CEN/CLC/JTC 22/WG 3 "Quantum Computing and Simulation"

Convenor: **Paul Alexandra Mme**

QEC_requirements_V11

Document type	Related content	Document date	Expected action
Project / Draft	Meeting: Madrid (Spain) 30 Oct 2025	2025-10-07	**COMMENT/REPLY** by 2025-10-30

Description

Dear members,

Please find attached the lastest version of the QEC requirements.

Kind regards,

CEN/CLC/JTC 22/WG 3 "Quantum Computing and Simulation"
WG Secretariat: **xxNSBxx**
Convenor: **xxWGCHAIRxx**

CEN-CLC-JTC 22_##_Text for Quantum Error-correction functionality

Document type	**Meeting**	**Document date**	**Expected action**
Contribution	JTC22-WG3	2025-10-06	For decision

Title	Proposal for adding text on quantum error correction and fault-tolerant quantum computation
Authors	Valentin Torggler (v.torggler@parityqc.com), Michael Fellner (m.fellner@parityqc.com) Rob F.M. van den Brink (rob.vandenbrink@delft-circuits.com), Michele Amoretti (michele.amoretti@unipr.it), Juan Carlos Boschero (juan.boschero@tno.nl),
Organisations	ParityQC, Delft Circuits, University of Parma, TNO
Representing	DIN, ASI, NEN, UNI
Work Item number	N/A
Work Item title	Cryogenic Solid-State Quantum Computing
Summary	This document contributes a literal text proposal for functional descriptions and functional requirements on Quantum Error Correction (QEC) and Fault-Tolerant Quantum Computation (FTQC)

The aim of this contribution is to develop functional descriptions and functional requirements on Quantum Error Correction (QEC) and Fault-Tolerant Quantum Computation (FTQC).

8.1 Functional descriptions

8.1.1 Instruction Set Architecture

8.1.2 Quantum Error Correction and Fault-Tolerant Quantum Computation

The aim of quantum error correction (QEC) is to protect the quantum information stored in the quantum device and/or correct for errors in the quantum information caused by quantum operations or by decoherence. QEC is essential for fault-tolerant quantum computation (FTQC), i.e., performing a quantum computation on protected quantum information. Together, QEC and FTQC reduce the effects of decoherence, noise, faulty gates, faulty state preparation, and faulty measurements.

Different approaches to QEC include:

- Software-level QEC, emulating logical qubits from a larger redundant set of physical qubits, using redundancy to protect the quantum state and/or make errors detectable and correctable. It is sometimes called measurement-based QEC.
- Hardware-level QEC, protecting the quantum state on the hardware level, using intrinsic properties of the dedicated types of qubits.
- Circuit-level QEC, encoding quantum information redundantly and using quantum circuits to detect and correct errors.

In software-level and circuit-level QEC, logical qubits are encoded into a QEC code comprising redundant physical qubits. The minimum number of physical qubits per logical qubit depends on the code type (especially its “threshold”) and the quality of the physical qubits, typically ranging from 10 to 1000.

Common examples of software-level QEC are based on stabiliser codes, such as the surface code. A stabiliser code comprises data qubits, which carry the redundant quantum information, and measurement qubits, which measure stabilisers of the code. A stabiliser measurement involves operations such as two-qubit gates between data and measurement qubits, optional single-qubit gates, and finally measuring the measurement qubit. This projects the physical state into one of several subspaces and yields *error syndromes*. These indicate whether the redundant information is consistent, or an error has occurred. A decoder uses syndromes to infer the most likely error subspace of the physical state. Based on this, the error-free state can be restored by applying single-qubit operations (typically Pauli operations). Importantly, stabiliser measurements do not reveal logical quantum information, and therefore preserve it.

In software-level QEC, the functions of defining the code, running stabiliser measurement cycles, decoding, and identifying correction operations are provided by a QEC unit implemented in higher layers (e.g., HAL). The QEC unit can comprise dedicated hardware (FPGA, CPU, GPU, …). The QEC unit connects to the ISA, which provides hardware information and measurement results to the QEC unit and executes stabiliser measurement cycles and correction operations. The interplay between QEC unit and ISA is detailed in the following steps:

- *Measurement*: A stabiliser measurement may involve resetting the measurement qubit and performing a sequence of gates and measurements. To perform such a cycle, the QEC unit

instructs the ISA to execute either a circuit or a predefined macro instruction. The ISA then returns the results.

- *Decoding*: The error syndromes are input to the decoder in the QEC unit. The decoder outputs the most likely error subspace of the data qubits. This step is computationally intensive and may run on dedicated hardware.
- *Correction*: Based on the most likely error subspace, the QEC unit identifies correction operations (e.g., Pauli flips). These can be sent directly to the ISA for execution or collected in a *Pauli frame* and may be applied later as a cumulative correction or accounted for in the measurements.

The schedule of QEC cycles may be autonomously defined by the QEC unit, depending on qubit performance, code choice, and parameters (e.g., code distance). Alternatively, scheduling may occur at a lower level (e.g., by the ISA) or be user-defined. The entire process, including decoding, must run during QPU runtime, which imposes strict requirements on hardware speed, data throughput, and control software.

For performing logical operations on encoded quantum states, FTQC techniques are employed. Here, a logical gate or measurement corresponds to a sequence of physical operations applied to the physical qubits. These sequences must be fault-tolerant, i.e., designed so that an initial error does not spread into an uncorrectable one. The sequences may comprise magic state distillation or gate teleportation sub-circuits, among others.

An FTQC unit accepts input from a higher layer, such as a logical quantum circuit provided by the user. The FTQC unit translates each logical gate into a sequence of ISA instructions for physical gates and measurements. In some cases, these instructions may adapt dynamically based on intermediate measurement results from the QPU.

8.2 Functional requirements

8.2.1 Instruction Set Architecture

8.2.2 Quantum Error Correction and Fault-Tolerant Quantum Computation

8.2.2.1 Initialization requirements

Quantum error correction requires means from the control software layer for its initialization. This can be achieved via initialization instructions supported by the ISA, such as:

- Instructions for inquiring the organization of qubits, such as *“"Get register set”, “Get register width”, “Get adjacency matrix”,* or the types of the qubits – With these instructions, the QEC unit should request information on the organization of the qubits from the ISA. The QEC unit should use this information e.g. to provide codes suitable for the organization of the qubits. For instance, the QEC unit should consider which qubits are assigned as data qubits and which qubits are assigned as measurement qubits by the ISA. When the ISA does not provide this information, the QEC unit should assign the roles of the qubits in the code.

- Instructions for inquiring supported gates, such as *“Get primitive gates”, “Get native gates”, “Get gate duration”* – With these instructions, the QEC unit should request information about the supported gates. The QEC unit should use this information e.g. to provide stabilizer measurement cycles or subcircuits for fault-tolerant logical gates and logical measurements adapted to the Quantum Processing Unit.
- Instructions for inquiring qubit or register performance, such as *“Get gate performance”, “Get instruction duration”, “Get qubit properties”* – With these instructions, the QEC unit can request information on qubit or register performance from the ISA. The QEC unit can use this information e.g. to choose a suitable code and/or suitable code parameters (e.g. code distance) for the given gate performance.
- Initialization instructions, such as *“Set pre-defined instruction sequences”* – The ISA should be initialized with a QEC code specific pre-defined instruction effecting a sequence of native or primitive gates and measurements (e.g., a pre-defined instruction for effecting a stabilizer measurement, or for subcircuits of fault-tolerant quantum computation such as magic state distillation or gate teleportation schemes).

8.2.2.2 Execution requirements

Quantum error correction requires means to change the status of qubits. This can be achieved via execution instructions supported by the ISA, such as:

- Preparation/initialization level instructions, such as *“Reset qubit”* – The ISA should be able to reset a qubit to the |0> state at any time, e.g., for initializing a measurement qubit for the next stabilizer measurement as instructed by the QEC unit or if necessary for performing FTQC subcircuits such as magic state distillation or gate teleportation schemes as instructed by the FTQC unit.
- Gate execution instructions, such as *“select gates”, “fire gates”* – The ISA should be able to select and fire gates to perform correction operations or operations necessary for performing a stabilizer measurement as instructed by the QEC unit and gate sequences for performing logical gates or FTQC subcircuits such as magic state distillation or gate teleportation schemes as instructed by the FTQC unit.
- Measurement instructions, such as *“select bases”, “fire measurements”* – The ISA should be able to perform a measurement on the measurement qubits for performing a stabilizer measurement as instructed by the QEC unit or for performing logical measurements or for FTQC subcircuits such as magic state distillation or gate teleportation schemes as instructed by the FTQC unit.

8.2.2.3 Latency requirements

Quantum error correction requires low latency responses from qubits through all lower layers of the stack. This is essential to perform real-time correction of quantum errors.

- The ISA may facilitate this for the *control software* layer by speed-optimizing multi-sequence instructions that have been defined via initialization instructions such as *Set pre-defined instruction sequences.*
- The *control electronics* layer may facilitate this via hardware support of multi-sequence instructions. It may minimize instruction overhead by firing a predefined *sequence* of pulses through a single command.
- The control highway layer may facilitate this via short signal channels between electronics and qubits to minimize propagation delay of signals in these channels.