

기생 및 RCL 감축을 통한 회로 시뮬레이션

소개

설계의 기하학적 구조가 축소되고 회로의 속도가 증가함에 따라, 기생 캐패시턴스와 저항의 중요성이 더욱 커졌습니다. 기생 성분은 회로의 동작 방식을 급격하게 바꾸어, 잘 동작하는 회로를 무용지물로 만들 수도 있습니다. 간단한 4 비트 시프트 레지스터를 사용하여 기생이 회로의 출력에 어떤 영향을 미치는지 살펴봅시다.

배경

기생 소자를 추가하면 전체 소자의 수가 간단하게 수십배 늘어났기 때문에, 기생 회로 시뮬레이션은 시간이 오래 걸렸습니다. 아래의 간단한 예제 입력 데크는 처음에는 소자가 104개에 불과했지만, 기생 저항과 캐패시터를 추가하여 그 수가 947개로 늘어났습니다. 이렇게 많은 소자를 추가하게 되면, 일반적인 저항과 캐패시터를 활용한 시뮬레이션도 시간이 크게 늘어날 수 있습니다. 그러나 SmartSpice에 내장된 RCL 감축 툴을 사용하여, 시뮬레이션의 정확도를 유지하면서 소요 시간을 크게 줄일 수 있습니다.

시뮬레이션

포함된 입력 데크는 실바코의 로직 데모 PDK의 모델을 사용하였습니다. 각 DFF 출력에서 전압에 대한 시뮬레이션 결과를 보면, 신호가 DFF를 통해 최종 출력 포트 (Out3)로 전파되는 것을 알 수 있습니다. 그림 1을 참조하십시오.

물리적 설계의 추출 및 백 애노테이션은 애플리케이션 노트 "포스트 레이아웃 시뮬레이션을 위한 기생 백 애노테이션"을 참조하십시오. 레이아웃이 완료된 후, 기생을 추출하여 백 애노테이션을 실행합니다. 그 다음, 그림 2에 표시된 결과와 동일한 타이밍을 활용하여 회로를 다시 실행합니다.

레이아웃 배선 및 인터넥트로부터 저항과 캐패시턴스가 추가되어 회로가 더 이상 기능할 수 없을 만큼 지연이 발생하였습니다. 사양에 맞게 레이아웃을 변경하거나 물리적 설계에 맞게 타이밍을 조정해야 합니다. 입력 신호의 주기를 두 배로 늘리면 그림 3처럼 DFF를 통해 신호가 다시 한번 전파될 수 있을 만큼 충분한 시간을 확보할 수 있습니다.

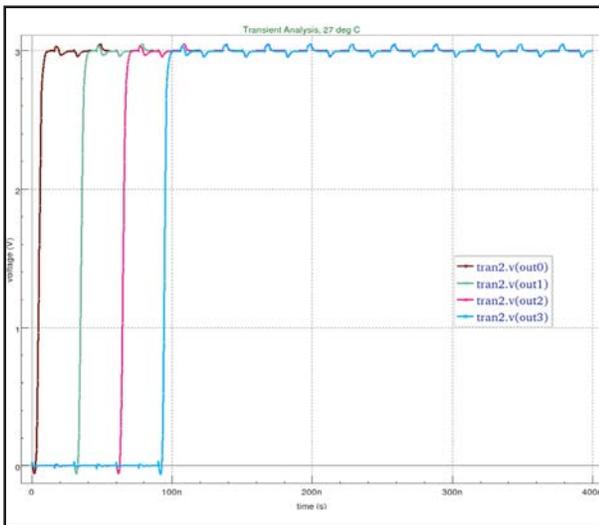


그림 1. 기생이 없는 시뮬레이션

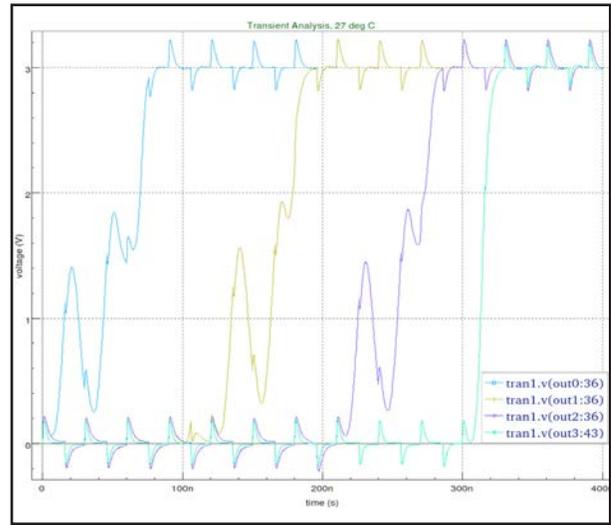


그림 2. 기생이 있는 시뮬레이션

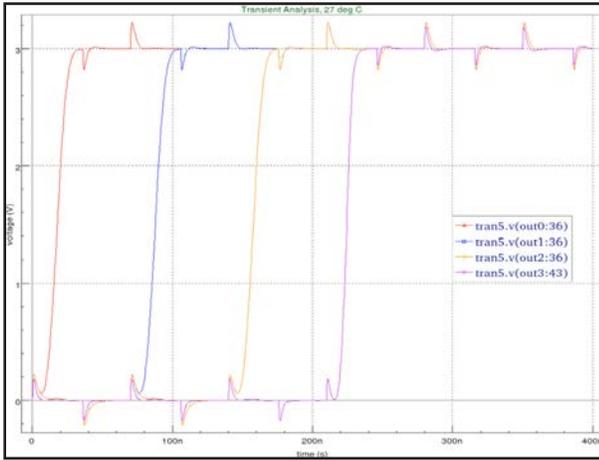


그림 3: 기생 및 저속 입력 신호를 사용한 시뮬레이션

신호

레이아웃 전과 후의 회로 시뮬레이션 시간은 레이아웃 후 시뮬레이션의 약 2.5배 차이가 있었습니다. 그러나 SmartSpice에 내장된 RCL 감소 툴을 사용하여 시뮬레이션 시간을 크게 단축하였습니다. 포스트 레이아웃 회로는 서브 회로 및 포스트 필터링에 대한 RCL 감소 옵션을 활성화하기 위해, 명령어에 스위치 "-rclevel 1"을 사용하여 다시 실행하였습니다. 이 옵션을 사용하여 시뮬레이션을 실행하면 과도 분석 시간이 30% 단축되지만, 시뮬레이션 결과는 0.007%만 차이가 있습니다. 두 포스트 레이아웃 시뮬레이션을 동일한 차트에 표시했을 때, 차이점은 거의 없었습니다. 회로 규모가 크거나 시뮬레이션이 길어질 경우, 시뮬레이션 시간을 더욱 단축할 수 있습니다.

결론

포스트 레이아웃 회로의 시뮬레이션은 설계 과정에서 매우 중요합니다. 시뮬레이션 결과에서 볼 수 있는 것처럼, 기생 캐패시턴스와 저항은 상당한 영향을 미칠 수 있습니다. 과거에는 포스트 레이아웃 시뮬레이션에 매우 많은 시간이 소요되었지만, SmartSpice RCL 감소 옵션을 활용하여 시뮬레이션 시간을 획기적으로 개선하면서도 높은 정확도를 유지할 수 있습니다.

"-rclevel"에는 다양한 옵션이 있습니다. "-rclevel" 스위치에 의한 시뮬레이션 방법은 SmartSpice 사용 설명서 1의 13장을 참조하십시오.

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 CMOSN L=2U W=10U AD=55P
  PD=31U AS=55P PS=31U M=1
M4 OUT IN2 VDD VDD CMOSN 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 CMOSN 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
```