Design of High Speed 32-bit Microarchitecture for Emulation of Quantum Computing Algorithms

Mayuresh Deshpande, Hardik Shah, Vinayak Kini, Chirag Bafna

Abstract— Quantum computing has emerged as a revolutionary concept in the field of cryptography and data operations. Quantum algorithms have proved themselves to be much faster and efficient than their classical counterparts. Though we are still years away from solid state realization of quantum computers [1], currently software emulations [5] are used for implementing quantum algorithms. However, present day processors are incapable of dealing with the immense parallel architecture required by the quantum computation. This research paper focuses on the design of a 32 bit floating point processor incorporating the superscalar architecture which is capable of running quantum operations. The unique feature of the processor is that it supports an instruction set dedicated to basic quantum operations required for algorithms. The quantum states are stored in the register memory in 32 bit floating point format. The execution unit consists of multiple FPUs in order to support the parallel quantum operations. The superscalar design of Quantum Computing Unit (QCU) is the key component of the processor which makes it superior to the existing architectures.

Index Terms— Quantum Computing; Superscalar Architecture; Floating Point Operations

1 INTRODUCTION

Over the past few years, Quantum computing has proved to offer a paradigm shift in the field of computing owing to its ability to tap in the fundamental fabric of reality. Quantum computer is a computation system which makes use of the properties of quantum mechanics for performing various operations on data. Some of the fundamental properties used in quantum computing are superposition, entanglement and quantum tunneling. It leads to the non-deterministic or probabilistic computation of data. Therefore quantum computers outperform classical computers in the fields of data search and encryption.

However, practical implementation of quantum computer has always been a challenging task for the electronics community. The daunting task in the quantum computation is to deal with the quantum states. Quantum states unlike classical states, exhibit superposition of states and are expressed in probabilistic terms using quantum bits widely known as qubits. Quantum processor based on few qubits have been demonstrated using nuclear magnetic resonance[4], cold ion trap[4], optical systems[4] and superconducting circuits[3]. We are still years away from developing a solid state quantum computer[1]. We know that we cannot increase computing power indefinitely because as conventional semiconductors get smaller and smaller they are going to hit a barrier within next ten years or so where they will no longer work as they do today. It will be no longer possible to scale semiconductor devices to smaller scales than now and that is where quantum computers are one possible solution to that problem.

The another way of harnessing benefits of quantum computing is emulating quantum algorithms on classical systems to improve their performance. In 1982 Feynman's pioneering paper on quantum computing was the first to indicate such possibility. Many attempts have been made in the last two decades to emulate quantum algorithms on software and hardware. Software emulators[5] employ explicitly defined quantum libraries to facilitate implementation of quantum algorithms. Emulation of few search and factoring algorithms have been demonstrated using software emulation[5]. However present day processors[5] are unable to cope up with immense parallelism required by quantum computing.

This project is centred around the development of high speed 32 bit special purpose processor capable of handling quantum computing more efficiently than present day systems. The processor is supposed to provide specific instruction set in order to support basic quantum operations with ease. The qubits are represented in 32 bit IEEE floating point format for high precision computation. The present state of qubits is stored in array of 32 bit registers. The memory organization of the current design supports N-qubit operations with higher efficiency. All the operations on the data are carried out in the execution unit which is called as Quantum Computing Unit.

The Quantum Computing Unit (QCU) is at the heart of the design and it incorporates superscalar architecture hence enabling parallel computation. The architecture employs 32 bit floating point unit to enable high speed multiplication and addition. The striking feature of the processor is its ability to eliminate redundant multiplications thereby demonstrating higher performance then other emulating models[9] of quantum algorithms. This is achieved by replacing complex operation of matrix multiplication by selective multiplication of quantum states depending on the quantum transform[7]. The QCU also enables entanglement of 2 distinct qubits at a time. However it can operate on previously entangled quantum data with higher efficiency. The proposed microarchitecture provides ideal emulation platform for quantum algorithms and presents a way to tap in the infinite computing power of quantum mechanics.

- *Authors are affiliated with Vivekanand Education Society's Institute of Technology, Chembur, Mumbai-74.*
- *E-mail: mayuresh2010@gmail.com, hardik.k.s@gmail.com, kini.vinayak94@gmail.com, bafnachirag94@gmail.com*

2 Quantum Computing Fundamentals

2.1 Representation of Quantum information

The major difference between quantum computing and classical computing arises from the types of values required to perform computing. While the basic information units for classical circuits are 0 and 1, quantum computing uses probabilistic numbers as bearers of information[8]. Hence, representing a single quantum information unit might require a large number of classical bits, depending on the precision required. In this case 32 bit floating point numbers[9] are used for representing basic quantum information.

2.2 Final Stage Quantum Bits

Quantum bits also known as qubits are the fundamental units of quantum computing. A single qubit is represented as superposition of two states '0' and '1' with corresponding amplitudes denoted by 'α' and 'β'.

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Here α and β are two complex numbers representing amplitude of states 0 and 1. The continuous nature of qubit coefficients enable infinite storage capacity of quantum elements. α and β are related to each other by following equation.

$$|\alpha|^2 + |\beta|^2 = 1$$

The superposition phenomena, by which the qubits simultaneously exist in states $|0\rangle$ and $|1\rangle$ is explained by considering $|\alpha|^2$ and $|\beta|^2$ as probabilities of being in states $|0\rangle$ and $|1\rangle$ respectively.

2.3 Quantum Entanglement

Quantum entanglement is a phenomenon in which two qubits interact in such a way that the state of each qubit cannot be defined independently. Hence these two qubits are required to be represented in the entangled state. There are certain quantum circuits which produce entangled output of the qubits at input. Quantum entanglement is very important operation required for quantum cryptography. The entanglement of two qubits is shown below.

$$|\Psi 1\rangle = \alpha 1|0\rangle + \beta 1|1\rangle$$

$$|\Psi 2\rangle = \alpha 2|0\rangle + \beta 2|1\rangle$$

$$|\Psi out\rangle = \alpha 1\alpha 2|00\rangle + \alpha 1\beta 2|01\rangle + \beta 1\alpha 2|10\rangle + \beta 1\beta 2|11\rangle$$

3 Basic Quantum Operations

Basic quantum operations performed on qubits to generate new quantum state. In quantum circuits, quantum operations are performed by quantum gates. Fundamental quantum gates which operate on a single qubit are as follow.

3.1 Bit-flip Gate

Bit flip gate also known as X gate functions as not gate in quantum computing. In this operation, the coefficients of states $|0\rangle$ and $|1\rangle$ are interchanged. The representation of bit-flip gate and matrix equation is shown in figure 1.

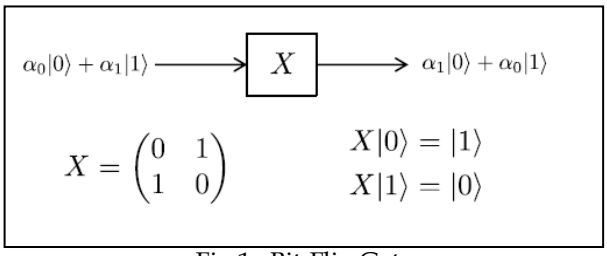
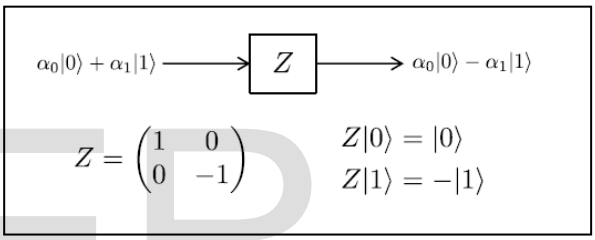


Fig.1 : Bit-Flip Gate

3.2 Phase Shift Gate

Phase Shift gate also known as Z gate operates on a single qubit and alters the sign of the second coefficient. The operation of the same is shown in figure 2.

Fig.2: Phase Shift Gate



3.3 Hadamard Gate

Hadamard gate is one of the most commonly used quantum gate and it corresponds to one qubit rotation. The hadamard gate can be implemented for multiple qubits. It involves 2 x 2 matrix multiplication with current quantum state. The matrix of the hadamard transform can be represented as below.

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Figure 3 shows representation and equation of hadamard gate for single qubit.

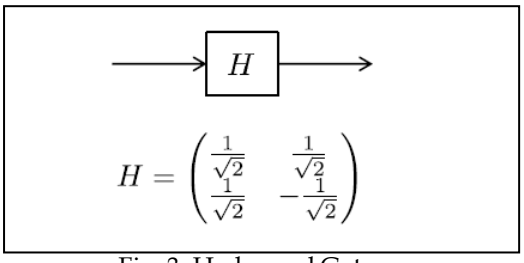


Fig. 3: Hadamard Gate

3.4 C-NOT Gate

C-NOT gate acts on two or more qubits and performs controlled NOT operation of second qubit depending on 1st qubit.

The output of the C-NOT gate is an entangled quantum state. It requires four complex multiplications for two qubits. The representation and matrix equation for two qubit C-NOT gate is shown in figure 4.

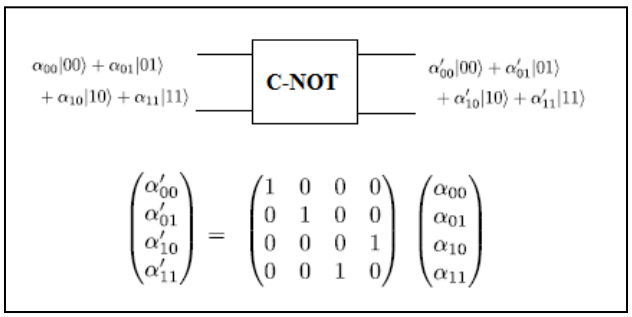


Fig. 4: C-NOT Gate

3.5 Swap Gate

As its name suggests, Swap Gate is used for swapping two qubits. In Swap gate the entanglement of the qubits is preserved and the coefficients for states are changed The equation for the same is shown in the figure 5.

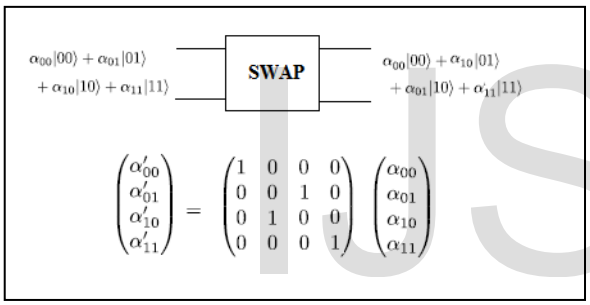


Fig. 5: SWAP Gate

4 Emulation of Quantum Information

4.1 Emulation of Quantum Bits

Quantum bits are represented as superposition of two states '0' and '1' with corresponding amplitudes denoted by complex numbers 'α' and 'β'.

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Hence each qubit is represented by four 32 bit floating point numbers as follow. Use of modified IEEE 754 floating point format in this case increases precision of the system and also accelerates multiplication of two numbers. In the modified format signed exponent field is used to enable representation of small numbers.

α_{real}

S	S_e Exp. (7 bit)	Mantissa (24 bit)

α_{imag}

S	S_e Exp. (7 bit)	Mantissa (24 bit)

β_{real}

S	S_e Exp. (7 bit)	Mantissa (24 bit)

β_{imag}

S	S_e Exp. (7 bit)	Mantissa (24 bit)

4.2 Emulation of Quantum Entanglement

Entanglement of two qubits is shown below. It is represented using 4 complex coefficients i.e. eight 32 bit floating point numbers.

$$|\Psi out\rangle = \alpha1\alpha2|00\rangle + \alpha1\beta2|01\rangle + \beta1\alpha2|10\rangle + \beta1\beta2|11\rangle$$

Representation of Entanglement of two qubits is as follow.

$\alpha1\alpha2$

S	S_e Exp.(7 bit)	Mantissa (24 bit)	
			Imag

$\alpha1\beta2$

S	S_e Exp. (7 bit)	Mantissa (24 bit)	
			Imag

$\beta1\alpha2$

S	S_e Exp. (7 bit)	Mantissa (24 bit)	
			Imag

$\beta1\beta2$

S	S_e Exp. (7 bit)	Mantissa (24 bit)	
			Imag

5 Architecture of Quantum Computing Unit

The Quantum Computing Unit (QCU) is a High performance Parallel Computing Device incorporating superscalar architec-

ture. The QCU is the heart of the entire design and performs functions similar to that of ALU in classical processors.

The said QCU is capable of carrying out basic quantum operations on input qubits. The operations performed by the QCU include Single qubit Bit-flip, Phase shift and Hadamard transform. It can also operate on entangled qubits and perform operations corresponding to Swap and C-NOT quantum gates.

The QCU incorporates superscalar architecture in order to enable parallel computation of quantum transforms. QCU can operate on two single qubits simultaneously depending on the control signal provided to it. Apart from basic quantum transforms, QCU can perform entanglement of two qubits or operate on two entangled qubits simultaneously.

32 bit complex floating point unit is the most fundamental block of the QCU architecture. It is required to carry out the multiplication and addition of coefficients of quantum states in the qubit [10]. Complex FPU employs 32 bit floating point addition and multiplication hardware and provides result in the complex number format.

The hardware required to carry out different operations involving quantum operands is shown in the table 1.

TABLE 1
HARDWARE REQUIREMENT OF QUANTUM COMPUTING UNIT

Operation	Computational Requirement	FPU Hardware Requirement
Complex Number Addition	Complex Number Adder	2 Floating Point Adders (32 bit)
Complex Number Multiplication	Complex Number Multiplier	2 Floating Point Adders 4 Floating Point Multipliers (32 bit)
Single Qubit Quantum Gate	4 Complex Number Multipliers 2 Complex Number Adders	16 Floating Point Multipliers 12 Floating Point Adders (32 bit)
Two Qubit Quantum Gate	8 Complex Number Multipliers 4 Complex Number Adders	32 Floating Point Multipliers 24 Floating Point Adders (32 bit)
Quantum Entanglement Unit	4 Complex Number Multipliers	16 Floating Point Multipliers (32 bit)

5.1 Architecture Schematic of QCU

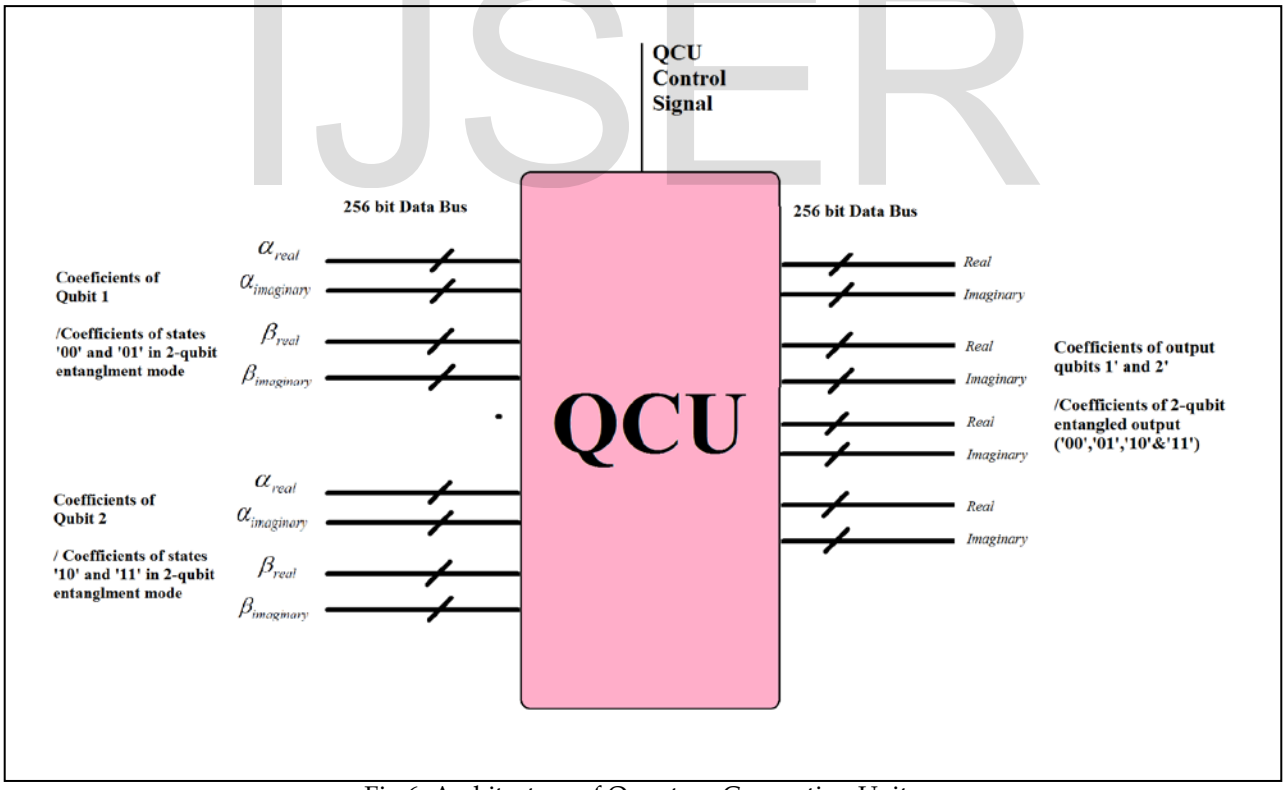


Fig 6: Architecture of Quantum Computing Unit

5.2 Architecture of Basic Quantum Gate

The operation of single qubit quantum gate can be represented by following matrix equation. Let inital qubit be $|\Psi\rangle$ and final output be $|\Psi'\rangle$.

$$|\Psi'\rangle = |\Psi\rangle \begin{bmatrix} M11 & M12 \\ M21 & M22 \end{bmatrix}$$

If $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$$\text{and } |\Psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$$

then

$$\alpha'|0\rangle + \beta'|1\rangle = (\alpha|0\rangle + \beta|1\rangle)\begin{bmatrix} M11 & M12 \\ M21 & M22 \end{bmatrix}$$

Therefore one can conclude that

$$\alpha' = \alpha * M11 + \beta * M21$$

and

$$\beta' = \alpha * M12 + \beta * M22$$

Hence computation of α' and β' comprises of 4 complex multiplications and two complex additions.

The architecture for the general quantum gate consists of 4 complex number floating point multipliers operating parallel and two complex number adders. The diagram of architecture is shown in the figure 8.

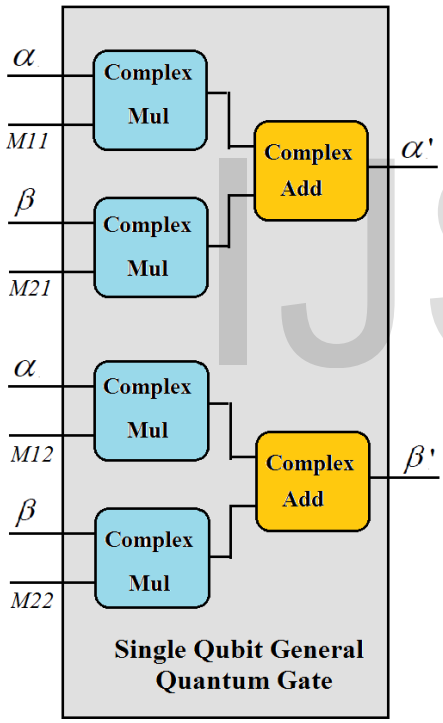


Fig. 8: Architecture of general Quantum Gate

5.3 Architecture of Quantum Entanglement Unit

As mentioned previously, the QCU is also capable of performing quantum entanglement of two qubit system at a time. Therefore it consists of an independent quantum entanglement unit performing at high speed.

The equation for the quantum entanglement of two qubits is shown below.

$$|\Psi 1\rangle = \alpha 1|0\rangle + \beta 1|1\rangle$$
$$|\Psi 2\rangle = \alpha 2|0\rangle + \beta 2|1\rangle$$

$$|\Psi out\rangle = \alpha 1\alpha 2|00\rangle + \alpha 1\beta 2|01\rangle + \beta 1\alpha 2|10\rangle + \beta 1\beta 2|11\rangle$$

Let

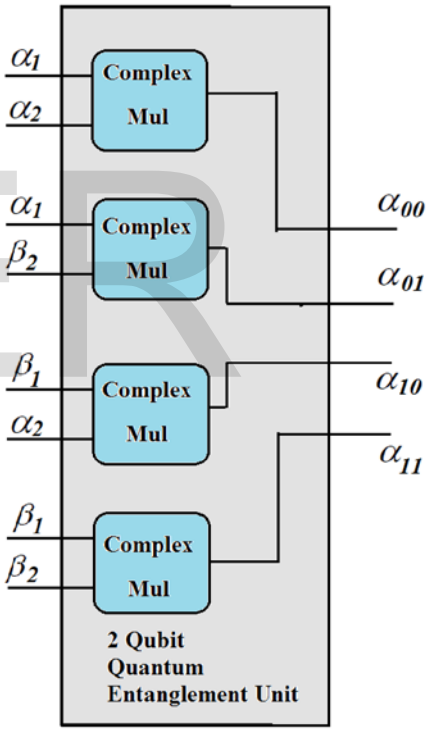
$$|\Psi out\rangle = \alpha 00|00\rangle + \alpha 01|01\rangle + \alpha 10|10\rangle + \alpha 11|11\rangle$$

Hence, we can conclude,

$$\alpha 00 = \alpha 1 * \alpha 2$$
$$\alpha 01 = \alpha 1 * \beta 2$$
$$\alpha 10 = \beta 1 * \alpha 2$$
$$\alpha 11 = \beta 1 * \beta 2$$

The architecture of Entanglement unit performing above operation is shown in the figure 9.

Fig. 9: Architecture of 2-Qubit Entanglement Unit



5.4 Architecture of QCU

The Quantum computing unit consists of two general quantum gates working simultaneously, auxiliary Entanglement unit and operation selection logic.

The operation carried out by QCU is determined by control signal generated by processor during decoding of instruction. The data of quantum state fetched from the register memory is applied to either general quantum gate or entanglement unit depending on the operation. The result of the QCU is 4 complex quantum coefficients which are forwarded to the memory for write-back operation.

In the schematic of QCU, it can be seen that it operates on 256 bit data bus. Input to the QCU are coefficients of two different qubits. These coefficients are then applied to the parallel quantum gates for quantum operations.

The transform operation performed on the qubit is determined by the matrix coefficients which are applied to the q-gate via select logic depending on the control signal. QCU can perform two quantum operations simultaneously because of two independently functioning quantum gates.

Quantum entanglement in the QCU has independent block functioning on the same inputs as that of q-gates. The result of both q-gates and entanglement unit are applied to the multiplexer which selects one output according to the QCU control signal.

Thus the proposed QCU operates on 256 bit data simultaneously and it is powered by parallel functioning of multiple 32-bit FPUs. Hence it plays key role in the superior performance of the designed processor. The schematic of quantum computing unit is shown in the figure 10.

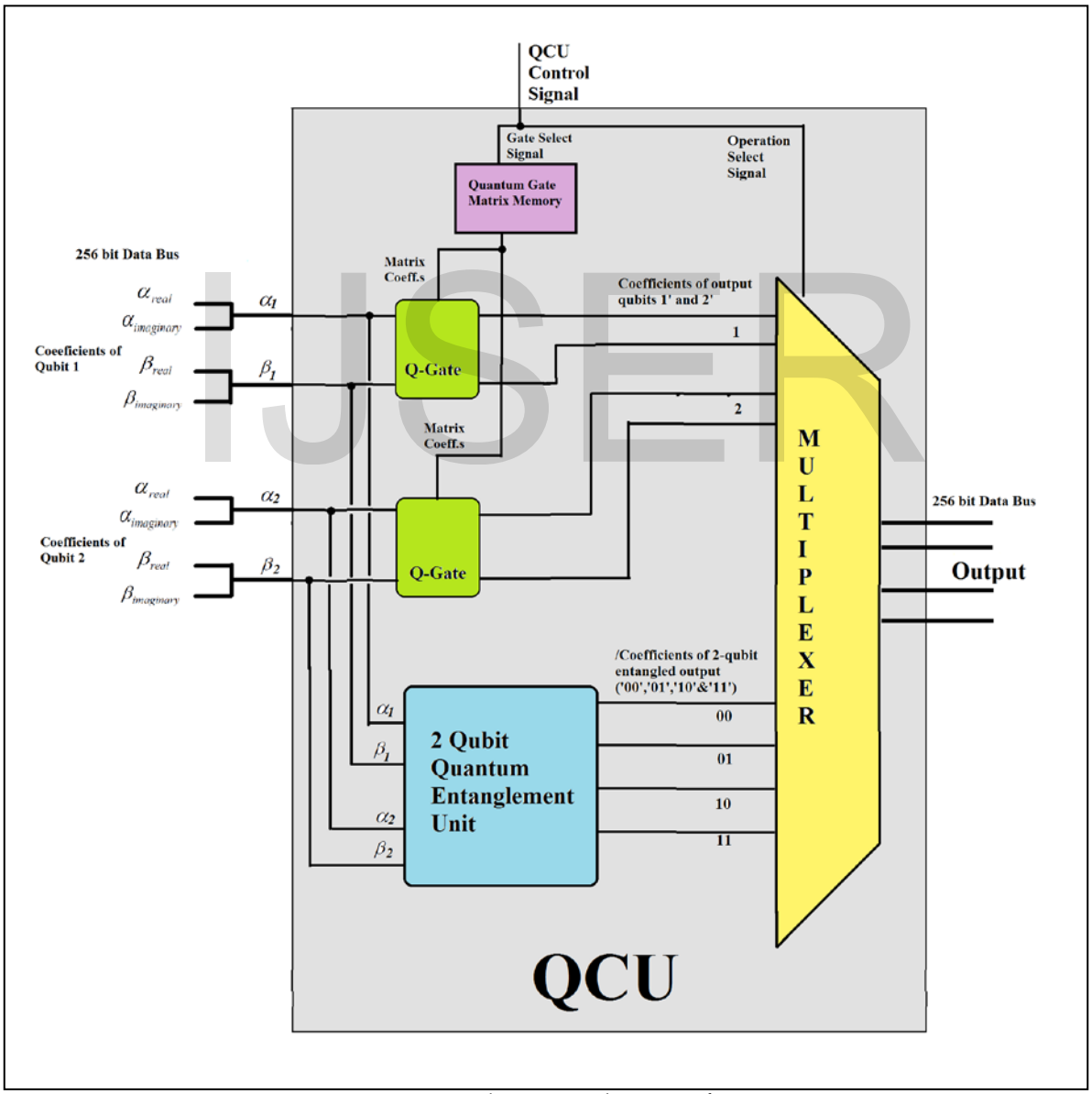


Fig. 10: Architecture schematic of QCU

6 Processor Architecture

The architecture of the processor follows the convention of classical processors. The processor architecture developed for quantum operations consists of 4 stages in its instruction cycle. The operation of the processor along with the architecture schematic is given below.

6.1 Instruction Fetch

This stage consists of program counter and the instruction memory. The program counter keeps track of the current instruction address and computes the next address on each clock cycle. During computation, the processor fetches two single qubit instructions in each cycle.

The instructions of the processor will be distinguished in two types. First type is configuration instructions. These instructions will be required for initializing the data in the quantum state registers for required operation. Second type of instructions is computing instructions. These instructions will specify operations to be performed and the operands required.

6.2 Instruction Decode and Control

In this stage, Instruction is decoded and the control signals for the execution stage are generated. The control signals determine the operation performed by the Quantum Computing unit (QCU) in the execution stage.

This stage also consists of Quantum state memory which holds the values of coefficients of different qubits on which the operation is to be performed. Depending on the address specified by the address field in the instruction, qubits are fetched and forwarded to the QCU.

6.3 Execution unit

Execution unit is the heart of the processor architecture and it consists of superscalar Quantum Computing Unit consisting of multiple FPU operating in parallel mode. The detailed description of the QCU is given in the section 5.

6.4 Register Write-back

The result obtained by the QCU is written back in the quantum state registers in order to update the qubits. The write operation of registers is carried out on different clock edge than that of read operation to avoid data hazards.

The State transitions of the proposed processor while executing quantum instruction are similar to that of single cycle classical processor. The state diagram is shown in figure 11.

Figure 12 shows the combined data-path of the processor.

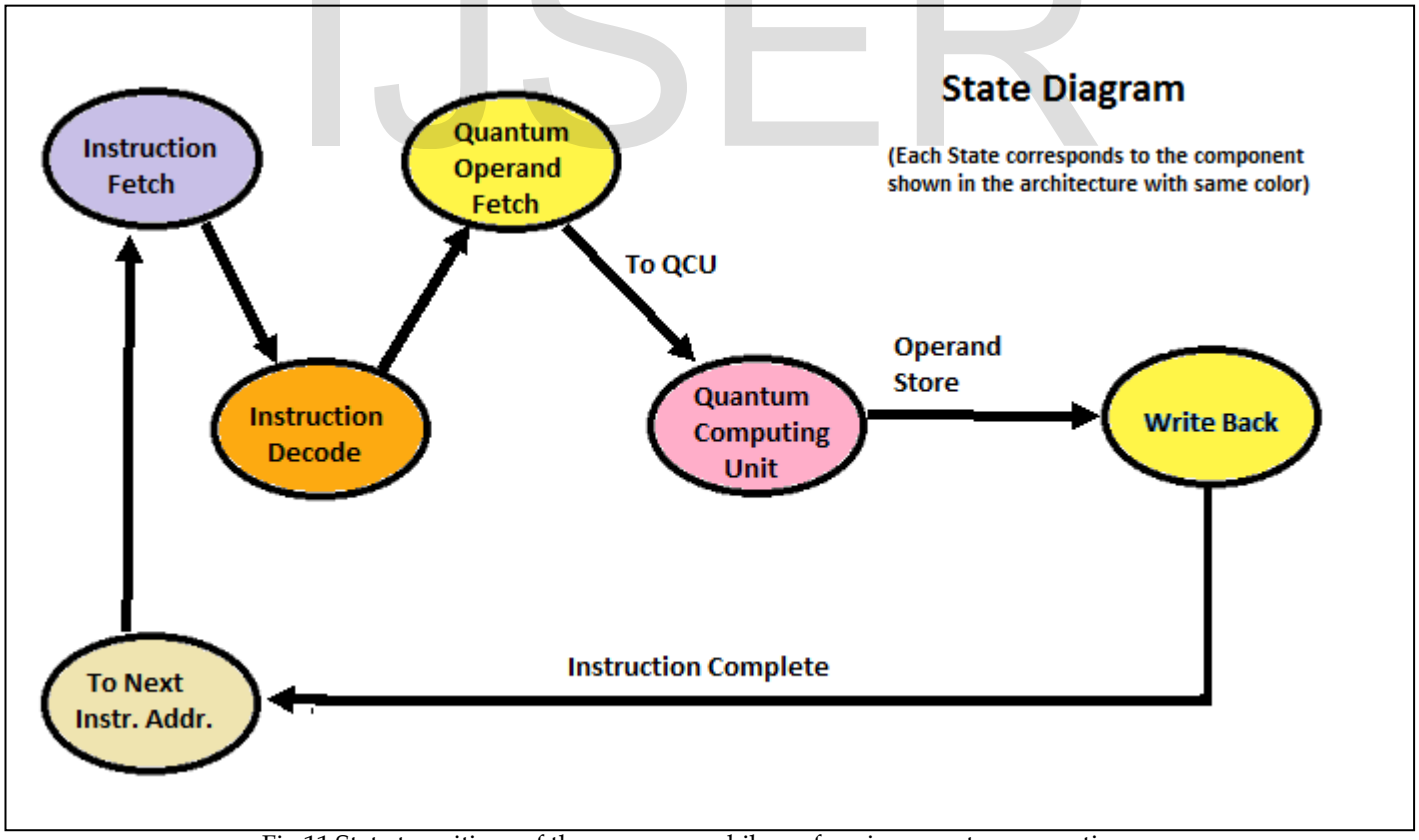


Fig.11:State transitions of the processor while performing quantum operation

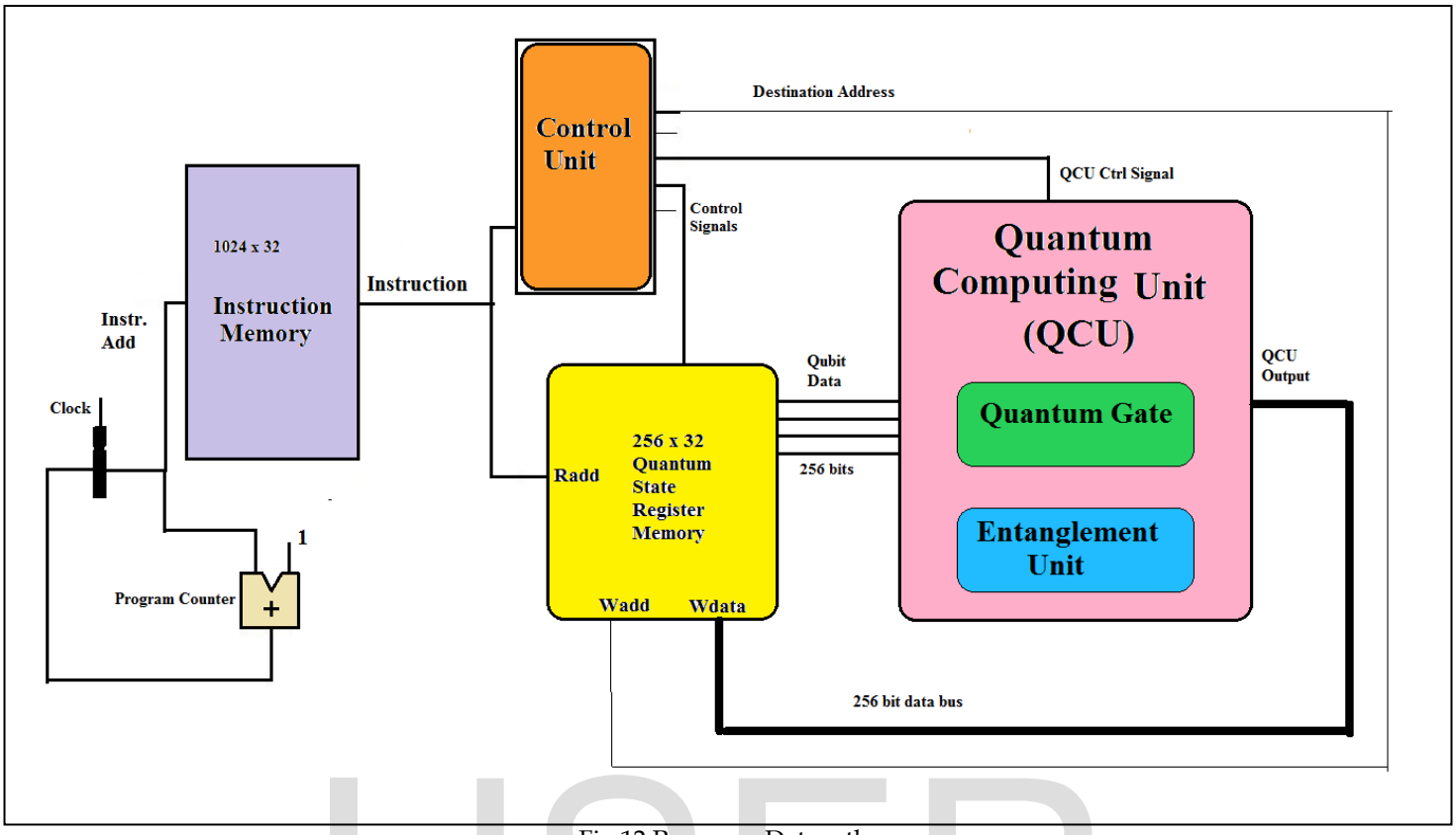


Fig.12 Processor Datapath

7. Future Scope

The quantum computing unit incorporated in processor can be used to perform Quantum Fourier transform and also in Grover's algorithm which is a quantum search algorithm that runs quadratically faster than any equivalent classical algorithm. The Instruction set architecture for the proposed microarchitecture will be developed to support complex instructions executing various quantum operations in different algorithms.

Various real-world applications of the quantum computing systems are as follow:

1. Quantum computing can play a significant role as an HPC (high performance computing) system, wherein it can be used as to optimize system goals within very less time.
2. Quantum computers can be used to simulate and predict folding patterns of proteins which can transform our understanding of complex biological systems.
3. Quantum computing algorithms can be incorporated in efficient transportation system so that sophisticat ed analysis of traffic patterns in air and ground will forestall bottlenecks and snarls.
4. Quantum computers will be able to analyze the vast amount of data collected by telescopes and con tribute to the space exploration.

8. Conclusion

The goal of this project was to develop an architecture capable of emulating Quantum Computing operations at high speed. The proposed design of Quantum Computing Unit increases the computation speed manifold owing to its parallel architecture. The representation of quantum data in 32 bit floating point format improves the precision and accuracy of the computation. Hardware requirement of such architecture is considerably large. However, it is far more superior than existing emulation systems of Quantum algorithms and it can be optimized during implementation phase. The architecture of the entire processor is planned to be implemented in Verilog and FPGA [10] with the help of HDL tools such as Xilinx 12.1. The prototype of Quantum emulation microarchitecture presented in this paper will certainly open further avenues in the practical implementation of quantum computing.

Acknowledgment

Authors would like to express their gratitude towards V.E.S Institute of Technology, Chembur, Mumbai; for providing necessary resources to conduct research and experimentation related to this project.

REFERENCES

[1] Han, J.; Jonker, Pieter, "On quantum computing with macroscopic Josephson qubits," *Nanotechnology, 2002. IEEE-NANO 2002. Proceedings of the 2002 2nd IEEE Conference on* , vol. 305, no. 308, pp.2002.
doi: 10.1109/NANO.2002.1032252G.

[2] Z. Zilic and K.Radecka, *The Role of Super-fast Transforms in speeding up Quantum Computations*, Proceedings of the 32nd Internatitonal Symposium on Multiple-Valued Logic (ISMVL02), 2002I. S.

[3] Jonker, Pieter; Han, J., "On quantum and classical computing with arrays of superconducting persistent current qubits," *Computer Architectures for Machine Perception, 2000. Proceedings. Fifth IEEE International Workshop on* ,vol. 69,no. 78,pp. 2000. doi: 10.1109/CAMP.2000.875960.

[4] DiCarlo, L.; Chow, J.; Gambetta, J.; Bishop, L.; Majer, J.; Blais, A.; Frunzio, L.; Girvin, S.; Schoelkopf, R., "Demonstration of two-qubit quantum algorithms with a solid-state electronic processor," *Lasers and Electro-Optics, 2009 and 2009 Conference on Quantum electronics and Laser Science Conference. CLEO/QELS 2009. Conference on* , vol. 1, no. 1, pp. 2-4. June 2009.

[5] "Libquantum", Online Quantum Library Documentation, http://www.enyo.de/libquantum/Electronic Publication: Digital Object Identifiers (DOIs):

[6] M. Steffen, L. M. K. Vandersypen and I.L. Chuang, *Toward Quantum Computation: A Five-Qubit Quantum Processor*, IEEE Micro, Vol. 27, No. 1, pp. 24-34, 2001.

[7] H. Goto, Y. Hasegawa, and M. Tanaka, "Efficient Scheduling Focusing on the Duality of MPL Representatives," Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.

[8] J.P. Hayes and I.L. Markov, Simulation, Synthesis and Testing of Quantum *Circuits*, DARPA QuIST annual research review, Beverly Hills, CA, June 2003.

[9] Montoye, R.K.; Hokenek, E.; Runyon, S.L., "Design of the IBM RISC System/6000 floating-point execution unit," *IBM Journal of Research and Development* , vol. 34, no. 1,pp.59,70, Jan. 1990 doi: 10.1147/rd.341.0059.

[10] Khalid, A.U.; Zilic, Z.; Radecka, K., "FPGA emulation of quantum circuits," *Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings. IEEE International Conference on* , vol. 310, no. 315, pp. 11-13. Oct. 2004.
doi: 10.1109/ICCD.2004.1347938