Quantum Channel Decoding

Shahab Hamidi-Rad
Emerging Technologies Lab
InterDigital Inc.
Los Altos, California
Shahab.Hamidi-Rad@InterDigital.com

John Kaewell
Emerging Technologies Lab
InterDigital Inc.
Conshohocken, Pennsylvania
John.Kaewell@InterDigital.com

***Abstract*—Channel Coding is the technique that enables reliable delivery of digital data over unreliable communication channels. For most high performance channel coding techniques, the existing classical algorithms are computationally expensive, making them impractical for throughput-demanding applications with large code sizes. Today's Noisy Intermediate-Scale Quantum (NISQ) computers, although limited due to a modest number of qubits, short coherence time, and poor gate fidelity, are useful tools for exploring and experimenting with possible solutions to a wide variety of computational problems.**

In this paper we show how careful initialization of qubits combined with a simple quantum circuit, enables us to perform channel decoding for different linear block codes. We first explain our novel qubit initialization technique which we call "Quantum Soft Decision". We then show how to build a simple quantum circuit based on the Generator or Parity-check matrix using another technique called "Quantum Generator". Using these universal concepts, we implement Quantum Decoders for two different types of linear block codes, namely Hamming codes and Polar codes. Our simple quantum circuits achieve decoding performances comparable with best classical algorithms such as Maximum Likelihood (ML) for Hamming codes and Successive Cancellation (SC) and Successive Cancellation List (SCL) for Polar codes. Using Qiskit, we implemented and compared the decoding performance at different code sizes and noise levels on simulated (both ideal and noisy) quantum computers. Also using Amazon Braket, we verified the algorithm on real quantum computers.

***Index Terms*—Quantum Channel Decoding, Polar code, Hamming code, Quantum Soft Decision, Quantum Generator, Successive Cancellation, 5G, Channel Coding**

I. Introduction

Quantum algorithms utilize essential features of quantum physics such as superposition and entanglement to solve some problems faster than classical computers. For example, Shor's algorithm [1] and Grover's algorithm [2] are two of the best-known quantum algorithms that solve problems such as factoring large numbers and searching unstructured data respectively. As another example, Quantum Approximate Optimization Algorithm (QAOA) [3] is a hybrid quantum/classical algorithm for approximating solutions to combinatorial optimization problems. Quantum algorithms are usually described and implemented in the circuit model where a quantum circuit acts on one or more qubits using quantum operators called gates.

The goal of channel coding is to devise codes that can be transmitted efficiently while enabling some error control capabilities such as error detection and error correction. Linear block codes are a category of codes with 2 main properties: They are applied to source bits in blocks and they are linear, which means modulo-2 sum of any two code-words is also a valid code-word. The Hamming codes [4] and Polar codes [5] are two different linear block codes used in this paper to demonstrate our quantum channel decoding approach.

Hamming code, a linear block code invented by Richard W. Hamming [4] in 1950, can detect one-bit or two-bit errors and correct one-bit errors. It is widely used in computer memory (i.e. RAM) where multiple bit errors happen very rarely. In Parity-check matrix of Hamming code, any two columns are pairwise linearly independent. The most accurate classical Hamming decoding algorithm is the exhaustive Maximum Likelihood (ML) whose computational complexity grows exponentially with message size.

Polar code is another type of linear block code which was proposed by Erdal Arikan [5] in 2009 and became famous due to some of its desirable characteristics. It can be proved explicitly that Polar codes approach Shannon capacity for a wide range of communication channels. The encoding process is significantly simpler compared to other methods such as Low Density Parity Check (LDPC) [6], [7]. However, due to decoding inefficiency for large code lengths, Polar codes are currently used only for control channels in the 5G standard [8], [9].

In this paper we demonstrate how to build quantum circuits capable of solving channel decoding problems by exploiting a) quantum superposition for initialization of the qubits and b) quantum entanglement for quantum implementation of a "Generator" or "Parity-check" matrix.

Although the quantum circuits presented here do not achieve quantum advantage, the methods explained for "Quantum Soft Decision" and "Quantum Generator" could be used in future research to improve performance and possibly achieve quantum advantage over the classical decoding algorithm.

II. Motivation and Background

As mentioned before, the best Hamming decoding algorithms are based on the computationally expensive Maximum Likelihood approach and the existing classical decoding algorithms for Polar codes are sequential or partially-sequential in nature [10]. The Successive Cancellation (SC) [5] and Successive Cancellation List (SCL) [11] algorithms involve navigating through a binary tree in a depth-first-search manner.

978-1-6654-9113-6/22/$31.00 ©2022 IEEE
DOI 10.1109/QCE53715.2022.00141

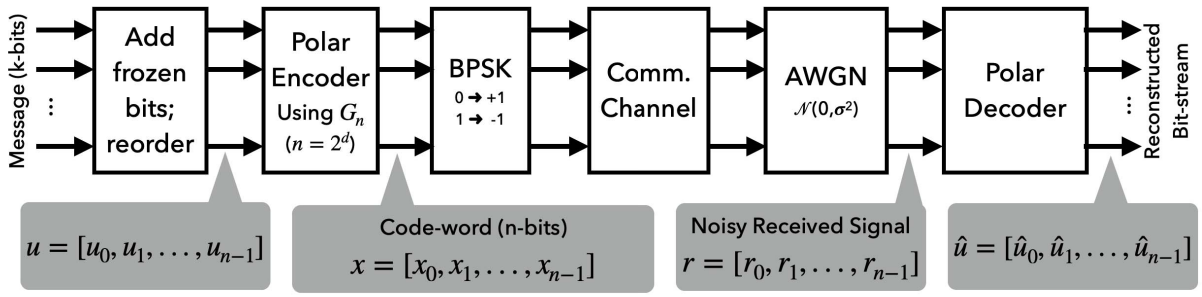


Fig. 1: Simplified communication pipeline for Polar code. Hamming codes use a similar (simpler) pipeline

At large code lengths, the decoding complexity leads to higher latency which eventually makes Polar codes impractical for most throughput-demanding applications.

In the rest of this document we consider bit-streams as row vectors with first element representing the most significant bit. All vector and matrix multiplications involving bit-streams are based on modulo-2 arithmetic unless otherwise specified. We use the (n, k) notation to specify a channel coding configuration with n-bit code-words and k-bit messages. We also use Frame Error Rate (FER) to measure the performance of different decoding algorithms where each k-bit message is considered a single frame.

A. Hamming codes: Problem Formulation

Hamming codes usually use code-words of size $n = 2^r - 1$ bits for messages of length $k = 2^r - r - 1$ bits for any integer $r \geq 2$. Consider a k-bit message $\boldsymbol{u}$ being encoded based on the Hamming code Generator matrix $\boldsymbol{G_{(n,k)}} \in \{0,1\}^{k \times n}$ to create the n-bit code-word $\boldsymbol{x}$:

$$\boldsymbol{x} = \boldsymbol{u}.\boldsymbol{G_{(n,k)}} \tag{1}$$

For the example of $(7, 4)$ Hamming code, the 4×7 Generator matrix $\boldsymbol{G_{(7,4)}}$ and the 3×7 Parity-check matrix $\boldsymbol{H_{(7,4)}}$ are:

$$\boldsymbol{G_{(7,4)}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\boldsymbol{H_{(7,4)}} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3}$$

The code-word $\boldsymbol{x}$ is then modulated and transmitted over the communication channel. For simplicity and without loss of generality, we are assuming Binary Phase Shift Keying (BPSK) modulation which uses +1 and -1 to signal bit values 0 and 1 respectively. At the receiving side, this signal is subject to an Additive White Gaussian Noise (AWGN) and we receive the noisy signal $\boldsymbol{r} \in \mathbb{R}^n$. This signal is then fed to the Hamming decoding algorithm which outputs the reconstructed bit-stream $\hat{\boldsymbol{u}}$.

B. Polar codes: Problem Formulation

Fig. 1 shows a simplified pipeline for Polar coding with message length k and code-word length $n = 2^d$. First the message needs to be embedded in the bit-stream which involves inserting $n - k$ zeros at frozen bit indexes and reordering the message bits to create the bit-stream $\boldsymbol{u}$. Throughout the rest of this paper we use $\mathcal{F}$ and $\mathcal{M}$ for the set of frozen and message bit indexes respectively, both of which are derived from a predefined "Reliability Sequence" for Polar codes [8], [9]. To obtain the transmitted code-word $\boldsymbol{x}$, the bit-stream $\boldsymbol{u}$ is multiplied by the Polar Generator matrix $\boldsymbol{G_n}$:

$$\boldsymbol{x} = \boldsymbol{u}.\boldsymbol{G_n} \tag{4}$$

where the Polar code Generator matrix is defined as:

$$\boldsymbol{G_n} = \boldsymbol{G_{2^d}} = \boldsymbol{G_2}^{\otimes d} \text{ and } \boldsymbol{G_2} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{5}$$

The code-word $\boldsymbol{x}$ is then modulated and transmitted over the communication channel and subjected to AWGN at the receiver where we receive the noisy signal $\boldsymbol{r} \in \mathbb{R}^n$. This signal is then fed to the Polar decoding algorithm which outputs the reconstructed bit-stream $\hat{\boldsymbol{u}}$. By removing the frozen bits from $\hat{\boldsymbol{u}}$ (using the indexes in $\mathcal{F}$) and reordering the remaining message bits (using the indexes in $\mathcal{M}$), we can obtain the k-bit predicted message.

III. Quantum channel decoder circuit design

As explained in the previous sections, the goal of a channel decoding algorithm is to output the reconstructed message $\hat{\boldsymbol{u}}$, given the noisy received signal $\boldsymbol{r}$. To implement a quantum algorithm, first we need to find a way to initialize our qubits in superposition states based on the analog values received in $\boldsymbol{r}$, and second, we need to use a set of quantum gates to simulate the effect of Generator or Parity-check matrix.

A. Quantum Soft Decision

Assuming a noise power of σ^2, we can calculate the probability of j^{th} BPSK-modulated code-word bit being '1' as:

$$P(x_j = 1 | r_j) = \frac{1}{1 + e^{2r_j/\sigma^2}} \tag{6}$$

Now suppose an $R_y(\theta_j)$ gate is applied to the j^{th} qubit in its initial state $|0\rangle$. The matrix representation of $R_y(\theta_j)$ is:

$$R_y(\theta_j) = \begin{bmatrix} \cos\theta_j/2 & -\sin\theta_j/2 \\ \sin\theta_j/2 & \cos\theta_j/2 \end{bmatrix} \tag{7}$$

The new state of the qubit after applying $R_y(\theta_j)$ is:

$$|\psi_{\theta_j}\rangle = \begin{bmatrix} \cos\theta_j/2 & -\sin\theta_j/2 \\ \sin\theta_j/2 & \cos\theta_j/2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta_j/2 \\ \sin\theta_j/2 \end{bmatrix} \tag{8}$$

If we measure this qubit in computational basis, we get a '0' with probability $\cos^2\theta_j/2$, and '1' with probability $\sin^2\theta_j/2$:

$$P(q_{meas_j} = 1|\theta_j) = |\langle 1|\psi_{\theta_j}\rangle|^2 = \left(\begin{bmatrix} 0 & 1 \end{bmatrix} . \begin{bmatrix} \cos\theta_j/2 \\ \sin\theta_j/2 \end{bmatrix} \right)^2$$

$$\Rightarrow P(q_{meas_j} = 1|\theta_j) = \sin^2\theta_j/2 \tag{9}$$

Now using (6) and (9) we can initialize the j^{th} qubit in a superposition state based on value of the j^{th} received signal r_j:

$$P(q_{meas_j} = 1|\theta_j) = P(x_j = 1|r_j)$$
$$\Rightarrow \sin^2\theta_j/2 = \frac{1}{1 + e^{2r_j/\sigma^2}}$$

Solving for θ_j:

$$\theta_j = 2\arcsin\sqrt{\frac{1}{1 + e^{2r_j/\sigma^2}}} \tag{10}$$

B. Quantum Generator

As mentioned before, the multiplication of a bit-stream vector by a Generator or Parity-check matrix uses modulo-2 arithmetic which is similar to the XOR operations (i.e. $1 \oplus 1 = 0$). This means the whole matrix multiplication process can be implemented using a set of XOR gates in a binary logic circuit. The entries in $\boldsymbol{G}$ or $\boldsymbol{H}$ matrices specify the pairs of bits for each XOR gate and the order these gates are applied.

The closest quantum gate to the logical XOR gate is the "Controlled-Not" gate CX which entangles its two input qubits. Algorithm 1 receives a Generator matrix as input and creates a list of qubit pairs for entanglement. Each pair in the list specifies the two qubits to be entangled (using CX gate) and the order in the list specifies the order these gates are applied. The state of qubits after these gates is a "soft" representation of $\hat{\boldsymbol{u}}$. A similar algorithm can be implemented for Haming codes.

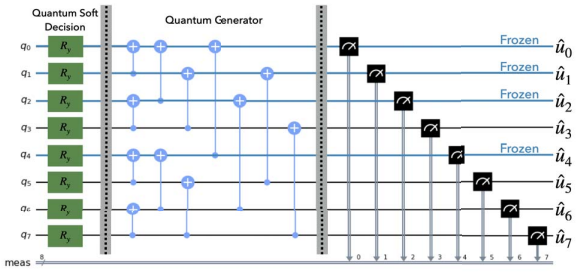
Fig. 2a shows the quantum circuit for the (8,4) Polar code. Notice that q_0, q_1, q_2, and q_4 represent the frozen bits in the (8,4) Polar code. Fig. 2b shows the simplified version. Line 10 in Algorithm 1 removes the qubit pairs from P_G when the target qubit is frozen and the control qubit is not frozen.

Using the ML approach we can improve the performance of our quantum decoder significantly at the cost of slight increase in computational complexity. In ML decoding, we first create a list of code-words $\hat{\boldsymbol{X}}$ containing a code-word $\hat{\boldsymbol{x}}$ for each predicted message $\hat{\boldsymbol{u}}$ in the list returned by the quantum circuit. Each code-word $\hat{\boldsymbol{x}}$ is obtained by encoding the predicted bit-streams $\hat{\boldsymbol{u}}$ using the Generator matrix.

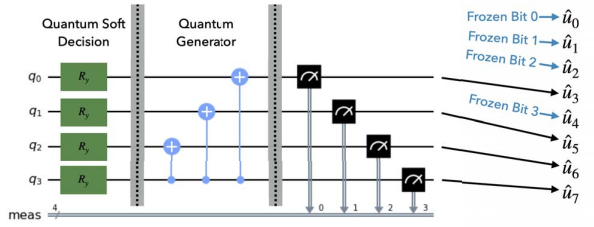
Algorithm 1 Quantum Generator for (n,k) Polar code

Input: $n \times n$ Generator matrix $\boldsymbol{G_n}$, k
Output: P_G ▷ A list of qubit pairs

1: $D \leftarrow \log_2 n$ ▷ n is always a power of 2
2: $P_G \leftarrow []$ ▷ Initialize with empty list
3: **for** $d \leftarrow 0$ to $D-1$ **do**
4: **for** $s \leftarrow 0$ to $2^d - 1$ **do**
5: **for** $i \leftarrow s$ to $n-1$ step 2^{d+1} **do**
6: Add $(i + 2^d, i)$ to P_G
7: **end for**
8: **end for**
9: **end for**
10: Remove the pairs with unused frozen qubits from P_G
11: **return** P_G



(a) Quantum Circuit for (8,4) Polar code



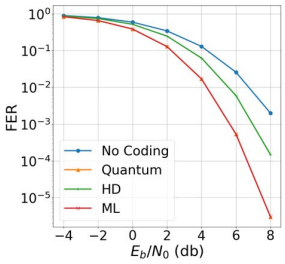
(b) Simplified Circuit for (8,4) Polar code

Fig. 2: Quantum circuits for (8,4) Polar code. (a) The circuit includes qubits corresponding to the frozen bits of Polar code. (b) A simplified version of the circuit with the frozen qubits removed. In this case the $\hat{\boldsymbol{u}}$ bit-stream can be retrieved by inserting zeros at the missing frozen bit indexes ($\hat{u}_f = 0, \forall f \in \mathcal{F}$).

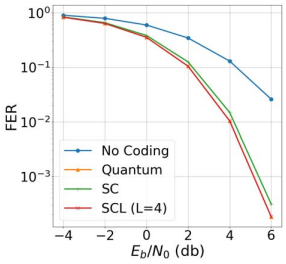
Once we have the list of predicted code-words $\hat{X}$, we can calculate the correlation between these code-words and the noisy received signal $\boldsymbol{r}$ and pick the code-word with the highest correlation.

IV. Experiments

We used different classical algorithms for Hamming and Polar codes and compared the decoding performance with our quantum approach at different code sizes and E_b/N_0 ratios. As you can see in tables I, II, and Fig. 3, the performance of quantum decoder is very close to the best classical algorithms (i.e. ML for Hamming and SCL for Polar codes). In these



(a) (15,11) Hamming code



(b) (16,11) Polar code

Fig. 3: Frame Error Rate at different E_b/N_0 ratios for (a) (15,11) Hamming code with Hard-Decision (HD), Maximum Likelihood (ML), and Quantum decoding and (b) (16,11) Polar code with Successive Cancellation (SC), Successive Cancellation List (SCL), and Quantum decoding

tables and charts, HD is the Hard Decision method, ML is the Maximum Likelihood algorithm. "No Coding" represents the case where no channel coding method was used. SC and SCL represent Successive Cancellation and Successive Cancellation List algorithms for Polar decoding respectively.

Note that the Maximum Likelihood algorithm provides the best possible solution for the Hamming decoding problem as it is based on exhaustive search. For Polar codes, SCL is considered the best algorithm.

TABLE I: FER for different Hamming Decoders

Hamming code (n,k)	E_b/N_0 **(dB)**	**Frame Error Rate (FER)** **HD**	**ML**	**Quantum**
(7,4)	0	0.261960	**0.178925**	0.179491
	4	0.036397	**0.011786**	0.011794
	8	0.000271	**0.000012**	**0.000012**
(15,11)	0	0.518580	**0.385033**	0.385035
	4	0.062622	**0.016960**	**0.016960**
	8	0.000147	**0.000003**	**0.000003**

TABLE II: FER for different Polar Decoders

Polar code (n,k)	E_b/N_0 **(dB)**	**Frame Error Rate (FER)** **SC**	**SCL (L=4)**	**Quantum**
(8,5)	0	**0.245138**	**0.245138**	0.245572
	4	**0.025443**	**0.025443**	0.025446
	8	**0.000127**	**0.000127**	**0.000127**
(8,4)	0	0.172802	**0.164553**	0.167320
	4	0.009933	**0.008466**	0.008500
	8	0.000005	**0.000003**	**0.000003**
(16,9)	0	0.324273	**0.315687**	0.318697
	4	0.015282	**0.013819**	0.013844
	8	**0.000001**	**0.000001**	**0.000001**
(16,11)	0	0.381585	**0.352366**	0.353487
	4	0.014791	**0.010326**	0.010345
	8	0.000002	<0.000001	<0.000001

The error rates shown in tables I, II, and Fig. 3 are based on Qiskit state vector simulator [12].

Table III compares the results between the ideal simulation based on state vector and the Qiskit's "FakeMontreal" noisy simulation. As you can see the results are almost identical which means our quantum circuits are robustly resistant to different types of quantum noise.

TABLE III: FER for Ideal and Noisy Quantum Simulation

Polar code (n,k)	E_b/N_0 **(dB)**	**Frame Error Rate (FER)** **State-Vector**	**FakeMontreal**
(8,4)	0	0.166900	0.164250
	2	0.054450	0.053800
	4	0.008450	0.008450
	6	0.000400	0.000400
(8,5)	0	0.24740	0.24733
	2	0.09793	0.09780
	4	0.02627	0.02627
	6	0.00467	0.00467

We also tried decoding different Hamming and Polar code configurations using our quantum decoder circuits on different quantum computers and managed simulators available on Amazon Braket [13][1].

REFERENCES

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[3] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[4] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[5] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[6] R. Gallager, "Low-density parity-check codes," *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.

[7] D. J. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics letters*, vol. 32, no. 18, p. 1645, 1996.

[8] 3GPP, "Technical Specification Group Radio Access Network; NR; Multiplexing and channel coding," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.212, 2018.

[9] V. Bioglio, C. Condo, and I. Land, "Design of polar codes in 5g new radio," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 29–40, 2020.

[10] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 289–299, 2012.

[11] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[12] M. S. ANIS, Abby-Mitchell, H. Abraham, AduOffei, R. Agarwal, G. Agliardi *et al.*, "Qiskit: An open-source framework for quantum computing," 2021.

[13] AWS, "Amazon braket - amazon web services," https://aws.amazon.com/braket/, December 2019.

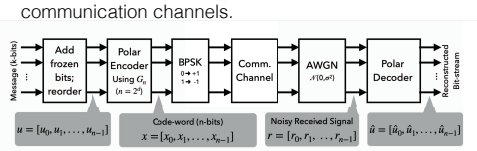
[1]The Python code for both Qiskit and Amazon Braket experiments is available at https://github.com/InterDigitalInc/QuantumChannelDecoding

Quantum Channel Decoding

Shahab Hamidi-Rad and John Kaewell

Problem

Channel Coding is the technique that enables reliable delivery of digital data over unreliable communication channels.



Code-words are obtained using a generator matrix: $\boldsymbol{x} = \boldsymbol{u} \,.\, \boldsymbol{G}$ (with modulo-2 additions)

$$\boldsymbol{G}_{(7,4)} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad G_4 = G_2 \otimes G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$G_{2^d} = G_2^{\otimes d}$$

Hamming Code — Polar Code

At the receiver, the noisy received signals must be decoded to obtain the original message.

Computational complexity of current classical decoding algorithms grow exponentially with code size.

Quantum Approach

1) **Quantum Soft Decision**

Use *superposition* to embed probabilistic information into qubit:

R_y gate:

$$R_y(\theta) = e^{(-i\theta Y/2)} = \begin{bmatrix} \cos\theta/2 & -\sin\theta/2 \\ \sin\theta/2 & \cos\theta/2 \end{bmatrix}$$

with:

$$\theta_j = 2\arcsin\sqrt{\frac{1}{1+e^{2r_j/\sigma^2}}}$$

$r_j \rightarrow$ j^{th} received signal
$\sigma^2 \rightarrow$ Noise power

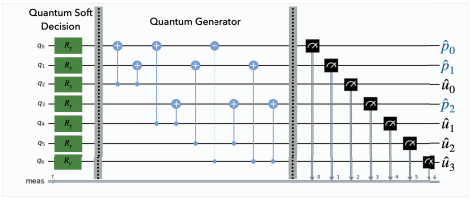
2) **Quantum Generator**

Use *entanglement* to mimic the generator matrix application (**CNOT** gates)

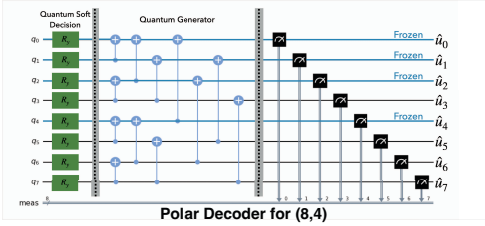
Design algorithms to generate a quantum circuit given a generator matrix.

Quantum Circuits

- r_j is an analog value representing the received noisy signal for the j^{th} bit.It is used to initialize qubits.
- Each **CNOT** gate mimics an XOR operation (modulo-2 addition)
- Measurements in computational basis provides the most likely decoding solution.



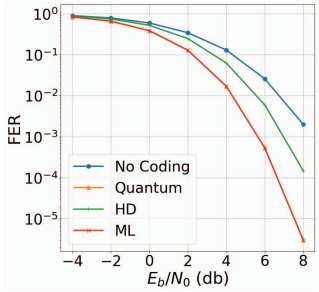
Hamming Decoder for (7,4)



Polar Decoder for (8,4)

Results

Frame Error Rate at different E_b/N_0 ratios with Hard-Decision (HD), Maximum Likelihood (ML), and Quantum decoding for (15,11) Hamming code with fixed σ^2



Frame Error Rate at different E_b/N_0 ratios with Successive Cancellation (SC), successive cancellation List (SCL) and quantum decoding for (16,11) Polar code with fixed σ^2