-ES_52: Traductor OpenQASM a construcciones V

Autor y proyecto: Víctor Estrada Díaz, ES_52, UNIHOLOG.

Texto: **CC BY-NC-SA 4.0**. Código: **HL3**. Principios éticos declarados por el autor: respeto a la dignidad humana, no uso lucrativo sin permiso y prohibición de usos dañinos. Prohibidos: guerras, violación de derechos humanos, daño ambiental, explotación, vigilancia masiva y discriminación. Declaración de licencia dual.

Qué construye

El traductor lee **OpenQASM 2.0** mediante el analizador de Qiskit; no ejecuta Qiskit como simulador. Genera un plan de conexiones V y código Python, C++ y Prolog para preparar ese plan y disparar su lectura. No calcula probabilidades, respuestas ni un registro de amplitudes.

Se conserva la preparación declarada de referencia cero, el orden de las operaciones y la correspondencia entre cada unidad y su bit clásico. Los ángulos son controles del circuito, no sorteos de resultados. Las definiciones de puertas se expanden respetando sus conexiones.

Las medidas terminales del QASM se agrupan como una lectura conjunta en $t = t_0$. Se exige exactamente una medida por unidad, con destinos clásicos distintos. No se agregan medidas ausentes ni se trasladan operaciones que estaban después de una medida.

El reloj se consulta al disparar. Antes de la lectura, la interfaz no expone valores internos y presenta @ como resultado desconocido. Reconsultar una lectura ya realizada no produce un nuevo disparo. Un disparo fallido consume la preparación: no se repite silenciosamente.

Alcance inicial

Se traducen H, X, CNOT, Toffoli, SWAP, CZ, Rx, Rz, fases P/U1, S, S inversa, T, T inversa, Z y fase controlada CU1/CP cuando el analizador admite su declaración. Identidad y barreras no cambian el plan.

Se rechazan reset, medidas intermedias, decisiones basadas en lecturas, medidas parciales, reutilización de destinos de medida y operaciones sin una traducción definida. OpenQASM 3, U/U2/U3, Y y Ry no están admitidos en esta versión. El rechazo es una limitación explícita del traductor, no una conclusión sobre las posibilidades de V.

Uso local

Desde la raíz SRC-VED, con el entorno de comparación ya instalado:

```
& 'DEMUESTRA/.venv-comparacion/Scripts/python.exe' -B `
  DEMUESTRA/comparacion/traductor_qasm_v.py `
  DEMUESTRA/comparacion/ejemplos_qasm/par_h_cnot.qasm `
  --salida DEMUESTRA/Proyectos/ES_52/traducciones/par_h_cnot
```

Produce `circuito_v.py`, `circuito_v.hpp`, `circuito_v.cpp`, `circuito_v.pl` y `circuito_v.json`. El JSON registra las conexiones, la medida y la huella del QASM, no estados internos. No sobrescribe salidas existentes.

Conexión a una realización V

Python: con `DEMUESTRA/comparacion` en la ruta de importación, `construir(realizacion)` configura la máquina temporal existente. La realización proporciona `preparar(plan)` sin devolver lecturas y `disparar(t0)` para la única lectura conjunta.

C++: el encabezado requiere una implementación de `Realizacion`. `MaquinaV` prepara una vez y usa `steady_clock` al disparar. Conserva únicamente la lectura posterior, además del plan de conexiones conocido.

Prolog: el módulo está orientado a SWI-Prolog. Se conectan los predicados multifile `qasm_v:realizar_preparacion/1` y `qasm_v:realizar_disparo/2`. `get_time/1` registra el reloj civil en segundos al disparar; no es un reloj monotónico ni una calibración del instante de conversión de un detector. Un mutex protege la medida única.

En los tres casos, la medida no recorre el circuito ni alternativas de salida. Solo dispara la realización y valida los bits recibidos. Leer o copiar los bits ya medidos no es recorrer los posibles estados.

Qué no afirma

La traducción de conexiones no sustituye la realización de las puertas. El traductor no incluye un detector inventado, una respuesta obtenida de Q ni el modelo numérico retirado. El contrato de interfaz tampoco puede garantizar qué hace internamente una realización externa: habrá que inspeccionarla y comprobarla con las mismas restricciones.

Las pruebas con detectores de ensayo verifican la traducción, la lectura única y el bloqueo previo; no comprueban la dinámica física de H/CNOT. No certifican ni descartan equivalencia Q-V, ni igualdad de recursos o tiempos. Esas comparaciones requieren ejecuciones de las realizaciones.

Analizador utilizado: documentación oficial de OpenQASM 2 y Qiskit.

Verificación local del 10 de octubre de 2026

La batería Python ejecuta correctamente 52 pruebas, incluidas ocho del traductor; las 14 del ensayo numérico retirado siguen omitidas. SWI-Prolog ejecuta correctamente tres pruebas sobre el ejemplo

traducido: resultado desconocido antes de medir, disparo único y rechazo de una segunda preparación. Sus lecturas son datos de ensayo declarados.

El encabezado C++ se ha generado, pero no se ha compilado: no se encontró un compilador C++ local en las rutas comprobadas. No se declara verificado ese destino al mismo nivel que los controladores Python y Prolog.