【悲報】量子コンピュータ、九九が解けない



自己紹介

- さめ (meg-cck)
  - ⚛️ VRChat物理学集会の主催
  - 👨‍🎓 社会人学生として通信制大学在学中
- 得意分野:
  - 📸 コンピュータビジョン（画像認識/点群処理）
  - 🌍 空間情報処理 （地理情報/リモートセンシング）
  - ☁️ クラウドインフラ設計/IaC (AWS, GCP)
- 学生時代は地球物理学を専攻
- 地球観測技術のエンジニアとして活動中



量子コンピュータの噂

- **素因数分解**: 数を素数の掛け算に分けること
  - 例: 15 = 3 × 5
- 掛け算は一瞬。でも**逆向きは難しい**
  - 巨大な素数 2 つを掛けた数 (例えば 617 桁) を分解するのは、 今のコンピュータでは現実的な時間で終わらない

> 「量子コンピュータなら、巨大な数の素因数分解も**あっという間に解ける**」と言われている

RSA 暗号と素因数分解

- **RSA 暗号**: Web の HTTPS (サーバ証明書の署名) などで広く使われている
- 鍵の材料は巨大な合成数 $N = p \times q$
  - N は**全世界に公開**、素因数 p と q は**秘密**
  - N を素因数分解できれば、暗号は破れる

- 安全性の根拠: 古典コンピュータで**効率よく素因数分解する方法が見つかっていない**こと
- 「不可能」と証明されたわけではない

量子で暗号が破られる？

- **正当な部分**
  - Shor のアルゴリズムは RSA を**原理的には**破れる
  - 今の通信を保存して将来解読される心配がある
  - 耐量子暗号への移行が始まっている (米国 NIST が 2024 年に標準化)
- **まだ先の部分**
  - 2048 bit の RSA には、誤りのある qubit が**約 100 万個**必要 (2025 年の見積もり)
  - 今日使う実機は **20 qubit**

実は、誰でも借りられる

- **Amazon Braket** (AWS) で本物の量子コンピュータを使える
- 例: **IQM Garnet** (超伝導、20 qubit)
  - 実行 1 回 0.30 USD + 1 ショット 0.00145 USD
  - 3000 ショットで **4.65 USD**

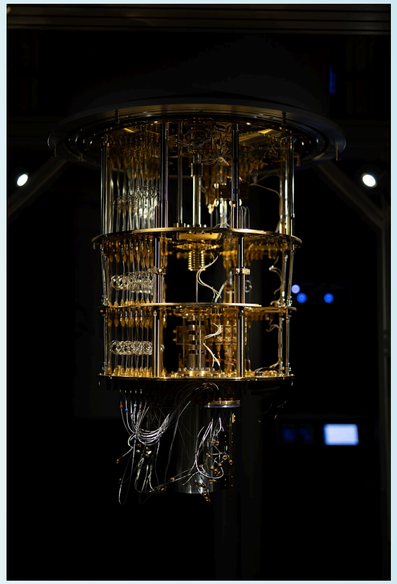


写真: Ragsxl, CC BY-SA 4.0, via Wikimedia Commons

> 噂のすごいやつを**自分の手で動かしてみたい**

じゃあ、お金を払えば
巨大な数も分解できる？

ところがどっこい

15＝3×5すら

解けませんでした

> 正確に言うと: 正しい答えに辿り着く割合で、**でたらめな数を返すだけの装置 (サイコロ) に負けた** (実機 69.5% 対 サイコロ 75%)

今日話すこと

- 素因数分解の量子アルゴリズム **Shor のアルゴリズム**を、 **本物の量子コンピュータ**で動かしてみた話
  - 題材は 15 = 3 × 5 (2001 年の最初の実験以来の定番)
  - 超伝導量子コンピュータ **IQM Garnet** にクラウド経由で投げる
  - 結果を**事前に書いておいた予測**と比べる

- **どう、どれだけうまくいかなかったか**を数字で確かめる
- **排出率0%のガチャ**を回して爆死してしまったのを笑いのネタにして成仏させる

今日話さないこと

- 量子誤り訂正、RSA 暗号がいつ破られるか
- 量子計算や Shor のアルゴリズムの厳密な説明
  - 位相推定や連分数展開の証明
- クラウド側のインフラ設計
  - 権限の分離、課金のガードレール、Terraform
  - ぶっちゃけるとここに一番手間と時間がかかってるのですが……

> 実装と判断の記録は GitHub のリポジトリと Wiki で公開 (最後にリンク)

量子コンピュータは何が違う？

超ざっくり版

古典コンピュータのゲート

- **ビット** (0 か 1) を**論理ゲート**で変換する
  - 例: **NAND** は「両方 1 のときだけ 0」
  - NAND を組み合わせれば、**どんな計算も作れる**

入力 A	入力 B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

> 2 つ入れて 1 つ出す → 出力 1 から入力は**戻せない**
> (情報を捨てている)

量子コンピュータのゲート

- **qubit** は 0 と 1 の**重ね合わせ** $\alpha|0\rangle + \beta|1\rangle$ を取れる
  - α と β は**振幅**。波の高さと位相のようなもの
- **量子ゲート**は振幅を回す操作
  - 例: **H ゲート**は $|0\rangle$ を 0 と 1 が半々の重ね合わせにする

	古典	**量子**
単位	ビット (0 か 1)	qubit (重ね合わせ)
ゲート	NAND など。入力を捨てる	H、CNOT など。逆回しできる
読み出し	何度読んでも同じ	測ると確率的に 0 か 1

測定1回=1ショット

- 回路を流して測ると、0 と 1 の列が **1 本**出る。これが **1 ショット**
  - どの列が出るかは**確率**で決まる
- 分布を知るには、**同じ回路を何千回も**流して頻度を数える
  - ショットが多いほど頻度のばらつきは小さい
- 実機は**ショット単位で課金**される

> 返ってくるのは**分布からのサンプル**。分布はショットを重ねてはじめて見える

重ね合わせと干渉

- n 個の qubit で、2^n 通りの入力の重ね合わせを**まとめて**扱える
- ただし測ると**答えは 1 つだけ**
  - 「全部同時に計算して全部読む」わけではない
- 振幅は波のように**干渉**する
  - 正解が強め合い、不正解が打ち消し合うように組む

- この解き方が効く問題は限られている
- 一番有名なのが素因数分解の **Shor のアルゴリズム** (1994 年)

SHOR のアルゴリズム

因数分解を「周期探し」に言い換える

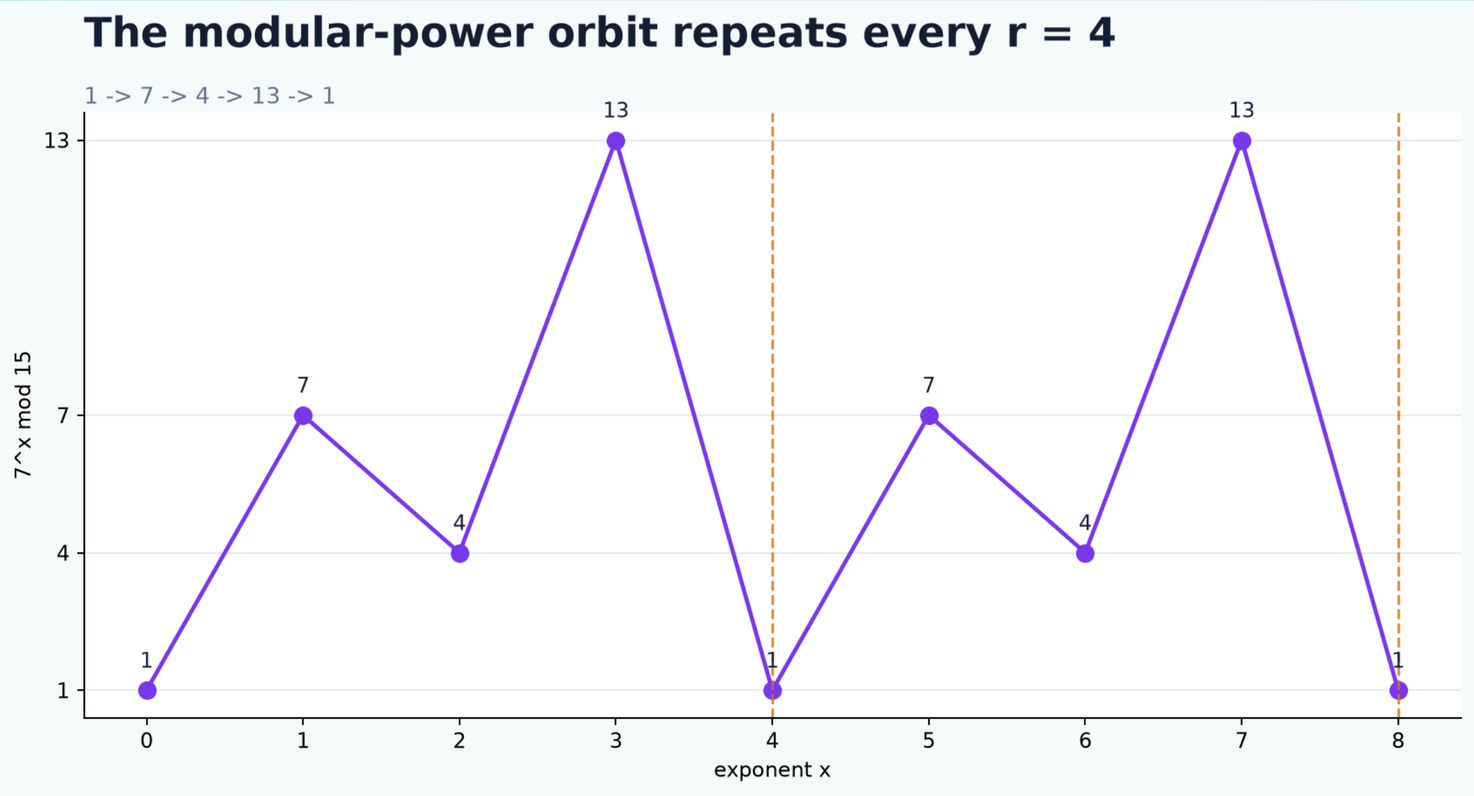
7 を掛けて 15 で割り続ける

- 15 と共通の約数を持たない数を選ぶ (今回は **7**)
- 7^x を 15 で割った余り ($7^x \bmod 15$) を計算する
  - 次の余りは「前の余り × 7」の余り ($4 \times 7 = 28 \to 13$)

x	**0**	**1**	**2**	**3**	**4**	**5**	**6**
7^x	1	7	49	343	2401	16807	117649
15 で割った余り	**1**	7	4	13	**1**	7	4

余りは 1 → 7 → 4 → 13 → 1 → … と **4 つごとに繰り返す**。この長さを**周期** r と呼ぶ ($r = 4$)

隠れた周期を探す



因数分解の問題は、この**周期を見つける問題**に言い換えられる (次のスライドから)

周期から「掛け算の形」を作る

- 周期が $4 \to 7^4$ を 15 で割った余りは 1
- つまり $7^4 - 1 = 2400$ は **15 で割り切れる** ($2400 = 15 \times 160$)
- 中学で習う因数分解 $A^2 - 1 = (A-1)(A+1)$ を、$A = 7^2 = 49$ で使う

$$7^4 - 1 = (7^2 - 1)(7^2 + 1) = 48 \times 50$$

> 15 の倍数を、**2 つの数の掛け算**に分けられた

48 も 50 も 15 の倍数ではない

- 48×50 は 15 の倍数。でも 48 も 50 も**単独では違う**
- → 15 の素因数が 48 と 50 に**分かれて**いる
  - 実際 $48 = 3 \times 16$、$50 = 5 \times 10$
- **最大公約数** (gcd) で取り出す

$$\gcd(48,\ 15) = 3, \qquad \gcd(50,\ 15) = 5$$

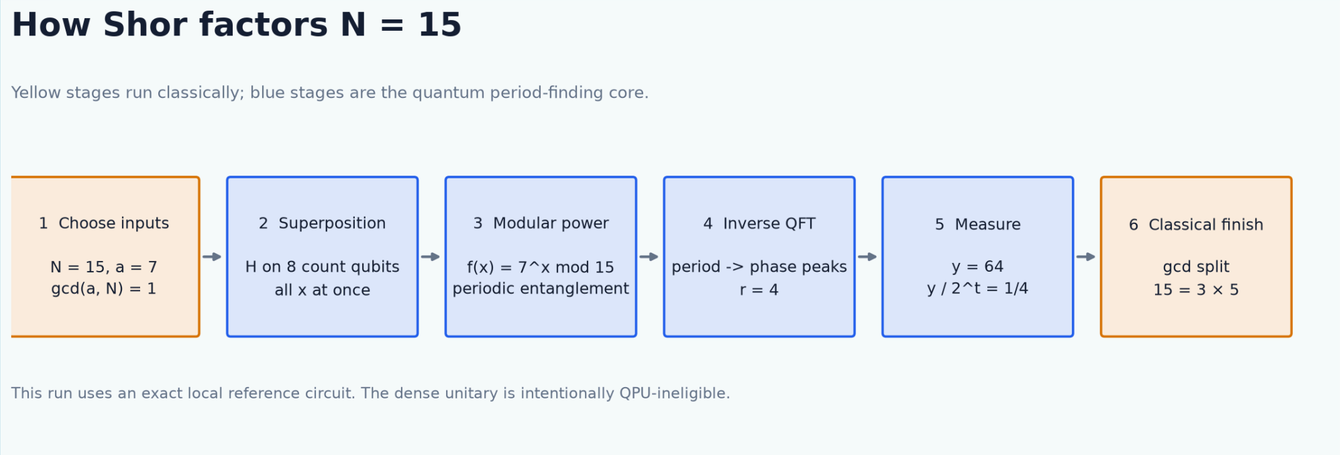
> 15 の素因数を知らなくても、**最大公約数を計算するだけ**で取り出せる

最大公約数は一瞬で計算できる

- **ユークリッドの互除法**: 割り算の余りを取り続けるだけ。600 桁でも一瞬 $\to O(\log \min(a, b))$
  - $50 = 3 \times 15 + 5$、$15 = 3 \times 5 \to$ gcd は **5**
  - $48 = 3 \times 15 + 3$、$15 = 5 \times 3 \to$ gcd は **3**
- 周期が奇数のときや片方が 15 の倍数のときは失敗
  - $\to$ 7 以外の数で**やり直す** (適当に選べば半分以上の確率で成功)

- 難しいのは**周期を見つける**ところ
  - (理想的な)量子コンピュータはこれが得意

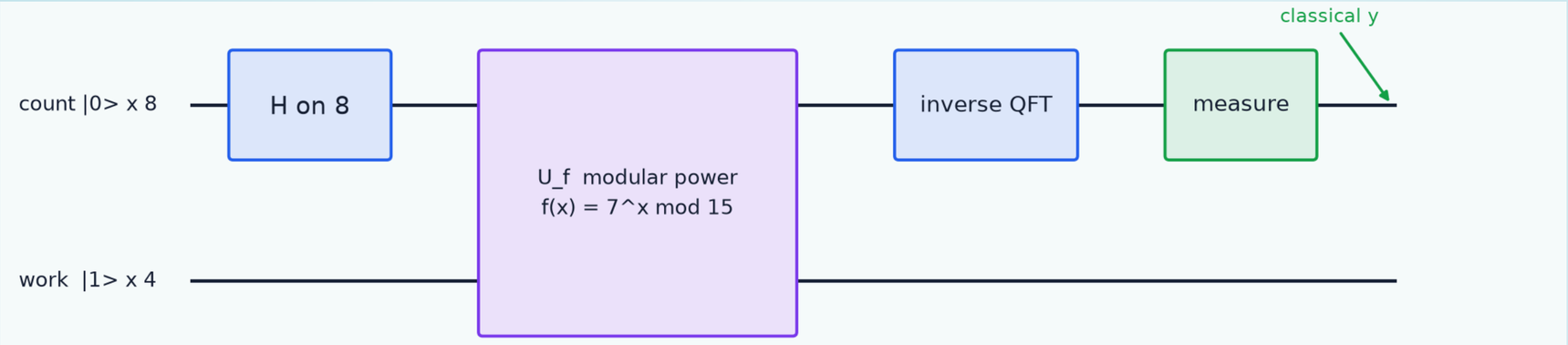
量子の担当は「周期探し」だけ



黄: 古典、青: 量子。量子が直接「3 と 5」を出すわけではない

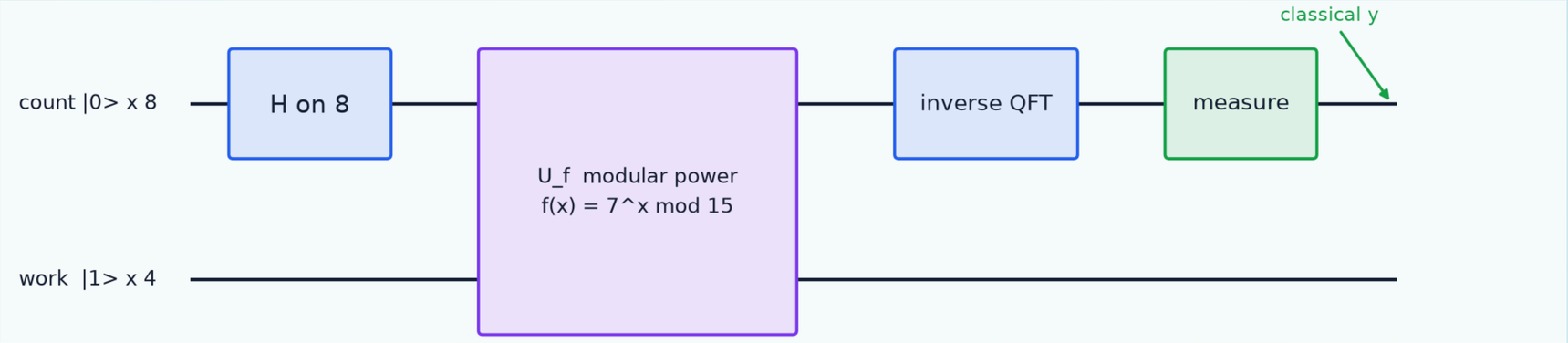
ステップ	やること	量子だと何がうれしいか
2: 重ね合わせ	x = 0〜255 を重ねる	256 通りを 8 ゲートで
3: べき乗の余り	余り $7^x \bmod 15$ を書く	全部の x をまとめて
4: 逆 QFT	縞 → ピーク	干渉で**周期だけ**読む

指数を入れる箱



- **count** (8 qubit、図の上の線): 指数 x を入れる
  - 8 qubit で 0〜255 の 256 通り
- H → 掛け算のスイッチ → 逆 QFT → **ここを測る**
- count / work は実装での呼び名 (教科書では第 1 / 第 2 レジスタ)

余りを書く箱



- **work** (4 qubit、図の下の線): 余り $7^x \bmod 15$ を書く。最初は 1
- 余りで x を消すと逆回しできない → **別の箱**へ
- 2 つの箱の状態は (count, work) の**表**。各マスに**振幅**が乗り、測ると **1 マス**だけ出る

> 量子の仕事: **どのマスを引いても役立つ表**に整える

ステップ 2: 重ね合わせ



横: count の値 x (0〜255)、縦: work の値 (使わない 12 行は省略)。
明るいほど確率が高い

- count の 8 qubit に H ゲートを 1 個ずつ
- work = 1 の行の **256 マスに同じ振幅** (確率 1/256)
  - いま測ると 0〜255 のどれかが均等に出る

- 古典: 256 通り並べるには 256 回か 256 台
- 量子: **ゲート 8 個**で 256 マス。1 qubit 足すと 2 倍

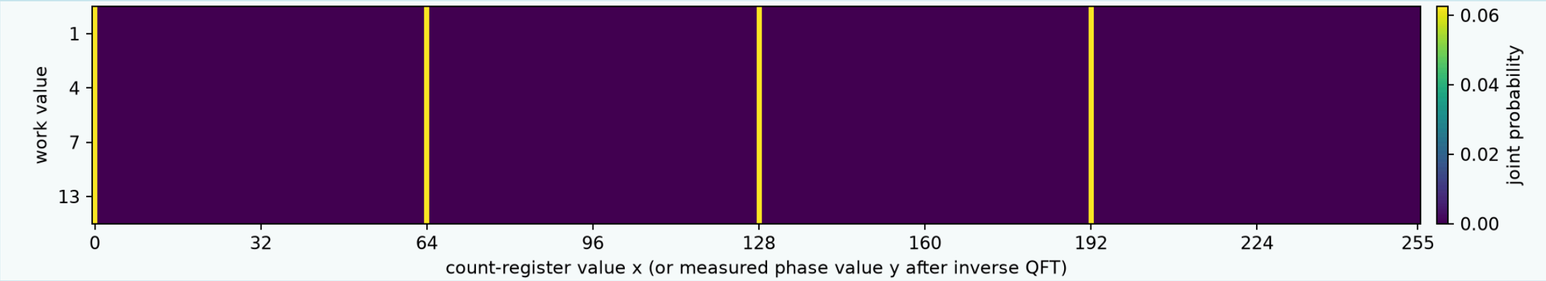
ステップ3: べき乗の余り



- count の各 qubit を**スイッチ**に、work へ $7, 7^2, 7^4, \ldots$ を掛ける
- 各マスが余りの行へ → **ずれた 4 つおきの縞**
- 2 つの箱の状態を**別々に言えない → エンタングルメント**

いま測っても (x, 余り) の組が 1 つ出るだけ。**古典で計算したのと同じ**

ステップ4: 逆フーリエ変換



- count に**逆量子フーリエ変換** (逆 QFT): 縞の「間隔」をピークの「位置」に変える
- 振幅が**干渉**: 0, 64, 128, 192 で強め合い、ほかは打ち消し合う
- 縞の**ずれ**は位相に逃げる → **どの行も同じ 4 列**

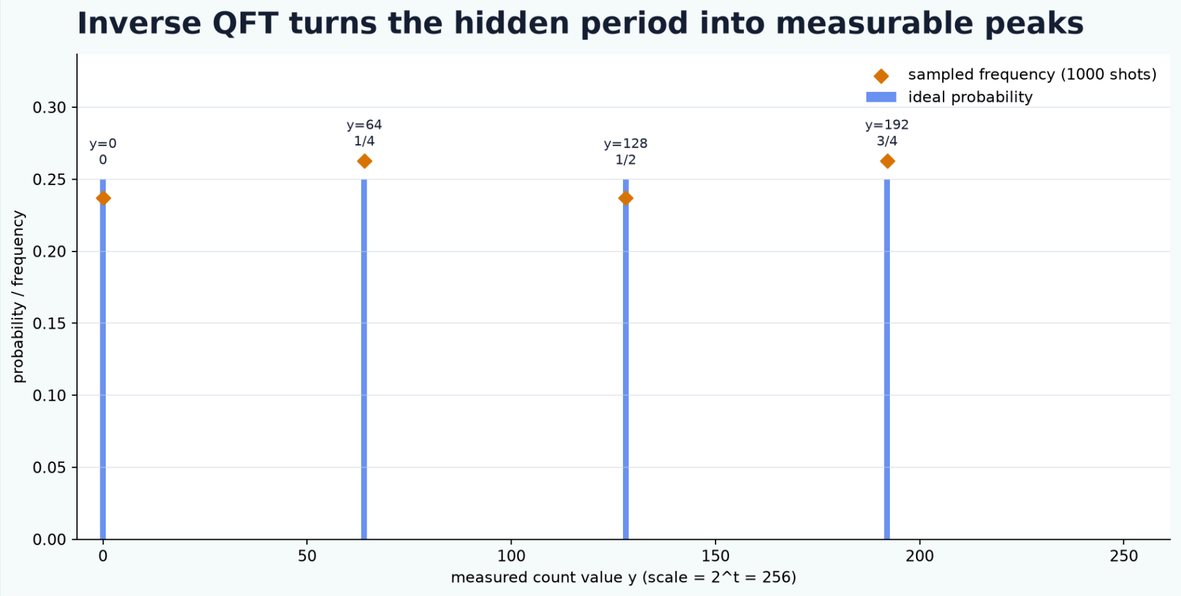
> 測った値 $y = 64 \to 64/256 = 1/4 \to$ 分母の **4 が周期**

なぜ量子だと速いのか

	古典	量子 (Shor)
周期の探し方	$7, 7^2, 7^3, \ldots$ と順に計算して、1 に戻るまで待つ	全部の x を重ね合わせ、フーリエ変換で周期だけ読む
手間	周期の長さ程度。617 桁の数では、周期も天文学的な長さになりうる	回路の大きさは桁数の 3 乗程度

- 表のマス数は qubit 1 個ごとに **2 倍**。12 qubit なら 4096 マスで、ノート PC でも計算できる
- 617 桁の数では、表は古典のメモリに**載らない**。qubit は桁数に比例する数で済む

1ショット＝光子1個



- 1 ショットで出るのは 4 本のどれか **1 つ**
- **分布の形**を見るには何千ショットも要る
- (電子と光子の違いはあれど) 有名な電子の二重スリット実験と基本は同じ
  - 回折格子は **フーリエ変換** そのもの

シミュレータでは完璧に動く

- ノイズのないローカルシミュレータ (12 qubit、1000 ショット)
- 理想の分布との距離は 3×10^{-16} — 数値誤差の範囲で一致
- 測定値から $r = 4$、そして $15 = 3 \times 5$

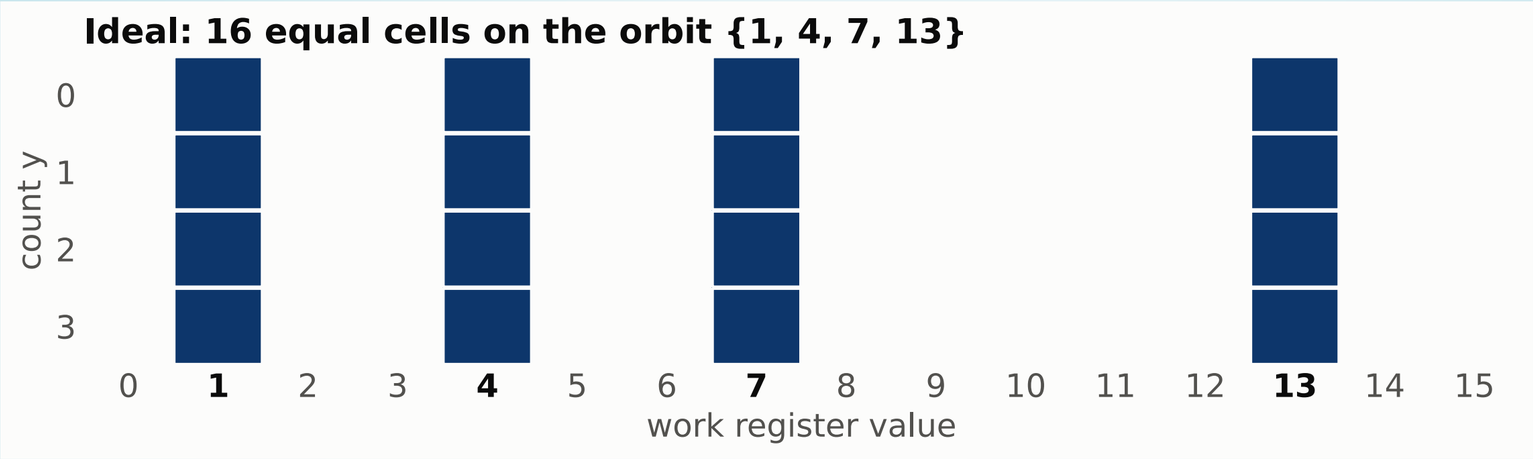
> - シミュレーションでうまくいった、勝ったな、ガハハ **(死亡フラグ)**

答えを使って回路を縮めた

- 教科書どおりの回路は**数千ゲート** → 実機では信号が消える
- そこで**近道を 2 つ**使った (詳細は Appendix)
  - 答え: 周期は 4 以下 → count を **2 qubit** に
  - 15 の形: **15 専用の掛け算回路**
- 答えを知っていれば「因数分解」を演出できてしまう (Smolin, Smith, Vargo 2013)

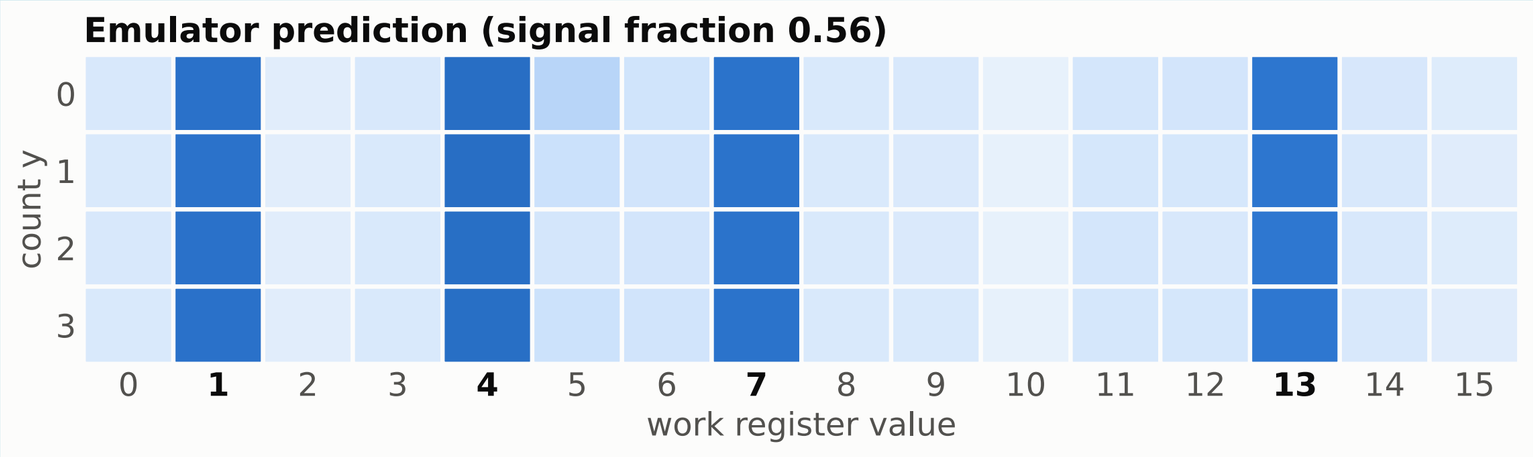
- **これは Shor のアルゴリズムに忠実に 15 を因数分解した実験ではない**
  - 15に特化して縮めた回路がどれだけ理想どおりに動くかを測る実験

縮めた回路の理想



- 理想: work が 1, 4, 7, 13 の **16 マスだけ**
- **信号残存率** λ: 縞の残り具合 (1 = 理想、0 = 乱数)
- 理想的 ($\lambda = 1$) にはこの16マスにしか出ない

縮めた回路の予測

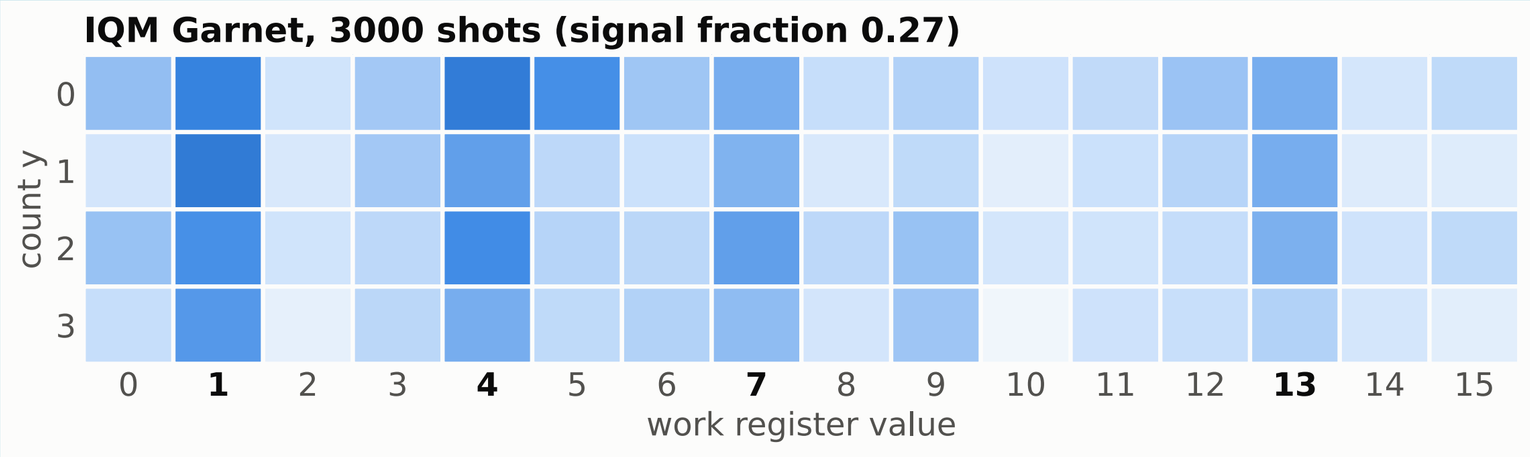


- エミュレータが回路の構成から $\lambda = 0.56$ と予測
- ノイズは混ざるが縞模様は確かに見える
- いけるか? (いけない)

実機の結果

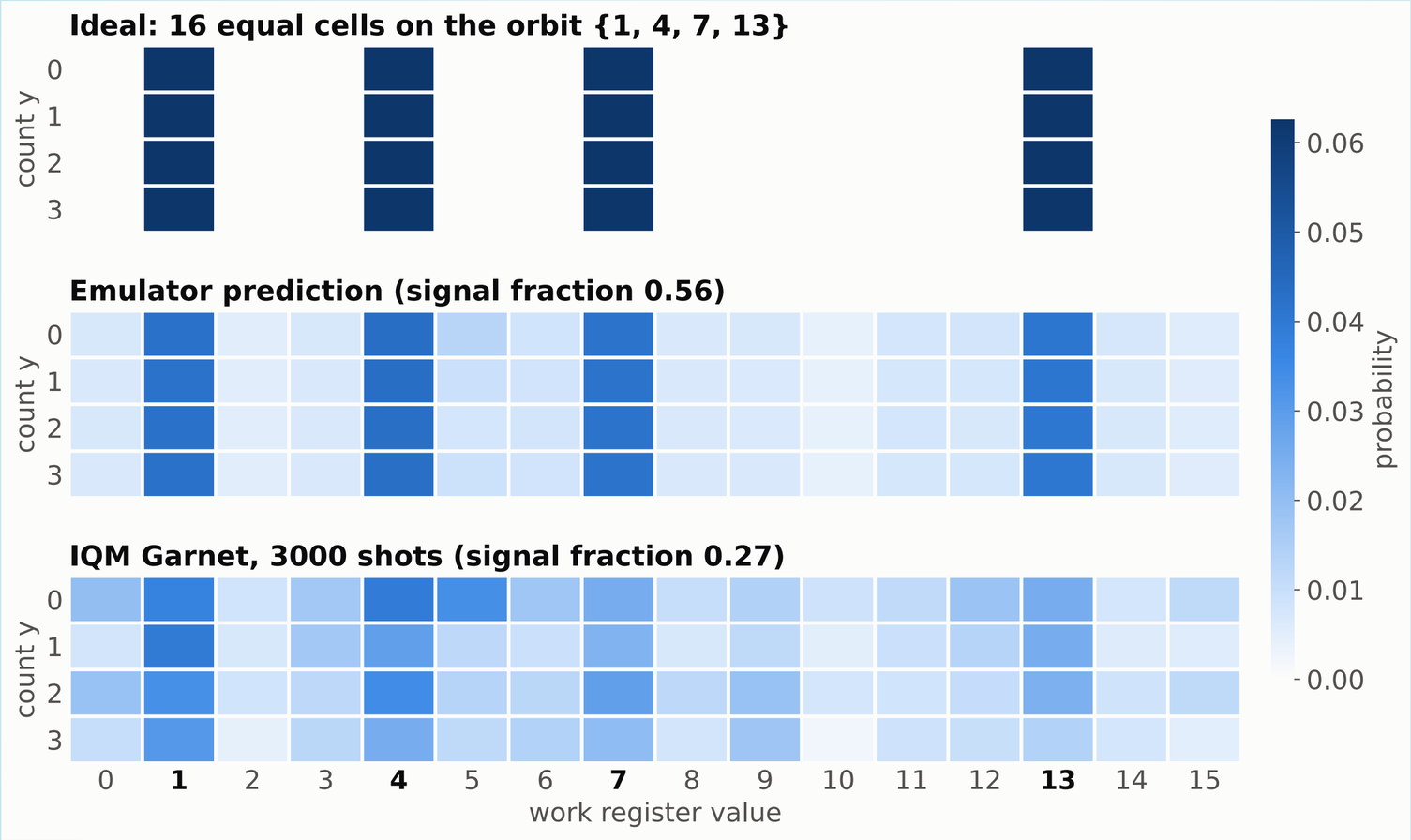
IQM GARNET、3000 ショット

結果:縞は見えるが、薄い



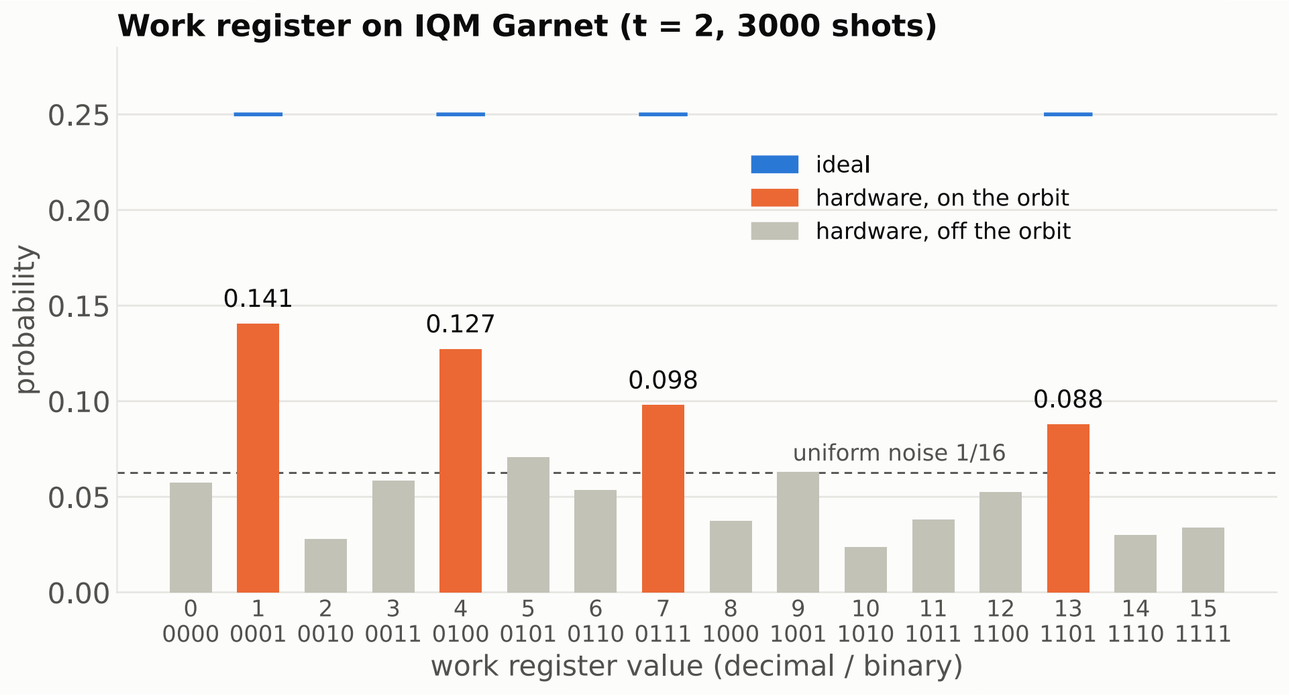
- 目を細めてぼやかせれば、かろうじて見えなくもないような気がしなくもないレベル
  - つまり見えない

理想、予測、実測の比較



上: 理想 (4 本の縦縞) / 中: エミュレータの予測 / 下: 実機

ピークの位置は合っている



- 上位 4 値は**理想の 4 値と一致**、解析ミスではない
- 外れた値の大半は、正解から **1 ビット違い**
  - 例: 0001 と 1 ビット違いの 0000, 0011, 0101, 1001 は、どの正解からも遠い値の**約 2 倍**

数字で見ると

指標	理想	予測	実機	乱数
work が正しい 4 値に乗った割合	1	0.673	0.454	0.25
信号残存率 λ	1	0.564	**0.272 ± 0.012**	0

- 0.454 は前のスライドの橙 4 本の合計
- **乱数** (64 マスに均等) でも **1/4** は乗る
- それを差し引き、乱数 0、理想 1 に揃えたのが λ

- 信号は**ある** (ゼロから 22σ)。でも予測の**半分以下** (予測から 24σ)

周期で採点すると

- Shor の最終的な答えは**周期**
- 1 ショットずつ、**周期 4** に辿り着けたかを採点

採点	予測	実機	乱数
信号残存率 λ	0.564	0.272	0
正しい周期に辿り着く割合	0.745	**0.695**	**0.750**

冒頭の「サイコロに負けた」はこの採点

- 乱数は $\lambda = 0$ でも **4 回に 3 回**当たる (後述)
- 実機も、**予測ですら**乱数に届かない

偏りに気づく

- **1 のビットが少ない値ほど出やすい**
  - work: 1 (0001) と 4 (0100) が、7 (0111) と 13 (1101) より多い
  - count: 00 が 0.305、11 が 0.203
- qubit が $|1\rangle$ から $|0\rangle$ に**落ちている**ように見える

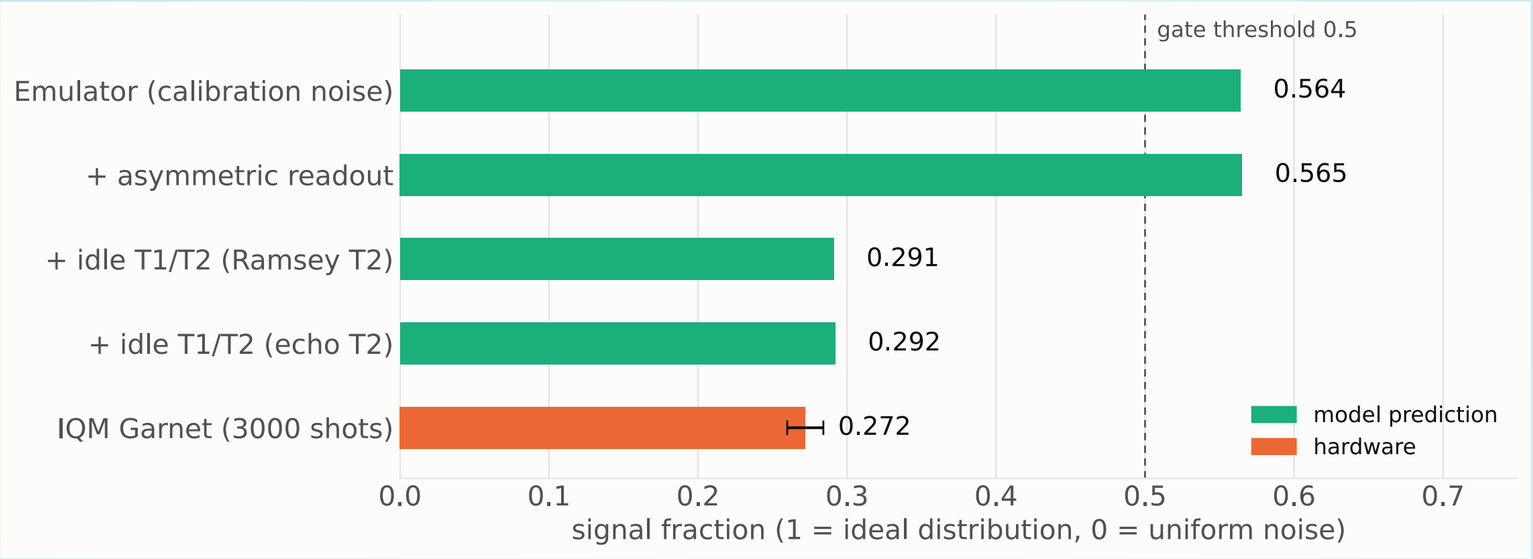
- **エネルギー緩和**の匂い
  - $|1\rangle$ が周りにエネルギーを捨てて $|0\rangle$ に落ちる
  - 波でいえば**振幅**の減衰 (**位相**の乱れは次で)
- エミュレータのノイズモデルには**無かった**

T1 と T2: 量子ビットの「寿命」

- **T1 (エネルギー緩和)**: $|1\rangle$ が $|0\rangle$ に落ちるまでの時間
- **T2 (位相緩和)**: 重ね合わせの**位相**が乱れるまでの時間
  - NMR の縦緩和・横緩和と同じ概念
- 今回使った qubit の T2 は **5.6〜28 μs**
- 回路全体は**約 6〜11 μs** (ゲート時間は仮定)

> ゲートを受けずに**待っている qubit** が、待っているあいだに崩れていく

待機中の減衰を予測に足す



- 読み出し誤りの偏り → **効かない**
- 待機中の T1/T2 → 予測が **0.29** に下がり、偏りも再現
- ゲート時間は**仮定**。これは**示唆**であって証明ではない

一番大事な気づき

測っていたのは「量子らしさ」で
はなかった

サイコロでも当たる

- 測定値から周期の候補を作り、**古典で検算**する
- でたらめな測定値でも、**1/4 か 3/4 の近く**に落ちれば周期 4 が出る

乱数しか返さない壊れた装置が	**1 回で周期 4 に辿り着く割合**
前半の例 (count 8 qubit)	約 12% (当たりの幅が全体の 1/8)
今回 (count 2 qubit)	**75%** (0, 1/4, 1/2, 3/4 のうち 0 以外)

1/2 は周期 2 の候補 → 検算で落ちるが、2 倍の 4 は通る

> **「因数が出た」を成功の条件にしてはいけない**

余りの箱が、指数を覚えている

count の値 x	**0**	**1**	**2**	**3**
掛け算のあとの work	1	7	4	13

- count 2 qubit では、work の値が**全部違う**
  - work の列ごとに x は **1 つ**だけ
- 前半の 8 qubit では、同じ work の列に **64 個の x** (4 つおきの縞)

- 足し合わさるのは**同じマス**の振幅どうしだけ
- 光なら work ごとに**別のスクリーン**
  - 8 qubit: **64 本の格子** / 2 qubit: **スリット 1 本**

1マスは均等に広がるだけ

count	逆 QFT の前 (work)					逆 QFT の後 (work)			
	1	4	7	13		1	4	7	13
0	●				→	●	●	●	●
1			●		→	●	●	●	●
2		●			→	●	●	●	●
3				●	→	●	●	●	●

- 1 マスには縞の**間隔**がない → **均等に広がる**だけ
- ● の確率: 前は 1/4、後は 1/16

> 後の表が縮めた回路の理想。はじめから**干渉なし**

重ね合わせが壊れても、同じ分布

- 重ね合わせが壊れる = 各 x の**位相**がばらばら (T2)
- 相手のいない振幅は、位相がずれても**同じ大きさ**
  - スリット 1 本の光は、位相をずらしても同じ

count	位相が揃っていれば	位相がばらばらなら
8 qubit (前半)	4 本のピーク	64 本が揃わず、均等
2 qubit (今回)	16 マスに均等	**16 マスに均等 (同じ)**

> 壊れていても**見分けがつかない**。λ が見ていたのは**掛け算回路の出来**だけ

15 という数の特殊性

- count の通り数は **2, 4, 8, …** (qubit 1 個で 2 倍)
- 15 の周期は **1, 2, 4 だけ** — 全部 2 のべき
  - count を**ぴったり**合わせられる → 1 マスずつ
- 周期 6 (例: $N = 21$、底 2) は合わせられない
  - 8 通りにすると $x = 0$ と 6 が**同じ余り**

> - 15 なら**干渉なしでも**周期が出せてしまう
> - 量子アルゴリズムの検証としては**甘い問題**

なぜ2量子ビットに縮めたのか

- 周期 4 なら count は **2 qubit** で足りる
  - 前半の 8 qubit と同じ 1/4 刻みの答え
- count を増やすほど、ゲートと**待ち時間**が増える
- 15 の実機実験は 2001 年 (NMR) 以来、**ほぼすべて答えを使って縮めた回路**
  - 今回もこの主流のやり方に倣った

> 代償: **乱数でも 75%** 当たり、**重ね合わせの壊れも見えない**

なぜ 15 が定番なのか

- Shor の手順が通る**最小の数**。次が 21
  - 奇数で、素数のべきでない数のうち最小
- 周期が全部 **2 のべき** → count が少なくて済む
- 掛け算が**ビットの回転と反転**だけ (Appendix 2)
  - 16 を 15 で割ると 1 余る → 2 倍はビットの回転
  - 21 は回転にならず、**3 qubit ゲート**が要る

> 実機に載る**一番浅い** Shor → 2001 年以来の定番

では 21 なら?

- 周期 6 は 2 のべきでない → **干渉が要る**
  - 同じ余りに複数の x → 壊れが数字に出る
- 1/6 刻みを読むには count が **7〜8 qubit**
  - 3 qubit では理想と乱数の差が**約 6 ポイント**
- 21 の実機例もある (光子 2012、IBM 2019, 2021)
  - どれも答えを使って縮めた回路

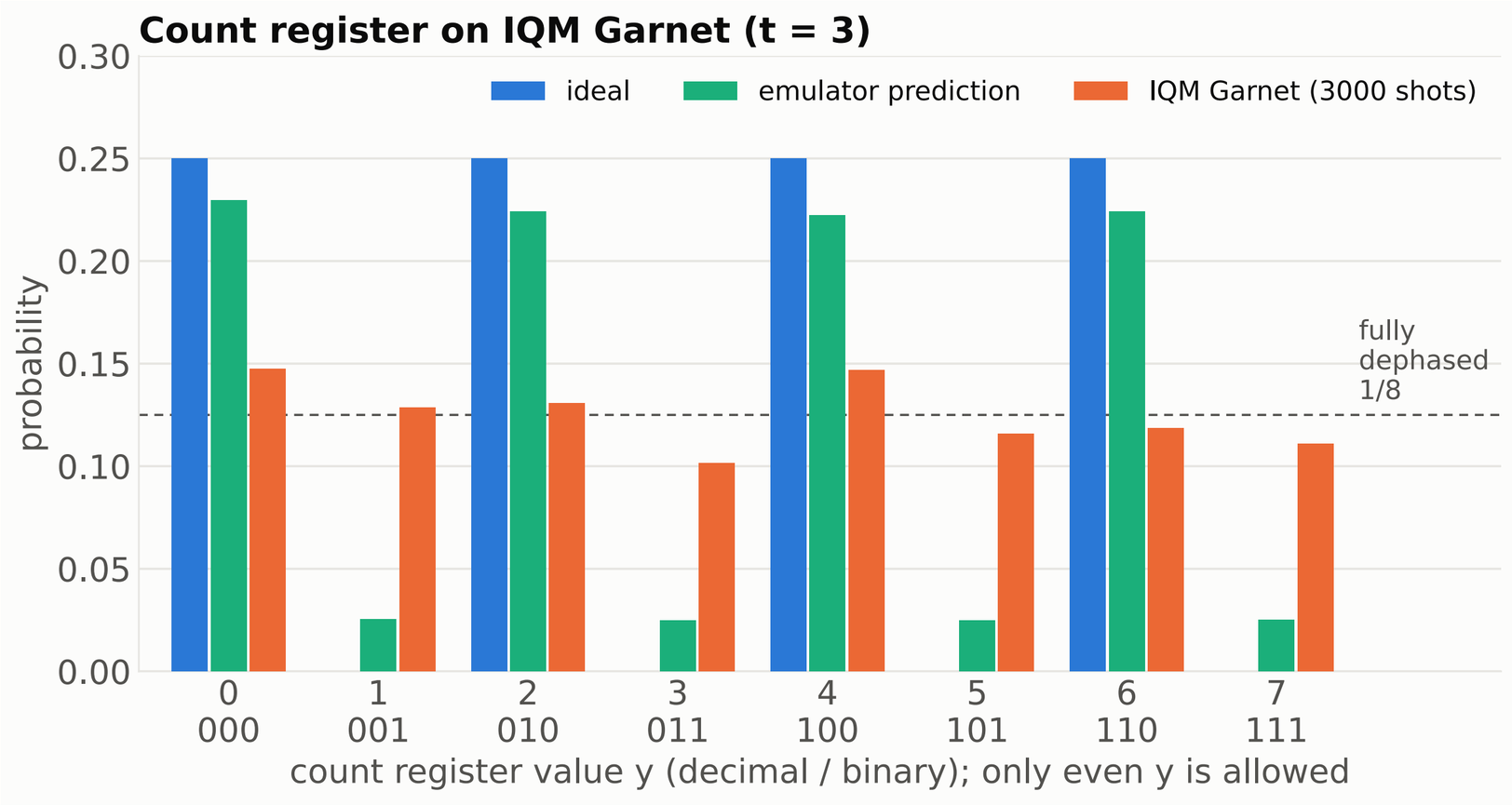
- 問われるのは、count の位相が**最後まで持つか**
- 15 も 1 qubit 足せば、**壊れが見える**

追加実験: 位相は保たれているか

- 増やした 1 qubit は、重ね合わせのまま**待機するだけ**
- 位相を保てていれば、測定値は**偶数だけ** (0, 2, 4, 6)
- 位相が壊れていれば、奇数も同じだけ出る
- **可視度** = 偶数の割合 − 奇数の割合 (理想 1、壊れると 0)

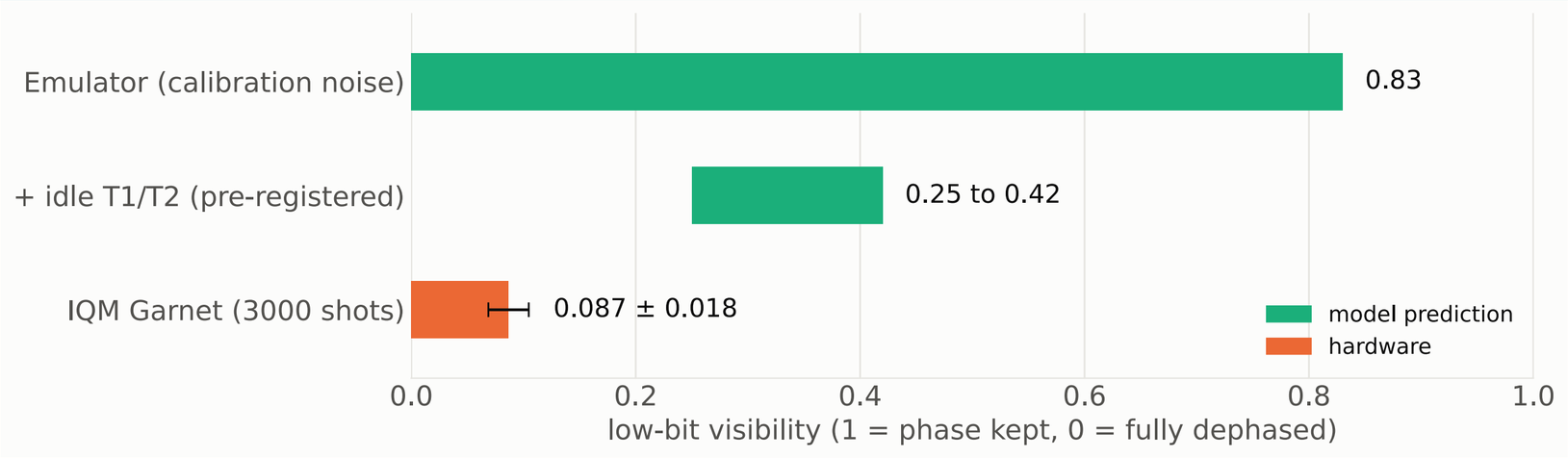
> 事前の予測: エミュレータ **0.83** / 待機の減衰を足すと **0.25〜0.42**

結果: 奇数がほぼ同じだけ出た



青: 理想 (偶数だけ) / 緑: エミュレータの予測 / 橙: 実機 (同日、3000 ショット、4.65 USD)

どちらの予測も外れた



- 可視度 **0.087 ± 0.018**
- 待機していた qubit は、回路のあいだに**ほぼ完全に位相を失った**

何が起きていたのか

1. **置き場所が悪かった**
   - 待機 qubit は、8 個のうち **T2 が最短 (5.6 µs)** の場所にいた
   - qubit の置き場所を決める自作の配置プログラムは、ゲートの誤り率しか見ていなかった
2. **それでも予測 (0.25) より悪い**
   - 候補: **隣の qubit で動くゲートが位相を乱す** (未検証)

> カタログ値の T2 は、動いている回路の中での**上限**でしかない

振り返り

かかった費用

実行	測定回数	費用
経路確認 (1 回目は権限設定で拒否)	10	0.31 USD
本測定 (count 2 qubit)	3000	4.65 USD
追加実験 (count 3 qubit)	3000	4.65 USD
合計		**9.61 USD**

> 見積りとサービス側の集計の差は 1 セントもなかった

9.61 ドルで分かったこと

- 校正データだけのエミュレータは、実機の誤りを**約 2 倍過小評価**した
- 欠けていたのは**待機中のデコヒーレンス**
- 15 相手では、測れるのは「掛け算回路の出来」と「待機 qubit の位相」まで
- 待機 qubit は、カタログ値の T2 より速く位相を失う

> どれも**実機を通さなければ見えなかった**

先行事例と比べると

- Amazon Braket での公開事例を調べた (2026-09-23 時点)
  - AWS 公式の教材は**シミュレータ**での実行
  - 実機の例は、回路の一部か、**答えを使って簡略化した回路**
  - 教材の回路は実機向けの**変換の段階で失敗**、という報告も

> 今の実機で Shor がうまくいかないのは**主流の結果**。
> 今回は**自分の数字で確かめた**

続けない判断と、次の一手

- 実機はここで**一区切り**
  - 予測が実測を説明できないまま投げても「またずれた」で終わる
- 21 はさらに先: **必要条件**を満たせていない
  - 追加実験が、結果としてそれを確かめていた
- 先にオフライン (無料) でやること
  - 配置プログラムに「待機時間と T2」を入れる
  - 隣のゲートの影響をモデルに入れる
- 次の実験の候補: **CHSH 不等式**
  - 2 qubit と数ゲートで、エンタングルメントの質を直接測る

完走した感想

- **「うまくいかない」を数字で踏む**のは、悔しいけど思ったより楽しく学べる
- **測る前に予測を書いておく**と、外れ方から学べる
- 「何を測っているつもりか」は、**測ったあとで**ひっくり返ることがある
  - 事前調査で潰すこともできたかもしれないけど、走らせたからこそ得られる教訓もある

- **10 ドルで教訓を買った**

まとめ

1. Shor の周期発見回路 (15 専用) を超伝導量子コンピュータで実行
2. 信号残存率は **0.27** (予測 0.56)。主因は待機中のデコヒーレンス (モデルの示唆)
3. しかも 15 相手では、その数字は**量子的な干渉を見ていなかった**
4. 待機 qubit の位相は**ほぼ完全に失われていた** (可視度 0.09)

> - 「15 = 3 × 5 が出た」は何の証拠にもならない
> - **「何を測ったのか？」** を考えたことが本当の成果

詳しく知りたい方へ

- 実装と記録: github.com/s-sasaki-earthsea-wizard/shor-braket
- Wiki の読み物
  - 図で追う Shor のアルゴリズム
  - N=15 を QPU 互換回路で
  - 実機で最初の 3000 ショット
  - Amazon Braket で Shor を動かした先行事例
- 開発には AI エージェント (Claude Code ほか) を活用

LT登壇者の募集

- 物理学集会ではLT登壇者を募集しています！
  - どんなジャンルでもOK！
- 興味のある方は物理学集会のDiscordサーバーまで！

APPENDIX

APPENDIX 1: 今の実機の制約

- **ゲートを掛けるたびに誤る**
  - 2 qubit ゲート 1 回で約 0.3〜2% (候補の 3 機)
- **隣どうしの qubit にしか 2 qubit ゲートを掛けられない** (超伝導の格子型)
- 教科書どおりの掛け算回路は、15 相手でも**数千ゲート**

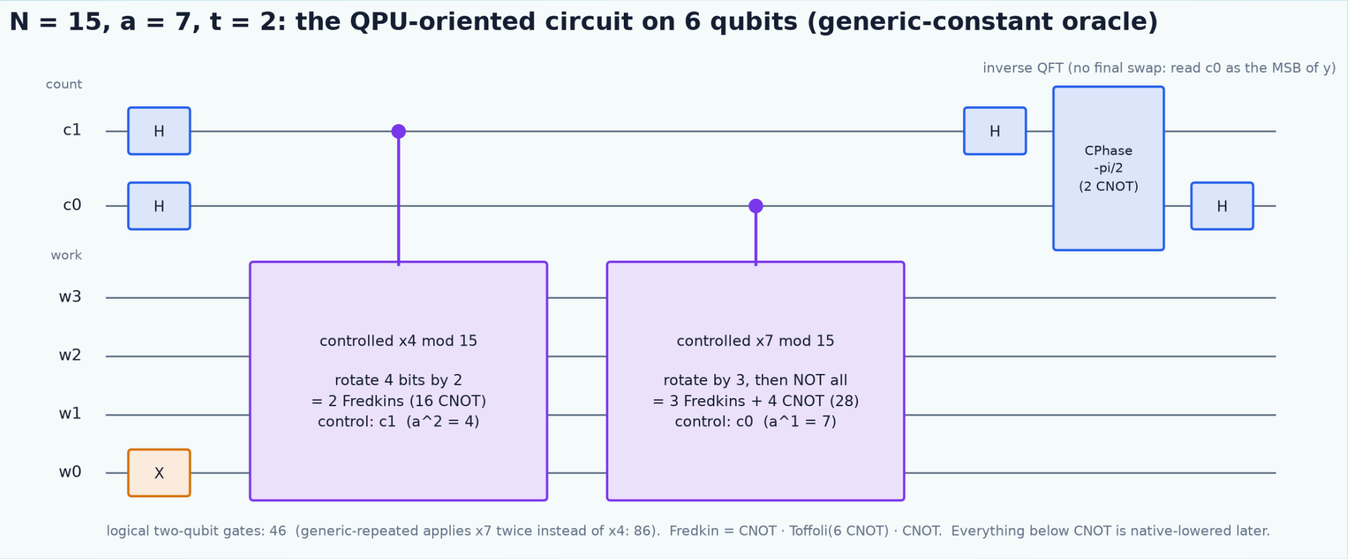
> 誤りが積もると、残る信号は $e^{-10} \approx 5 \times 10^{-5}$ 程度。**乱数と区別できない**

APPENDIX 2:15 専用の掛け算回路

- 4 ビットの数を 2 倍して 15 で割った余り = **ビットを 1 つ左に回すだけ**
  - あふれた最上位ビットが最下位に戻る ($15 = 1111_2$)
- 7 倍は「3 ビット回して、全ビット反転」
  - 例: $0001 \to 1000 \to 0111$ (1 × 7 = 7)
- 掛け算が**ビットの入れ替えと反転だけ**で書ける

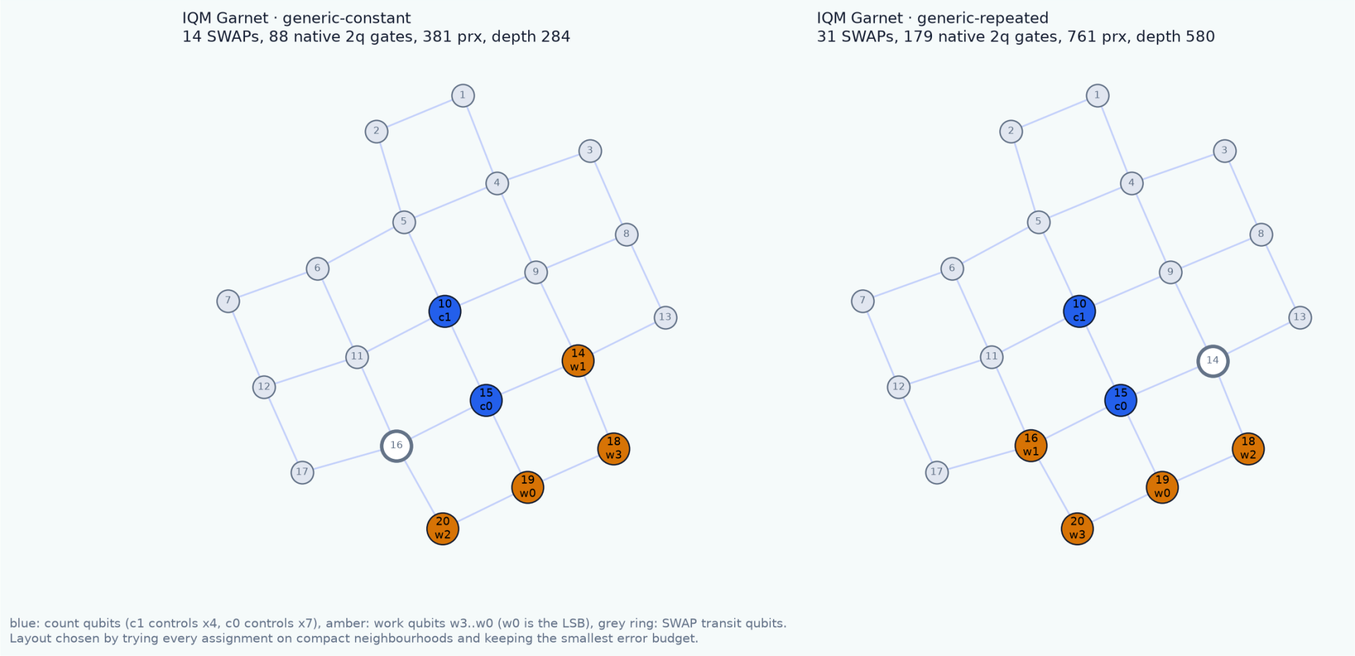
> 数千ゲート → **2 qubit ゲート 46 個**。ただし **15 にしか使えない**専用回路

APPENDIX 3: COUNT を 2 QUBIT に



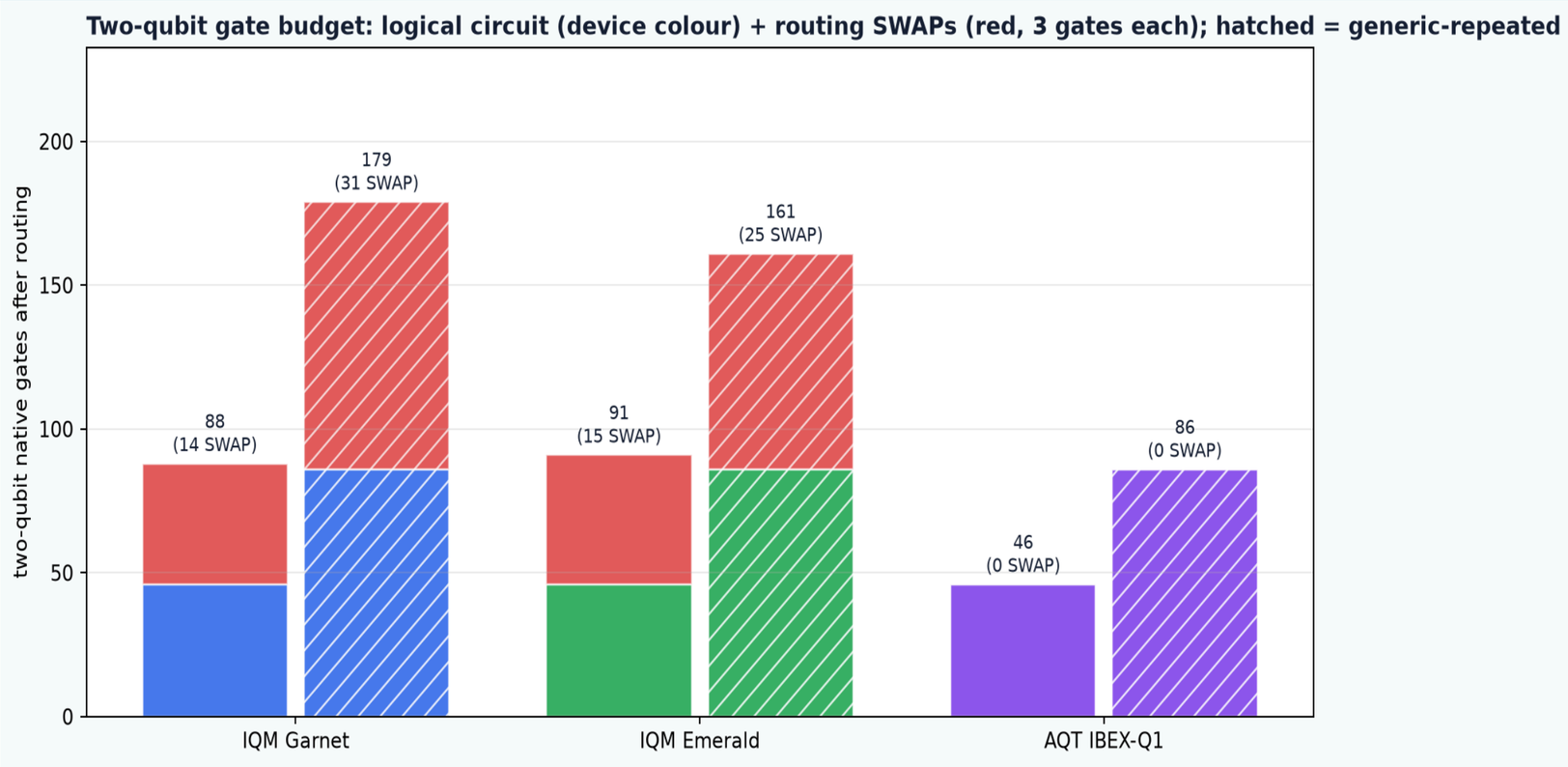
- count を **2 qubit** に (教科書の推奨は 9)。合計 **6 qubit**
  - 「周期は 4 以下」と知っているからできる選択
- 近道を 2 つ使った以上、**この実験は「因数分解」ではない**

APPENDIX 4: 格子で回路が倍に



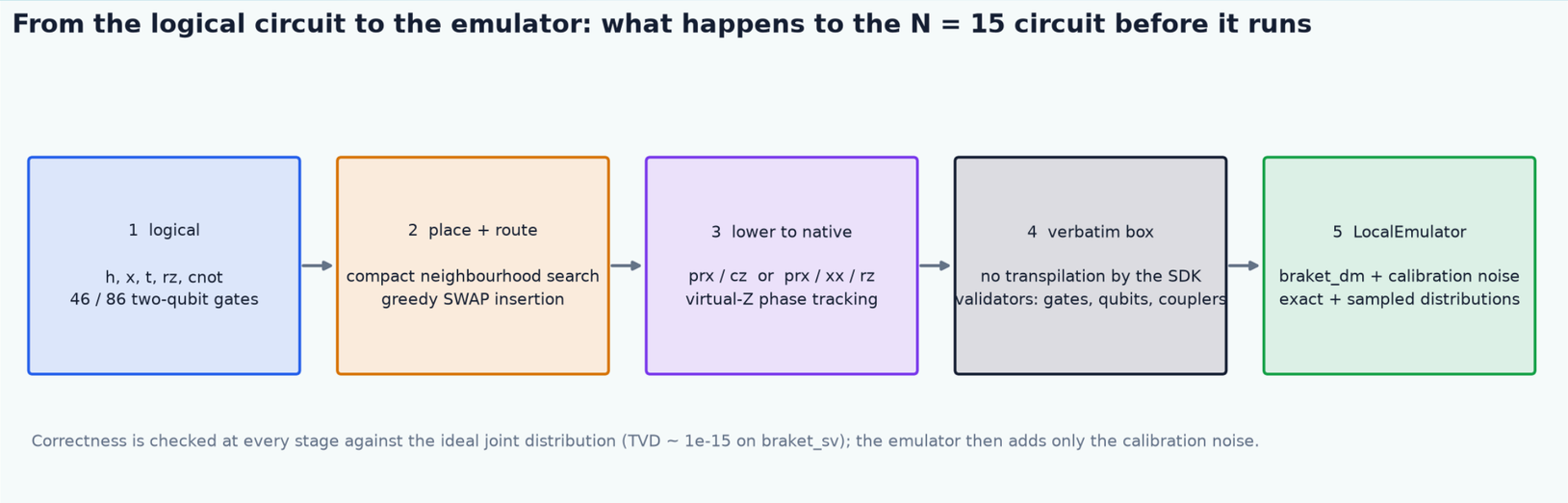
- IQM Garnet (左が今回の回路)。ゲートは**線で結ばれた隣どうしだけ**
- 離れた qubit は入れ替えて運ぶ → 2 qubit ゲート **46 → 88 個**

APPENDIX 5: 3 機のゲート数



赤が入れ替え (SWAP) による増分。格子型の IQM ではほぼ倍になり、全結合の IBEX ではゼロ

APPENDIX 6: 論理回路が実機の形になるまで

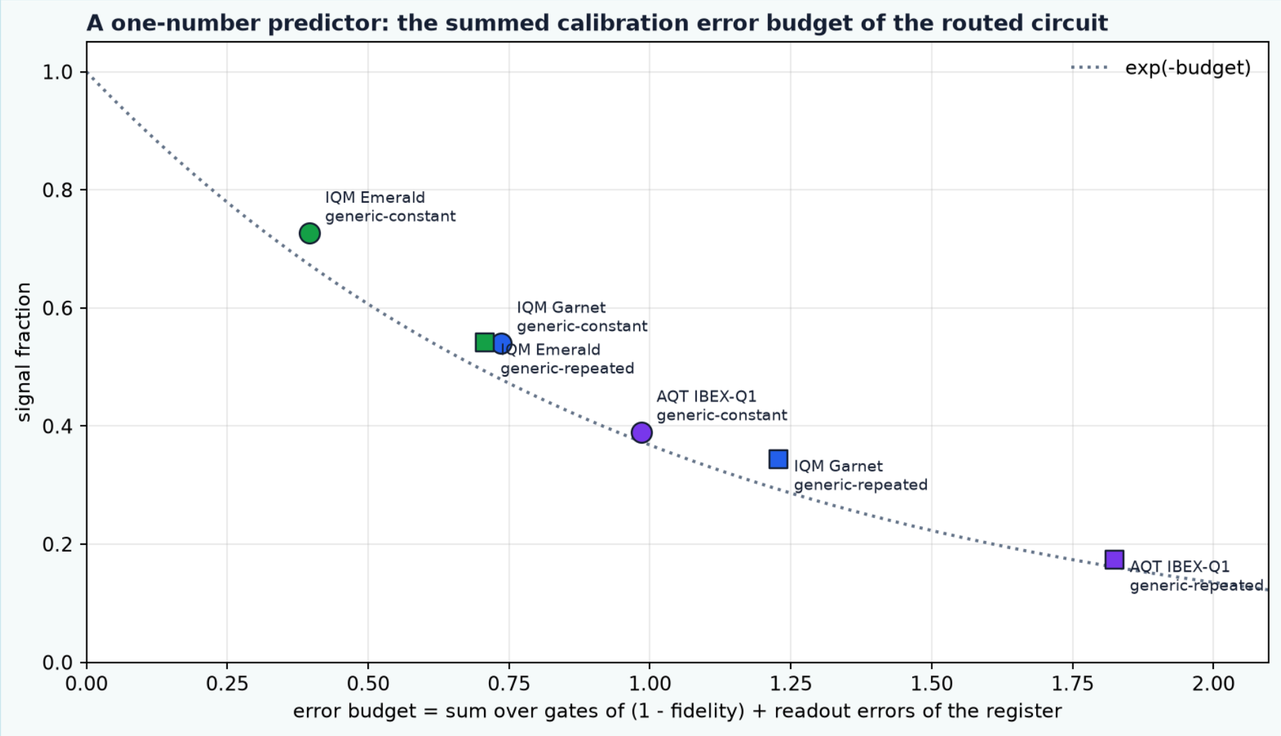


配置と経路探索、機械固有のゲートへの分解は自作。各段階で理想分布との一致を確かめている

APPENDIX 7: 反復型の位相推定は効かなかった

- 測定結果で次の操作を切り替える方式なら、count を 1 qubit で使い回せる
- エミュレータで比べると、λ は標準の方式と **±0.03 以内**
  - IQM Emerald: 0.70 対 0.73 / IQM Garnet: 0.53 対 0.54
- 入れ替えの回数は掛け算の側で決まり、count の数に依らないため

APPENDIX 8: 残る信号の見積もり



- 回路が使う全ゲートの誤り率の合計 = **誤り予算** B
- 残る信号はおよそ $\lambda \approx e^{-B}$

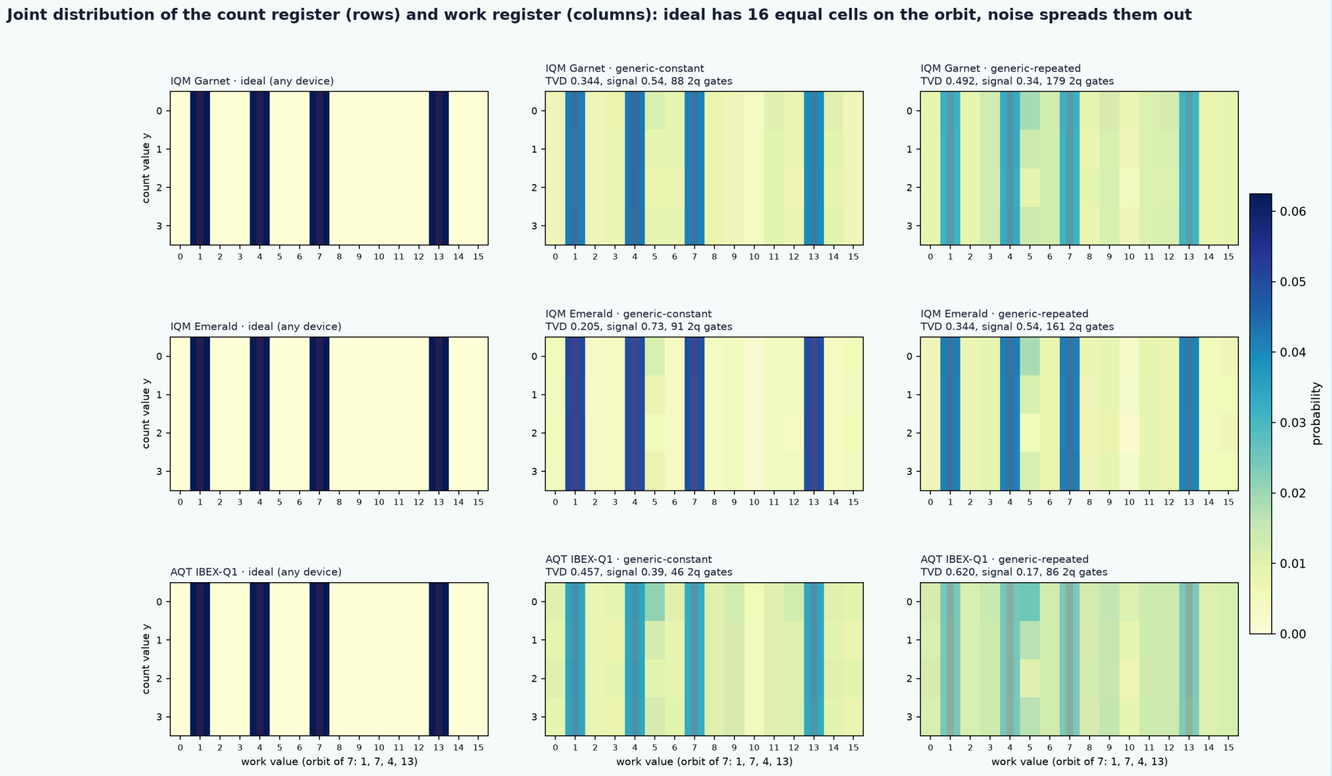
APPENDIX 9: 3 機の予測と機種選び

機械	方式	ゲート数	予測 λ	測定 1 回
IQM Emerald	超伝導 54 qubit	91	0.73	0.00160 USD
IQM Garnet	超伝導 20 qubit	**88**	**0.54**	**0.00145 USD**
AQT IBEX Q1	イオントラップ 12 qubit	46	0.39	0.02350 USD

ゲート数は 2 qubit ゲートの個数。2026-09-13 時点の校正データによる予測。本編の 0.56 は実行当日 (2026-09-23) の校正データによる

- 一番短い回路 (IBEX) が一番良いとは限らない
- **安くてやり直しのきく Garnet から始める**

APPENDIX 10: エミュレータが予測した分布



左: 理想 / 中: 今回の回路 / 右: 近道を 1 つ減らした回路

APPENDIX 11: 実行条件

デバイス	**IQM Garnet** (超伝導、20 qubit)
実行日	2026-09-23
回路	$N = 15$、底 7、count 2 + work 4 = 6 qubit
2 qubit ゲート / 深さ	88 / 284
測定回数 (ショット)	3000
投入から完了まで	約 3 秒
費用	4.65 USD

APPENDIX 12: お金を溶かさない仕組み

- 実機は**課金される**: 3000 回の測定で 4.65 USD
- シミュレータの検証を通った回路**だけ**を投げられる仕組みを自作
- クラウド側にも支出の上限と権限の分離を設定

> 余談: 最初の 1 回は、**自分で作った安全装置に自分が止められた** (課金なし。Appendix 13)

APPENDIX 13: 自分の安全装置に止められた話

- 投入は**多要素認証 (MFA) を通した一時的な権限**に限定
- その権限の中では「MFA なし」と判定される仕様だった
- 自分で書いた「MFA なしは拒否」のルールが発動
- 事前の検証 22 項目は全部通っていた

> **シミュレータは、与えた前提の下での答えしか返さない**

APPENDIX 14: 信号残存率の求め方

- 実機の分布を「理想」と「乱数」の混合とみなす

$$p_{\text{device}} \approx \lambda \cdot p_{\text{ideal}} + (1 - \lambda) \cdot p_{\text{uniform}}$$

$$\lambda = \frac{m - f}{1 - f}$$

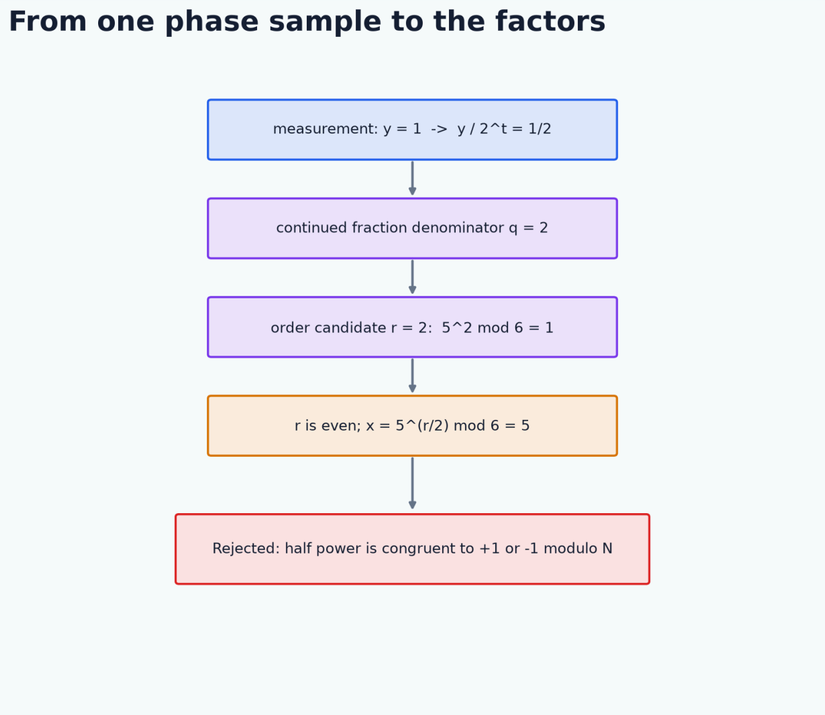
- m: 理想で確率を持つマスに乗った割合
- f: そのマスが全体に占める割合
- 本測定: $m = 0.454$、$f = 16/64 \to \lambda = 0.272$

APPENDIX 15: 追加実験のモデル比較

モデル (count 3 qubit)	可視度	work の正答率
エミュレータ	0.80	0.673
＋ 待機 T1/T2 (Ramsey T2、20 / 40 ns)	0.254	0.461
＋ 待機 T1/T2 (Ramsey T2、40 / 80 ns)	0.099	0.367
実機	**0.087**	**0.424**

括弧内は T2 の種類と、仮定したゲート時間 (1 qubit / 2 qubit)。ゲート時間を伸ばすだけでは両方を同時には説明できない

APPENDIX 16: 6 が縮退する理由



- 周期 2 は得られるが、$5^1 \equiv -1 \pmod 6$ で棄却される
- 片方の gcd から出る 2 は「6 は偶数」と同じ情報

参考文献

- A. Tonomura et al., Am. J. Phys. 57, 117 (1989)
- P. W. Shor, SIAM J. Comput. 26, 1484 (1997)
- L. M. K. Vandersypen et al., Nature 414, 883 (2001)
- E. Martín-López et al., Nat. Photonics 6, 773 (2012)
- J. A. Smolin, G. Smith, A. Vargo, "Oversimplifying quantum factoring", Nature 499, 163 (2013)
- M. Amico, Z. H. Saleem, M. Kumph, Phys. Rev. A 100, 012305 (2019)
- U. Skosana, M. Tame, Sci. Rep. 11, 16599 (2021)
- C. Gidney, arXiv:2505.15917 (2025)
- NIST, FIPS 203 / 204 / 205 (2024)
- Bagourd et al., arXiv:2512.15330
- Amazon Braket Algorithm Library, "Shor's Algorithm" notebook