

Universidade de São Paulo

C4AI

Sub Relatórios

Pedro Calligaris Delbem

Orientador: Fillipo Ghiglieno

Agosto de 2024

Sumário

1 Parte 1: Somador Completo

1.1 Introdução

Para estudar o somador completo quântico (quantum full adder), é necessário - primeiro - compreender, e desenvolver, o somador completo clássico.

Deste modo, implementou-se o somador completo clássico que consiste em um circuito que recebe dois de dois bits e um "carry in" resultando em um bit de saída e um "carry out" de modo a representar a soma de dois bits com um possível bit extra vindo de uma soma anterior.

1.2 Materiais e Métodos


No processo utilizou-se o Intel® Quartus® Prime Lite Edition Design Software Version 21.1.1 para fazer o design de hardware além de um FPGA DE0-CV Cyclone V 5CEBA4F23C7N onde programou-se o mesmo.

Primeiramente buscou-se implementar apenas uma porta "and" para compreender a utilização do software Quartus. Em seguida, desenvolveu-se o somador completo que funcionou perfeitamente ao ser testado na FPGA.

Posteriormente buscou-se medir os tempos de reação do hardware - ao mudar os valores de entrada - utilizando o simulador de FPGA "ModelSim", junto ao programa Quartus, mas por problemas do Software (dos quais não encontrou-se solução) não foi realizada a simulação.

1.3 Resultados e Discussão

Na imagem 1, vê-se o hardware implementado para o somador completo clássico



`imagens/full_adder.png`

imagem 1: implementação do circuito de um somador completo no programa
Quartus

1.4 Conclusão

O procedimento como um todo foi essencial para o desenvolver uma compreensão completa das propriedades do somador completo clássico. Tais entendimentos serão fundamentais no entendimento, desenvolvimento de um somador completo quântico, além de possibilitar comparações entre as vantagens e desvantagens de cada um.

2 Parte 2: Somador Completo Quântico

2.1 Introdução

Para desenvolver uma rede neural celular quântica, faz necessário entender somador completo quântico. Ao contrário do caso do somador completo clássico - todas as portas lógicas na computação quântica devem ser reversíveis.

Deste modo, cada operação deve contar com um qubit auxiliar de controle que garante a reversibilidade. Buscou-se entender este processo e implementar o somador completo quântico.

2.2 Materiais e Métodos

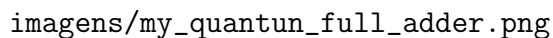
Utilizando o Circuit Composer no site da IBM, que permite utilizar as portas lógicas quânticas em um sistema de "arrastar" implementou-se o somador completo quântico.

As principais portas lógicas utilizadas são Controlled-NOT que dado dois qubit's retorna o inverso do qubit que não é o de controle e a Controlled-Controlled-NOT (Toffoli's Gate) que dado três qubit's retorna a adição entre os qubit's que não são os de controle.

Por fim, faz-se importante simular os resultados antes de processá-los no hardware real. Uma vez que - para circuitos mais complexos a saída correta não será tão óbvia. Então utilizando a biblioteca "Qiskit" para Python, da IBM - foi possível montar o mesmo circuito e simulá-lo através do código.

2.3 Resultados e Discussão

Na imagem 1, vê-se o hardware implementado para o somador completo quântico

A rectangular box containing the text "imagens/my_quantun_full_adder.png", which serves as a placeholder for a quantum circuit diagram.


imagens/my_quantun_full_adder.png

imagem 2: implementação do circuito de um somador completo mo IBM Circuit Composer

Note que o qubit é sempre inicializado em 0, assim para utilizar o valor 1, devemos invertê-lo. Além disso é notório que a operação "medida" (ícone cinza) apesar de não-reversível é necessária para trazermos o resultado quântico para o "mundo clássico".

Ademais, é importante perceber que o uso da porta "RX(π)" é necessário para manter a reversibilidade do "carry out" que poderá ser utilizado em outra soma de qubit's.

Ao rodar o circuito em um dos computadores quânticos da IBM, é sempre importante observar qual tem a menor fila de espera e foi reiniciado a menos tempo - para que se garanta um resultado com menor ruído.

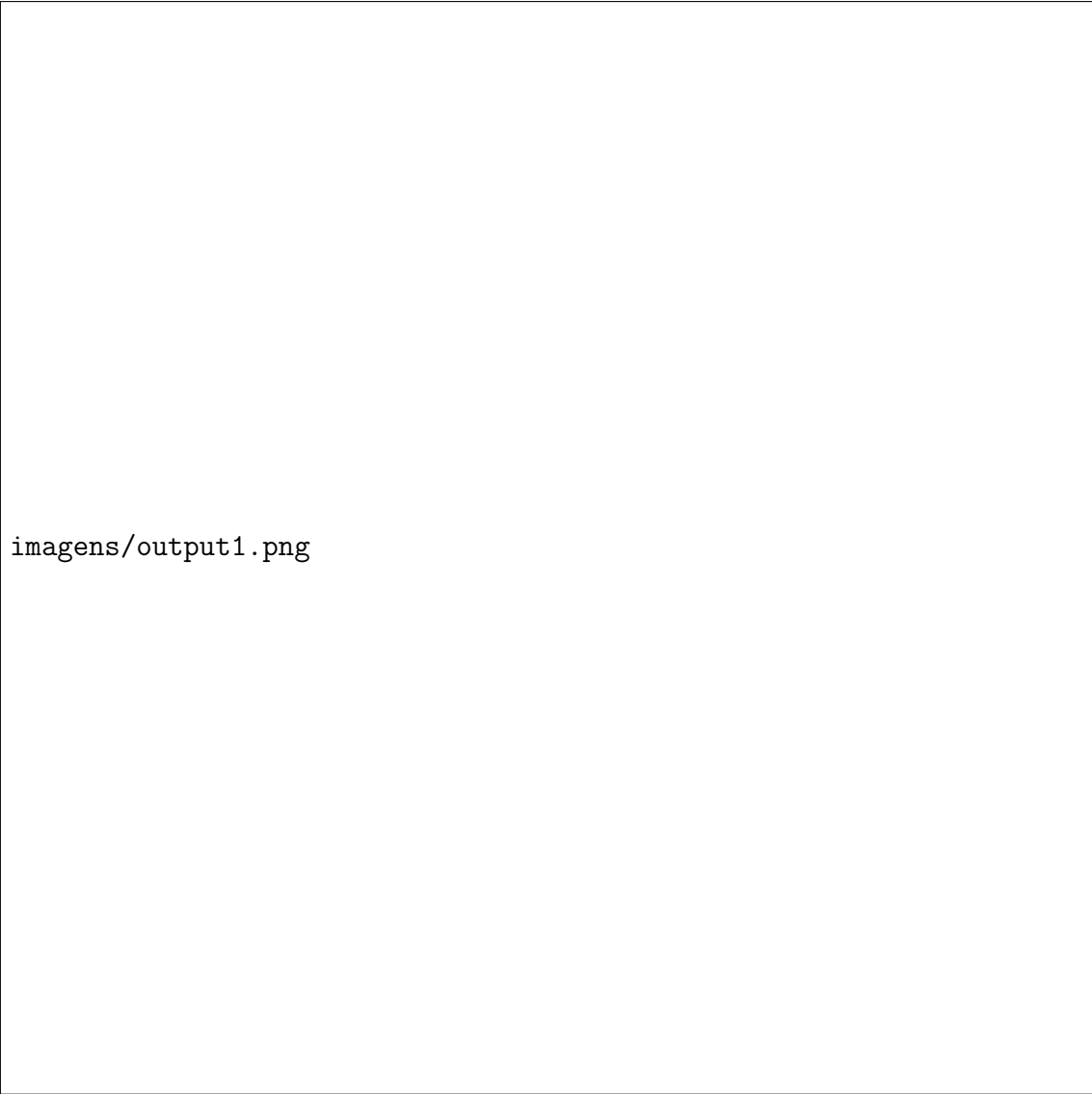


`imagens/select_quantum_computer.png`

imagem 3: janela de escolha de hardware no IBM Circuit Composer

Para exemplificar isso, foram feitas três rotinas de 1024 disparos nos quais apenas a saída da soma (c3 na imagem 2) teve -em duas das três rotinas - uma resposta majoritária diferente do esperado.

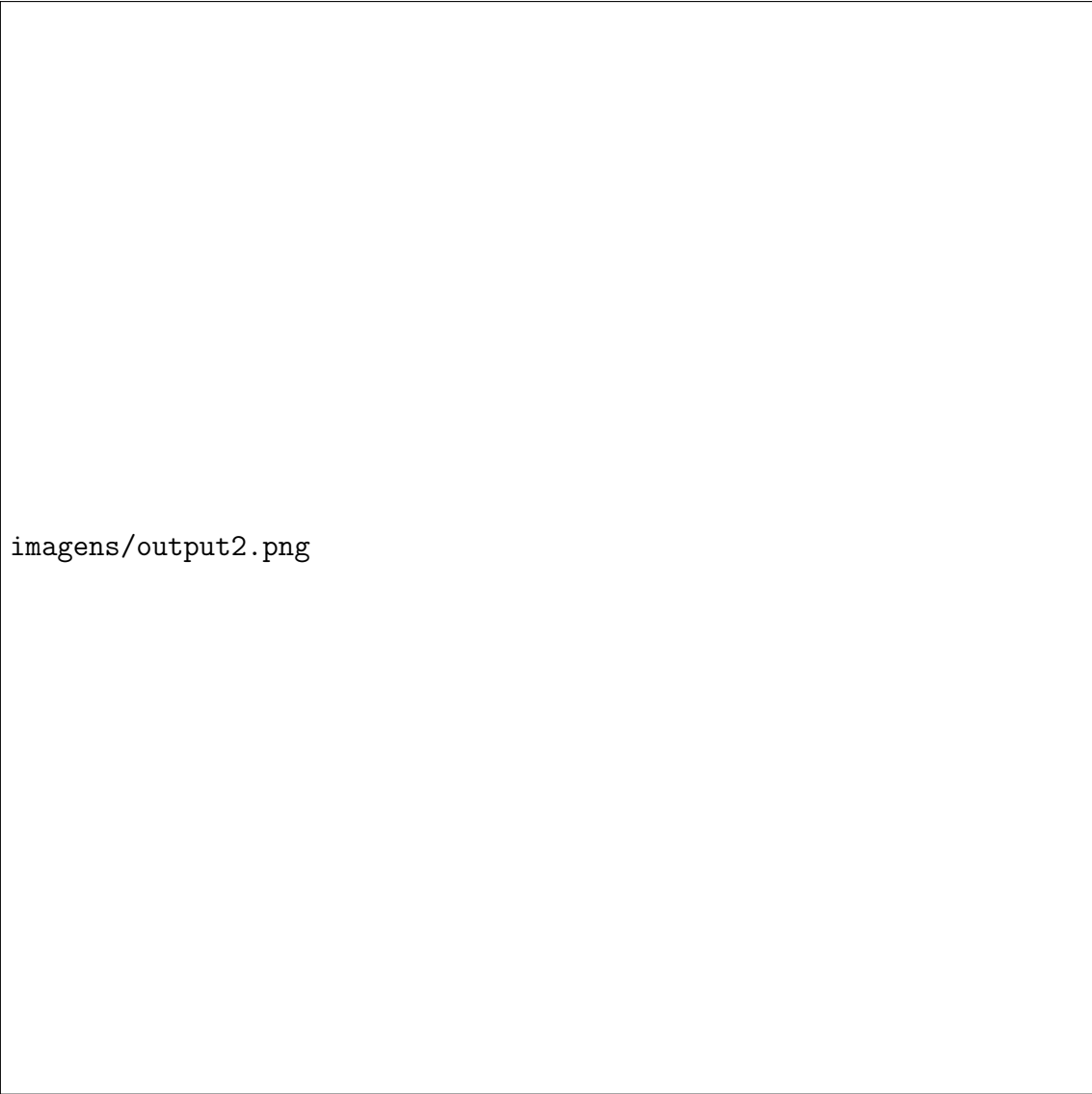
A primeira ficou mais de 5 horas na fila o que facilmente justifica o resultado majoritário ser 0 ao invés de 1



imagens/output1.png

imagem 4: saída correspondente à soma dos qubits

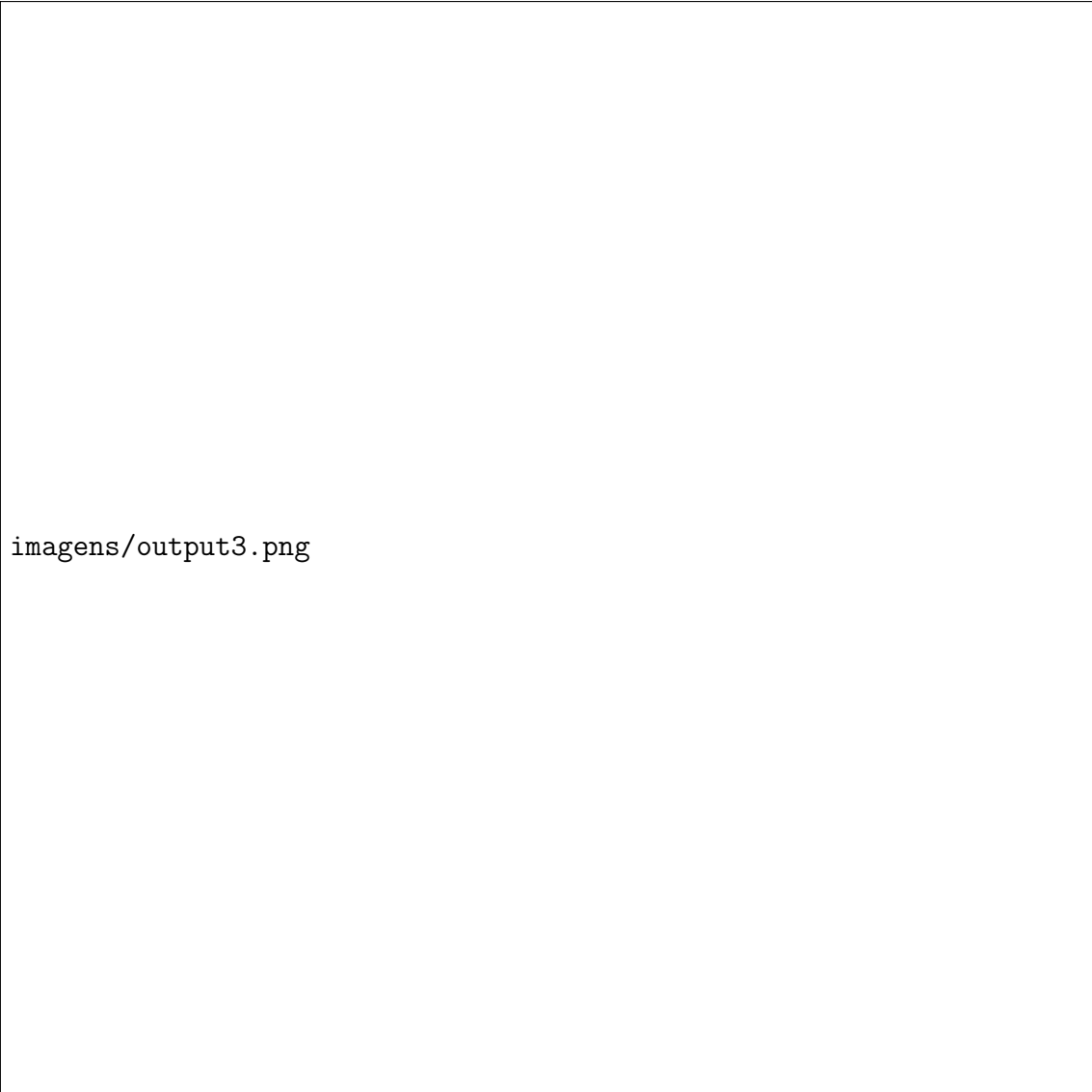
Já a segunda rotina ficou com um "empate" entre 0 e 1, o que se justifica pois o hardware escolhido havia sido reinicializado pela última vez a mais de 2 horas.



imagens/output2.png

imagem 5: saída correspondente à soma dos qubits


Por fim, escolhendo um hardware reinicializado a pouco tem e com fila pequena obteve-se a saída esperada como resultado majoritário.



imagens/output3.png

imagem 6: saída correspondente à soma dos qubits

Ademais, é importante notar que em um hardware quântico nem todos os qubit's estão conectados - deste modo, o circuito realmente implementado não é o que foi passado - assim, para obtermos a melhor resposta gastando o menor tempo possível é útil analisar o hardware real e tentar explorá-lo da melhor forma.



imagens/arquitetura_real.png


imagem 7: parte da arquitetura real de um dos computadores quânticos da IBM

Posteriormente, - utilizando a biblioteca da IBM para o Python, Qiskit, - realizou-se ambas as simulações, para um e dois full adders, além de rodar no hardware real.

Para os casos com três e quatro full adders foi apenas simulado sem ruído (pois a simulação com ruído desses casos tem o tempo de execução maior até que o tempo de espera para utilizar o hardware real) e rodados na hardware real.

As imagens 8, 9 e 10 representam a rotina de simulação sem ruído, com ruído e a rotina no hardware real - respectivamente

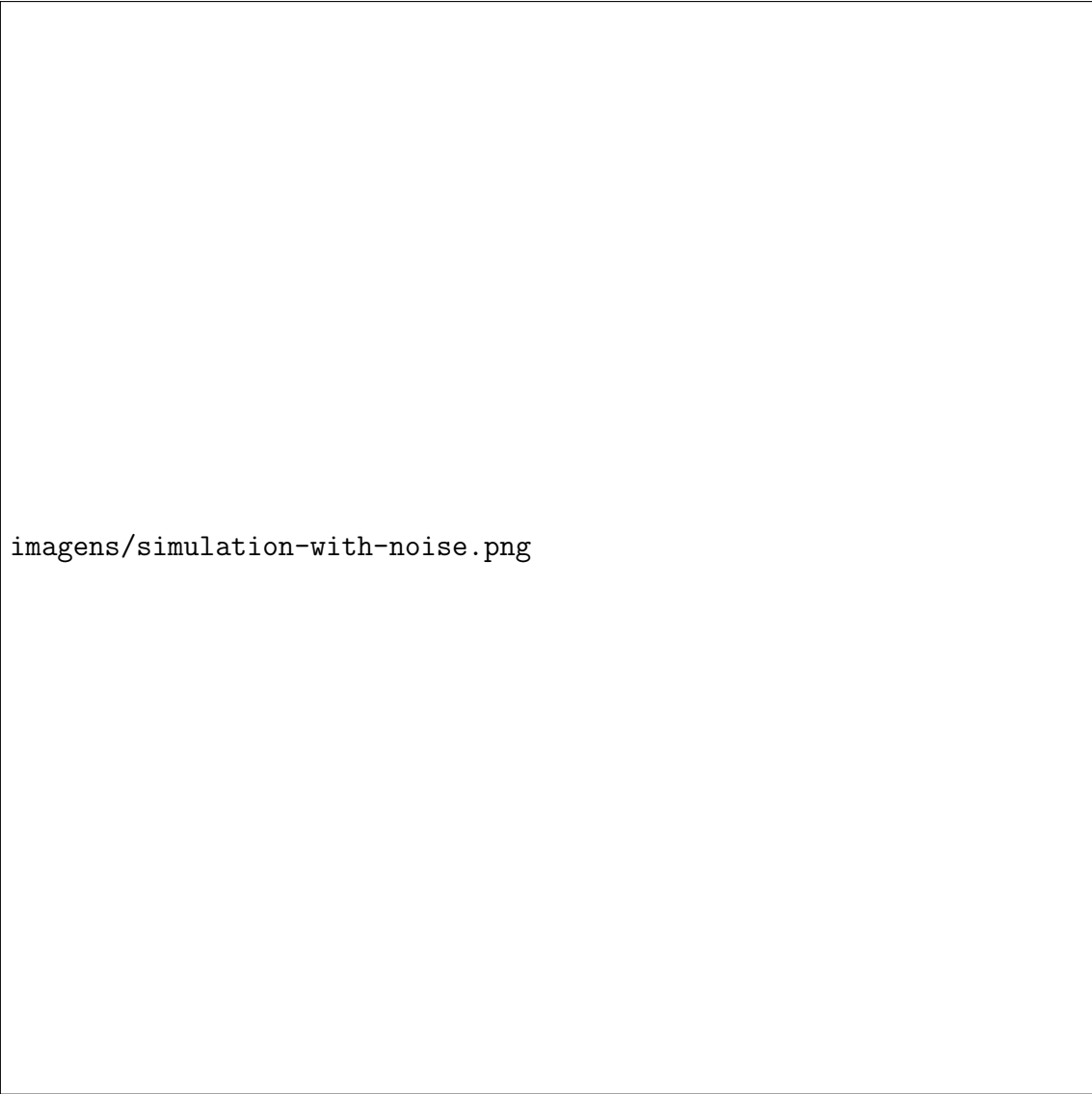
Para dois full adders (onde o primeiro recebe 1 no carry in e soma 1 com 1 e o segundo recebe como carry in o carry out do primeiro e faz a mesma soma) obtivemos os seguintes resultados presentes dos gráficos das imagens 9 a 11.



`imagens/simulation-without-noise.png`

imagem 9: resultados da simulação sem ruído

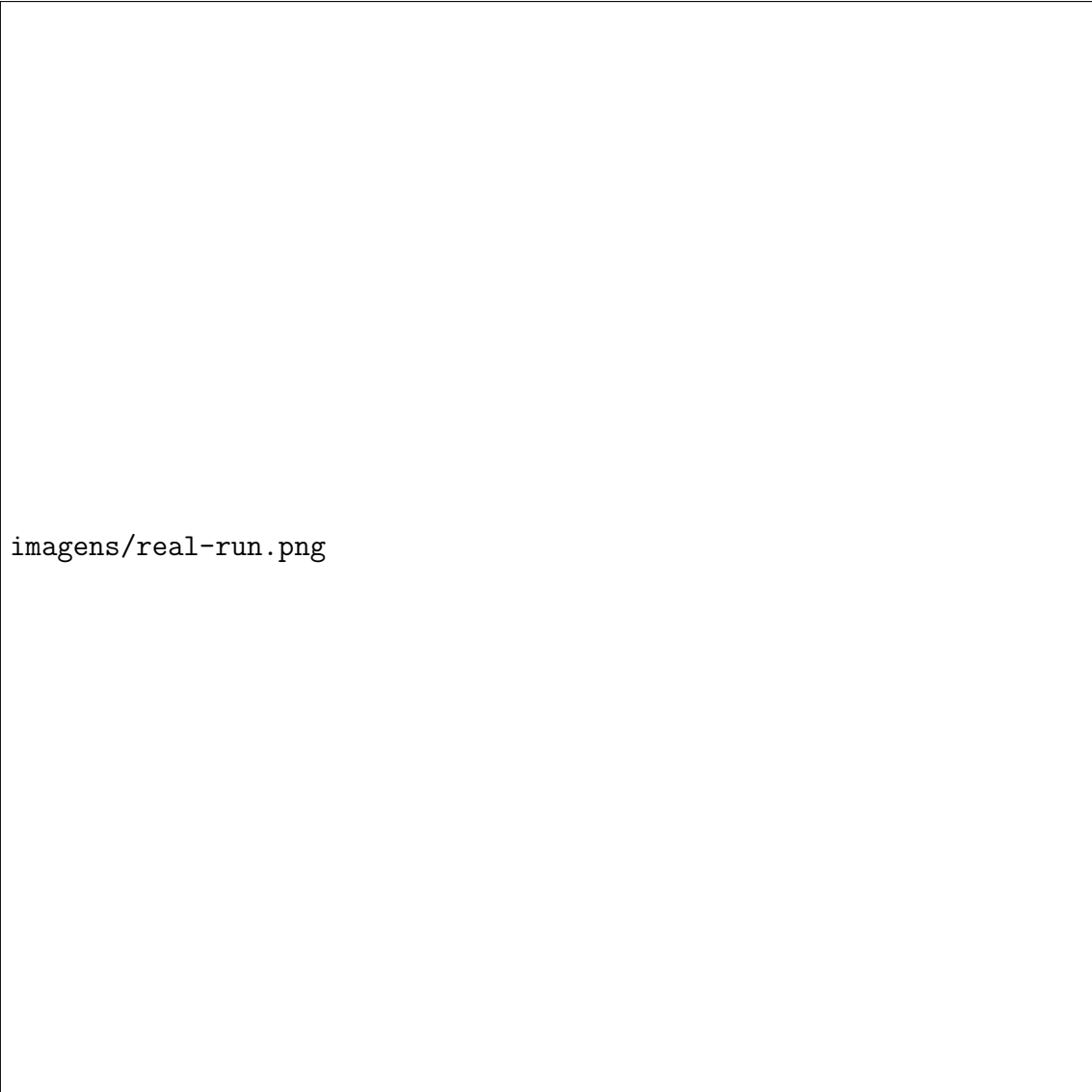
Como esperado todos os qbit's medidos tem valor 1.



`imagens/simulation-with-noise.png`

imagem 10: resultados da simulação com ruído

O ruído simulado gerou alguns resultados diferentes - contudo o resultado majoritário ainda é o correto.



imagens/real-run.png

imagem 11: resultados da rotina no computador quântico da IBM (filtrando apenas os resultados com frequência superior a 10)

Note que apesar de haverem vários resultados com alta frequência - o resultado de maior frequência é o resultado correto.

2.4 Conclusão

O procedimento como um todo foi essencial para o desenvolver uma compreensão completa das propriedades da computação quântica, bem como suas diferenças para a computação clássica. Tais entendimentos serão fundamentais no entendimento, desenvolvimento de uma rede neural celular quântica.