8Q - RoadMap

Why Open Source?

Quantum Computing is currently relegated to the very high-end industrial and scientific markets, there isn't really anyone working on quantum computing for mere mortals like us. As we have seen in every other tech revolution in history, industrial giants tend to hog technology, locking it down with things like DRM, non-standard hardware, and doing some less than ethical things to get their customers to upgrade, all of this hurts all of us. The entire reason the Open-Source Movement was started was because of hardware giants like IBM having incredibly restrictive licensing agreements (No offense to IBM, they've changed a lot since the 80's) which prevented users from actually using their computers and, were it not for the open and free-source movements that sprung up as a result, we would be missing out on a lot of the technologies we take for granted today.

Quantum Computing is an incredibly powerful tool that has the potential to fundamentally alter our understanding of the universe and will undoubtedly herald in a new era of technological advancements. This project will most likely not do that, but hopefully it will act as a springboard for those inventors, tinkerers, and hackers who can take quantum computing into areas we've never seen before.

Primary Goals

Our two primary goals are to:

1. Creating an all-in-one development system tailored for Quantum Computing via the KLM Protocol
2. Integrate an 8-Qubit QPU based on the KLM Protocol

The reason for the separation of the goals is that while we are optimistic the QPU will be successful, until we have it fully tested, we have no way of knowing for certain if it will work. The overall purpose of this device is to bring quantum computing into the hackersphere. We have the known-good KLM Protocol which is perfect for application-specific circuits that we can develop and test for, and we have our Quantum Processor Unit which is based on the same protocol (but not yet tested). In the event the QPU development is unsuccessful, all the development and testing work surrounding both the processor and the framework can still be put to use for its' intended purpose: experimenting with quantum computing. In that case, the user will have to implement their own circuit for each algorithm, and with the QEDA project (in development) this won't introduce a significant hurdle for beginners.

On the flipside though, if the QPU is a success but users have, for instance, budgetary or performance concerns, they can still build and test application-specific circuits from the same development board without having to switch to an entirely new development system.

Success Criteria

Documentation Success

For this project to be successful, it is imperative that the documentation be rock-solid. A total beginner, someone without any experience in electronics or quantum mechanics, should be able to pick up a build-guide for the QPU and build their own using store-bought discrete components. The beginner should also be able to follow an installation guide to install the necessary software/firmware (if not bundled with the operating system), as well as follow a programming guide to program, compile, and run a basic program consisting of not less than two qubits and a bell-state.

Development Board Success

Minimum Specs

The Development Board should be able to act as a stand-alone development environment. To maximize beginner friendliness, the board should be able to run beginner-friendly linux distributions such as Ubuntu Mate. The Minimum specs we should consider are:

1. 4GB DDR3/DDR4 SDRAM
2. 16GB Flash Storage
3. Bootable USB Flash Drive
4. 32 Bit Architecture
5. 1440 x 900 or higher display resolution

Peripheral Capabilities

To offer a streamlined experience for users, they should be able to write and compile their code directly on the development board if they choose to.

1. USB - Keyboard

2. USB - Mouse (optional)

3. HDMI - Monitor

4. 3mm Jack - Speakers & (optional) Mic

Networking Capabilities

As some will not want to work directly on the machine itself (preferring their current work stations), a means of transferring the compiled code to run from a workstation to the development board in question is required. Additionally, the user may wish to access an internet browser from the development board, or even adapt the board itself into a standalone product requiring internet access. To allow greater flexibility for users, there should be both wired and wireless communication capabilities for the device.

Minimum Networking Capabilities:

1. USB
2. WiFi

Communication Interfaces

GPIO

Communication Interfaces

No development computer/board is complete without the ability to communicate with other devices, as the core target of this device is hackers and students who are likely interested in integrating quantum capabilities in their own hardware designs, it is imperative to supply them with the industry standard communication interfaces they are most likely to require in their day to day use. For that reason the device should supply at minimum the following interfaces:

1. SPI - 1x
2. UART - 1x
3. TWI - 1x
4. PWM - 4x

Analog I/O

In additional to the above communication interfaces, hardware developers are likely to make use of a plethora of sensors, many of which rely on analog signals. For that reason our device should accept analog signals and have a built-in ADC. Occasionally, a hardware developer may need to convert a digital signal to an analog signal. This use case is not as common, but some basic support should be given if possible

1. 4 Analog Input Pins
2. 4 Analog I/O Pins

Digital I/O

The device should also have general purpose digital I/O pins, with 8 pins being the bare minimum and more pins being preferred. A generous target might be 32 digital I/O pins in addition to the aforementioned GPIO.

Non-Goals

It is worth mentioning due to the already large scope of this project that there are certain non-goals which while they would offer significant improvements to the user experience, are not going to be bundled with this project.

1. A customized linux distro

   This would add too much to the workload for this project, instead any customization focus should be on general linux compatibility, or specifically for supporting functions within Ubuntu-MATE.

2. Quantum IDE
   A quantum IDE would be a wonderful addition to this project, but is not a requirement at this time.

Participating In The Project

Quantum Computing is extremely difficult, even within the academic realm it is regarded as one of the most complex engineering disciplines, so we should not be making it harder for anyone interested in this field to pursue. It is sad that I even have to say this, but given the current social climate it is important to state this so that we are all on the same page:

To ensure this project is as inclusive as possible to all individuals regardless of socioeconomic, political, racial, religious beliefs, sexual orientation, or any other way humans have devised to delineate one group of people from another, we all, as a community, need to make the effort to be compassionate, patient, and respectful of all the members of the community. There will be zero tolerance for bullying, racism, harassment, or other maliciousness from anyone in the community towards anyone else in the community online or in person.