



Issue Date:	2021-12-16	CEN-CLC/FGQT	N154
Deadline:	n/a	Supersedes:	
Status:	FOR APPROVAL		

TITLE                    **Modularity and layering of Quantum Computing Hardware Stack**  
PROJECT                FGQT Roadmap  
REFERRING TO        "N146, Quantum Computing break-out group"  
SOURCE                Rob F.M. van den Brink (Delft Circuits, The Netherlands)  
CONTACT              [Rob.vandenBrink@Delft-Circuits.com](mailto:Rob.vandenBrink@Delft-Circuits.com)  
                              <https://delft-circuits.com/>

---

**With valuable input from and supported by:**

Niels Neumann (TNO)	<a href="mailto:niels.neumann@tno.nl">niels.neumann@tno.nl</a>
Nicolas Spethmann (PTB)	<a href="mailto:nicolas.spethmann@ptb.de">nicolas.spethmann@ptb.de</a>
Amado Bautista (PTB)	<a href="mailto:amado.bautista@ptb.de">amado.bautista@ptb.de</a>
Thomas Monz	<a href="mailto:thomas.monz@uibk.ac.at">thomas.monz@uibk.ac.at</a>
Andreas Spoerl (German Aerospace Center)	<a href="mailto:Andreas.Spoerl@dlr.de">Andreas.Spoerl@dlr.de</a>
Pim Venderbosch (QuiX)	<a href="mailto:p.venderbosch@quix.nl">p.venderbosch@quix.nl</a>
Ben Kassenberg (QuiX)	<a href="mailto:b.kassenberg@quix.nl">b.kassenberg@quix.nl</a>
Frank Wilhelm-Mauch (UdS/FZ Jülich)	<a href="mailto:f.wilhelm-mauch@fz-juelich.de">f.wilhelm-mauch@fz-juelich.de</a>

---

**ABSTRACT**

**Explanation of changes**

During FGQT18 in Berlin, a breakout group worked together to build consensus on a workable subdivision of quantum computing into several architectures and associated modularity. The outcome of that effort is presented in this contribution and aimed to replace the full section 7.1.3 of the Roadmap Document (N020f, or its successors). We have also identified that further expertise is needed to elaborate some of these architectures in more detail and we also do not exclude that more architectures may be identified in future.

**Why are the changes needed?**

We need a way forward, to get progress on several implementations of Quantum Computers. Since some QC hardware implementations are more mature than others, it makes sense to break the problem down along these lines so that we get progress on those that are more mature.

---

## 1. Situation sketch

Quantum computing is an area covering very different implementations and each implementation has its own dedicated solution and maturity. The aim is a stack of software and hardware layers, where higher layers are more agnostic for these differences than lower layers.

At the bottom (hardware side) of this stack, solutions are very dedicated to these differences. Figure 1 may be illustrative for that part of the stack. These hardware layers are controlled from software layers above them, but the description of software layers is out of scope of this contribution.

We propose to replace the present section 7.1.3 of the roadmap document (N020f, or its successors) with the text in this contribution, and to fill that section only with generic/functional descriptions on the various architecture families and their members of quantum computing. For those families where further details are available, we propose to keep that describing in *different* annexes.

The present version of the roadmap document has already dedicated annexes for (a) Cryogenic solid-state quantum architectures and (b) photonic quantum solutions, but more are not excluded.

### Proposal for inclusion in section 7.1.3

#### 7.1.3 Modularity and layering of hardware stack

Quantum computing is an area covering many different implementations. A convenient way of specifying its requirements is via a stack of layers, as shown in figure 7.1. The layers are chosen in such a manner that the functionality of each layer can be described in an independent manner. This causes that the interworking between these layers can be described through well-defined interfaces at the boundaries of these layers. Note that such an interface can be *virtual* (hidden internally within the implementation of the same origin) or *real* (between implementations of different origin).

The stack covers both hardware and software layers, while some layers are a mix of both. The software layers are drawn in figure 7.1 above the hardware layers with another color. These software layers may include (from bottom to top) software drivers, hardware abstraction layers, register level programming layers, assembly, etc. The aim is that higher software layers are more agnostic to differences in hardware architectures.

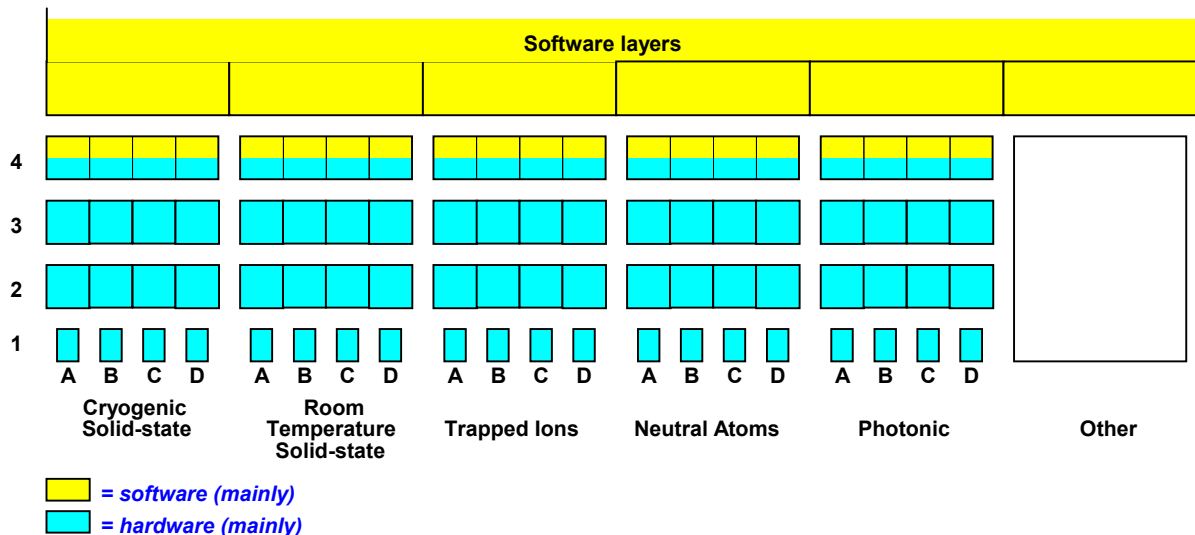


Figure 7.1. A possible break-down of quantum computing into layered stacks, accounting for different architectures

The layered approach allows for using different hardware stacks for specifying the requirements of different architecture families. Each architecture family can have multiple members (A, B, C, ...) and the description of its hardware layers (1,2,3,4) may account for differences between these members. The diagram in figure 7.1 has illustrated this symbolically by drawing different “boxes” in the layers for different members of the same family.

All these hardware stacks can make use of the same layering approach:

- *Device holder*, which may include one or more quantum devices, housing, shielding, magnets, i/o connection, etc
- *Control Highway*, which may include wiring and/or fibres, free-space optics, active devices (amplifiers), passive devices (attenuators, filters, couplers), opto-electronic (photo diodes), thermalization means, vacuum feed-through, i/o connection, etc
- *Control Electronics/Optics*, which may include output generators, input analyzers, signal processing, i/o connection, as well as low-level firmware to guide the generation of signals and the read-out of their response.
- *Control Software*, which may include calibration means, low-level code to translate instructions from higher software layers into commands for guiding the control electronics/optics.

**Editorial note:** Maybe a better name for “Device holder” could be proposed. A brainstorm on alternative names for this module resulted in names like: device holder / housing shell / environment / module / unit / block / socket / container / vessel / basket for this modules. Good naming suggestions are invited here.

So far, the following architecture families have been identified (in arbitrary order):

- Cryogenic Solid State based
- Room Temperature Solid state based
- Trapped Ions
- Neutral Atoms
- Photonic Quantum Computing
- Other solutions

These architectures are described below in further detail.

A module is an implementation that may be constructed from (smaller) modules and components. It could offer the functionality of a single layer, of multiple layers, or just of a fragment of a layer. A module may also support different operating modes, such that it complies with the requirements of multiple members and/or multiple architecture families. As such, the functionality of a module may cover multiple layers and/or families and/or members.

### 7.1.3.1 Cryogenic solid-state based quantum computing architectures

The members of this architecture family have in common that they all make use of a cryogenic fridge, where the quantum devices in a holder are controlled from outside the fridge by room-temperature electronics. Consequently, a huge amount of control channels is required to interconnect those two, especially when many qubits are to be controlled in a single fridge.

The following members have been identified within this architecture family:

- superconducting
  - transmons
  - flux qubits
- semiconductor spin qubits
- topological qubits
- artificial atoms in solids

The layers in the stack describing this architecture family involves:

- [HL1] *Quantum devices* - The modules in hardware layer HL1 are typically operating at cryogenic temperatures and may be implemented as chip and/or on PCB. It could have tough requirements on shielding, operating temperature, magnetic aspects, etc
- [HL2] *Control highway* – Hardware Layer HL2 covers all infrastructure needed for transporting microwave, lightwave, RF and DC signals (via electrical and/or optical means) between the control electronics at room temperature and the quantum device at cryogenic temperatures. It is a mix of transmission lines, filtering, attenuation, amplification, (de)multiplexing, etc. A huge number of control channels are required to control many qubits in a single fridge (which clarifies the name) and this can easily become very bulky. It could have tough requirements on aspects like heat-flow, thermal noise and vacuum properties.
- [HL3] *Control electronics* – Hardware layer HL3 covers all electronics for generating, receiving, and processing microwave, RF and DC signals. Some implementations make use of routing/switching and/or multiplexing of control signals at room temperatures. It may have some firmware on board to guide the signal generation and signal processing.
- [HL4] *Control software* - Layer HL4 can be a mix of hardware and low-level driver software for instructing the control electronics. It also has a software interface to higher layers for receiving sequences of instructions about when, where and what pulses are to be generated, how to process and read-out the response and means for performing calibration.

Further details about solid-state based quantum computing have been elaborated in annex E

*Editorial note:* Input from more experts is needed here. Especially on topological qubits, as well as “gated-based” versus “adiabatic” architectures. Contributions invited

### 7.1.3.2 Room-temperature solid-state based quantum computing architectures

The members of this architecture family have in common that solid-state qubits are all operating at room temperatures. Examples of members in this architecture family are:

- Artificial atoms in solids, such as NV Centers
- optical Quantum Dots

*Editorial note:* Input from more experts is needed here.

### 7.1.3.3 Trapped-Ion quantum computing

The members of this architecture family can operate either at room temperature or at cryogenic temperature (4K). Quantum devices are controlled by electronics operating either at room temperature or under cryogenic conditions. In the case of large number of qubits the amount of routing signals become bulky, and efficient thermal management, low-noise electrical and magnetic components are required.

- Room temperature
  - Optical qubits
  - Raman qubits
  - Spin (microwave) qubits
- Cryogenic (4K)
  - Optical qubits
  - Raman qubits
  - Spin (microwave) qubits

The layers in the stack describing this architecture family involves:

- [HL1] *Quantum devices* - The modules in hardware layer HL1 are typically operating either at room temperature or at cryogenic temperatures and may be implemented stand-alone, as chip and/or on PCB. The device may have though requirements on materials compatibility, shielding, operating temperature, electrical and magnetic aspects, vacuum properties, etc
- [HL2] *Control highway* – Hardware Layer HL2 covers all infrastructure needed for routing light, microwave signals, RF and DC signals between the control electronics at room temperature and the quantum device at room temperature or cryogenic temperatures. It is a mix of transmission lines, filtering, attenuation, amplification, (de)multiplexing, etc. The number of control channels required to control many qubits in single vacuum housings rapidly grows with the complexity and capabilities of the quantum devices. Stringent requirements to convey on efficient thermal management, low-noise (electrical and magnetic) environment and materials with low outgassing (especially for the room temperature) vacuum components need to be taken into account during design, development and production.
- [HL3] *Control electronics* – Hardware layer HL3 covers all electronics for generating, receiving, and processing microwave, RF and DC signals. Some implementations make use of routing/switching and/or multiplexing of control signals at room temperatures and at cryogenic. It may have some firmware on chip/board to route signal preparation, control, and processing/readout.
- [HL4] *Control software* - Layer HL4 can be a mix of hardware and low-level driver software for instructing the control electronics. It also has a software interface to higher layers for receiving sequences of instructions about when, where and what pulses or signals are to be generated, how to process and read-out the response, and means for performing calibration.

*Editorial note:* Input from more experts is needed here.

Linear vs. quantum CCD should be reflected in some way Examples are:

- Linear Architecture
  - Cryogenic or Room Temperature
  - Optical or Microwave
- Quantum CCD
  - Cryogenic or Room Temperature
  - Optical or Microwave

This architecture family deserves a dedicated annex for describing further details, so contributions about that are invited as well

#### 7.1.3.4 Neutral Atoms

Examples of members in this architecture family are:

- Collision-based
- Rydberg-based

*Editorial note:* Input from more experts is needed here.

#### 7.1.3.5 Photonic Quantum Computing

These architectures have in common that they rely on optical principles. The following hardware layers have been identified:

The layers in the stack describing this architecture family involves:

- [HL1] *Quantum photonic devices* - The modules in hardware layer HL1 contains sources, detectors, (non)linear optical devices, and feedback mechanisms. These devices are combined into a quantum photonic architecture.
- [HL2] *Control highway* – Hardware layer HL2 covers all wiring infrastructure (optical and electrical) needed for controlling the quantum photonic devices.
- [HL3] *Control electronics* – Hardware layer HL3 contains all electronics and/or optics to control and read-out the quantum photonic devices. It may have some firmware on board to guide the signal generation and signal processing.
- [HL4] *Control software* – Layer HL4 can be a mix of hardware and low-level driver software for instructing the control electronics. It also has a software interface to higher layers for receiving sequences of instructions about when, where and what signals are to be generated, how to process their response

Further details about photonic-based quantum computing have been elaborated in annex G

*Editorial note:* Input from more experts is needed here.

Some questions to be solved are:

is structuring along computational models a good way forward? How about

- continuous variables
- cluster states / measurement based
- Knill-LaFlamme-Milburn
- Boson sampling

### End of literal text proposal

#### 7.1.4 Modularity and layering of software stack

*Editorial note:* Experts, who are more dedicated to software are invited to elaborate on relevant text and associated naming.