Digital VLSI Design - Project 1

Leakage Current Estimation of ISCAS 74182 circuit

Contributions

Gokulraj R (2020102042) - Leakage Estimation and Report
Ruhul Ameen S (2020102031) - Leakage Estimation and Report
Sushil Kumar Yalla (2020102071) - Ngspice netlist for gates and the final circuit
Chinmay Deshpande (2020102069) - Ngspice netlist for gates and the final circuit

Parameters

- Tech Node: 22nm MGK
- Supply Voltage: 0.8V
- Temperature: 25°C

Stage-1: Single MOS leakage currents

- For single MOS, leakage currents are calculated for all the cases (NMOS on, NMOS off, PMOS on, PMOS off), for varying V_{DS}.
- Both drain voltage and gate voltage are swept from 0V to 0.8V in steps of 0.05V, and the leakage currents are stored in matrices for all possible V_{DS}.

Stage-2: Intermediate Voltage in stack of 2 MOSFETs

- The voltage of the intermediate node in a stack of 2 MOSFETs is obtained for all cases by simulating the corresponding netlist (NMOS stack and PMOS stack for inputs 00, 01, 10, 11).

Stage-3

Two Input Gates

- Since we are building the final circuit using 2 input gates, we estimate the leakage currents of the NAND, NOR, NOT gates first.
- To get the leakage currents of a gate, we get the intermediate voltage from the output of stage-2, and use the voltages (Vd and Vs) of all the transistors to look up the corresponding single MOSFET leakage currents from the output matrices of stage-1.
- Following are the comparisons of the estimated leakages and the leakages from simulating a netlist for a gate. Estimated leakage is an array of `[total_supply_leakage, total_gate_leakage]`.
- **NAND:**
    - Input 00:
        - Estimated from the matrices: `array([2.97757140e-08, 2.94479625e-08])`

    - Simulation:
```
i(vdd) = -2.96412e-08
i(vin1) = 1.550720e-08
i(vin2) = 1.392888e-08
```
  - Input 01:
    - Estimated from the matrices: `array([1.99372144e-08, 2.04640615e-08])`
    - Simulation:
```
i(vdd) = -1.98049e-08
i(vin1) = 1.548900e-08
i(vin2) = -4.97305e-09
```
  - Input 10:
    - Estimated from the matrices: `array([1.72662209e-08, 1.48124605e-08])`
    - Simulation:
```
i(vdd) = -1.69817e-08
i(vin1) = -1.14087e-12
i(vin2) = 1.476574e-08
```
  - Input 11:
    - Estimated from the matrices: `array([3.03749084e-08, 1.60006213e-08])`
    - Simulation:
```
i(vdd) = -3.02927e-08
i(vin1) = -7.98297e-09
i(vin2) = -7.99209e-09
```
- **NOR:**
  - Input 00:
    - Estimated from the matrices: `array([5.80306315e-08, 4.89855324e-08])`
    - Simulation:
```
i(vdd) = -5.75700e-08
i(vin1) = 2.448619e-08
i(vin2) = 2.444923e-08
```
  - Input 01:
    - Estimated from the matrices: `array([4.81225252e-08, 3.28728575e-08])`
    - Simulation:
```
i(vdd) = -4.76244e-08
i(vin1) = 2.291142e-08
i(vin2) = -9.95387e-09
```
  - Input 10:
    - Estimated from the matrices: `array([1.45714001e-08, 7.97156436e-09])`
    - Simulation:
```
i(vdd) = -1.53600e-08
i(vin1) = -8.12494e-09
i(vin2) = 1.145047e-11
```

- Input 11:
  - Estimated from the matrices: `array([1.71906893e-09, 1.51135762e-08])`
  - Simulation:
```
i(vdd) = -1.52818e-09
i(vin1) = -5.03569e-09
i(vin2) = -1.00208e-08
```

- **NOT:**
  - Input 0:
    - Estimated from the matrices: `array([1.99372144e-08, 1.54909205e-08])`
    - Simulation:
```
i(vdd) = -1.98140e-08
i(vin) = 1.548899e-08
```
  - Input 1:
    - Estimated from the matrices: `array([1.51874542e-08, 8.00031063e-09])`
    - Simulation:
```
i(vdd) = -1.51772e-08
i(vin) = -7.99804e-09
```

- In all the above cases, we observe that the estimated leakage current is very close to the leakage currents obtained using simulations.

Final Circuit

- Now that we have the leakage currents of the individual 2-NAND, 2-NOR and NOT gates, we build the final circuit by only using the two input gates. The following is the logic of the final circuit using two input gates.

```
def CLA(P0, P1, P2, P3, G0, G1, G2, G3, Cn):
    K1 = invert(Cn)
    K2 = P0 & G0
    K3 = G0 & K1
    Cnx = invert(K2 | K3)

    K4 = P1 & G1
    K5 = K2 & G1
    K6 = K3 & G1
    K7 = K4 | K5
    Cny = invert(K6 | K7)

    K8 = P2 & G2
    K9 = K4 & G2
    K10 = K5 & G2
    K11 = K6 & G2
    K12 = K8 | K9
```

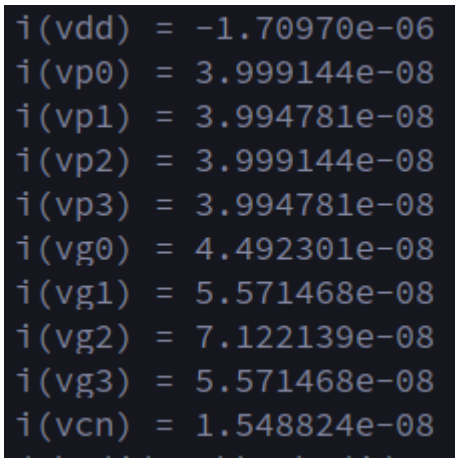
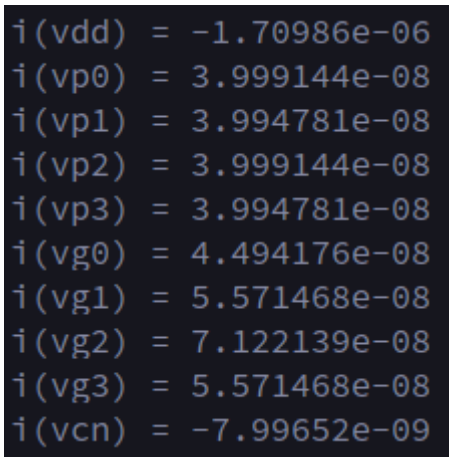
```
    K13 = K10 | K11
    Cnz = invert(K12 | K13)

    K14 = P3 & G3
    K15 = K8 & G3
    K16 = K9 & G3
    K17 = G0 & G1
    K18 = G2 & G3
    K19 = K17 & K18
    K20 = K14 | K15
    K21 = K16 | K19
    G = K20 | K21

    K22 = P0 | P1
    K23 = P2 | P3
    P = K22 | K23

    return Cnx, Cny, Cnz, G, P
```

- The AND and OR gates in the above logic are made using the NAND, NOR and NOT gates, for which we have already estimated and verified the leakage currents.
- Now, the leakage currents in the final circuit is:

```
leakage = (  not_leakage(Cn)
           + nand_leakage(P0, G0) + not_leakage(invert(P0 & G0))
           + nand_leakage(G0, K1) + not_leakage(invert(G0 & K1))
           + nor_leakage(K2, K3)

           + nand_leakage(P1, G1) + not_leakage(invert(P1 & G1))
           + nand_leakage(K2, G1) + not_leakage(invert(K2 & G1))
           + nand_leakage(K3, G1) + not_leakage(invert(K3 & G1))
           + nor_leakage(K4, K5) + not_leakage(invert(K4 | K5))
           + nor_leakage(K6, K7)

           + nand_leakage(P2, G2) + not_leakage(invert(P2 & G2))
           + nand_leakage(K4, G2) + not_leakage(invert(K4 & G2))
           + nand_leakage(K5, G2) + not_leakage(invert(K5 & G2))
           + nand_leakage(K6, G2) + not_leakage(invert(K6 & G2))
           + nor_leakage(K8, K9) + not_leakage(invert(K8 | K9))
           + nor_leakage(K10, K11) + not_leakage(invert(K10 | K11))
           + nor_leakage(K12, K13)

           + nand_leakage(P3, G3) + not_leakage(invert(P3 & G3))
           + nand_leakage(K8, G3) + not_leakage(invert(K8 & G3))
```

```
            + nand_leakage(K9, G3) + not_leakage(invert(K9 & G3))
            + nand_leakage(G0, G1) + not_leakage(invert(G0 & G1))
            + nand_leakage(G2, G3) + not_leakage(invert(G2 & G3))
            + nand_leakage(K17, K18) + not_leakage(invert(K17 & K18))
            + nor_leakage(K14, K15) + not_leakage(invert(K14 | K15))
            + nor_leakage(K16, K19) + not_leakage(invert(K16 | K19))
            + nor_leakage(K20, K21) + not_leakage(invert(K20 | K21))

            + nor_leakage(P0, P1) + not_leakage(invert(P0 | P1))
            + nor_leakage(P2, P3) + not_leakage(invert(P2 | P3))
            + nor_leakage(K22, K23) + not_leakage(invert(K22 | K23))
    )
```

Verification with simulation results

- The ngspice netlist of the final circuit is simulated to obtain the leakage currents for various input combinations (P0, P1, P2, P3, G0, G1, G2, G3, Cn) and they are used for the verification of the estimated leakage currents.
- Input 000000000:
  - Estimated: `array([1.5176009e-06, 1.2280603e-06])`
  - Simulation: 
```
i(vdd) = -1.70970e-06
i(vp0) = 3.999144e-08
i(vp1) = 3.994781e-08
i(vp2) = 3.999144e-08
i(vp3) = 3.994781e-08
i(vg0) = 4.492301e-08
i(vg1) = 5.571468e-08
i(vg2) = 7.122139e-08
i(vg3) = 5.571468e-08
i(vcn) = 1.548824e-08
```
- Input 000000001:
  - Estimated: `array([1.52268964e-06, 1.22955359e-06])`
  - Simulation: 
```
i(vdd) = -1.70986e-06
i(vp0) = 3.999144e-08
i(vp1) = 3.994781e-08
i(vp2) = 3.999144e-08
i(vp3) = 3.994781e-08
i(vg0) = 4.494176e-08
i(vg1) = 5.571468e-08
i(vg2) = 7.122139e-08
i(vg3) = 5.571468e-08
i(vcn) = -7.99652e-09
```

- Input 000101010:
  - Estimated: `array([1.36634473e-06, 1.07492542e-06])`
  - Simulation:

```
i(vdd) = -1.54645e-06
i(vp0) = 3.999144e-08
i(vp1) = 3.992901e-08
i(vp2) = 3.841803e-08
i(vp3) = -1.79280e-08
i(vg0) = 4.490420e-08
i(vg1) = -1.98924e-08
i(vg2) = 7.120258e-08
i(vg3) = -2.29083e-08
i(vcn) = 1.548824e-08
```

- Input 111100110:
  - Estimated: `array([1.13765985e-06, 9.16456797e-07])`
  - Simulation:

```
i(vdd) = -1.32021e-06
i(vp0) = -5.03672e-09
i(vp1) = -1.00207e-08
i(vp2) = -1.30122e-08
i(vp3) = -1.79962e-08
i(vg0) = 4.575955e-08
i(vg1) = 5.655122e-08
i(vg2) = -3.08850e-08
i(vg3) = -2.89402e-08
i(vcn) = 1.548824e-08
```

- We observe that in all the input cases, the estimated supply leakage roughly matches the supply leakage obtained from the simulation.
- The gate leakage estimated is the total gate leakage from all the individual gates.
- The gate leakage however doesn't match. This is because in the estimation we add the gate leakages of all the individual gates whereas in the simulation the gate leakages only include leakages from the first layer of gates.

Conclusion

- Overall, we observe that the estimated leakage currents match the leakage currents obtained from the simulations with reasonably small errors.