

### QuantumRegister

qasm	Return OPENQASM string for this register
check_range(j)	Check that j is a valid index into self

### QuantumCircuit

add( *regs )	add registers
barrier( *tuples )	apply barrier to tuples (reg, idx)
ccx( ctl1, ctl2, tgt )	apply Toffoli
ch( ctl, tgt )	apply CH from ctl to tgt
combine( rhs )	if self has rhs's regs return self + rhs
crz( ctl, tgt )	apply crz from ctl to tgt with angle theta
cswap( ctl, tgt1, tgt2 )	apply Fredkin
cu1( theta, ctl, tgt )	apply cu1 from ctl to tgt with angle theta
cu3( theta, phi, lam, ctl, tgt )	apply cu3 from ctl to tgt with angle theta, phi, lam
cx( ctl, tgt )	apply CNOT from <b>ctl to tgt</b>
cx_base( ctl, tgt )	apply CX from <b>ctl to tgt</b>
cy( ctl, tgt )	apply CY from <b>ctl to tgt</b>
cz( ctl, tgt )	apply CZ from <b>ctl to tgt</b>
extend( rhs )	append rhs to self if self contains rhs's registers modify and return self
first_atomic_gate_host()	return the host list of the leaf gate on the left edge
get_cregs()	get the cregs from the registers
get_qregs()	get the qregs from the registers
h( q )	apply H to q
has_register( register )	return True or False
iden( q )	apply Identity to q

