Qasm exporter fails with `ECRGate` `QasmError: "Duplicate declaration for gate 'rzx'` #XXXX

Edit New issue

Closed **ANONYMOUS** opened this issue 8 days ago · 1 comment

ANONYMOUS commented 8 days ago

Environment

- **Qiskit Terra version**: 0.19.1
- **Python version**: 3.8
- **Operating system**: Ubuntu 18.04.6 LTS

What is happening?

The qasm exporter on a circuit with an `ECRGate` outputs an invalid QASM code which cannot be parsed back.

How can we reproduce the issue?

Run this circuit:

```
import qiskit
from qiskit import QuantumRegister
from qiskit import Aer, transpile, execute
qc = QuantumCircuit(2, 2, name='qc')
qc.ecr(0, 1)
qc.measure_all()
qc.qasm(formatted=True)
```

Output

```
OPENQASM 2.0;
include "qelib1.inc";
gate rzx(param0) q0,q1 { h q1; cx q0,q1; rz(-pi/4) q1; cx q0,q1; h q1; }
gate rzx(param0) q0,q1 { h q1; cx q0,q1; rz(pi/4) q1; cx q0,q1; h q1; }
gate ecr q0,q1 { rzx(pi/4) q0,q1; x q0; rzx(-pi/4) q0,q1; }
qreg q[2];
creg c[2];
creg meas[2];
ecr q[0],q[1];
barrier q[0],q[1];
measure q[0] -> meas[0];
measure q[1] -> meas[1];
```

Read the QASM back:

```
qc = QuantumCircuit.from_qasm_str(qc.qasm())
```

Output

```
QasmError                                 Traceback (most recent call last)
/tmp/ipykernel_35760/1770840892.py in <module>
----> 1 qc = QuantumCircuit.from_qasm_str(qc.qasm())
      2 counts = execute(qc, backend=Aer.get_backend('qasm_simulator'), shots=1024).result().get_counts(qc)
      3 counts

/qiskit/circuit/quantumcircuit.py in from_qasm_str(qasm_str)
   2362         """
   2363         qasm = Qasm(data=qasm_str)
-> 2364         return _circuit_from_qasm(qasm)
   2365
   2366     @property

/qiskit/circuit/quantumcircuit.py in _circuit_from_qasm(qasm)
   4695     from qiskit.converters import dag_to_circuit
   4696
-> 4697     ast = qasm.parse()
   4698     dag = ast_to_dag(ast)
   4699     return dag_to_circuit(dag)

/qiskit/qasm/qasm.py in parse(self)
     51         with QasmParser(self._filename) as qasm_p:
     52             qasm_p.parse_debug(False)
---> 53             return qasm_p.parse(self._data)

/qiskit/qasm/qasmparser.py in parse(self, data)
```

Assignees
No one assigned

Labels
bug qasm status: duplicate

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

Notifications Customize
Unsubscribe
You're receiving notifications because you authored the thread.

2 participants

```
/qiskit/qasm/qasmparser.py in parse(self, data)
   1138     def parse(self, data):
   1139         """Parse some data."""
-> 1140         self.parser.parse(data, lexer=self.lexer, debug=self.parse_deb)
   1141         if self.qasm is None:
   1142             raise QasmError("Uncaught exception in parser; " + "see previous messages for details.")

/ply/yacc.py in parse(self, input, lexer, debug, tracking, tokenfunc)
    331             return self.parseopt(input, lexer, debug, tracking, tokenfunc)
    332         else:
--> 333             return self.parseopt_notrack(input, lexer, debug, tracking, tokenfunc)
    334
    335

/ply/yacc.py in parseopt_notrack(self, input, lexer, debug, tracking, tokenfunc)
   1118                           del symstack[-plen:]
   1119                           self.state = state
-> 1120                           p.callable(pslice)
   1121                           del statestack[-plen:]
   1122                           symstack.append(sym)

/qiskit/qasm/qasmparser.py in p_gate_decl_2(self, program)
    613             )
    614         self.pop_scope()
--> 615         self.update_symtab(program[0])
    616
    617     def p_gate_scope(self, _):

/qiskit/qasm/qasmparser.py in update_symtab(self, obj)
     70         if obj.name in self.current_symtab:
     71             prev = self.current_symtab[obj.name]
---> 72             raise QasmError(
     73                 "Duplicate declaration for",
     74                 obj.type + " '" + obj.name + "' at line",

QasmError: "Duplicate declaration for gate 'rzx' at line 4, file .\nPrevious occurrence at line 3, file "
```

What should happen?

I would have expected the final qasm to define a "general" `rzx` gate and reuse it:

```
gate rzx(param0) q0,q1 { h q1; cx q0,q1; rz(param0) q1; cx q0,q1; h q1; }
gate ecr q0,q1 { rzx(pi/4) q0,q1; x q0; rzx(-pi/4) q0,q1; }
```

Any suggestions?

The problem is in the `qasm()` function of the `QuantumCircuit`, in particular it fails to convert `ecr` which has two identical `rzx` instructions. I noticed that by having two `RZXGate` directly in the circuit, gives a correct result (even if it seems a bit odd, given that the parametrization is not used):

```
qc = QuantumCircuit(2, 2, name='qc')
qc.rzx(0.78, 0, 1)
qc.rzx(-0.78, 0, 1)
```

```
OPENQASM 2.0;
include "qelib1.inc";
gate rzx(param0) q0,q1 { h q1; cx q0,q1; rz(0.78) q1; cx q0,q1; h q1; }
gate rzx_140348495844112(param0) q0,q1 { h q1; cx q0,q1; rz(-0.78) q1; cx q0,q1; h q1; }
qreg q[2];
creg c[2];
creg meas[2];
rzx(0.78) q[0],q[1];
rzx_140348495844112(-0.78) q[0],q[1];
barrier q[0],q[1];
measure q[0] -> meas[0];
measure q[1] -> meas[1];
```

Maybe, as a quick workaround, we should add this part below within the function `_add_sub_instruction_to_existing_composite_circuits` so that the two rzx gates get two identifiers.

qiskit-terra/qiskit/circuit/quantumcircuit.py
Lines 1689 to 1693 in `ee0d760`

```
1689     if instruction.name in [
1690         instruction.name for instruction in existing_composite_circuits
1691     ]:
1692         # append instruction id to name of instruction copy to make it unique
1693         instruction = instruction.copy(name=f"{instruction.name}_{id(instruction)}")
```

Insert it above this line:

qiskit-terra/qiskit/circuit/quantumcircuit.py
Line 4738 in `ee0d760`

```
4738     existing_composite_circuits.insert(0, sub_instruction)
```

Although, the most elegant solution would be to actually get a parametrizable `rzx`

```
gate rzx(param0) q0,q1 { h q1; cx q0,q1; rz(param0) q1; cx q0,q1; h q1; }
```

Looking forward to hearing your feedback on this odd situation, thanks in advance

ANONYMOUS added the bug label 8 days ago

QISKIT DEV commented 8 days ago — Contributor

Thanks for the report! Right now there are systemic failures in the QASM 2 and QASM 3 exporters with slightly exotic parameterised gates, which stem from design choices in how the `Instruction` class stores its definition. If a gate that the exporter doesn't have special handling for takes a parameter, but the first time it is encountered the parameter is already bound, there's no way for the exporter to retrive the general definition from the instruction instance.

The QASM 3 exporter should at least be "smart" enough to munge the gate names to avoid a collision (see #7335 and its workaround in #7336), but the QASM 2 exporter has had this problem for a long time, so we didn't prioritise folding in a change during the 0.19 rush when the QASM 3 exporter was written.

The new approach to parameters that we're moving towards (see #7624) will let us solve this problem. I'll update #7335 to mention the QASM 2 exporter as well, but then I'll close this as duplicate of that, just to make it a little easier for us to track.

1

QISKIT DEV closed this 8 days ago

QISKIT DEV added qasm status: duplicate labels 8 days ago

ANONYMOUS mentioned this issue 4 days ago

Wrong order of gate definitions in QASM #XXXX — Open

ANONYMOUS mentioned this issue 3 days ago

Qasm exporter fails with `RYYGate` in subcircuit (with inverse) `QasmError: "Duplicate declaration for gate 'ryy'` #XXXX — Open

QISKIT DEV mentioned this issue yesterday

rzx defined twice in QASM export #XXXX — Closed

Write | Preview

Leave a comment

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

Comment

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

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About