Ngspice Netlist Cheat Sheet

Based on Ngspice User's Manual (version 43)

Quick Reference: Ngspice netlists describe circuits using device instances, models, parameters, and control statements in plain text format.

1 General Syntax

1.1 Basic Structure

- **Netlist**: Plain text file with "cards" (lines)
- **Separators**: spaces, commas, =, (), []

1.2 Numbers & Units

- **Formats**: Integer (10), float (2.5), scientific (1e-12)
- **Suffixes**: f, p, n, u, m, k, meg, g, t (case-insensitive)
- **Example**: 1k = 1000, 4.7u = 4.7×10^{-6}

1.3 Node Names

- Alphanumeric; can't start with digit
- Reserved chars: = % ( ) , [ ] ¡ ¿ ∼ ! &
- Ground: `0` or `gnd`

1.4 Expressions

- **Arithmetic**: +, -, *, /, %, ** (power)
- **Logical**: &&, ——, !
- **Relational**: ==, !=, ¡=, ¿=, ¡, ¿
- **Ternary**: condition ? val1 : val2
- **Functions**: sin, cos, exp, log, log10, sqrt, abs, pwr, time, temper, hertz
- **Compile-time**: Use {} (e.g., {param+2})

1.5 Parameters & Vectors

- `.PARAM name=value` (global/local in .SUBCKT)
- **Vectors**: v(node), i(device), db(v(1))
- **Arrays**: param[1]=5

1.6 Continuation & Encoding

- Start line with `+` or end with \
- UTF-8 supported
- **Reserved words**: or, sqrt, sin, cos, exp, ln, log, time, temper, hertz

2 Netlist Order

1. **Title Line** (optional)
2. **Headers**: .TITLE, .INCLUDE, .LIB, .PARAM, .FUNC, .OPTIONS
3. **Models**: .MODEL, .SUBCKT ... .ENDS
4. **Circuit**: Device instances (R, C, L, M, etc.)
5. **Analysis**: .DC, .AC, .TRAN, .OP, .NOISE
6. **Output**: .PRINT, .PLOT, .SAVE
7. **Control**: .CONTROL ... .ENDC
8. **End**: .END (required last line)

Key Notes:

- Models before instances
- .CONTROL executes after parsing
- Batch mode: `ngspice -b file.sp`

3 Comments

- **Line**: * at column 1 (or ; or //)
- **End-of-line**: `$ comment` or `; comment`
- **Example**: `R1 n1 n2 1k $ resistor`
- **Multi-line**: Prefix each with *
- **PSPICE mode**: $ not a comment

4 Dot Commands

Case-insensitive, start with . to control simulation

4.1 File & Structure

- `.TITLE "text"` – Set title
- `.END` – End netlist (required)
- `.INCLUDE "file.sp"` – Insert file
- `.LIB "lib" model` – Include library

4.2 Definitions

- `.MODEL name type(params)` – Device model
- `.SUBCKT name nodes ... .ENDS` – Subcircuit
- `.GLOBAL node1 node2` – Global nodes
- `.PARAM name=val` – Parameters
- `.FUNC f(x) {expr}` – User functions

4.3 Simulation Setup

- `.OPTIONS opt=val` – Tolerances, methods
- `.TEMP 27` – Temperature (°C)
- `.IC v(node)=val` – Initial conditions
- `.OSDI "file.osdi"` – Load Verilog-A

4.4 Analysis

- `.TRAN tstep tstop [tstart [tmax]]` – Transient
- `.AC DEC|OCT|LIN npts fstart fstop` – AC sweep
- `.DC src start stop step` – DC sweep
- `.OP` – Operating point
- `.NOISE v(out) vsrc [dec|lin npts fstart fstop]` – Noise
- `.TF v(out) vsrc` – Transfer function
- `.SENS v(out)` – Sensitivity

4.5 Output & Control

- `.SAVE v(node) i(dev)` – Save variables
- `.PRINT type v(out)` – Print results
- `.PLOT type v(out)` – Plot results
- `.MEAS type result TRIG...` – Measurements
- `.FOUR freq v(out)` – Fourier analysis
- `.CONTROL ... .ENDC` – Script block

4.6 Conditionals

- `.IF {expr} ... .ELSE ... .ENDIF`

5 Capitalization

Case-Insensitive: Device names, models, dot commands, parameters, expression keywords

Case-Sensitive: Filenames, node names (interactive mode)

Variables: Lowercase unless `setcs` used

6 Device Instances

6.1 Format

`[prefix][name] [nodes...] [model] [params...]`

Examples:

- `R1 n1 n2 1k` – Resistor
- `C1 n1 0 10u IC=5` – Capacitor (w/ init cond)
- `L1 n1 n2 1m` – Inductor
- `V1 n+ n- DC 5` – Voltage source
- `I1 n+ n- AC 1m` – Current source
- `D1 anode cathode DMOD` – Diode
- `Q1 c b e QMOD` – BJT
- `M1 d g s b NMOS W=10u L=1u` – MOSFET
- `X1 out in gnd AMP gain=10` – Subcircuit

6.2 Device Prefixes

- **R** Resistor, **C** Capacitor, **L** Inductor
- **V** Voltage src, **I** Current src
- **D** Diode, **Q** BJT, **M** MOSFET, **J** JFET
- **K** Mutual inductance, **T** Transmission line
- **X** Subcircuit, **B** Behavioral, **E/F/G/H** Dependent src
- **A** XSPICE, **N** OSDI/Verilog-A

6.3 Special Sources

- **Behavioral**: `B1 n+ n- V={expr}` or `I={expr}`
- **Waveforms**: SIN, PULSE, PWL, EXP, SFFM, AM, TRNOISE

- **Example**: `V1 n+ n- PULSE(0 5 1n 1n 1n 50n 100n)`
- **XSPICE**: Prefix A, ports: %v, %i, %vd, %id
- **Verilog-A**: Prefix N, load with .OSDI

6.4 Modifiers

- `m=val` – Multiplier (parallel for R/C/L, scale for M)
- `scale=val` – Geometry scaling
- `temp=val` – Device temperature
- `dtemp=val` – Temperature offset

7 Topological Rules

- No voltage source/inductor loops
- No current source/capacitor cut-sets
- Every node: DC path to ground, $\geq$2 connections

8 Best Practices

- Define models before use
- Use {} for compile-time params in expressions
- Avoid reserved words as identifiers
- Check node connectivity (no floating nodes)
- Use meaningful node/device names
- Add .SAVE before analysis in .CONTROL blocks
- Ground reference required (node 0)

9 Compatibility Modes

`set ngbehavior=mode`

- **ps** – PSPICE
- **lt** – LTSPICE
- **lta** – Full LTSPICE

Example Netlist

RC Low-Pass Filter with Amplifier – Demonstrates Key Ngspice Syntax

Circuit Description: Input signal → RC filter → Amplifier → Output. Demonstrates parameters, subcircuits, models, multiple analyses, and control scripts.

```
* RC Low-Pass Filter with Amplifier Example
* Demonstrates: parameters, subcircuits, models,
* analyses, and control blocks

.TITLE "RC Filter and Amplifier Demo"

*=============================================
* HEADER SECTION: Parameters and Options
*=============================================

.PARAM VCC=5        $ Supply voltage
.PARAM fc=1k        $ Cutoff frequency
.PARAM Rval=1k      $ Filter resistance
.PARAM Cval={1/(2*3.14159*fc*Rval)} $ Calc C

.OPTIONS reltol=0.001 abstol=1e-12
.TEMP 27            $ Temperature in Celsius

.GLOBAL vcc gnd     $ Global nodes

*=============================================
* MODEL DEFINITIONS
*=============================================

* Simple diode model for clamping
.MODEL DCLAMP D(IS=1e-14 RS=10 CJO=1p)

* NPN transistor model
.MODEL QNPN NPN(BF=200 IS=1e-15 VAF=100)

*=============================================
* SUBCIRCUIT DEFINITIONS
*=============================================

* Simple amplifier subcircuit
.SUBCKT AMP in out vcc gnd gain=10
  * Parameters: gain (default=10)
  .PARAM Rf={gain*1k}
  .PARAM Rin=1k

  * Voltage follower with gain
  R1 in ninv {Rin}
  R2 ninv out {Rf}
  R3 inv gnd 10k

  * Ideal opamp (behavioral source)
  B1 out gnd V={gain*(v(in)-v(inv))}

  * Supply decoupling (just for show)
  C1 vcc gnd 100n
.ENDS AMP

*=============================================
* MAIN CIRCUIT
*=============================================

* Input voltage source with AC and transient
Vin in gnd DC 0 AC 1 SIN(0 1 1k 0 0)

* RC Low-Pass Filter
R1 in filt {Rval}   $ Resistor
C1 filt gnd {Cval}  $ Capacitor

* Protection diodes (demonstrates model usage)
D1 filt vcc DCLAMP  $ Clamp to VCC
D2 gnd filt DCLAMP  $ Clamp to GND

* Amplifier instance (subcircuit call)
X1 filt out vcc gnd AMP gain=5

* Load resistor with multiplier
Rload out gnd 10k m=2 $ Two 10k in parallel

* Power supply
Vcc vcc gnd DC {VCC}

*=============================================
* INITIAL CONDITIONS
*=============================================

.IC v(filt)=0 v(out)=0

*=============================================
* ANALYSIS COMMANDS
*=============================================

* DC Operating Point
.OP

* AC Analysis (frequency sweep)
.AC DEC 20 10 100k

* Transient Analysis
.TRAN 10u 5m 0 10u

* DC Sweep
.DC Vin -2 2 0.1

*=============================================
* OUTPUT CONTROL
*=============================================

.SAVE v(in) v(filt) v(out) i(Vcc)

*=============================================
* CONTROL BLOCK (Scripting)
*=============================================

.CONTROL
  * Run all analyses
  run

  * AC Analysis plots
  set hcopydevtype=postscript
  plot vdb(filt) vdb(out) xlog $ Bode plot
  plot vp(filt) vp(out) xlog $ Phase plot

  * Transient plots
  plot v(in) v(filt) v(out)

  * Measurements
  meas tran vout_max MAX v(out)
  meas tran vout_min MIN v(out)
  meas ac cutoff WHEN vdb(out)=-3
```

```
 * Conditional execution
 if vout_max > 3
   echo "Warning: Output exceeds 3V"
 end

 * DC transfer characteristic
 plot v(out) vs v(in)

 * Print operating point info
 op
 print all
.ENDC

*=========================================
* CONDITIONAL NETLIST (Example)
*=========================================

.IF {VCC > 3.3}
* High voltage configuration
Rbias vcc out 100k
.ELSE
* Low voltage configuration
Rbias vcc out 47k
.ENDIF

*=========================================
* END OF NETLIST (Required)
*=========================================

.END
```

Example Breakdown

Key Features Demonstrated

1. Structure & Order

- Title line (comment)
- .TITLE command
- Parameters (.PARAM)
- Options and temperature
- Global nodes
- Models (.MODEL)
- Subcircuits (.SUBCKT/.ENDS)
- Circuit instances
- Initial conditions (.IC)
- Analyses (.OP, .AC, .TRAN, .DC)
- Output control (.SAVE)
- Control block (.CONTROL/.ENDC)
- Conditionals (.IF/.ELSE/.ENDIF)
- .END statement

2. Syntax Elements

- Comments: *, $, end-of-line
- Parameters with expressions {...}
- Device instances: R, C, D, V, X, B
- Node names and ground (gnd)
- Scale suffixes: k, m, n, u
- Subcircuit parameters
- Model references
- Device multiplier (m=)
- Waveforms: SIN
- Behavioral source (B)

3. Analysis Types

- .OP – Operating point
- .AC – Frequency response
- .TRAN – Time domain
- .DC – DC sweep

4. Control Block Features

- run command
- plot commands with options
- meas statements
- Conditional (if/end)
- echo for messages
- print for data output

5. Best Practices Shown

- Section comments for organization
- Meaningful node/device names
- Parameters for reusability
- .GLOBAL for supply nodes

- .SAVE before run
- Calculated parameter values
- Ground reference (node 0/gnd)
- Models defined before use

Running the Example

Batch mode:

```
ngspice -b example.cir -o output.log
```

https://github.com/Numeric-Solutions/Ngspice-Netlist-Cheat-Sheet

Interactive mode:

```
ngspice example.cir
```

With modifications:

```
ngspice
ngspice 1 -> source example.cir
ngspice 2 -> alter VCC = 3.3
ngspice 3 -> run
ngspice 4 -> plot v(out)
```