

orders of magnitude smaller than the largest signal present in the circuit of the same kind.

9. If, by a process of elimination, you identify a nonlinear capacitance that seems to have a discontinuity, try to simplify or even just modify the nonlinear capacitor model. This can sometimes eliminate the discontinuity.
10. If all else fails, loosen `trtol` (`lteratio` in Spectre) or `reltol` and widen transitions in the stimulus waveforms. This sometimes causes the simulator to just jump past the convergence difficulties (good luck).

## 4.4 Applications of Transient Analysis

In this, the last major section of this chapter the focus shifts from transient analysis algorithms to circuits. Particular types of circuits are used to illustrate nonobvious issues involved when applying transient analysis to real circuits.

### 4.4.1 Computing the DC Operating Point

If it is not possible to use DC analysis to compute a DC operating point because of convergence problems, then on most simulators transient analysis can often be used instead. Even on those simulators that employ a pseudo-transient analysis in their DC analysis it is still worthwhile to try using transient analysis. In the pseudo-transient analysis the simulators often do not use the capacitors and inductors specified for the circuit. Instead they generally install 1 F capacitors from every node to ground, or across every nonlinear component. The idea is that the solution process is faster if all capacitors are linear and the same size. However, the 1 F capacitors may cause a circuit to oscillate that would not normally oscillate. In this case, transient analysis is able to find the equilibrium point, where the pseudo-transient is not. Even if the circuit oscillates during transient analysis, because the real circuit is being analyzed and the waveforms are available, it is often possible to study the circuit, determine what is causing the oscillation, and disable it.

To compute a DC operating point with a transient analysis, start by modifying all the independent sources to start at zero and ramp to their DC values. Then, run the transient analysis well beyond the time when all the sources have reached their final value. Using a huge stop time is not be expensive as long as the circuit does not oscillate because the simulator takes large time steps once the solution reaches the equilibrium point. With SPICE, set `uic` on the transient analysis statement to avoid having the transient analysis perform a DC analysis in order to compute its initial conditions. With Spectre, use the transient analysis option `skipdc` for the same purpose (see below for more information on this option).

Use the solution at the final time point to generate a set of node-sets. To make the transient analysis more efficient, set the integration method to backward Euler if possible and eliminate the local-truncation error criterion by setting the option `lvltim=1` (this is the *only* time you should use this option). Spectre does not allow you to disable the local-truncation error criterion, but you can de-emphasize it by increasing `lteratio`, say to 50 or even 500. To tell SPICE2 to use backward Euler, set the undocumented option `mu` to zero. With Spectre, set the transient analysis options `method=euler`.

Occasionally, this approach fails or is very slow because the circuit contains an oscillator. Try to disable the oscillator before using this approach.

Spectre provides several capabilities that make using transient analysis to compute a DC operating point considerably easier. First, the transient analysis provides the `skipdc` option. If `skipdc=yes`, then the transient analysis avoids using the DC analysis to compute unspecified initial conditions. Instead transient analysis simply starts from the specified initial conditions. Any unspecified initial conditions are taken to be zero. With this setting, the independent sources act the same as they would in a normal transient analysis. However, `skipdc` can also change the behavior of the independent sources. For example, with `skipdc=waveless`, the initial DC analysis is skipped and the independent source values are fixed to their initial values (not their DC values). With `skipdc=rampup`, the independent source values start at 0 and ramp up to their initial values

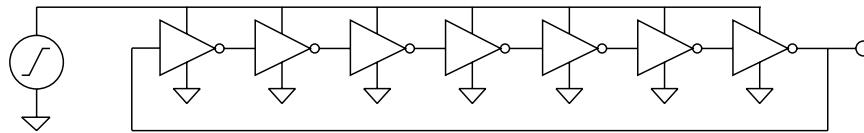
in the first 10% of the analysis interval. After that their values remain constant. The ramp-up option is best if no initial condition is given, and waveless is best when a non-zero initial condition is specified. If `skipdc=autodc` then Spectre automatically selects either `waveless` or `rampup` depending on whether a non-zero initial condition is specified.

When Spectre finishes the transient analysis, it can be directed to automatically write the state of the circuit at the final time point to a file so that it can be recalled to act as a nodeset for a later DC analysis. To do so, use the `writefinal` transient analysis option. In addition, the final point of the transient analysis can be directly used as an operating point for a subsequent analysis. To use it as an operating point for a small signal analysis, the small signal analysis must follow the transient analysis and the `prevoppoint` option on the small signal analysis must be set to yes (see Section 3.4.4 on page 113). It is also possible to print operating point information about the final transient analysis point by simply following the transient analysis by an `info` statement.

### 4.4.2 Oscillators

There are two issues that must be considered when simulating oscillators. First, it is necessary to manually start the oscillator. Second, oscillators are more sensitive to simulator error than most nonautonomous circuits, and so it is necessary to be more careful.

**Starting the Oscillation** The DC analysis finds an equilibrium point for the circuit. By definition, the circuit does not drift away from an equilibrium point unless it is perturbed. If the equilibrium point is stable and the perturbation is small, the circuit drifts back to the equilibrium point afterwards. Oscillators have unstable equilibrium points. Unlike the latch of Figure 2.1 on page 17, when an oscillator is perturbed from its equilibrium point it breaks into an oscillation rather than drifting to a stable equilibrium point. However, a perturbation must be supplied to cause the oscillator to drift away from its equilibrium point and oscillate. In a physical circuit, thermal noise or the turn-on transients are sufficient to start the



**Figure 4.44:** A 7-stage ring oscillator used to explore characteristics of various approaches to simulating oscillators.

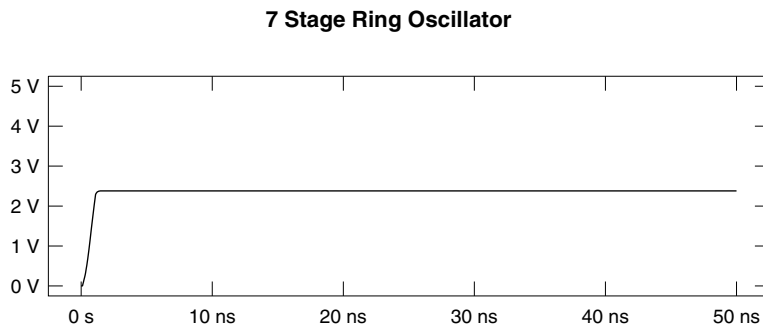
---

oscillator. In a circuit simulator, neither stimulus exists and so an explicit perturbation is needed and must be supplied by the user.

One starts an oscillator by either inserting a time-varying independent source, such as one that generates a short pulse, or by specifying an initial condition on a capacitor or inductor. In addition there are usually several places where sources can be installed in the circuit, and several components for which initial conditions can be specified. In general, it is best to choose the method and location that provides the best coupling into the mode of oscillation, and the least coupling to other undesirable response modes. In particular, try to avoid exciting the very slow time constants associated with any bias circuitry.

When using sources to start an oscillator, there is a tendency to use a convenient source such as a power supply. Often these sources are poorly coupled to the mode of oscillation. For example, consider ramping the supply voltage in an attempt to start a 7-state ring oscillator, such as the one shown in Figure 4.44. If all 7 stages are identical, then the pulse would couple equally and identically to each stage. This balanced perturbation is orthogonal to the mode of oscillation in which the outputs of the stages alternate between high and low. A balanced stimulus on a balanced circuit is unlikely to generate an unbalanced response. Figure 4.45 on the facing page shows the response of the 7-stage ring oscillator when its power supply is quickly ramped to its final value.

A better approach for this circuit would be to use an initial condition



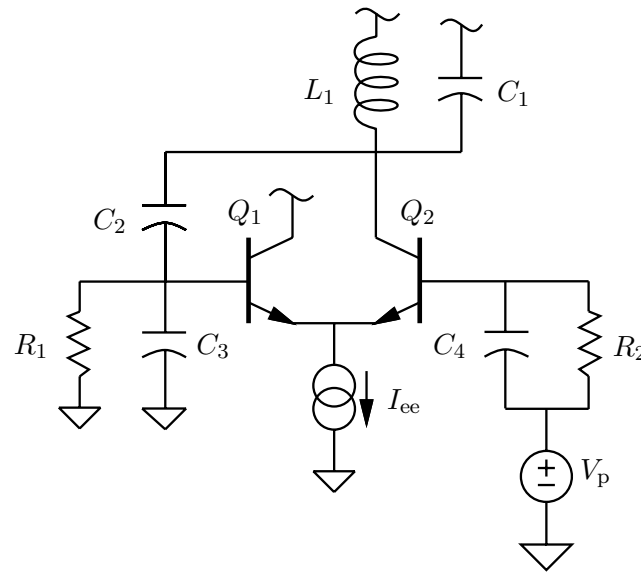
**Figure 4.45:** Response of the 7-stage ring oscillator when its power supply is quickly ramped to its final value. Oscillator does not start because the perturbation affects all stages identically and so does not couple into the mode of oscillation.

---

to set on of the nodes in the ring to either ground or to the supply voltage. The actual value used for the initial condition is not very important as long as it is in the normal operating range and it is sufficiently far from the equilibrium point. In this case, an initial condition couples strongly to the mode of the oscillation and does not excite any significant undesirable modes. In addition, using initial conditions has the advantage of not adding any new components to the circuit.

A circuit where initial conditions are not as effective is shown in Figure 4.46 on the next page and Netlist 4.2. Using `uic`-based initial conditions requires that you specify initial conditions for every capacitor and inductor in the circuit. For this circuit alone that would be easy. However, if this circuit were just a piece of a much larger circuit being simulated, specifying initial conditions for every capacitor and inductor in the circuit could easily be impractical.

When `uic` is not specified, SPICE uses voltage sources and  $1\ \Omega$  resistors to force initial conditions. Consider trying to apply an initial condition of 1 V to  $C_1$ . The simulator connects the series combi-



**Figure 4.46:** An emitter-coupled pair oscillator.

nation of a 1 V source and a 1  $\Omega$  resistor in parallel with  $C_1$ . The simulator then computes the DC solution to this modified circuit and uses it as the initial condition for the original circuit. Unfortunately the inductor shorts the source and resistor used to force the initial condition and so  $V_{C_1} = 0$ . In addition, the current from the source and resistor used to force the initial condition travels through  $L_1$ , and so it starts with an initial condition of 1 A. The huge initial current for  $L_1$  cause the oscillation to start with a very large amplitude. This is shown in Figure 4.47 on page 214. On some circuits, it takes a long time for the amplitude to decay to the point where the oscillator functions normally and the large amplitude can cause a severe loosening of tolerances (see Section 4.3.2.1 on page 183). In general, having the simulator compute non-uic-based initial conditions on an  $LC$  resonator (series or parallel) is likely to result in huge oscillations.

---

```
// BJT ECP Oscillator

global gnd
simulator lang=spectre

Vcc (vcc gnd)  vsource dc=5 type=dc \
               val0=0 val1=5 rise=10ns delay=10ns
Iee (e gnd)    isource dc=1mA
Q1 (vcc b1 e vcc) npn
Q2 (out b2 e out) npn
L1 (vcc out)   inductor l=1uH
C1 (vcc out)   capacitor c=1pf ic=1
C2 (out b1 )   capacitor c=272.7pF
C3 (b1 gnd)    capacitor c=3nF
R1 (b1 gnd)    resistor r=10k
C4 (b2 in )    capacitor c=3nF
R2 (b2 in )    resistor r=10k
Vp (in gnd)    vsource mag=1 type=dc \
               ampl=0.1 freq=10MHz damp=5Mnepers

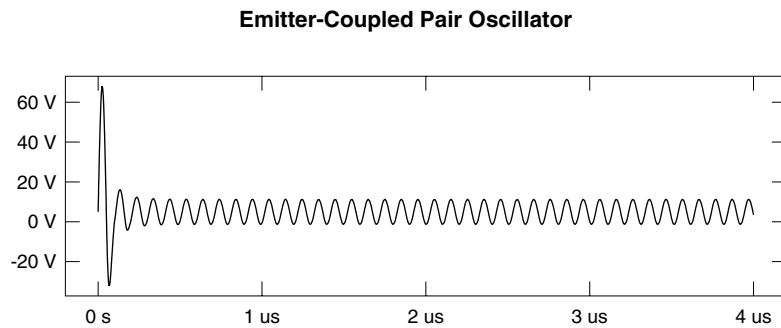
ic b1=1

model npn bjt type=npn bf=80 rb=100 vaf=50 \
        cjs=2pf tf=0.3ns tr=6ns cje=3pf cjc=2pf

osc1 tran stop=4us ic=dev maxstep=10ns
osc2 tran stop=80us ic=node maxstep=10ns
EnablePerturbation alter dev=Vp param=type value=sine
osc3 tran stop=10us ic=dc maxstep=10ns
```

**Netlist 4.2:** Emitter-coupled pair oscillator and the three analyses that generate the results for the following three figures.

---



**Figure 4.47:** Assigning a non-uic-based initial condition to a capacitor in the  $LC$  resonator of Figure 4.46 on page 212 results in a huge response because of the large current forced through the inductor.

Assigning initial conditions to  $C_2$ ,  $C_3$ , or  $C_4$  trades the problems just discussed with another serious problem. Initial conditions on  $C_2$ ,  $C_3$ , or  $C_4$  disturb the bias point of the oscillator, and the slow time constants at the base results in the circuit taking a very long time to recover. For example, setting an initial condition of 1 V on  $C_3$  results in all bias current initially going through  $Q_1$  with  $Q_2$  turned off completely. The time constant at the base of  $Q_1$  is roughly  $30 \mu\text{s}$ . It takes several time constants for the circuit to become balanced enough to support oscillation. This is shown in Figure 4.48.

A better approach to starting this oscillator is to use an independent source ( $V_p$ ) to couple a signal into the base of  $Q_2$ . Exciting the circuit with a small damped sinusoid at the estimated frequency of oscillation is particularly appropriate and effective in this case because

1. It couples strongly to the mode of oscillation, increasing the likelihood of starting the oscillation.
2. The stimulus has a small DC level relative to its amplitude and so does not excite the slow time constants in the bias circuitry