QASM2 Exporter: register name clashes with names `qelib1.inc` #10459

Edit New issue

Open **ANONYMOUS AUTHOR** opened this issue last week · 2 comments

ANONYMOUS AUTHOR commented last week • edited

Environment

- **Qiskit Terra version**: 0.43.1 meta package, terra 0.24.1
- **Python version**: 3.10
- **Operating system**: docker continuumio/miniconda3

What is happening?

When I declare a register named with the same name of a gate and then export it in QASM, the exporter generates invalid QASM where the register name clashes with the std gates (due to lack of escaping of the register name).

How can we reproduce the issue?

Run this python script:

```
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister
qr = QuantumRegister(2, name="q")
crz = ClassicalRegister(1, name="crz")
qc = QuantumCircuit(qr, crz)
qc.x(qr)
qc.measure(qr[0], crz[0])
qc.qasm(filename="my.qasm")

QuantumCircuit.from_qasm_file("my.qasm").draw(output='mpl')
```

Produces this error:

```
qiskit.qasm.exceptions.QasmError: "Duplicate declaration for creg 'crz' at line 4, file my.qasm.\nPrevious occu
```

The error happens only when importing the qasm file, whereas I believe the mistake is done by the exporter, which generates a file in which `crz` is defined both as standard gate (in `qelib1.inc`) and as register (via my code).

This is the genrated QASM:

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[2];
creg crz[1];
x q[0];
x q[1];
measure q[0] -> crz[0];
```

What should happen?

I would have expected the QASM exporter to generate a valid qasm, ready to be reimported.

Any suggestions?

I would suggest to improve the exporter by escaping any register name to avoid clashes with standard libraries.

Add tasklist

ANONYMOUS AUTHOR added the bug label last week

Abhiraj-Shrotriya commented last week • edited

It would work if we made the exporter to

> escape any register name

but I think it would also be good if we have a convention that classical registers should start with '_' or something like that. We can do this for Quantum registers too. The exporter can automatically add these 'convention symbols' while exporting.

Please correct me if my idea is wrong

Assignees
No one assigned

Labels
bug

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

Notifications Customize

Unsubscribe

You're receiving notifications because you authored the thread.

3 participants

Please correct me if my idea is wrong.

jakelishman commented last week Member

We don't typically want to universally escape names, because that makes the output harder to read for users, and it means that reading an exported OQ2 string *back* into the program gives things surprising names (or we have to have code to attempt "demangling" of names, etc, except we might not be given an OQ2 file with mangled names, so we need to have another "don't demangle" case, etc, etc).

In this case, the problem is mostly just that the OQ2 exporter doesn't put *all* its defined variables into a global symbol table, so it doesn't notice when there are clashes. It'd probably be fairly easy to a) ensure that the register name-escaping code takes into account any gates defined in the header files and b) enter the used register names as "reserved" into the object that functions as a sort of symbol-table for gate definitions, so any *subsequent* gate definitions don't conflict.

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close issue Comment

Remember, contributions to this repository should follow its contributing guidelines, security policy, and code of conduct.