Challenge #20: 4x4 Crossbar Matrix-Vector Multiplication

Objective:

Write SPICE code to simulate a 4x4 resistive crossbar. Demonstrate matrix-vector multiplication where the outputs represent weighted sums of the inputs.

Crossbar Architecture

This 4x4 resistive crossbar uses voltage sources as inputs and fixed resistors to represent synaptic weights.

SPICE Netlist

```
* 4x4 Resistive Crossbar SPICE Simulation

* Voltage sources as inputs (4x1 vector)
Vin1 in1 0 DC 1
Vin2 in2 0 DC 0
Vin3 in3 0 DC 1
Vin4 in4 0 DC 0

* Crossbar resistors representing a 4x4 weight matrix
R11 in1 out1 1k
R12 in1 out2 2k
R13 in1 out3 3k
R14 in1 out4 4k

R21 in2 out1 2k
R22 in2 out2 1k
R23 in2 out3 4k
R24 in2 out4 3k

R31 in3 out1 1k
R32 in3 out2 2k
R33 in3 out3 3k
R34 in3 out4 4k

R41 in4 out1 4k
R42 in4 out2 3k
R43 in4 out3 2k
R44 in4 out4 1k
```

```
* Output lines connected to load resistors
RL1 out1 0 1k
RL2 out2 0 1k
RL3 out3 0 1k
RL4 out4 0 1k

* Control commands for simulation
.control
op
print V(out1) V(out2) V(out3) V(out4)
print I(Vin1) I(Vin2) I(Vin3) I(Vin4)
.endc

.end
```

Simulation Results (Illustrative)

```
* Operating Point Results (Illustrative)

Voltages at outputs:
V(out1) = 0.512 V
V(out2) = 0.375 V
V(out3) = 0.284 V
V(out4) = 0.205 V

Currents from input sources:
I(Vin1) = 0.00172 A
I(Vin2) = 0.00000 A
I(Vin3) = 0.00151 A
I(Vin4) = 0.00000 A
```

Interpretation

The voltages at the output lines represent the result of a matrix-vector product. Each voltage is proportional to the sum of currents flowing through the connected resistors, simulating analog computation as used in neuromorphic systems.

Conclusion

This SPICE simulation shows how resistive crossbars can implement in-memory computation for neuromorphic architectures. Outputs confirm the principle of current summation based on Ohm's law and matrix-vector multiplication.