキテクチャはそのような競争を生き抜いたものである。そのアーキテクチャを踏襲しつつ、量子に拡張することは、量子計算機のアーキテクチャとして提案する上でも有力な方法の1つである。

Why RISC-V

RISC-Vは、オープンソースであるため、数多くの優れたオープンソースソフトウェアがRISC-V に対応しており、**ツールがとても充実**しています。

- gcc(GNUコンパイラ)を含む riscv-gnu-toolchain
- RISC-V Foundation 謹製のシミュレータ spike
- 任意のプログラミング言語に対応可能なコンパイラ基盤 LLVM・オープンソースのプロセッサエミュレータ QEMU 等々



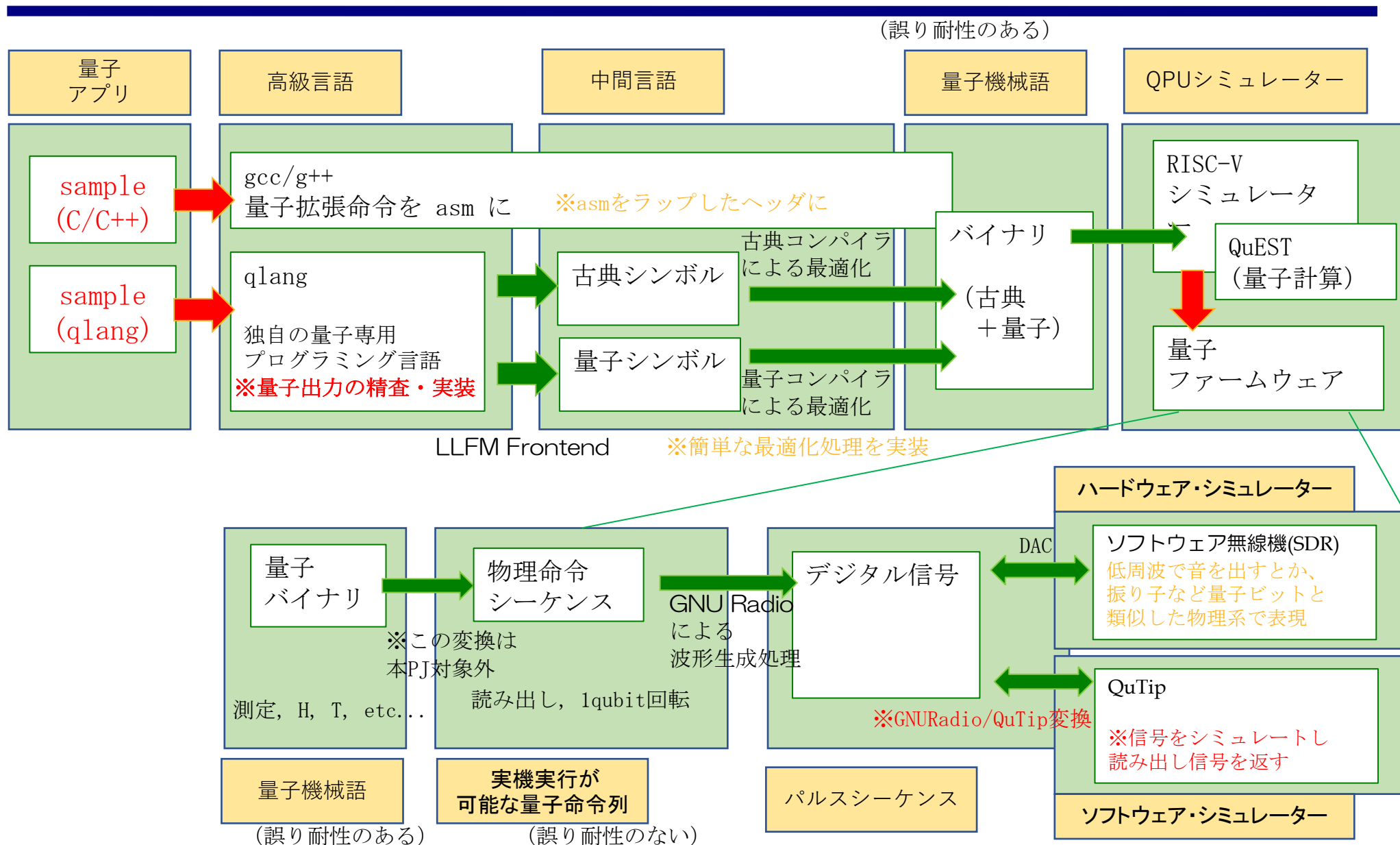
RISC-Vの特徴と本開発のアーキテクチャにRISC-Vを選定した理由

- RISC-Vがシンプルな命令セット・アーキテクチャ (ISA)
- “基本命令”とそれに付随する“拡張命令”によって構成
- RISC-Vは非営利団体「RISC-V Foundation」が管理している標準仕様
- 拡張命令を使った実装が期待できる



- 拡張提案や参照実装が容易
- OSSソフトウェアとも相性が良い

ソフトウェアの全体像



開発したアプリケーション

「RISC-V量子拡張の参照実装とマイクロ波制御量子ファームウェアの開発」として、次のアプリケーションと検証用アプリケーションを開発しています。

(開発している本体のアプリケーション)

RISC-V量子拡張が動作する
参照実装としての

「**量子プロセッサ・シミュレーター**」

Spikeを改造

RISC-V量子拡張のための

「**量子コンパイラ**」

gccを改造、独自言語(qlang)の開発

マイクロ波制御のための

「**量子ファームウェア**」

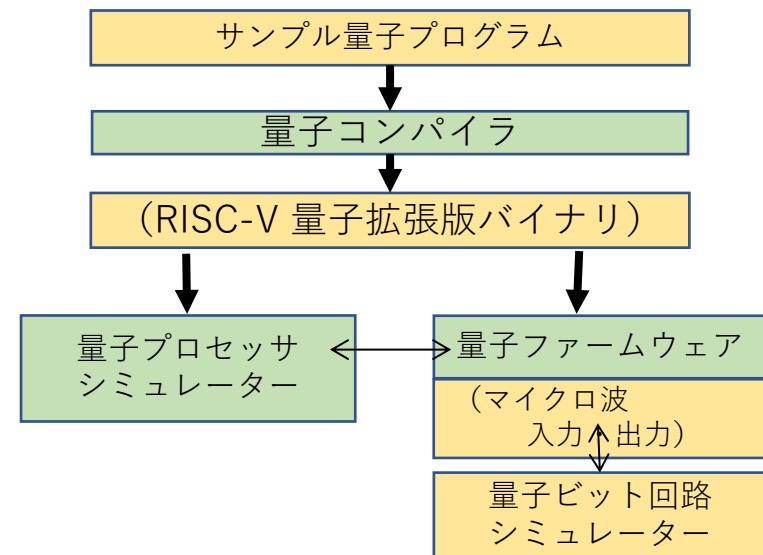
GNU Radio の改造

(検証用アプリケーション)

簡単なサンプルとなる量子プログラム

量子ビット・シミュレーター

(各コンポーネントの関連図)



展望など

■「量子プログラミング言語」の利用促進

RISC-V量子拡張の仕様は、Creative Commons Attribution 4.0 Internationalのライセンスのもと riscv-k-spec という名称で Github にて公開中

■開発したプログラムの次期バージョンへの展開

本アプリケーションは、開発後も継続して次のような具体的なビジョンがあります。

「量子コンパイラ」

- ・NISQに対応するためのISAの追加
- ・QPU profileを定義してPass機構（コンパイラの内部機能）を使う仕組みの開発

「量子プロセッサ・シミュレーター」

- ・RISC-V量子拡張(QPU) on RISC-V
- ・メニーコア実行環境で量子拡張シミュレータの動作

「量子ファームウェア」

- ・研究機関と共同で実際の超伝導型量子ビットに対する制御を試したい
- ・教育用として「視覚的・間隔的なプレゼンテーション」の作成

開発環境

■ OS Ubuntu 18.04/Mac OS, Docker

■ コンパイラ riscv gcc (最新版)

LLVM 10 (最新版)

■ 開発PC Windows/Mac

■ 開発IDE Visual Studio Code/vim

注) gcc/LLVMともにソースからフルビルド(危険)

RISC-V ISA

量子拡張命令セット

RISC-V量子拡張

「RISC-V量子拡張」は、RISC-Vのカスタム拡張領域を使って、現行の仕様から無理なく、拡張定義して、RISC-V量子プロセッサ・シミュレーターを開発、実装。

●量子レジスタ

q0 : 初期化に利用, q1 ... q31 : 32量子ビットの量子レジスタ

●ユニタリ演算を表す命令セット(拡張領域 custom0 のみを利用して拡張)

qmeas.k	rd	rs1	rs2	qimm6	31=1	14..12=0	6..0=0x0b	# quantum measurement
qtelep.k	rd	rs1	rs2	qimm6	31=0	14..12=0	6..0=0x0b	# quantum teleportation
qtocx.k	rd	rs1	rs2	qimm6	31=1	14..12=4	6..0=0x0b	# quantum two-qubit operator CX-gate
qootdg.k	rd	rs1	rs2	qimm6	31=0	14..12=1	6..0=0x0b	# quantum one-qubit operator $T^\dagger$ -gate
qoosdg.k	rd	rs1	rs2	qimm6	31=0	14..12=2	6..0=0x0b	# quantum one-qubit operator $S^\dagger$ -gate
qooz.k	rd	rs1	rs2	qimm6	31=0	14..12=3	6..0=0x0b	# quantum one-qubit operator Z-gate
qoox.k	rd	rs1	rs2	qimm6	31=0	14..12=4	6..0=0x0b	# quantum one-qubit operator X-gate
qoos.k	rd	rs1	rs2	qimm6	31=0	14..12=5	6..0=0x0b	# quantum one-qubit operator S-gate
qoot.k	rd	rs1	rs2	qimm6	31=0	14..12=6	6..0=0x0b	# quantum one-qubit operator T-gate
qooh.k	rd	rs1	rs2	qimm6	31=0	14..12=7	6..0=0x0b	# quantum one-qubit operator H-gate

RISC-V Quantum Extension

<https://quantum.siprop.org/risc-v-quantum-extension/>

RISC-Vシュミレータ(Spike)

定義した命令セットの拡張をRISC-Vシュミレータ Spikeに組み込みました。

● Spikeの起動例

```
spike -k -i192.168.0.2 -s7000 -o7001 -r32 -q4 pk test_qfunc
-k      量子ファームウェア処理ON/OFF(default : QuEST simulator)
-i      量子ファームウェア側アドレス(default: localhost)
-s      量子ファームウェア通信用送信ポート(default : 7000)
-o      量子ファームウェア通信用受信ポート(default: 7001)
-r      量子レジスタ数
-q      量子ビット数
```

● Spike改造ポイント

- RISC-Vエミュレータに量子系命令セットを追加
- 量子用のレジスタをエミュレーター上に定義
- 量子用の演算はプロセッサ毎に実行
- 量子用の演算にはQuESTを利用してシンプル化(量子用外部アクセラレータとしても)
- 通常のgccでアセンブラとしてQ系命令を関数的に実行することで、普通のC/C++言語で作られたプログラムコードに超簡単に量子系の命令を行うことができる

RISC-Vシュミレータ(Spike) ～ 改造ポイント

● Spike改造したソースコード部分など

- ・ RISC-V ISA SIM(Spike) <https://github.com/riscv/riscv-isa-sim>
- ・ 量子エミュレータにはQuEST (<https://quest.qtechtheory.org/>) を組み込み
- ・ わりと簡単に改造できてゴキゲン
- ・ 命令セット拡張・ブログの方で説明
QuESTをRISC-V上で動かす <https://quantum.siprop.org/?p=105>
Quantum compute on RISC-V emulator(Spike) <https://quantum.siprop.org/?p=183>
RISC-V spike vs QuEST <https://quantum.siprop.org/?p=552>

RISC-V GCC大改造

量子プログラミング可能なように、gcc に量子命令セットに対応した拡張を加える。

● gcc の改造

```
riscv64-unknown-elf-gcc -march=rv64imafdkv test.c -o test  
-march k : 量子拡張の追加
```

● gcc 改造ポイント

- gccのコードを読みながら、ほぼ気合で改造
- MPIのようなマクロで量子演算範囲の規定も考えられたが...ヤバそうなのでやめた
- 命令セットの拡張定義 (kオプション)
- 量子レジスタ定義の追加
- 量子命令セット定義の追加
- アセンブリの解釈の実装

RISC-V GCC大改造 ～ 改造のポイント

● gcc 修正ソースコードなど

- ・ RISC-V ISA SIM(Spike) <https://github.com/riscv/riscv-gnu-toolchain>
- ・ ビルド&テストで時間がすげーかかった...ビルド時間が長すぎる...
- ・ ブログの方で説明
RISC-V Quantum Extension <https://quantum.siprop.org/?p=770>

量子コンパイラ (C/C++)

GNU Toolchain(GCC) コンパイラを改造して、RISC-V量子拡張命令セットをインラインアセンブラで直接記述して実行できるようにしています。

関数化までは行っていません。

既存のC/C++プログラムに量子プログラミングを直接コーディングする(例)

```
#include <stdio.h>

int main() {
    int i, j, ret;

    ... (古典プログラミング処理) ...

    asm volatile(
        "qooh.k    qa0, qt1,qzero, 1;"
        "qmeas.k  %0, qt1, qzero, 1;"
        : "=r"(ret) : );

    printf("%d\n", ret);
    return 0;
}
```

古典プログラミングにH-gate、Measurementを直接記述した例。
既存の古典プログラミング資産(C/C++プログラム)に手を加えることなく、容易に量子回路を組み込めるようにしています。

量子コンパイラ (qlang)

フロントエンドとしてLLVMを利用した言語 qlang を開発。

GO-Likeな独自言語で動作させようとしている量子プログラム(例)

```
func main() {  
  qint x, y;    // primitiveな型  
  var z;  
  
  x = 0;  
  y = 1;  
  x = y;        // quantum teleportation  
  
  if x == 1 {   // Implicit Measurement  
    write x;  
  }  
}
```

```
x = 1;  
  
while x <= 10 { // Implicit Measurement  
  write x;  
  x = x + 1;  
}  
  
if x <> 1 {      // Implicit Measurement  
  write x;  
}
```

既存の記述を拡張しつつ、量子に実行できないものは読み替えやコンパイルエラーを出すなど
人間にとって直感的で、意図しないミスがなく実行可能なプログラムを開発するプログラミングを可能にします。

量子コンパイラ (qlang)

● コンパイル

```
qlang -O[0-3] example/test.q
```

● gcc 修正ソースコードなど

- ・ フロントエンドを対象
- ・ バックエンドは改造版gcc
- ・ LLVMサイコーに便利(ビルドやばい)

- ・ ブログの方で説明

LLVMを使う <https://quantum.siprop.org/?p=802>

LLVM最適化パスを使ったゲート最適化 <https://quantum.siprop.org/?p=878>

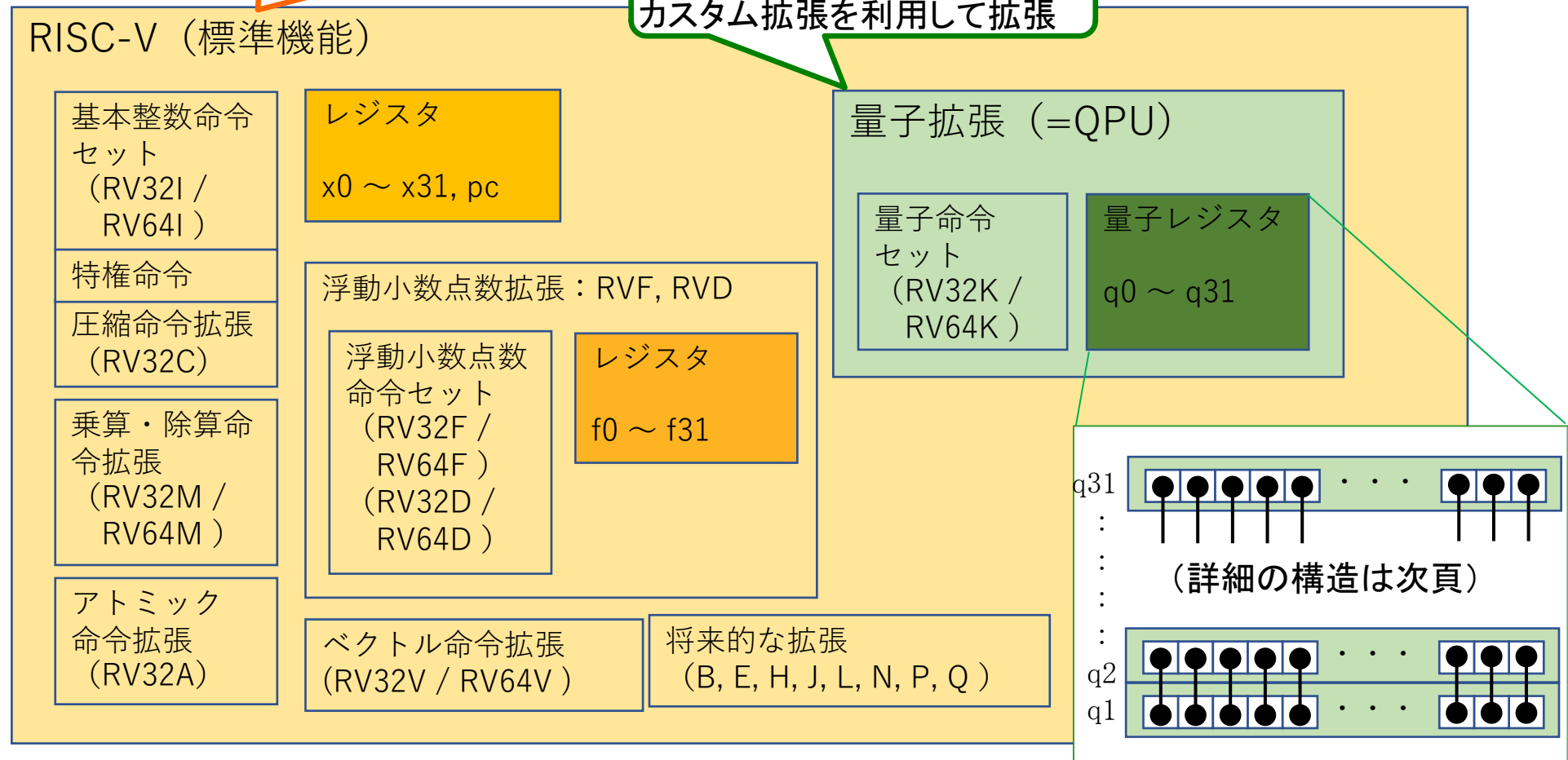
量子プロセッサ・シミュレーター

(「QPUシミュレーター」とも呼びます)

RISC-V量子拡張された量子プロセッサを仮想マシン(Virtual Machine, VM)として開発し、
「RISC-V量子拡張が動作する参照実装」をDockerで用意

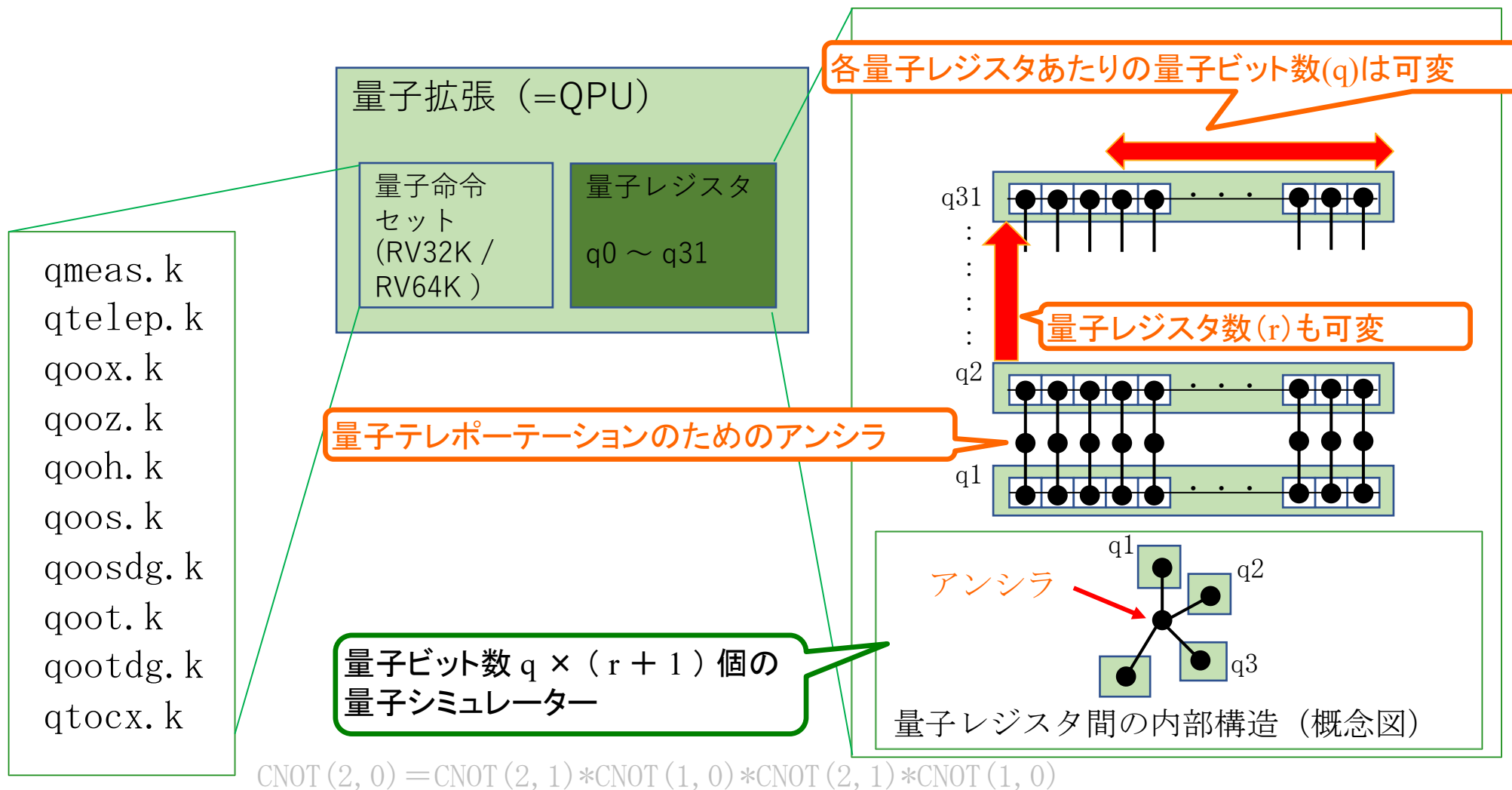
RISC-Vの標準機能はそのまま

RISC-Vの標準機能にある
カスタム拡張を利用して拡張



RISC-V量子拡張の構造

実装は、RISC-Vの純正のシミュレーター「Spike」を改造しました。
量子計算の実装は、既存のOSSライブラリ「QuEST」を組み込みました。



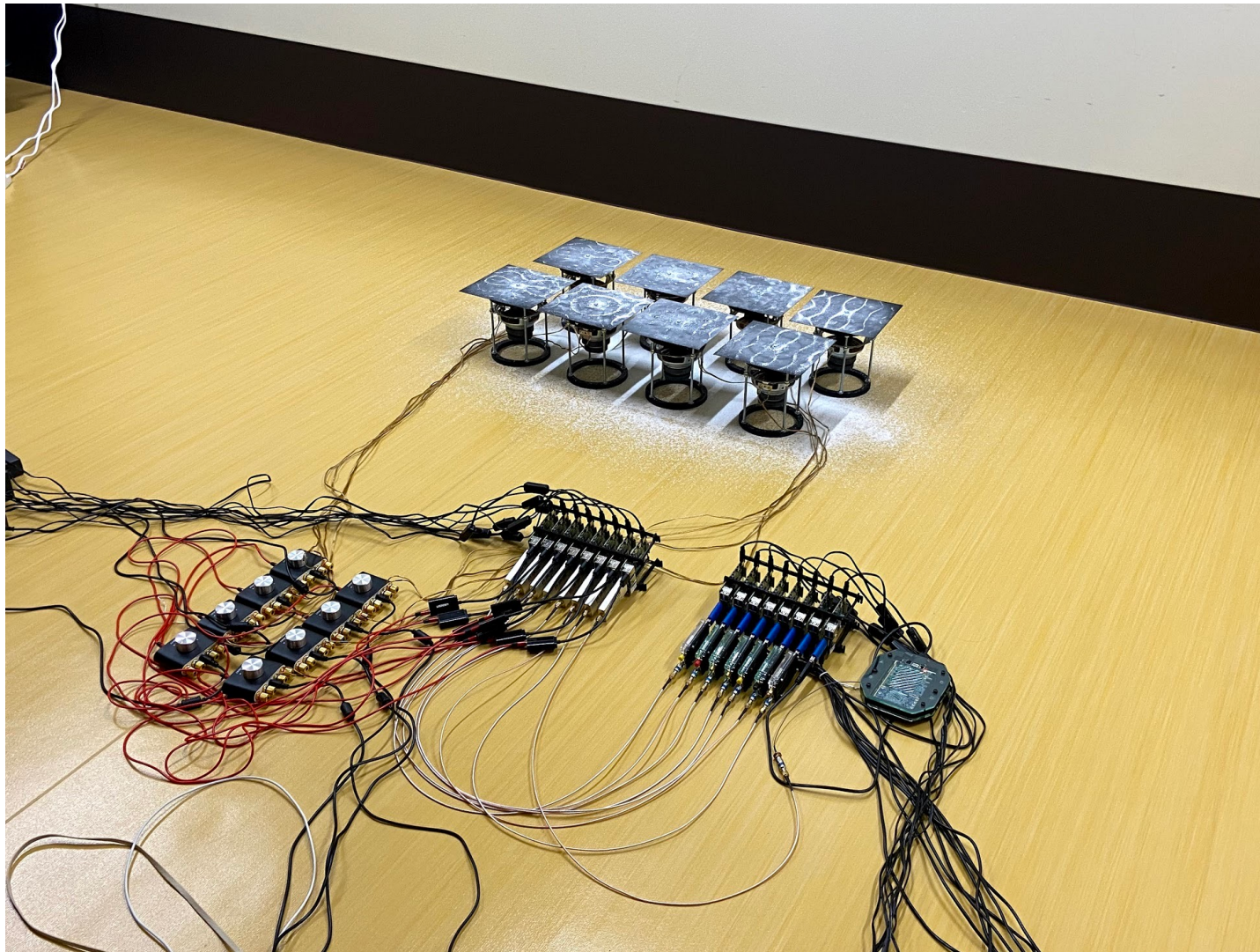
デモ

なにこれ？

量子コンパイラ基盤と 量子計算を体感する プレゼンテーション

超電導量子コンピュータの 動作原理でもある、パルスを用いたアプリケーション

8bit

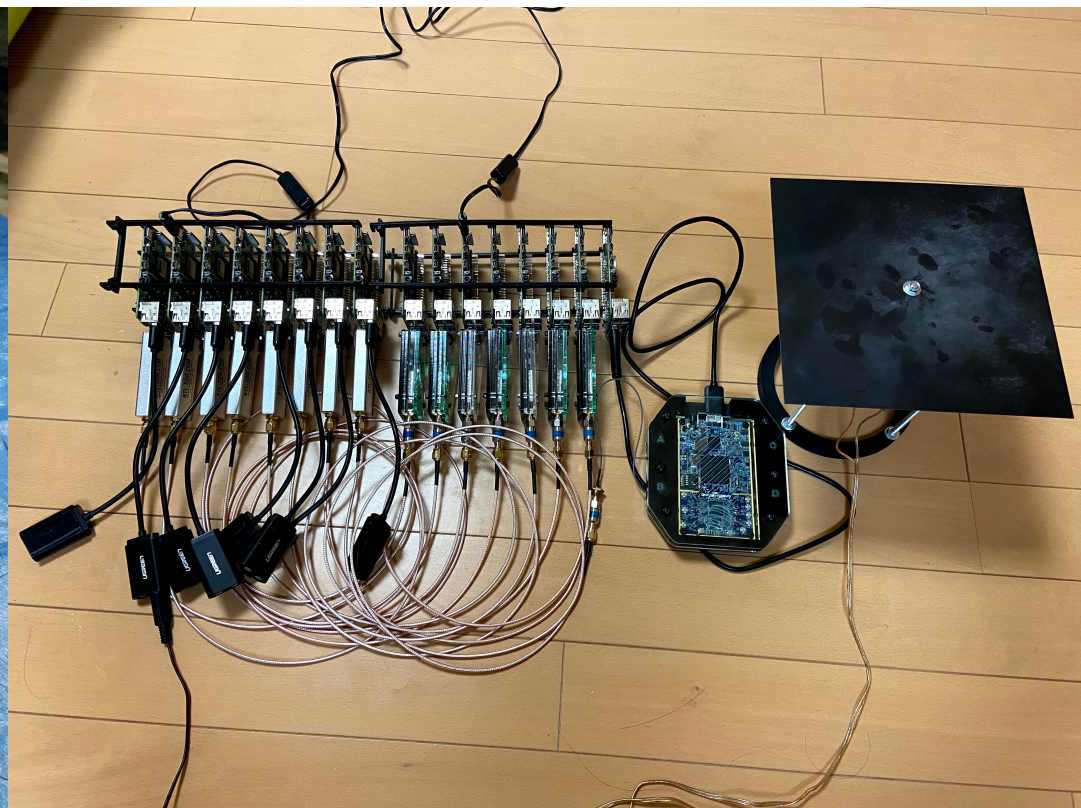


動画 <https://www.youtube.com/watch?v=ts4WP3nBXMc>
制作内容 <https://quantum.siprop.org/?p=1433>

制作



インターテスラ社の工場にて初回テスト



自宅でひたすらテスト

展示



Maker Faire Tokyo 2020 にて

thank you