Qasm exporter doesn't generate the definition of `C3SXGate` which is given for granted causing `QasmError: "Cannot find gate definition for 'c3sx'` #XXXX

Edit New issue

Closed **ANONYMOUS** opened this issue 7 days ago · 2 comments

ANONYMOUS commented 7 days ago

Environment

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

What is happening?

When exporting to qasm a circuit that contains the `C4XGate`, the newly generated code is incorrect, since it doesn't contain the definition of the `c3sx` gate and reading that qasm output back raises the error ``QasmError: "Cannot find gate definition for 'c3sx'".

How can we reproduce the issue?

Run this program

```
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit.circuit.library.standard_gates import C4XGate
from qiskit import transpile
qr = QuantumRegister(5, name='qr')
cr = ClassicalRegister(5, name='cr')
qc = QuantumCircuit(qr, cr, name='qc')
qc.append(C4XGate(), qargs=[qr[0], qr[1], qr[2], qr[3], qr[4]], cargs=[])
qc.measure(qr, cr)
qc = transpile(qc)
qc.draw(fold=-1)
```

Output:

```
qr_0: ──■──┤M├──────────
        │  └╥┘┌─┐
qr_1: ──■───╫─┤M├───────
        │   ║ └╥┘┌─┐
qr_2: ──■───╫──╫─┤M├────
        │   ║  ║ └╥┘┌─┐
qr_3: ──■───╫──╫──╫─┤M├─
      ┌─┴─┐ ║  ║  ║ └╥┘┌─┐
qr_4: ┤ X ├─╫──╫──╫──╫─┤M├
      └───┘ ║  ║  ║  ║ └╥┘
cr: 5/══════╩══╩══╩══╩══╩═
            0  1  2  3  4
```

If we convert to QASM, it generates:

```
qc.qasm(formatted=True)
```

Output:

```
OPENQASM 2.0;
include "qelib1.inc";
gate rcccx_dg q0,q1,q2,q3 { u2(-2*pi,pi) q3; u1(pi/4) q3; cx q2,q3; u1(-pi/4) q3; u2(-2*pi,pi) q3; u1(pi/4) q3;
gate rcccx q0,q1,q2,q3 { u2(0,pi) q3; u1(pi/4) q3; cx q2,q3; u1(-pi/4) q3; u2(0,pi) q3; cx q0,q3; u1(pi/4) q3;
gate mcx q0,q1,q2,q3,q4 { h q4; cu1(pi/2) q3,q4; h q4; rcccx q0,q1,q2,q3; h q4; cu1(-pi/2) q3,q4; h q4; rcccx_c
qreg qr[5];
creg cr[5];
mcx qr[0],qr[1],qr[2],qr[3],qr[4];
measure qr[0] -> cr[0];
measure qr[1] -> cr[1];
measure qr[2] -> cr[2];
measure qr[3] -> cr[3];
measure qr[4] -> cr[4];
```

And then finally reading back the same generated QASM creates a problem, because the definition of `cs3x` at line 5 is never

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.

2 participants

declared:

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

Output:

```
QasmError                                 Traceback (most recent call last)
/tmp/ipykernel_4861/2855096827.py in <module>
----> 1 qc = QuantumCircuit.from_qasm_str(qc.qasm())

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)
   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_op_2(self, program)
    797         # 1. id is declared as a gate in global scope
    798         # 2. everything in the id_list is declared as a bit in local scope
--> 799         self.verify_as_gate(program[1], program[2])
    800         self.verify_bit_list(program[2])
    801         self.verify_distinct([program[2]])

qiskit/qasm/qasmparser.py in verify_as_gate(self, obj, bitlist, arglist)
    136         """Verify a user defined gate call."""
    137         if obj.name not in self.global_symtab:
--> 138             raise QasmError(
    139                 "Cannot find gate definition for '" + obj.name + "', line",
    140                 str(obj.line),

QasmError: "Cannot find gate definition for 'c3sx', line 5 file "
```

What should happen?

The `c3sx` gate should have been defined somewhere in the output qasm file, so that it becomes a valid qasm file, ready to be imported.

Any suggestions?

During conversion it seems that `c3sx` should be available among the "basic" gates (`existing_gate_names` list):

qiskit-terra/qiskit/circuit/quantumcircuit.py
Line 1635 in `ee0d760`

```
1635     "c3sx",
```

Nevertheless, if we have a look at what is imported with the `qelib1.inc` external library, we do not find any `c3sx`.

https://github.com/Qiskit/qiskit-terra/blob/c816be80a7713af7d39550887f6f8e57e22e09e7/qiskit/qasm/libs/qelib1.inc

So, maybe we should either drop `c3sx` from the list of `existing_gate_names` in the `def qasm()` method or define it properly in the `qelib1.inc`. At the moment these are the two conclusion I came to, but I am looking forward to your feedback.
Thanks in advance

ANONYMOUS added the bug label 7 days ago

QISKIT DEV commented 7 days ago • edited Contributor

Thanks for the report. This is a duplicate of #7148, which does have a fix proposed in #7241 but there's some slight complicating factors around Terra's non-standard version `qelib1.inc` (see also #6125 (comment)). It's stalled a little, in part because sorting out awkward deprecation periods and new features for old QASM 2 libraries that current users are relying on is tricky, and a low priority because we expect most use cases to transpile the circuit to a supported basis before emitting them.

1

QISKIT DEV closed this 7 days ago

QISKIT DEV commented 7 days ago • edited Contributor

Duplicate of #7148

QISKIT DEV marked this as a duplicate of **#7148** 7 days ago

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