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

Convenor: **PAUL Alexandra MME**

Input - HAL_MappingRouting_V02

Document type	Related content	Document date	Expected action
Meeting / Document for discussion	Meeting: VIRTUAL 17 Jun 2026	2026-06-17	

Description

Dear members,

Please find attached the input for the EAS on HAL regarding "Mapping Routing" by Alejandra Ruiz Lopez (Technalia).

Best regards,

Simon Del Nin

Library on Mapping and Routing

DRAFT V02

Date of submission:	2026-06-11
Submitted by:	Alejandra Ruiz - (alejandra.ruiz@tecnalia.com)
Supported by:	Rob van den Brink
Expected action:	Vote
Expected date:	2026-06-17 (JTC22/WG3 meeting)
WG3-Project:	HAL

Abstract

Document N253 gives the latest Draft (V04) for a Technical Specification (TS) about the Hardware Abstraction Layer. Section 8.3 is aimed for a library about Mapping and Routing, but its content is still to be elaborated.

This contribution proposes literal text for inclusion into a section 8.3 of the HAL, and describes a library for Mapping and Routing.

Literal text proposal

Start of literal text proposal

8.3. Library on Mapping and Routing

The Library for Mapping and Routing is a specialized software component within the HAL designed to translate the logical qubit layout and intended interactions of a quantum program into the actual physical topology of the underlying hardware. Its primary purpose is to abstract the complexities of physical qubit connectivity, enabling higher-level software and compilers to operate independently of hardware-specific details and constraints.

Mapping variables to physical qubits is a transversal challenge across all quantum hardware architectures, including gate-based and annealing systems. By providing a standardized mechanism for this process, the HAL aims to improve the scalability and efficiency of quantum computing solutions. The mapping process shall aim to minimize the number of physical qubits used to mitigate the impact of noise and increase the probability of high-quality solutions.

This library is an optional component of the HAL. Depending on the vendor implementation, a mapping library may provide support for a single paradigm (e.g., exclusively for gate-based systems) or comprehensive support for multiple paradigms. Both approaches are acceptable, as the HAL is designed to be flexible and provide information regarding the underlying

architecture—such as "gate-based", "annealing", or "simulation"—to ensure coherence across the system.

8.3.1 Functional Requirements

To ensure that mapping and routing operations are transparent to the user and higher software layers, the library shall adhere to the following requirements:

- **Topology-Awareness and Transparency**: The library shall utilize HAL inquiry mechanisms, specifically the `"Get adjacency matrix"` instruction, to automatically retrieve the physical connectivity of quantum registers. This ensures the mapping process is handled internally by the HAL, presenting a unified interface and hiding implementation-specific details.
- **Mapping Strategy Control**: The HAL shall enable higher layers to define the mapping strategy through operational modes. The user shall be able to select from different libraries from different providers and select the mode. If there is no specific mode selected it is considered default mode, the library shall automatically apply the system's default embedding or transpiling algorithm to ensure a seamless execution flow for standard problems. The library shall expose a set of available mapping algorithms, allowing the user or compiler to select the most appropriate heuristic based on problem characteristics and hardware topology or a specific mapping defined by the user.
- **Algorithmic Flexibility**: The library shall be designed to accommodate various embedding and routing heuristics, including but not limited to:
    - Iterative reduction of physical qubit usage through search algorithms, with [2,3] or without [1,4] intelligent initialization.
    - Structured embedding methods for fully connected or dense instances [5,6,7].
    - Reinforcement learning based methods [8,9,10].
    - Iterative refinement of embeddings through probabilistic decision-making [10].
- **Plug-in Extensibility**: In alignment with the general HAL architecture, this library shall be plug-in extendable. This allows for the integration of proprietary mapping algorithms via a uniform interface and the use of dynamic linking to support heterogeneous platforms without requiring the recompilation of higher layers.
- **Integration with Execution Flow**: The mapping functionality shall be integrated into the HAL's instruction set, utilizing commands such as `"set mapping"` to construct the qubit topological mapping and `"def registers"` to group qubits. These instructions shall operate transparently, allowing compilers to optimize execution without direct interaction with proprietary hardware interfaces.

8.3.2 Interface and Integration

In alignment with the general HAL architecture, the Mapping and Routing library shall:

- Operate as a preprocessing step that translates logical instructions into physical-layer assignments before they are dispatched to the control software layer.
- Be selectable via a "Select Library" instruction, enabling the HAL to switch between different mapping library or versions dynamically.

- Provide reports to higher layers regarding the quality or feasibility of the generated mapping (e.g., number of physical qubits required versus available).

[1] J. Cai, W. G. Macready, A. Roy, A practical heuristic for finding graph minors (2014).
[2] J. P. Pinilla, S. J. Wilton, Layout-aware embedding for quantum annealing processors, in: International Conference on High Performance Computing, Springer, 2019, pp. 121–139.
[3] S. Zbinden, A. Bärtschi, H. Djidjev, S. Eidenbenz, Embedding algorithms for quantum annealers with chimera and pegasus connection topologies, in: International Conference on High Performance Computing, Springer, 2020, pp. 187–206.
[4] Zou, H., Treinish, M., Hartman, K., Ivrii, A., & Lishman, J. (2024). LightSABRE: A lightweight and enhanced SABRE algorithm. *arXiv preprint arXiv:2409.08368*.
[5] T. Boothby, A. D. King, A. Roy, Fast clique minor generation in chimera qubit connectivity graphs, Quantum Information Processing 15 (2016) 495–508.
[6] T. D. Goodrich, B. D. Sullivan, T. S. Humble, Optimizing adiabatic quantum program compilation using a graph theoretic framework, Quantum Information Processing 17 (2018) 1–26.
[7] Jüttner, A., & Madarasi, P. (2018). VF2++—An improved subgraph isomorphism algorithm. *Discrete Applied Mathematics*, *242*, 69-81.
[8] Ngo, H., Do, N., Vu, M., Jeter, T., Kahveci, T., & Thai, M. (2025). CHARME: A chain-based reinforcement learning approach for the minor embedding problem. *ACM Transactions on Quantum Computing*, *7*(1), 1-28.
[9] Dubal, A., Kremer, D., Martiel, S., Villar, V., Wang, D., & Cruz-Benito, J. (2025). Pauli network circuit synthesis with reinforcement learning. *arXiv preprint arXiv:2503.14448*.
[10] Kremer, D., Villar, V., Paik, H., Duran, I., Faro, I., & Cruz-Benito, J. Practical and efficient quantum circuit synthesis and transpiling with reinforcement learning (2024). *arXiv preprint arXiv:2405.13196*, *50*.
[11] D. E. Bernal, K. E. Booth, R. Dridi, H. Alghassi, S. Tayur, D. Venturelli, Integer programming techniques for minorembedding in quantum annealers, in: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Springer, 2020, pp. 112–129.

End of literal text proposal