Qiskit Metal: Superconducting Circuit Simulation

The superconducting circuit simulation uses `Qiskit Metal` to model transmission lines and qubits. Here, the `RouteMeander` component connects superconducting qubits with coplanar waveguides (CPWs).

- The `connect` function automates CPW creation between qubit pins, allowing customization of:
  - **Meander asymmetry**: Adjusts the waveform shape.
  - **Fillet radius**: Defines rounded corners for better transmission.
  - **Lead extensions**: Adds straight sections at the ends.

Integration Example

Below is an example integrating STM32 and Qiskit Metal. This conceptual model simulates CPW behavior based on control parameters received from STM32:

Simulated CPW Tuning with STM32

1. Use I2C to send/receive parameters (e.g., `lead_start`, `asymmetry`) to/from the STM32 microcontroller.
2. Update `Qiskit Metal` CPW models dynamically with the received parameters.

```
from qiskit_metal.qlibrary.tlines.meandered import RouteMeander
from qiskit_metal import Dict

# Function to connect qubits with CPWs
def connect_with_i2c(component_name, component1, pin1, component2,
pin2, length, asymmetry, flip):
    """Connects two pins with a coplanar waveguide (CPW)."""
    myoptions = Dict(
        pin_inputs=Dict(
            start_pin=Dict(component=component1, pin=pin1),
            end_pin=Dict(component=component2, pin=pin2)
        ),
        lead=Dict(start_straight='0.1mm', end_straight='0.1mm'),
        total_length=length,
        fillet='90um',
        meander=Dict(asymmetry=asymmetry),
    )
    if flip:
        myoptions.meander.lead_direction_inverted = 'true'
    return RouteMeander(design, component_name, myoptions)
```

```
# Example data received from STM32 via I2C
i2c_received_data = {
    "lead_start": "0.13mm",
    "lead_end": "0.13mm",
    "asymmetry": "150um",
}

# Use received data in Qiskit Metal
cpw1 = connect_with_i2c('cpw1', 'Q1', 'd', 'Q2', 'c', '6.0mm',
i2c_received_data['asymmetry'], flip=False)
gui.rebuild()
gui.autoscale()
```