Journal of Electrical Engineering Vol. 74, No. 3, 2023, pp. 154-166

sciendo

PAPER

Entaglement and communication in digital qubits using FPGAs

Bhuvan Hawargi, Akshay K B, Kaustav Bhowmick

This paper focuses on usage of digital qubits on a digital quantum computing platform implemented on FPGAs. Modifications have been made to existing digital qubit standards to account for complex probability amplitudes and not only real ones. This paper introduces an improved transient effect ring-oscillator based TRNG used in the scope of this project. The objective of this was to simulate the addition of environmental noise in a digital setting. Using the newly designed digital qubits, a few quantum logic gates have been designed to work on the FPGA platform using Verilog. These logic gates have then been used to implement entanglement on a digital hardware platform. Furthermore, this paper implements communication between two digital quantum computers over both wired and wireless media by transmitting alphabetical messages as qubits and compares the same with transmission involving classical digital bits only.

Keywords: Video Surveillance, Human Detection, Human activity recognition (HAR), Abnormal behavior, 3D Convolutional Neural Network (CNN), Deep Neural Architecture

1. Introduction

In the recent past, the knowledge of the principles of Quantum mechanics, namely superposition and entanglement, has led to new novel computational paradigms which utilize phenomena like the polarization of light waves [1]. Quantum computing makes use of these physical effects at small, quantum levels, providing significant improvements in terms of computational speed-up and super-dense coding [2]. However, the dependence of quantum computers on unstable physical processes can lead to the decay of information. To achieve consistent speed-ups and run algorithms needing several hundred qubits, quantum computers could work simultaneously, like a cluster. This leads to the need for a reliable method of information transmission between the quantum nodes [3]. For this, the concept of digital qubits may be used in the capacity of co-processors to speed-up the present quantum-computing nodes.

While real quantum computers are probabilistic devices that are analog in nature, FPGA based quantum co-processors are of digital nature. These devices make use of digital qubits, which in turn are based on classical logic circuits with probabilistic outputs [4]. Digital qubits allow for the offloading of some calculations that can benefit from parallelization. Performing majority of the calculation on a classical computer will reduce the queuing times, allowing the real quantum computer to focus on the core parts which cannot be otherwise emulated. The qubit state θ can be represented as movement of a unit radius vector, as shown in Fig. 1.

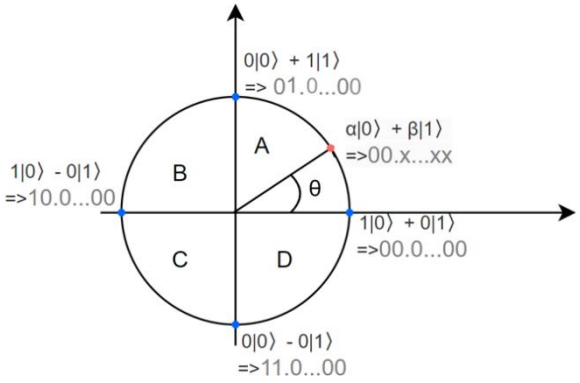


Fig. 1. Polar coordinate system for θ from [4]

Department of Electronics and Communication Engineering, PES University, Bengaluru 560064, India
bhuvanhawargi@pesu.pes.edu

https://doi.org/10.2478/jee-2023-0021, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

As indicated in Fig. 1, change in position of the vector affects a single parameter in polar coordinates, angle θ. The radius remains unchanged. The state of the qubit itself can be represented by a binary sequence of some fixed length. The work done in this paper builds upon the concept of digital qubits from [5] and uses the same by transmitting qubit information over either a wired or wireless medium and compares some observed characteristics.

In the scope of this paper, two methods have been used for the introduction of noise in digital qubits. The methods used are: a) Pseudo Random Number Generator, and b) True Random Number Generator. The implemented PRNG follows the Xorshift architecture introduced in [6] and [7] while the TRNG is based on a transient effect ring oscillator [8] and [9]. Both the noise models used introduce unpredictable interference, serving as the various quality factors found in true analog quantum computers [3]. This paper also considers two methods for communication between the digital quantum nodes. The wired medium makes use of a UART bridge [10], while the wireless mode makes use of the popular IoT platform ThingSpeak, connected the boards through the cloud. In section II, some fundamental concepts regarding qubits and digital qubits are discussed.

2. Mathematical basis for quantum computers

Quantum bits, also known as qubits, hold information in a complex sphere [11]. A single qubit with state $|\psi\rangle$ can be expressed as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where both α, β are of complex nature and $|\alpha|^2+|\beta|^2 = 1$. Expression (1) can also be expressed vectorially as $[p_0\ p_1]$. The states $|0\rangle$ and $|1\rangle$ can be considered as the ground and excited states, respectively, with binary 0 and 1 being their closest parallels from classical computation [11].

The state of a qubit may be changed in two fundamental ways: a) by unitary operations and b) by measurement.

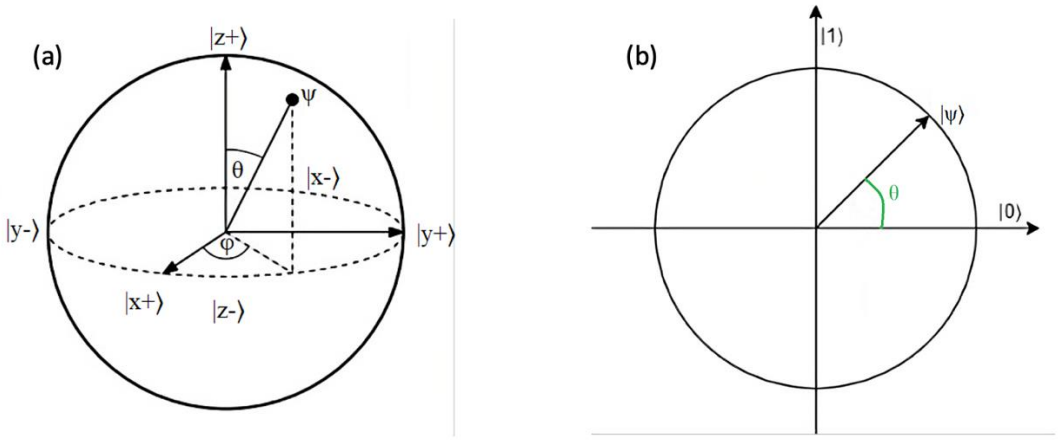


Fig. 2. (a) Bloch sphere representation of a qubit with complex amplitudes (b) Qubit representation with real amplitude

Unitary operations on the qubit are performed by constructs called quantum gates. The effects of such an operation can by represented by a displacement on the surface of a Bloch Sphere, a unit radius sphere with complex amplitudes [12] as shown in Fig. 2a. If only real amplitudes are considered, the Bloch sphere can be simplified as a unit circle, as shown in Fig. 2b and the probabilities of the pure states can be measured as following - probability $\cos^2\theta$ for $|0\rangle$ and $\sin^2\theta$ for $|1\rangle$. In the Fig. 2b, it must be observed that the pure state $|0\rangle$ is along the horizontal axis and $|1\rangle$ is along the vertical axis. The angle θ is measured from the horizontal axis in an anticlockwise manner.

Quantum logic gates unlike their classical counterparts are essentially matrices [13]. Applying these gates requires a multiplication between the matrices of the gates and the state vector of the qubit. However, when multiple qubits are involved, an operation called the Kronecker or tensor product takes place. The product of the individual state-vectors results in a matrix with larger dimensions. One such example can be seen in equation 2.

$$|mn\rangle = m \otimes n = [p_{00}\, p_{01}\ p_{10}\ p_{11}] \tag{2}$$

In equation (2), *m* and *n* are two qubits, whose state-vectors are multiplied, resulting in a larger vector of dimensions 1x4. As yet, quantum computers do not work in a standalone manner - they work alongside a classical computer and merely operate as quantum co-processors [14]. Quantum computers can also offload some calculations required by an algorithm to an FPGA, making use of its highly parallel nature [15] [16]. For example, the efficiency of Shor's Algorithm can be improved by implementing the required Quantum Fourier Transform on a FPGA, allowing multiple lines of the circuit to be run simultaneously [15] [17]. This dependence of quantum computers on their classical counterparts requires the concept of digital qubits discussed in Sections 3 and 4 of this paper.

3. Earlier work

In the previous works of V. Hlukhov and B. Havano, digital qubits were implemented only for the case of real amplitudes [5]. Out of the possible approaches - Cartesian and Polar coordinates, [5] focuses on using the Polar system. As the total probability of the individual elements of the state vector is equal to 1. This way, it is necessary to only specify and define the angle θ, thus making it simpler than tracking x and y coordinates. An example of this can be observed in Fig. 3a.

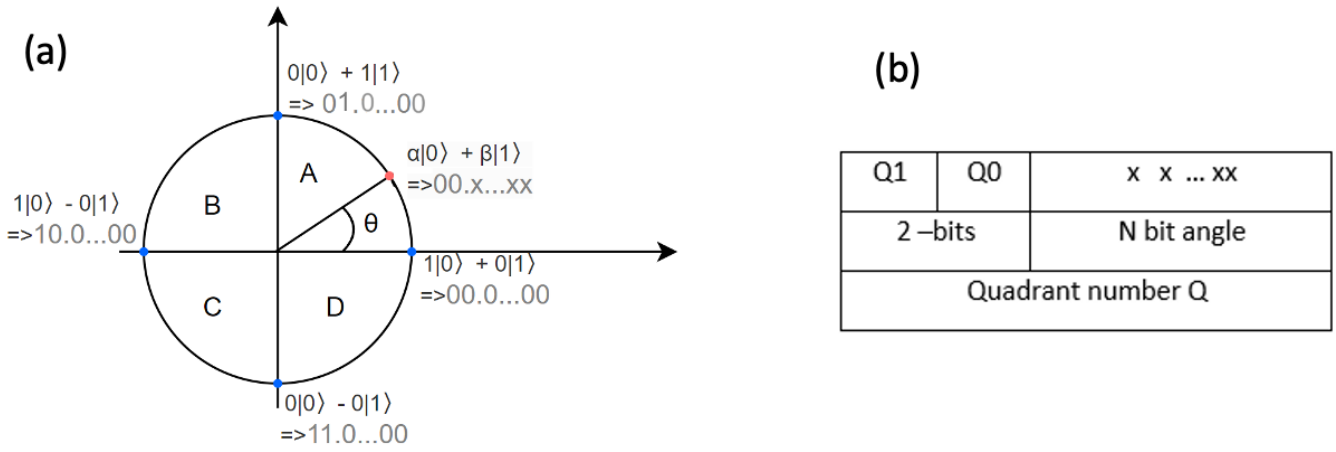


Fig. 3. (a) Unit circle qubit representation from [14]. (b) Data formatting used to represent qubits in unit circle [5].

The co-processor receives data from the classical computer in the format QQS...S, as present in Fig. 3b, where the first two bits, denoted by Q, represents the quadrant that the vector falls under. The remaining bits, denoted by a sequence of S, corresponds to the sector within each quadrant. The midpoint of this range is taken exactly as QQ100...0. During measurement, all the angles are reduced to angles in the first quadrant, with angle $\theta \in (0, \pi/2)$. The values indicated by QQ0...00 are peculiar. They belong to both quadrants simultaneously and can inscribe both state $|0\rangle$ & $|1\rangle$.

However, this model cannot account for complex amplitudes of the pure state of the qubit. While arguments about how a 2 qubit state with complex coefficients can be converted to a real valued 3 qubit system has been discussed in the works of [18], making a few changes to the model described by V. Hlukhov in [5] and [4] would allow for the use of these complex probability amplitudes. This ideology is comprehensively discussed in the subsection 4.5 of this paper.

In the works of [19], ring oscillator based true random number generator was implemented. This circuit made used of inverter gates, MUXes and a D flip-flop (see Fig. 4).

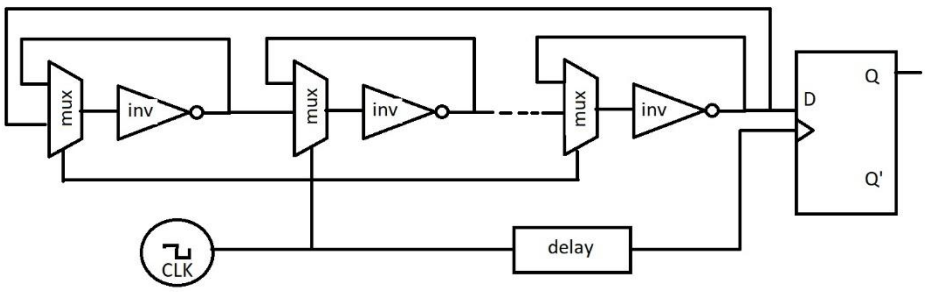


Fig. 4. Meta ring-oscillator structure from [19]

In Fig. 4, an odd number of inverter gates is used. The switching blocks, MUX, are being used to switch the overall circuit between a meta-stable and generating mode. The final output is rendered at the out port of the D flip-flop.

4. Methodology

This section of the paper introduces the FPGAs and other components used. In a sequential manner, the input to the digital quantum co-processor, introduction of noise in the system, modifications to the digital qubit, python interfacing, entanglement and transmission are discussed.

4.1 Hardware components

The digital quantum co-processor in this paper is implemented on an Avnet Digilent ZedBoard. The ZedBoard ships with a Zynq-7000 SoC tightly coupled with an ARM processor, allowing for multiple UART bridges and custom designed Verilog quantum logic gates. The FPGA also comes with an onboard Analog to Digital converter - XADC, which is capable of 1 mega samples per second. The XADC has a 12-bit resolution and supports voltages between 0 and 1 volts [20].

The module used for wireless transmission in this paper was an ESP8266-01. This module is a low cost SoC that connects the ZedBoard to a wireless network and operates in the 2.4 GHz spectrum [21].

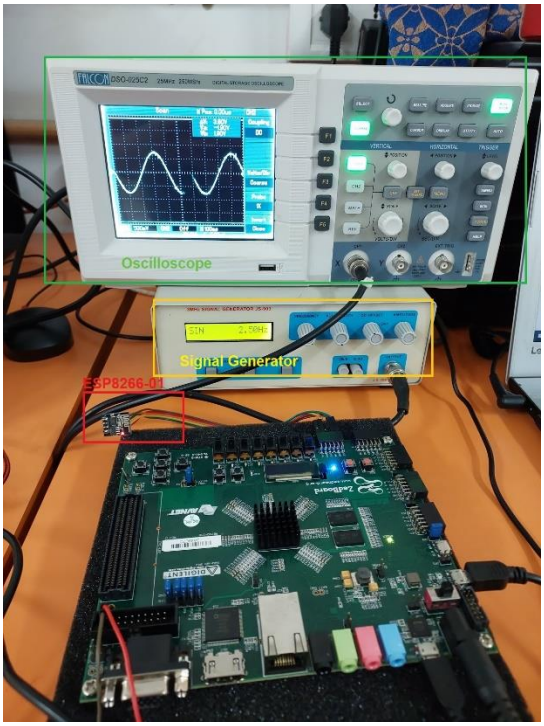


Fig. 5. Important components used in this paper - ZedBoard FPGA, ESP 8266-01, signal generator, and oscilloscope.

In Fig. 5, the following components have been marked: a) signal generator, b) oscilloscope, c) FPGA d) ESP 8266-01. However, in Fig. 5 the TRNG is not pictured and is show in in Fig. 7b.

4.2 Input

To provide input to the quantum co-processor, a real world signal was emulated. A real world signal is characterized by its random unpredictable frequencies and amplitudes. However, a more controlled setup was used. First, a periodic sinusoidal signal with frequency 2.5 Hz was taken from a signal generator. A periodic signal however does not contain any information due to its repetitive nature [22].

To introduce the randomness element, a software defined PRNG was used initially. The PRNG works on the principle of XOR-shift introduced in [6] and [7]. A shift register of size 8 was defined. On every clock-cycle, the initial fill of the register was XORed with another previously defined sequence of the same length. The resulting sequence was then shifted left by one bit before the cycle repeated itself. Unfortunately, due to the operating principle of the PRNG, the output of the signal can be predicted if the initial fill and modifier are known. Although it does not pose a serious threat in the application discussed in this paper, it is a serious drawback to cryptographic applications. For this purpose, a hardware based TRNG was chosen and implemented, discussed in section IV-C. In Fig. 6, the graph indicated in red is the sampled, unaltered signal which acts as the base.

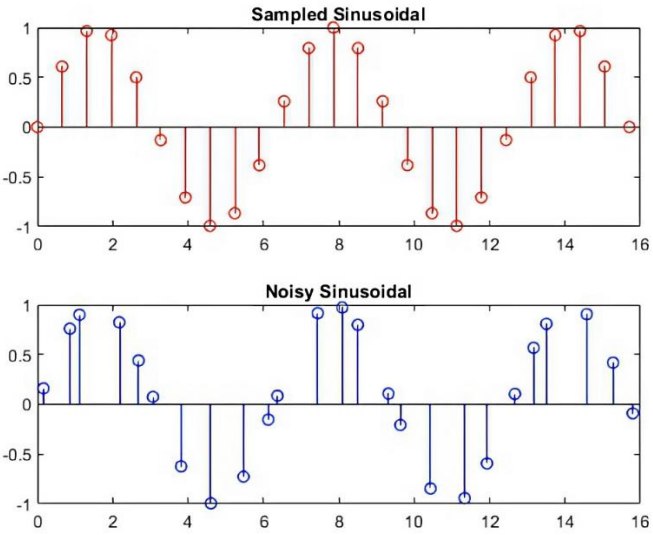


Fig. 6. Input to the system visualized in MATLAB

The graph marked with blue sampled in Fig. 6 is the noisy, emulated real world signal. This is created by introducing an error of $\pm 2^8$. This number arises from the length of the noise sequence.

4.3 True random number generator

A true random number generator (TRNG) does not depend on an algorithm but rather on a physical process. TRNGs provide statistically random ”noise” signal. The generator used in the scope of this paper is uses a transient effect ring oscillator coupled with a counter [9,23]. The ring oscillator is comprised of NOT gates connected sequentially as a ring. The ring oscillator as a whole is then connected to a NAND gate and JK flip flops as indicated by the schematic in Fig. 7a. This schematic was designed using Proteus circuit simulator [24].

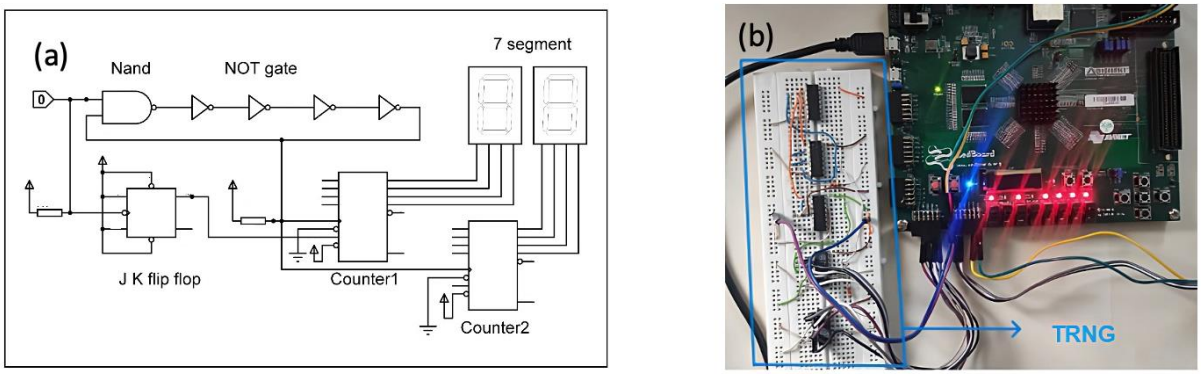


Fig. 7. (a) Proteus schematic of TRNG implemented, (b) ZedBoard with external TRNG circuit.

As seen in Fig. 7a, the oscillator module flips the state of the flip flop. This leads to an update signal to two separate 4-bit counters asynchronously. By concatenating the outputs of the individual counters, a noise signal of 8-bits was realized. This particular design for the random number generator was designed as a circuit matching the required length of noise sequence was not available in literature.

To verify the randomness of the sequence produced by the TRNG, the circuit was tested against NIST's guidelines [25] using a pre-built test suite by S. K. Ang [26]. A sequence of a thousand bits from the TRNG were recorded and used as the input to the test suite. 16 unique tests based on p-value were tested to check for randomness. The calculated scores and their implication are discussed in section V. As the TRNG was not the target of the project, an external TRNG was used instead of an onboard one, enabling ease of operation. Due to its asynchronous design, the overall inter-working remains unaffected.

4.4 Digital qubits

As discussed in section 3, a binary string representation was used for the digital qubit. In this system, a 5-bit binary sequence was used for representation indicated by Fig. 8. Two bits of the total five indicate the quadrant and the remaining 3 bits for the sectors within the quadrants as previously shown in [5].

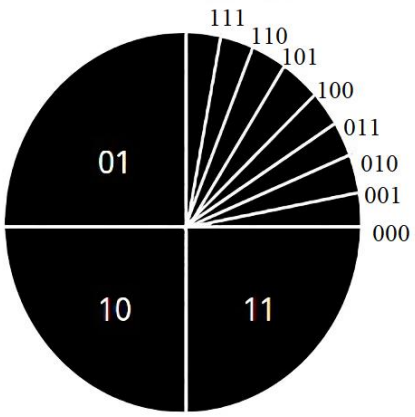
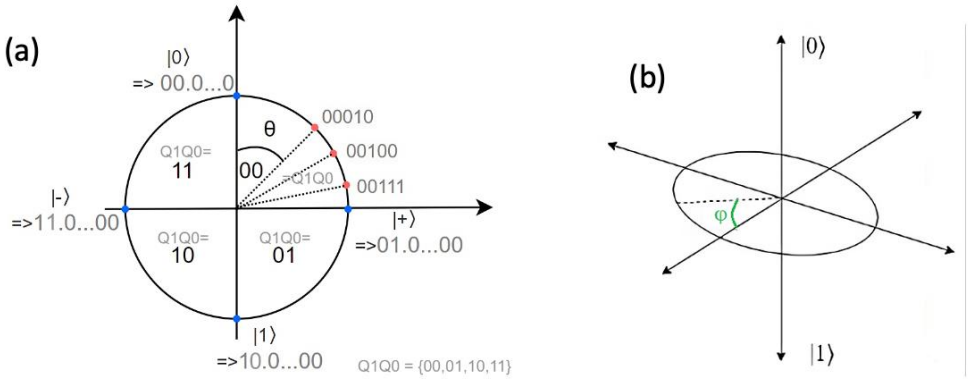


Fig. 8: Implemented unit circle qubit

From Fig. 8, the number of sectors per quadrant is dictated by the number of bits being used. As 3 bits are used to represent angle θ, we get $2^3-1 = 7$ sectors. The midpoint of this quadrant is indicated by the 100 line. To allow for complex amplitudes, this representation is repeated and doubled. This effectively makes the digital qubit 10-bits in length. By projecting two unit circles, the entire Bloch Sphere is covered. The $|0\rangle$ state lies along the positive z axis while the $|1\rangle$ lies along the negative z axis as indicated in Fig. 9a. To enable the rotation of this unit circle on the XY plane, a new angle - ϕ is introduced. This angle represents the phase of the qubit.

The system for phi follows a similar structure to that of Fig. 8. The line becomes a top-down view of the vertical unit circle on the XZ plane, shown in Fig. 9b. In Fig. 9a, sector numbers increase while moving in the clockwise direction. The quadrant numbers are also similarly affected. The data formatting method used does not change and remains similar to Fig. 3b. However, this causes two superposition states to lie on the horizontal axis, thus giving them a unique representation each, as seen in Fig. 9c.



(c)

State	Q1	Q0	XXX
$\|0\rangle$	0	0	000
$\|1\rangle$	1	0	000
$\|+\rangle$	0	1	000
$\|-\rangle$	1	1	000

Fig. 9. (a) Modified representation of a qubit on unit circle, (b) Angle Phi on the horizontal XY plane, (c) Modified data formatting used to represent qubits

4.5 Python interfacing

For visualization purposes, the FPGA was connected to a PC terminal over UART. The ZedBoard would then send back 10bit strings using the pySerial Library [27]. These string were plotted onto the Bloch Sphere. The angles θ and ϕ were calculated here in degrees, allowing for reconstruction of the qubit from 2 angles. In Fig. 10, the 10-bit binary string represents the qubit at a particular instance.

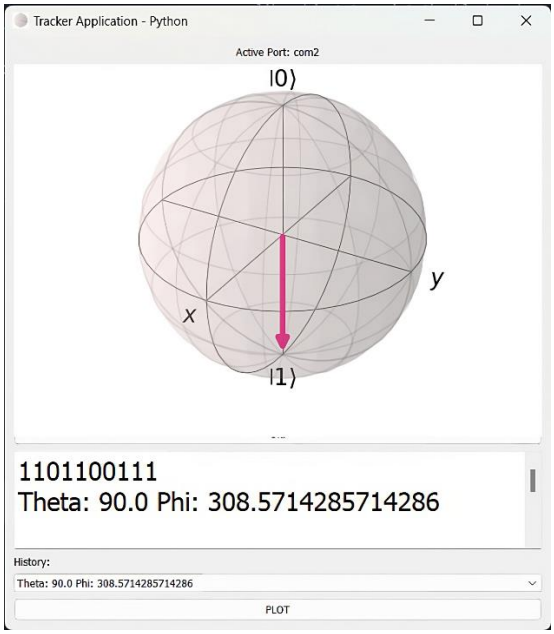


Fig. 10. Python GUI for interfacing with FPGA. Initial 5 bits handle angle phi. Theta is handled by remaining 5 bits.

In Fig. 10, the two angles – theta and phi were calculated to allow for the recreation of the qubit. The interface in Fig. 10 is also capable of visualizing all instances of the qubit from the recorded logs.

4.6 Entaglement

To create all the maximally entangled states for 2 qubits, the process of creating the $|\Phi^+\rangle$ state was followed [28]. For this circuit, the Hadamard and CNOT gates were implemented in Verilog to execute on the FPGA. The H gate was applied to both qubits. This step was followed by the application of a CNOT gate, with the first qubit acting as the control and the latter as the target. This generated the four possible entangled states for 2 qubits as indicated by Fig. 11.

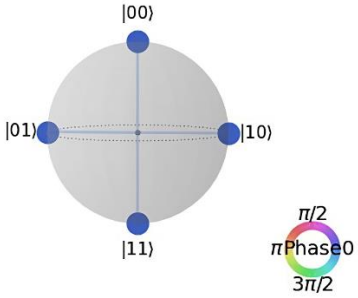


Fig. 11. All maximally entangled states for 2 qubits visualized as a Q-state sphere.

In Fig. 11, the states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ all have the same phase of 0°. The state vector after this operation can be expressed through equation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (3)$$

In equation (3), from left to right the matrices indicate CNOT, two-qubit parallel Hadamard gate and the starting state $|00\rangle$.

4.7 Transmission

For the wireless mode of transmission, the cloud IoT platform ThingSpeak was used. Using the ESP 8266-01 (see Fig. 5, the ZedBoard connects to the internet and transmits data by making use of AT commands. The HTTP POST method sends the individual qubits to an API token, which is then retrieved by the receiving board. In order to set a baseline for comparison and establish a controlled set of readings, the two FPGAs were connected directly using a GPIO defined UART bridge as seen in Fig. 12.

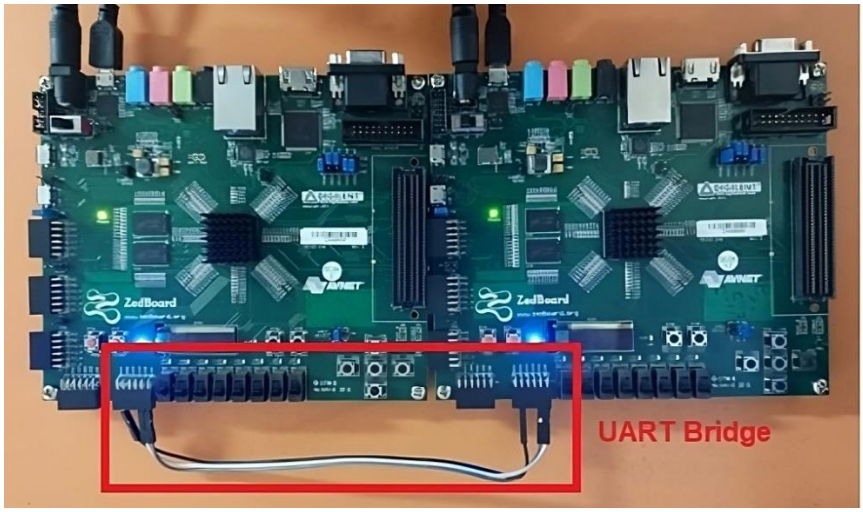


Fig. 12. Wired channel connection of ZedBoard

To, illustrate the transmission process, consider the word ‘hello’ as the message to be transmitted. The overall process is shown in Fig. 13. Each character in the message word is first converted to its binary equivalent. The generated binary bits are taken individually converted to its qubit equivalent. Both bits 0 and 1 have fixed qubit representations. The values shown in Fig. 13 are merely a sample and do not reflect the exact values obtained. See Fig. 15.

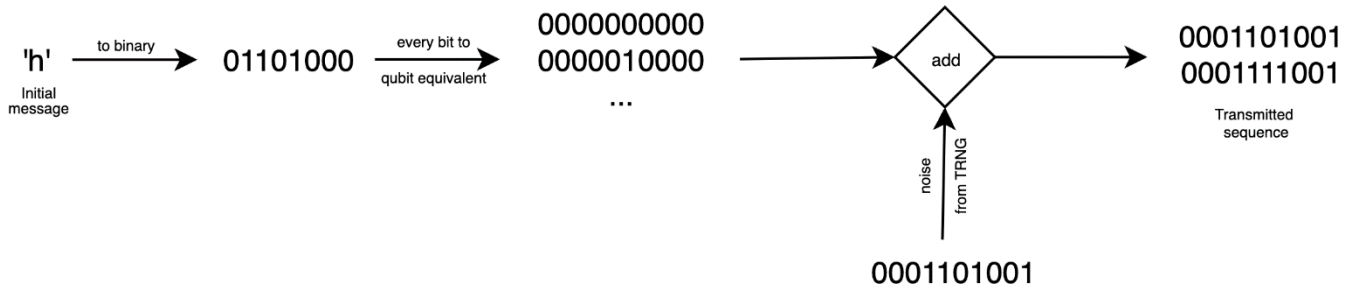


Fig. 13. Message to qubit conversion process

The binary qubit values are then added with an input from the TRNG, which introduces the noise element. The overall resulting binary string from Fig. 13 is then transmitted as a baseband signal without modulation.

5. Results and discussion

The TRNG designed and implemented in the scope of this paper was subjected to 16 tests based on NIST guidelines. Some of these tests have been enumerated in Tab. 1. It must be noted the ’approximate entropy test’ focuses on the frequency of overlapping m-bit patterns across the entire sequence. Overlap may be generated by small sub-sequences with length 2/3 bits, causing the failure of this test. In comparison to the overall sampled sequence, this is insignificant and may be ignored.

Table 1 Randomness test results

Test Type	p-value	Result
Frequency test (monobit)	0.5298	Random
Run test	0.0665	Random
Linear Complexity	0.9197	Random
Approximate Entropy	0.0012	Non-random
Binary Matrix Rank	0.4812	Random

In the case of entanglement as shown in Fig. 11, all the four entangled states |00⟩, |01⟩, |10⟩ and |11⟩ were created with equal probabilities of 25%. However, on the hardware side, the entangled states cannot be shown simultaneously. Hence, they were indicated sequentially. These states can be seen in the waveforms in Fig. 14.

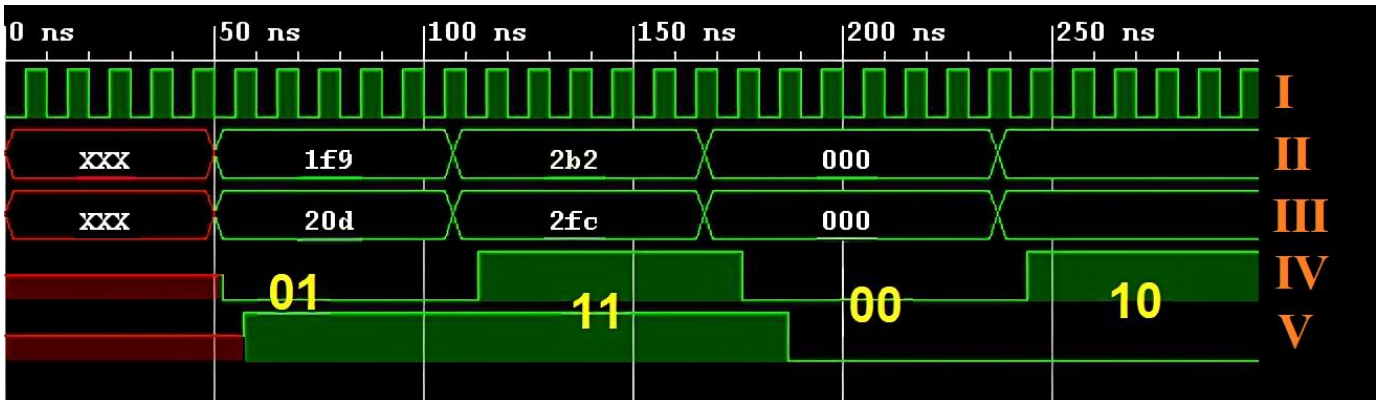


Fig. 14. Waveform obtained by simulating entanglement on hardware. (States marked in yellow). The line marked in Roman Numerals are as follows – I) clock II) statein1 III) statein2 IV) qubit1 V) qubit2.

It must be noted that in Fig. 13, statein1 and statein2 hold the 10-bit qubit representations of both the qubits. q1 and q2 hold the data of their collapsed pure states. The states are indicated in yellow however their order is not necessarily fixed.

Regarding the transmission of information between the FPGA nodes, a five-character word was sent between them over both wired and wireless media. All transmitted signals are the unaltered, base-band signals and not encoded using any carrier. The experimentation procedure was repeated twice and tested the time taken by the full experiment cycle.

This cycle is defined as a) converting the alphabetical word as either bit or qubit, b) transmitting the generated binary sequences, c) receiving on a different node, d) converting and reconstructing the initial message. The average transmission time per alphabetical character is enumerated in Table 2. In the able, the values observed were done so by repeating the same transmission for three hundred times. The total time was found to be 404922 ms in the case of bits and 496388 ms in the case of qubits. As the word itself consisted of 5 characters, we arrive at 269.948 ms per character in the first case and 330.925 ms for the second both. It must be noted that both of these calculations apply to the wired mode of transmission. In the case of wireless transmission, the same cycle took averaged at 299.596 ms for bit mode and 781.95 ms for qubit mode. An example of such a transmission can be observed in Fig. 15.

In Fig. 15a, the message “hello” is first converted into qubits. This sequence is then transmitted. Fig. 15b displays the received sequence and the reconstructed message. In both Fig.s, the original and reconstructed messages are highlighted in yellow for the reader’s convenience. In Fig. 15c, the waveform shows the word “hello” when converted to its direct binary equivalent (40-bits). This waveform is then converted into qubits, shown in Fig. 15d, and constitutes the transmitted signal. Figure 15e is the signal as received on the other FPGA. It must be noted that a timescale of 5ns has been used. The overall process of message conversion and addition of noise is illustrated through Fig. 13.

The error % has been calculated as the number of wrong transmissions out of the total number of transmissions. The transmitted messages in bit form were found to have lower error rates in both media - wireless media resulted in about 1.66% while wired resulted in approximately 0% errors. For the qubit mode cases, the average number of wrong transmission per cycle was observed to be 7 with the wired medium, giving 2.33%. In the wireless medium, it was significantly higher, with 25.6 wrong transmissions per cycle amounting to 8.53%. As the signals transmitted in both cases were the original base-band signals, the qubit transmission became especially susceptible to higher error rates. Existing modulation methods maybe be considered to reduce the error rate, however, that remains outside the scope of this paper.

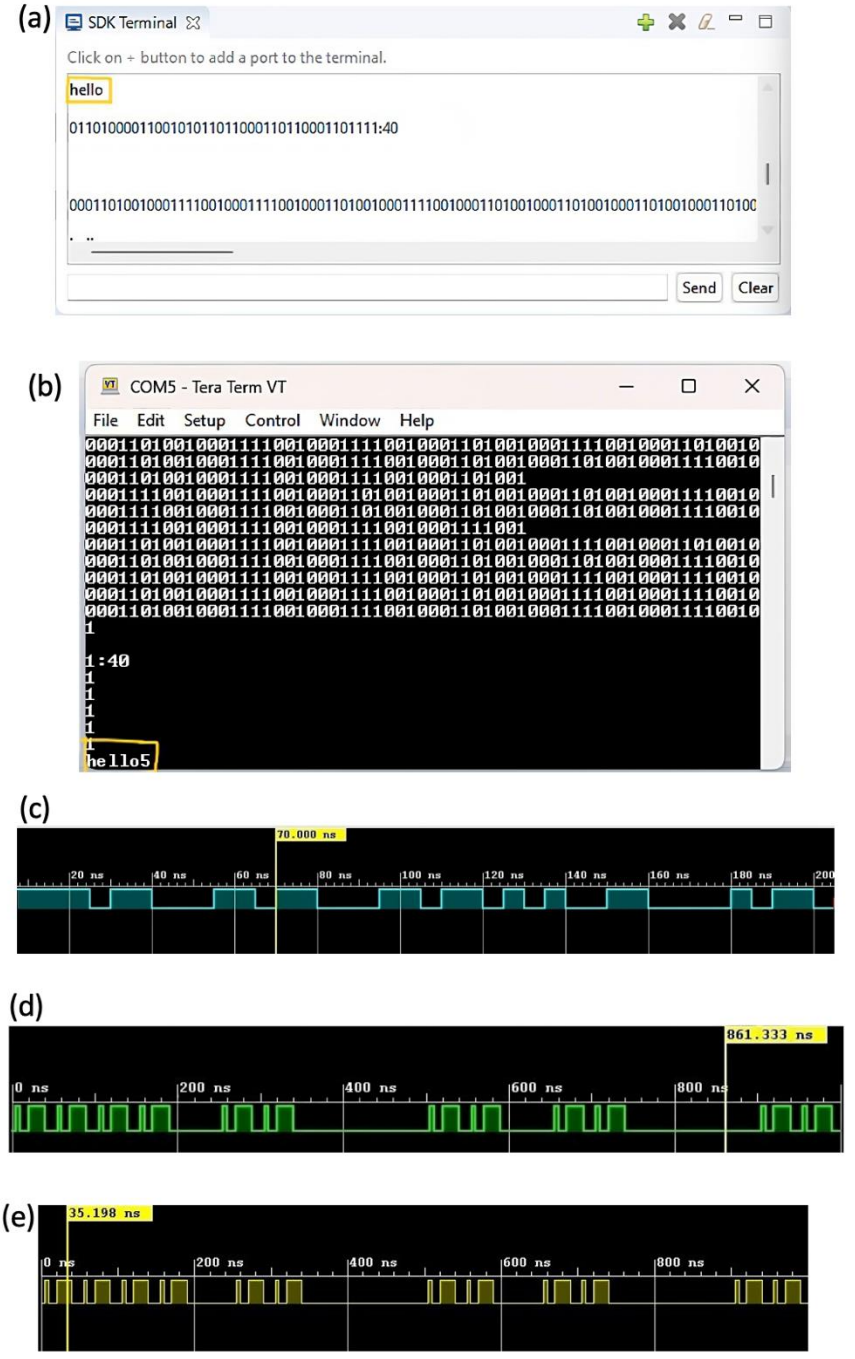


Fig. 15. (a) Sender side message in binary. (b) Receiver side message (Message marked in yellow. 5 indicates the length of reconstructed message). (c) Direct conversion to binary. (d) Message converted to qubit [timescale 5 ns]. (e) Received message in qubit form.

Table 2 Observed transmission characteristics (for 300 experiment cycles)

Transmission Type		Transmission Time (ms)	Error %
Wired	Bit	269.95	~0
	Qubit	330.93	2.33
Wireless	Bit	299.59	1.66
	Qubit	781.95	8.53

6. Conclusions

In this paper, modifications were made to existing digital qubit standards to account for complex probability amplitudes. By extending the same concept, entanglement was achieved on a digital hardware platform thus creating all the maximally entangled states possible for 2 qubits. However, the created entangled pair does not leave the digital quantum computer on which it was created thus displaying the limitation with this aspect of the paper. Nevertheless, entanglement finds its application in quantum teleportation, which may be extended to work with digital qubits in the future.

This paper also showcases how qubit information was successfully transmitted between two digital quantum computers over both wired and wireless media. Alphabetical messages of around 10 words (English alphabet) were transmitted in the form of qubits on the one boards and retrieved and converted on the receiving board.

References

[1] V. Hlukhov, "FPGA based digital quantum computer verification," *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 05 2020.

[2] A. Harrow, P. Hayden, and D. Leung, "Superdense coding of quantum states," *Physical review letters*, vol. 92, no. 18, p. 187901, 2004.

[3] J. C. Bardin, D. H. Slichter, and D. J. Reilly, "Microwaves in quantum computing," *IEEE Journal of Microwaves*, vol. 1, pp. 403–427, 2021.

[4] V. Hlukhov, "Digital qubits for fpga-based homogenous quantum coprocessor," *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 09 2021.

[5] V. Hlukhov and B. Havano, "Fpga-based digital quantum coprocessor," *Advances in Cyber-Physical Systems*, vol. 3, pp. 67–83, 11 2018.

[6] G. Marsaglia, "Xorshift rngs," *Journal of Statistical Software*, vol. 8, 2003.

[7] S. Vigna, "Further scramblings of marsaglia's xorshift generators," *Journal of Computational and Applied Mathematics*, vol. 315, pp. 175– 181, 05 2017.

[8] P. Haddad, V. Fischer, F. Bernard, and J. Nicolai, "A physical approach for stochastic modeling of tero-based trng," *Lecture Notes in Computer Science*, pp. 357–372, 2015.

[9] M. K. Mandal and B. C. Sarkar, "Ring oscillators: Characteristics and applications," *Indian Journal of Pure Applied Physics*, vol. 48, pp. 136– 145, 2010.

[10] D. V. Gadre, S. Gupta, D. V. Gadre, and S. Gupta, "Universal asynchronous receiver and transmitter (uart)," *Getting Started with Tiva ARM Cortex M4 Microcontrollers: A Lab Manual for Tiva LaunchPad Evaluation Kit*, pp. 151–167, 2018.

[11] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge Cambridge University Press, 2019.

[12] I. Glendinning, "The Bloch sphere," *QIA Meeting*, Research Gate, 2005

[13] C. P. Williams and C. P. Williams, "Quantum gates," *Explorations in Quantum Computing*, pp. 51–122, 2011.

[14] V. Hlukhov and B. Havano, "Principles of digital quantum coprocessor based on a fpga, which operates under the control of a classical computer," *2019 9th International Conference on Advanced Computer Information Technologies (ACIT)*, 06 2019.

[15] H. M. Waidyasooriya, H. Oshiyama, Y. Kurebayashi, M. Hariyama, and M. Ohzeki, "A scalable emulator for quantum fourier transform using multiple-fpgas with high-bandwidth-memory," *IEEE Access*, vol. 10, pp. 65103–65117, 2022.

[16] W. Li and X. Yang, "A parallel and reconfigurable united architecture for fibonacci and galois lfsr," *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 08 2015.

[17] L. Hales and S. Hallgren, "An improved quantum fourier transform algorithm and applications," in *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pp. 515–525, IEEE, 2000.

[18] I. Bautista, V. Kreinovich, O. Kosheleva, and H. P. Nguyen, "Why it is sufficient to have real-valued amplitudes in quantum computing," in *Soft Computing: Biomedical and Related Applications*, pp. 131–136, Springer, 2021.

[19] I. Vasyltsov, E. Hambardzumyan, Y.-S. Kim, and B. Karpinskyy, "Fast digital trng based on metastable ring oscillator," *Cryptographic Hardware and Embedded Systems – CHES 2008*, pp. 164–180.

[20] "7 series fpgas and zynq-7000 soc xadc dual 12-bit 1 msps analog-todigital converter user guide", UG480, version 1.11, 2022

[21] "ESP8266 Technical Reference", version 1.7, Expressif, 2020.

[22] J. Price, T. Goble, "10 – signals and noise," *Telecommunications Enigineer's Referencebook (F. Mazda, ed.)*, pp. 10-1-10-15, Butterworth-Heinemann, 1993.

[23] "Pcb design and circuit simulator software - proteus."

[24] A. L. Rukhin, N. I. O. Standards, and T. (U.S, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology ; Washington, D.C, 2010.
[25] S. K. Ang, "Nist randomness testsuit," 08 2022.
[26] C. Leitchi, "pySerial Documentation", Release 3.4, Pyserial, 2021.
[27] N. Zou, "Quantum entanglement and its application in quantum communication," *Journal of Physics: Conference Series*, vol. 1827, p. 012120, 03 2021.
[28] S. Gorn, R. W. Bemer, and J. Green, "American standard code for information interchange," *Communications of the ACM*, vol. 6, no. 8, pp. 422–426, 1963.

Bhuvan Hawargi is currently pursuing a B.Tech degree in Electronics and Communication Engineering from PES University, graduating in 2023. His research interest are quantum technologies, micro-controllers and single-board-computers.

Akshay K B received his Diploma in Electronics Communication from KS Polytechnic in 2018, and is currently pursuing his B.Tech degree in Electronics and Communication Engineering from PES University and is expected to graduate in 2023. His research interests are embedded systems, hardware design and product development.

Kaustav Bhowmick received his PhD from the University of Nottingham, Nottingham, United Kingdom in 2013. He heads the Photonics and Quantum Technologies Laboratory at PES University. He is currently an Associate Professor in the Department of Electronics and Communication Engineering at PES University, Bengaluru, India. His research focuses on photonics and related quantum technologies.

Received 26 February 2023