Simulating Circuits with Parasitics and RCL Reduction

Introduction

As design geometries get smaller and circuit speeds increase, parasitic capacitance and resistance become more and more important considerations. Parasitics can dramatically alter the way a circuit behaves, changing a circuit from one that runs well to one that does not function at all. A simple 4 bit shift register can easily demonstrate how parasitics may effect a circuit's output.

Background

In the past simulating circuits with parasitics was very time consuming as the addition of parasitic devices can easily increase the total number of devices by an order of magnitude. With the simple example input deck below the number of devices was initially only 104, but adding parasitic resistors and capacitors brought the total number of devices up to 947. The addition of this many devices can significantly increase the amount of time required to run a simulation even with generic resistors and capacitors. However, with SmartSpice's built in RCL reduction tool, the time required can be markedly reduced while still maintaining the simulation accuracy.

Simulation

The included input deck was run using the models included with Silvaco's Logic Demo PDK. Looking at the simulation results for the voltage at the output of each DFF the signal can be seen propagating through the DFF's to the final output port (out3). See Figure 1.

See the application note "Parasitic Back Annotation for Post Layout Simulation" for instructions on extraction and back annotation of physical designs. After the layout was completed the parasitics were extracted and back annotated. The circuit was then rerun using the same timing with the results displayed in Figure 2.

Obviously, the addition of the resistance and capacitance from the layout wiring and interconnects caused enough delay to make the circuit no longer functional. Either the layout would need to be changed to match the required specifications or the timing would need to be adapted to match the physical design. Doubling the period of the input signal allows the signal enough time to propagate through the DFF's once again as in Figure 3.

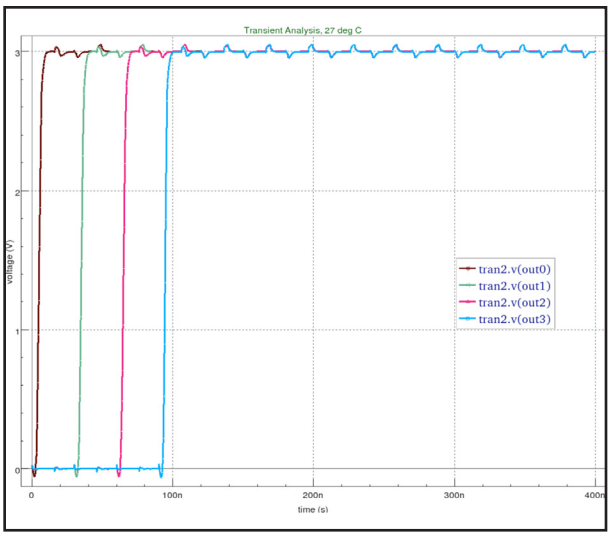

Figure 1: Simulation without Parasitics.

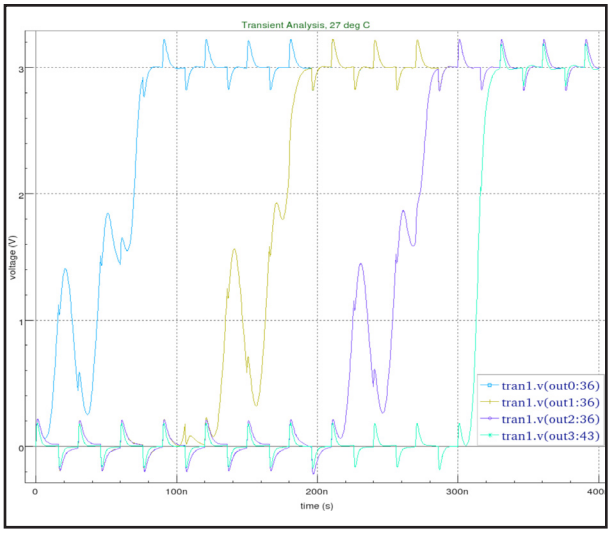

Figure 2: Simulation with Parasitics.

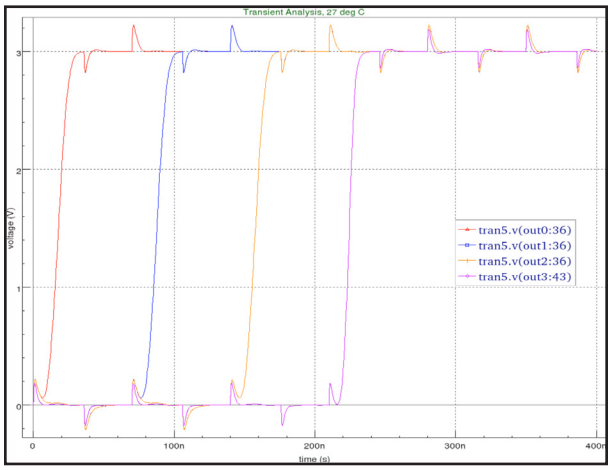

Figure 3: Simulation with Parasitics and Slower Input Signal.

Signal

The difference in time to simulate the pre-layout and post-layout circuits was approximately 2.5 times as long for the post-layout. However, by using Smartspice's built in RCL reduction tool, the time to simulate was greatly improved. The post-layout circuit was re-run using the command line switch "-rclevel 1" which activates the options RCL reduction on subcircuits and post filtering. Running the simulation with these options reduced the transient analysis time by 30% and changed the simulation results by an insignificant 0.007 percent. When the two sets of post-layout simulations were plotted to the same chart it was nearly impossible to see the difference. Further reductions in simulation time can be achieved on larger circuits or longer simulations.

Conclusion

Simulating the post-layout circuit is a very important step in the design process. As seen in the simulation results the parasitic capacitance and resistance can have a dramatic effect. While in the past post-layout simulations could be very time consuming, with the Smartspice RCL reduction option simulation time can be much improved while high accuracy is maintained.

There are many different options for "-rclevel". For more details on running simulations with the "-rclevel" switch see Chapter 13 of the Smartspice User's Manual 1.

Simulated schematic input deck

```
* Schematic name: shift_register
*
R5 OUT3 GND 10Meg
R6 NET4 GND 10Meg
R7 NET5 GND 10Meg
R8 NET3 GND 10Meg
R9 NET1 GND 10Meg
V1 VDD GND DC 3
V2 C GND PULSE(0 3 0 0.01ns 0.01ns 15.98ns
    30ns)
V3 NET7 GND DC 3
X1 C NET7 OUT0 NET1 VDD GND DFF
X2 C OUT0 OUT1 NET3 VDD GND DFF
X3 C OUT1 OUT2 NET5 VDD GND DFF
X4 C OUT2 OUT3 NET4 VDD GND DFF
*
* Schematic name: DFF
*
.SUBCKT DFF C D Q Q_bar VDD VSS
*
X1 NET13 Q_bar Q VDD VSS NAND2
X2 Q NET10 Q_bar VDD VSS NAND2
X3 NET10 D NET8 VDD VSS NAND2
X4 NET12 C NET13 VDD VSS NAND2
X5 NET8 NET13 NET12 VDD VSS NAND2
X10 NET13 C NET8 NET10 VDD VSS NAND3
*
.ENDS DFF
*
* Schematic name: NAND2
*
.SUBCKT NAND2 IN1 IN2 OUT VDD VSS
*
M1 OUT IN1 NET2 VSS CMOSN L=2U W=5U
    AD=27.5P AS=27.5P PD=21U PS=21U M=1
M2 NET2 IN2 VSS VSS CMOSN L=2U W=5U
    AD=27.5P AS=27.5P PD=21U PS=21U M=1
M3 OUT IN1 VDD VDD CMOSP L=2U W=10U AD=55P
    PD=31U AS=55P PS=31U M=1
```

```
M4 OUT IN2 VDD VDD CMOSP L=2U W=10U AD=55P
    PD=31U AS=55P PS=31U M=1
*
.ENDS NAND2
*
* Schematic name: NAND3
*
.SUBCKT NAND3 IN1 IN2 IN3 OUT VDD VSS
*
M1 OUT IN1 NET2 VSS CMOSN L=2U W=5U
    AD=27.5P AS=27.5P PD=21U PS=21U M=1
M2 NET2 IN2 NET1 VSS CMOSN L=2U W=5U
    AD=27.5P AS=27.5P PD=21U PS=21U M=1
M3 OUT IN1 VDD VDD CMOSP L=2U W=10U AD=55P
    PD=31U AS=55P PS=31U M=1
M4 OUT IN2 VDD VDD CMOSP L=2U W=10U AD=55P
    PD=31U AS=55P PS=31U M=1
M5 NET1 IN3 VSS VSS CMOSN L=2U W=5U
    AD=27.5P AS=27.5P PD=21U PS=21U M=1
M6 OUT IN3 VDD VDD CMOSP L=2U W=10U AD=55P
    PD=31U AS=55P PS=31U M=1
*
.ENDS NAND3
*
* End of the netlist
*
* Markers to save
*
.INC '../models/SBCD.lib'
.IC V(OUT0)=0
.IC V(OUT1)=0
.IC V(OUT2)=0
.IC V(OUT3)=0
.TRAN 0.1ns 400n
.SAVE ALL(I)  ALL(V)

.END
```