gEDA/gaf Switcap Symbols and Netlister

Dan McMahill

April 13th, 2003

Contents

1 Overview

This document describes the symbol library and gnetlist backend which supports driving SWITCAP simulations from the gEDA/gaf system. SWITCAP is a switched capacitor circuit simulator available from Columbia University. It is used in many classroom and research environments. One drawback to SWITCAP is the lack of a freely available schematic capture frontend. The gEDA/gaf SWITCAP symbol library and gnetlist backend tries to fill that gap.

The basic steps involved with using gEDA as the frontend for SWITCAP simulations are:

1. Create schematics of the circuit.
2. Create an analysis file.
3. Extract the netlist.
4. Run the SWITCAP simulation.
5. Run `sw2asc` to extract the results.
6. View the results with `gwave`.

2 Requirements

You will need the following programs to be installed.

1. A recent version of gEDA/gaf. To see if your version is recent enough, see if the directory `$prefix/share/gEDA/sym/switcap` exists. `$prefix` is the installation prefix for gEDA on your system.
2. SWITCAP. The executable is usually called `sw`. If you do not have SWITCAP available on your system, you will need to contact Columbia University to obtain a copy. The gEDA/gaf SWITCAP support was tested with SWITCAP Version A.5R Release 21-Sep-87.
3. Although it is optional, you may wish to install a tool which can be used for plotting the output data. SWITCAP produces both ASCII data listings as well as ugly ASCII plots (note the release date of the version of SWITCAP used). Suitable tools are:
   - Gwave. Gwave is an analog waveform viewer. It is fairly basic, but easy to use, includes cursors, and has zoom/pan features. See `http://www.geda.seul.org`
   - Scilab. Similar to matlab. Powerful, but no cursors or panning. See `http://www-rocq.inria.fr/scilab`
   - Octave. Similar to matlab. See `http://www.octave.org`
   - Grace. See `http://plasma-gate.weizmann.ac.il/Grace/`

3 Creating Schematics

3.1 Required Symbols

This section assumes you are familiar with using gschem to create and edit schematics. SWITCAP netlisting is only supported for the components contained in the SWITCAP symbol library as well as the ground symbol found in the 'power' library which comes with gEDA. All allowed SWITCAP elements except for subcircuits are supported. You *must* include the following elements on your schematic:

1. One instance of the switcap-timing symbol. This symbol will set the master clock period for your simulations.
2. One or more instances of the switcap-clock symbol. This symbol defines a clock with a particular phase and period. The reference designator of the clock symbol is used by the switches to set what phase they switch on.
3. One or more instances of the switcap-analysis symbol. This symbol defines an analysis by specifying a file to include in the SWITCAP netlist. By including multiple instances of this symbol, multiple analysis files may be included.

3.2 Optional Symbols

You can also optionally add the following SWITCAP special symbols to your schematic:

1. Zero or one instance of the switcap-title symbol. This will add a TITLE: line to the SWITCAP netlist and will appear in the output file.
2. Zero or one instance of the switcap-options symbol. By editing the OPTIONS attribute on this symbol you can set the various options which can be passed to SWITCAP.

3.3 Net Names

When creating schematics to drive SWITCAP, you should name all nets that you wish to plot. To avoid possible conflicts with unnamed nets, you should avoid using purely numerical names for nets because all unnamed nets will be assigned (somewhat randomly) numbers without checking for possible conflicts with explicitly named nets. SWITCAP limits the length of node names to 7 characters.

3.4 Switches

When placing switches on your schematic, you will need to define which clock they are controlled with. This is done by setting the clock attribute on the switch to the reference designator of the clock which should control it.

4 Extracting the SWITCAP Netlist

To extract the SWITCAP netlist, run

```
gnetlist -g switcap -o test.scn file1.sch [file2.sch ...]
```

For the example file contained in this archive, you can run:

```
gnetlist -g switcap -o example.scn ckt.sch \
    clocks.sch analysis.sch
```

The netlist will be left in `example.scn`.

5 Running SWITCAP

I typically use something like:

```
printf "example.scn\nexample.out\n" | sw
```

so I can use command history to rerun SWITCAP without having to manually type the file names each time.

Refer to the SWITCAP manual for more details.

A Symbols in the Library

A.1 Capacitors (switcap-capacitor)

Ideal capacitor. Attributes:

- **C**=capacitance. Required. Specifies filename to be included.
- **refdes**=reference designator. Required. Must start with "C" and be unique.

A.2 Switches (switcap-switch)

Ideal switch. Attributes:

- **clock**=Controlling clock. Required. Specifies which clock controls this switch.
- **refdes**=reference designator. Required. Must start with "S" and be unique.

A.3 Independent Voltage Sources (switcap-vsrc)

Attributes:

- **refdes**=reference designator. Required. Must start with "V" and be unique.

A.4 Dependent Voltage Sources (switcap-vcvs)

Attributes:

- **gain**=gain. Required. Specifies the gain of the controlled source.
- **refdes**=reference designator. Required. Must start with "E" and be unique.

A.5 Clock Specification (switcap-clock)

Attributes:

- **PSTART**=starting clock phase. Required. Specifies on what phase of the master clock this clock turns on.
- **PSTOP**=ending clock phase. Required. Specifies on what phase of the master clock this clock turns off.
- **PERIOD**=clock period. Required. Specifies the period of the clock in terms of master clock cycles.
- **refdes**=reference designator. Required. The switches that are controlled by this clock will refer to it by the reference designator. As such, avoid running any reference designator renumbering tools.

A.6 Master Timing Specification (switcap-timing)

Attributes:

- **PERIOD**=clock period. Required. Specifies the period of the master clock in seconds.

Only a single instance of this symbol is allowed.

A.7 Analysis File Include (switcap-analysis)

This symbol will cause a specified file containing SWITCAP analysis commands to be included in the output netlist. Attributes:

- **file**=filename. Required. Specifies filename to be included.

A.8 Simulation Title Specification (switcap-title)

Attributes:

- **TITLE**=switcap title. Required. Specifies the TITLE line for the SWITCAP netlist.

Only a single instance of this symbol is allowed.

A.9 Simulation Options Specification (switcap-options)

Attributes:

- **OPTIONS**=switcap options. Required. Specifies the OPTIONS line for the SWITCAP netlist. See the SWITCAP manual for allowed values.

Only a single instance of this symbol is allowed.

B Example

This appendix provides a simple example of the entire process of generating a schematic, producing a SWITCAP netlist, running a simulation, and plotting the results.

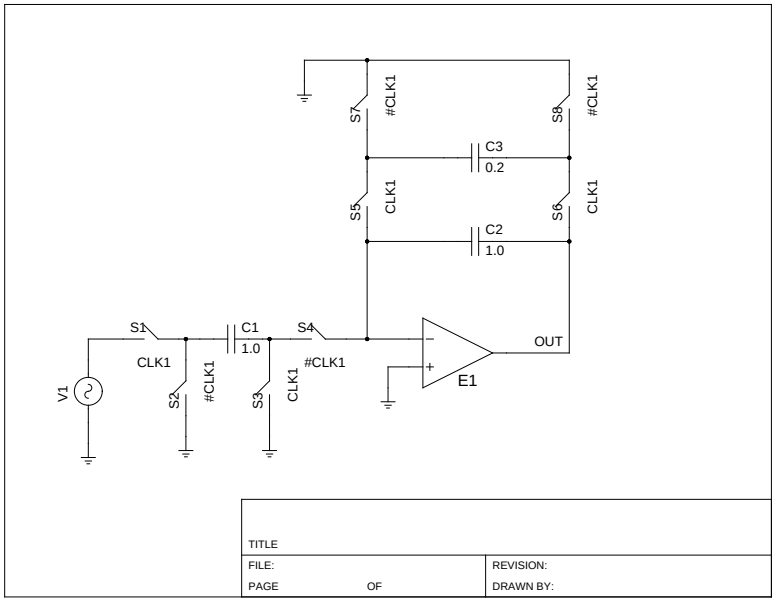

Figure 1: Simple Switched Capacitor Circuit

B.1 Example Schematics

Figure 1 shows the schematic of a simple switched capacitor circuit. Note that some switches, S1 and S3 for example, are controlled by CLK1 while others, S2 and S4 for example, are controlled by the complement of CLK1 (#CLK1).

Figure 2 shows the definition of a clock and the master clock. Here we define a master clock period (mcp) of 1.0 μs in the timing block. In the clock definition symbol, we define a clock called CLK1 that has a period equal to 1 master clock period (mcp). The phase of CLK1 turning on switches is 0 and the phase of CLK1 turning off switches is 3/8 mcp. Additional clock phases can be defined by creating more instances of the clock definition symbol.

Figure 3 shows an instantiation of the title block symbol which will cause "my title" to be used in the TITLE line in the SWITCAP netlist. Figure 3 also shows an instantiation of an analysis block which directs the netlister to include the contents of the file `test.ana` in the output netlist. Figure 4 shows the contents of the `test.ana` file.

B.2 Netlist the Design

To netlist the design, run:

```
gnetlist -g switcap -o example.scn ckt.sch \
```

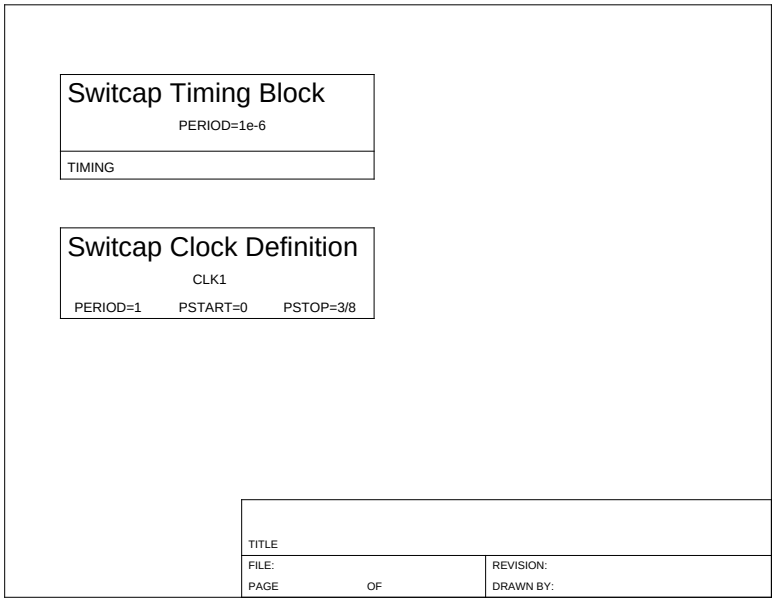

Figure 2: SWITCAP Clock Definition Schematic

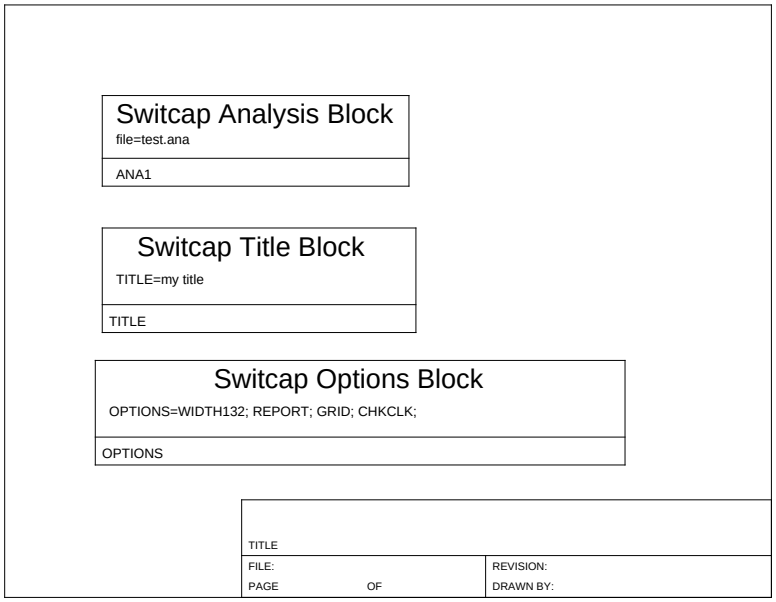

Figure 3: SWITCAP Analysis Definition Schematic

```
ANALYZE SSS;
     INFREQ 100.0 1.0E6 LOG 501;
     SET V1 AC 1.0 0.0;
     PRINT VDB(OUT) VP(OUT);
     END;

ANALYZE TRAN;
     TIME 0+ 500 1
     SET V1 PULSE 0 5 10e-6 5e-6 5e-6 100e-6 500e-6;
     PRINT V(OUT);
     END;
```

Figure 4: SWITCAP Analysis File, `test.ana`

```
clocks.sch analysis.sch
```

B.3 Run the Simulation

Run the simulation with:

```
printf "example.scn\nexample.out\n" | sw
```

B.4 Process the Results

Convert the SWITCAP output file to something gwave can read by running:

```
sw2asc example.out
```

B.5 Plot the Results

Start up the `gwave` program and load the first sinusoidal steady state result by running:

```
gwave example.out.SSS.1.asc
```

Drag the two waveforms onto the two waveform panels and change the x-axis to a log scale. Figure 5 shows the output. Start up the `gwave` program and load the transient result by running:

```
gwave example.out.TRAN.1.asc
```

Drag the output waveform onto the waveform panel. Figure 6 shows the output.

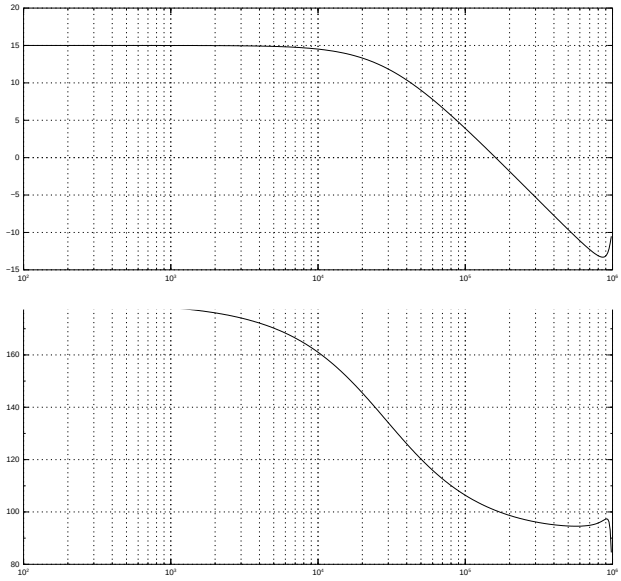

Figure 5: Simulation Results – Sinusoidal Steady State

C Document Revision History

April 13th, 2003	Created switcap.tex

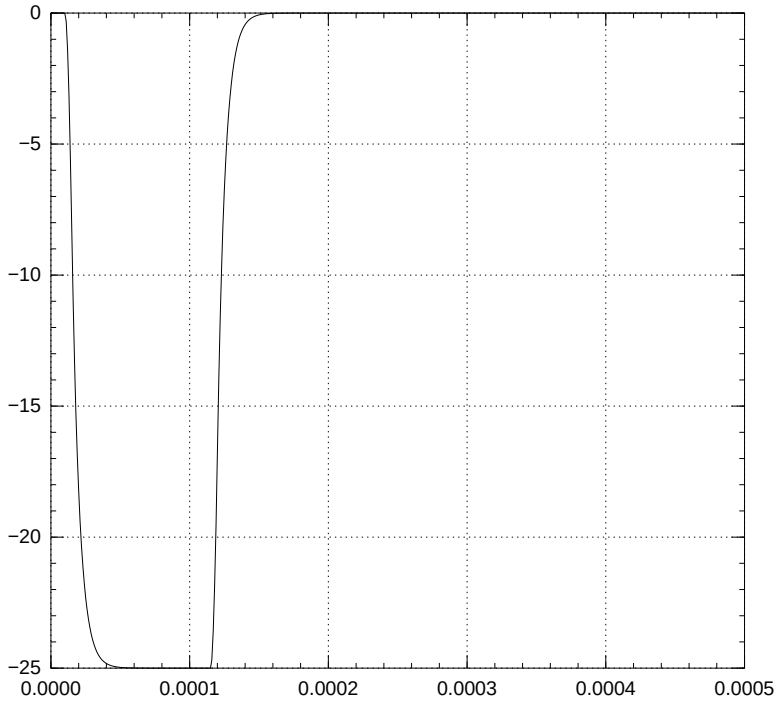

Figure 6: Simulation Results – Transient