

# SmartSpice: 인터커넥트 RC 네트워크 축소 기능 내장

SmartSpice 4.0.1.R 버전부터 인터커넥트 RC 네트워크 축소 기능이 있습니다. 이를 위해 `int_rc_method` 옵션을 1 또는 2로 설정해야 합니다. 이 값은 구현 알고리즘을 선택합니다. `int_rc_method` 값에 따라, 접지된 캐패시터 주변의 RC 네트워크를 분석하여 회로 토플로지를 변경합니다.

축소 기능을 제어하기 위해 두 가지 추가 옵션을 지정할 수 있습니다.

`int_rc_cmin` 옵션은 캐패시턴스 기준값을 지정합니다. 이보다 작은 접지 캐패시터는 모두 제거됩니다. `int_rc_cmin` 기본값은  $1e-22$ 입니다.

`int_rc_rmin` 옵션은 저항의 기준값을 지정합니다. 이보다 작은 저항은 단순화된 등가 네트워크로 축소됩니다. `int_rc_rmin` 기본값은  $1e-3$ 입니다.

축소 기능이 활성화되면, SmartSpice는 축소된 등가 인터커넥트 RC 네트워크로 입력 네트리스트를 처리하며 다음 정보를 출력합니다:

```
Internal RC-reduction starting...
-----
Internal RC-reduction: detecting nodes with
grounded capacitors and number of connections <
4 ...
Internal RC-reduction: grounded capacitors with
capacitance less than 1e-15 are : 5841 Internal
RC-reduction: analyzing RC networks... Internal
RC-reduction: RC network chains: 5758 Internal
RC-reduction: maximum length of RC networks: 3
Internal RC-reduction: TOTAL STATISTICS :
-----
Internal RC-reduction: capacitors before :
62706 Internal RC-reduction: capacitors
removed : 5697 Internal RC-reduction: capacitors
reduction ratio (%): 9.08525 %
Internal RC-reduction: resistors before :
100046 Internal RC-reduction: resistors
removed : 815 Internal RC-reduction: resistors
reduction ratio (%): 0.814625
Internal RC-reduction: nodes before : 83101
Internal RC-reduction: nodes removed : 815
-----
```

## 두 알고리즘에 대한 세부 설명

### 두 알고리즘의 공통 부분

두 알고리즘의 공통 부분은 사전에 정의된 특정 조건에 맞는 RC 네트워크를 탐지하는 것입니다. 먼저 SmartSpice는 접지 캐패시터와 연결 수가 4개 미만인 노드를 감지합니다. 둘째, SmartSpice는 캐패시턴스가 `int_rc_cmin` 보다 큰 캐패시터만 보존합니다. 마지막으로 SmartSpice는 그러한 캐패시터에 연결된 모든 RC 네트워크를 감지하여 연결합니다. 출력에서, 이러한 체인의 총 갯수는 "Internal RC-reduction: RC network chains: 5758"처럼 표시됩니다. 체인의 최대 길이도 알 수 있습니다. SmartSpice는 체인의 모든 RC 네트워크를 확인하고, 축소 기술을 통해 체인에 있는 모든 RC 소자의 토플로지를 변환합니다.

### 축소 방법 1

첫 번째 축소 방법은 `int_rc_method=1` 옵션을 사용하여 정의합니다. 그리하면, 체인의 모든 캐패시터가 제거됩니다. 체인의 모든 내부 노드를 분석하여, 가능한 모든 내부 저항을 분리합니다. 처리할 RC 체인의 외부 노드에 연결된 최종 저항은 그대로 유지되어, 저항은  $R_{total} = \text{Sum}(R_i)$ 로 조정됩니다. 여기서  $R_i$ 는 `int_rc_rmin`보다 큰 내부 직렬 저항 모두의 저항값입니다. 그 후, 연결 갯수가 0인 불필요한 내부 노드가 모두 제거됩니다. 최종 저항이 해당 체인의 외부 노드에 연결됩니다. 이 과정에 대한 설명은 그림 1, 그림 2를 참조하십시오.

### 축소 방법 2

두 번째 축소 방법은 `int_rc_method=2` 옵션을 사용하여 정의합니다. 먼저 저항을 분석한 후 모든 내부 직렬 저항이 제거되어, 해당 RC 체인의 외부 노드에 연결된 노드로 최종 저항이 새로 생성됩니다. 둘째, 최종 저항의 저항  $R_{total}$ 은 내부 직렬 저항 모두를 기준으로 계산됩니다.

축소 후, SmartSpice는 제거한 캐패시터, 저항 및 노드의 갯수와 함께 최종 축소 통계를 나타냅니다. 기본 축소를 더 할 수 없는 경우, SmartSpice는 토플로지를 변경하지 않습니다.

## 예제

### Example 1

```
.option int_rc_method=1 int_rc_cmin=le-18 int_
rc_rmin=0.01
...
R1 Extnode1 Node1 0.001
C1 Node1 0 1e-19
R2 Node1 Node2 10K
C2 Node2 0 1e-19
R3 Node2 Extnode2 10K
C3 Extnode2 0 1e-19
...
```

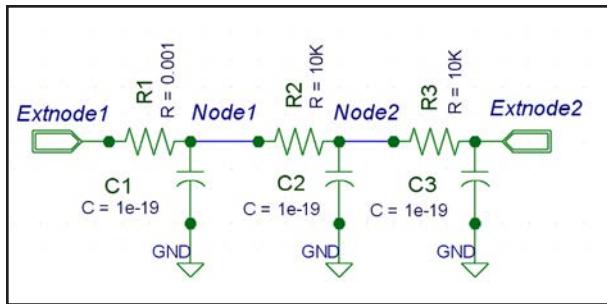


그림 1. 예 1 - 기준 블록

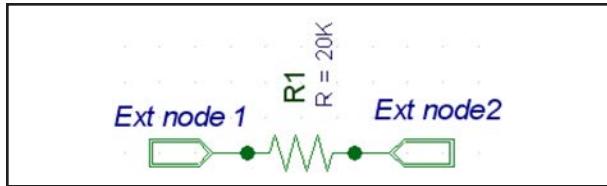


그림 2. 예 1 - 축소 블록

### SmartSpice output:

```
Internal RC-reduction starting...
-----
Internal RC-reduction:detecting nodes with
grounded capacitors and t_numconn < 4 ...
Internal RC-reduction:grounded capacitors with
capacitance less than 1e-018 are: 3
Internal RC-reduction: analyzing RC networks...
Internal RC-reduction: RC network chains: 1
Internal RC-reduction: maximum length of RC
networks: 3
Internal RC-reduction: TOTAL STATISTICS :
-----
Internal RC-reduction: capacitors before : 3
Internal RC-reduction: capacitors removed : 3
Internal RC-reduction: capacitors reduction ra-
tio (%): 100
Internal RC-reduction: resistors before : 3
Internal RC-reduction: resistors removed : 2
Internal RC-reduction: resistors reduction ra-
tio (%): 67
Internal RC-reduction: nodes before : 5
Internal RC-reduction: nodes removed : 3
-----
```

### Example 2

```
.option int_rc_method=2 int_rc_cmin=le-18
int_rc_rmin=0.5
...
R1 Extnode1 Node1 1
C1 Node1 0 1e-19
R2 Node1 Node2 10K
C2 Node2 0 1e-19
R3 Node2 Extnode2 10K
C3 Extnode2 0 1e-19
...
```

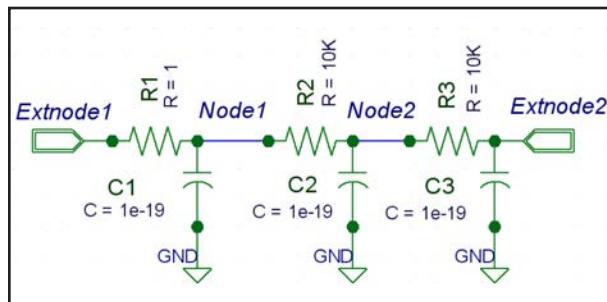


그림 3. 예 2 (방법 2) - 기준 블록

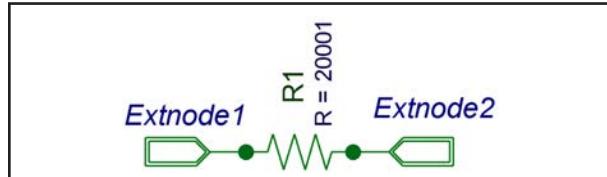


그림 4. 예 2 (방법 2) - 기준 블록

## 결론

사용자는 모델의 부하 및 LU 분해에서 성능의 향상을 확인할 수 있습니다. ".option acct=2" 구문은 최종 회로 방정식의 수를 출력합니다. RC 기준값을 사용하면 축소 비율을 제어할 수 있습니다. 기준값이 클수록 더 압축된 인터커넥트 등가 회로가 생성됩니다. 기준값을 너무 크게 지정하면, 실패할 수 있습니다. SmartSpice는 RC 기준값을 제시하지 않습니다. 사용자는 RC 축소 기능을 적용하고 결과를 확인하여, 시뮬레이션 오류를 허용할 수 있는지 판단해야 합니다.