Qspice - Reference Guide by KSKelvin

KSKelvin Kelvin Leung

Created on 8-4-2023
Last update on 11-13-2023

QSPICE

- QSPICE
  - Arthor : Mike Engelhardt
  - Download : https://www.qorvo.com/design-hub/design-tools/interactive/qspice

Schematic Editor Keyboard Shortcuts

HELP > Schematic Capture > Schematic Editor > Keyboard Shortcuts

Key	Command
␣ (spacebar)	Zoom to fit
B[1]	Behavioral source
C[1]	Capacitor
D[1]	Diode
E[1]	E-source
F	F-source
G[2]	Ground, G-source
H	H-source
I	Current Source
J[1]	JFET
L[1]	Inductor
M[1]	MOSFET
Q[1]	Bipolar Transistor
R[1]	Resistor
S[1]	Voltage Controlled Switch
T[3]	Place Text
V[1]	Voltage Source
W	Start a wire
Y	Piezoelectric Crystal
Z[1]	MESFET

Ctrl-A	Draw an arc(graphical annotation)
Ctrl-B	Draw a box(graphical annotation)
Ctrl-C	Copy selected object(s) to clipboard
Ctrl-F	Find
Ctrl-G	Toggle display of grid dots
Ctrl-L	Draw a line(graphical annotation)
Ctrl-M	Mirror selected object(s)
Ctrl-R	Rotate selected object(s)
Alt-Ctrl-R	Rotate in 45° increments
Ctrl-V	Paste
Ctrl-X	Cut
Ctrl-Y	Redo
Ctrl-Z	Undo
Ctrl-3	Draw a triangle(graphical annotation)
;	Toggle a text graphic's comment status
F2	Toggle visibility of the Symbol and IP Browser pane.
F3	Toggle visibility of the Symbol Properties pane.
F4	Toggle visibility of the output console.
F5	Run the simulation.

[1] Repeated depressions of the key cycles through different versions of the symbol.
[2] Repeated depressions of 'G' cycles through different versions of the ground symbol and then G-source symbols.
[3] The period key, '.', is accepted as a synonym for 'T'.

Symbol Editor and Waveform Viewer Keyboard Shortcuts

HELP > Waveform Viewer > Keyboard Shortcuts / HELP > Schematic Capture > Symbol Editor > Keyboard Shortcuts

Waveform Viewer Keyboard Shortcuts

Key	Command
Delete	Delete attached cursor if pointing to a readout or delete selected plot labels
F	Zoom to fit(all panes)
F4	Toggle visiblity of the console display
F5	Rerun the simulation
␣	Reload Plot configuration file
←	Move attached cursor left
→	Move attached cursor right
↑	Move attached cursor to next step } for .step
↓	Move attached cursor to previous step } for .step
Ctrl-A	Add a trace
Ctrl-C	Copy
Ctrl-D	Delete a plotting pane
Ctrl-F	Find
Ctrl-G	Turn Grid On/Off
Ctrl-V	Paste
Ctrl-P	Print
Ctrl-W	Add a plotting pane
Ctrl-X	Cut
Ctrl-Y	Redo
Ctrl-Z	Undo

** Probe Differentiate Voltage : Hold Alt and click differentiate nodes

Symbol Editor Keyboard Shortcuts

Key	Command
Ctrl-A	Draw an arc defined by three points
Shift-Ctrl-A	Draw an arc defined by four points
Ctrl-B	Draw a box(or a box for an image)
Ctrl-C	Copy selection(s) to clipboard
E	Draw an Ellipse
Ctrl-F	Find
Ctrl-L	Draw a line
Ctrl-M	Mirror selected objects
P	Place a pin
Ctrl-R	Rotate selected objects
T	Place a text attribute
Ctrl-V	Paste
Ctrl-X	Cut
Ctrl-Y	Redo
Ctrl-Z	Undo
Ctrl-3	Draw a triangle
F3	Toggle visibility of the Properties pane.

Waveform Viewer Functions and Keywords (.func , .meas)

HELP > Waveform Viewer > Waveform Expressions

The following functions, constants, and keywords are recognized in expressions of waveform data.

Waveform Viewer Functions and Keywords

Syntax		Description
ABS(x)	$\lvert x \rvert$	Absolute value of x
ACOS(x)	$\cos^{-1} x$	Inverse cosine of x
ACOSH(x)		Inverse hyperbolic cosine of x
ARCCOS(x)		Inverse cosine of x
ARCCOSH(x)		Inverse hyperbolic cosine of x
ARCSIN(x)		Inverse sine of x
ARCSINH(x)		Inverse hyperbolic sine of x
ARCTAN(x)		Inverse tangent of x
ARCTANH(x)		Inverse hyperbolic tangent of x
ASIN(x)	$\sin^{-1} x$	Inverse sine of x
ASINH(x)		Inverse hyperbolic sine of x
ATAN(x)	$\tan^{-1} x$	Inverse tangent of x
ATAN2(x,y)[1]		Four quadrant inverse tangent of x
ATANH(x)		Inverse hyperbolic tangent of x
BUF(x)		x > .5 ? 1 : 0
CBRT(x)	$\sqrt[3]{x}$	Cube root of x
CEIL(x)		x rounded up to nearest integer
COS(x)	$\cos x$	Cosine of x
COSH(x)		Hyperbolic cosine of x
COT(x)		Cotangent of x
D(x)		Derivative of x
DD(x)		Second derivative of x
D²(x)		Second derivative of x
E		2.7182818284590452354
ERF(x)		Error function of x
ERFC(x)		Complementary error function of x
EXP(x)	e^x	e raised to the x power
EXP10(x)	10^x	10 raised to the x power
FABS(x)		Absolute value of x
FLOOR(x)		x rounded down to nearest integer
FREQ	←	Frequency
FREQUENCY	←	Frequency
GAMMA(x)		Gamma function of x
HYPOT(x,y)	$\sqrt{x^2+y^2}$	√(x²+y²)
IF(x,y,z)		(x > .5) ? y : z
ILOGB(x)		Unbiased exponent of x
IM(x)	$\mathrm{im}(x)$	Imaginary part of x
IMAG(x)		Imaginary part of x
INT(x)		x rounded to nearest integer
INV(x)		x > .5 ? 0 : 1
INVSQRT(x)		1÷√x
ISNAN(x)		One if x is not a number, otherwise zero
J	←	√-1
J0(x)		Zero order Bessel function of the first kind at x
J1(x)		First order Bessel function of the first kind at x
JN(x,n)		N^{th} order Bessel function of the first kind at x
K		1.380649e-23 J/°K
LGAMMA(x)		Log-gamma function of x
LIMIT(x,y,z)		Mutually intermediate value of x,y, and z
LN(x)	$\log_e x$	Natural logarithm of x
LOG(x)	$\log_e x$	Natural logarithm of x
LOG10(x)	$\log_{10} x$	Logarithm of x in base 10
LOG1P(x)		Natural logarithm of (x + 1)
LOG2(x)		Logarithm of x in base 2
LOGB(x)		LOG2(ABS(x))
MAG(x)	$\lvert x \rvert$	Absolute value of x
MAX(x,y)		Maximum of x and y
MAXMAG(x,y)		x or y with maximum magnitude
NAN		A value guaranteed to be not a number
MIN(x,y)		Minimum of x and y
MINMAG(x,y)		x or y with minimum magnitude
PH(x)	$\angle x$	Phase of x
PHASE(x)	$\angle x$	Phase of x
PI	←	3.14159265358979323846
POW(x,y)	x^y	x raised to the y power
POWN(x,y)		x raised to the nearest integer value of y
PWR(x,y)	$\lvert x \rvert^y$	Absolute value of x raised to the y power
PWRS(x,y)		x >= 0 ? x^y : $-x^y$
Q		1.602176487e-19 Coulomb
RE(x)	$\mathrm{re}(x)$	Real part of x
REAL(x)		Real part of x
RINT(x)		x rounded to the nearest integer
ROUND(x)		x rounded to the nearest integer
SGN(x)		Sign of x
SIGN(x)		Sign of x
SIN(x)	$\sin x$	Sine of x
SINH(x)		Hyperbolic sine of x
SQRT(x)	$\sqrt{x}$	Square root of x
TABLE(x,x1,y1,...)		Interpolate the table given as x1,y1, x2,y2,... at point x
TAN(x)	$\tan x$	Tangent of x
TANH(x)		Hyperbolic tangent of x
TAUGRP(x)		Group delay of x
TBL(x,x1,y1,...)		Interpolate the table given as x1,y1, x2,y2,... at point x
TEMP	←	Circuit temperature
TG(x)		Group delay of x
TIME	←	Time
TRUNC(x)		Integer part of s
URAMP(x)		x > 0 ? x : 0
USTEP(x)		x > 0 ? 1 : 0
Y0(x)		Zero order Bessel function of the second kind at x
Y1(x)		First order Bessel function of the second kind at x
YN(x)		N^{th} order Bessel function of the second kind at x

[1] For complex data, the syntax is ATAN2(z). The meaning is ATAN2(IMAG(z),REAL(z)).

← simulation variable

← important constant

Function and Operators for Behavioral V and I Sources

HELP > Simulator > Device Reference > B. Behavioral Sources

Functions

Name		Description
abs(x)	$\lvert x \rvert$	Absolute value of x
acos(x)	$\cos^{-1} x$	arc cosine of x
arccos(x)		Synonym for acos()
acosh(x)		arc hyperbolic cosine of x
asin(x)	$\sin^{-1} x$	arc sine of x
arcsin(x)		Synonym for asin()
asinh(x)		Arc hyperbolic sine
atan(x)	$\tan^{-1} x$	Arc tangent of x
arctan(x)		Synonym for atan()
atan2(y,x)		Four quadrant arc tangent of y/x
atanh(x)		Arc hyperbolic tangent
buf(x)		1 if x > .5, else 0
ceil(x)		Integer equal or greater than x
cos(x)	$\cos x$	Cosine of x
cosh(x)		Hyperbolic cosine of x
ddt(x)	$dx/d\text{time}$	Time derivitive x
delay(x,y)		x delayed by y
delay(x,y,z)[1]		x delayed by y, but store no more than z history
dlim(x,y,z)		x bounded by y which it asymptotically starts to approach at y+z as a first inverse order Laurent series
exp(x)	e^x	e to the x
floor(x)		Integer equal to or less than x
hypot(x,y)	$\sqrt{x^2+y^2}$	sqrt(x^2 + y^2)
idt(x,y,z)	$\int x\, d\text{time} + y$	Time integral of x with initial condtion of y reset when z > .5
if(x,y,z)		If x > .5, then y else z
int(x)		Convert x to integer
inv(x)		0. if x > .5, else 1.
limit(x,y,z)		Intermediate value of x, y, and z
ln(x)	$\log_e x$	Natural logarithm of x
log(x)	$\log_e x$	Alternate syntax for ln()
log10(x)	$\log_{10} x$	Base 10 logarithm
max(x,y)		The greater of x or y
min(x,y)		The smaller of x or y
pow(x,y)	x^y	x^y
pwr(x,y)	$\lvert x \rvert^y$	abs(x)^y
pwrs(x,y)		sgn(x)*abs(x)^y
random(x)		Random number from 0. to 1. depending on the integer value of x. Interpolation between random numbers is linear for non-integer x.
sin(x)	$\sin x$	Sine of x
sinh(x)		Hyperbolic sine of x
sqrt(x)	$\sqrt{x}$	Square root of x
table (x,a,b,c,d,...)		Interpolate x from the look-up table given as a set of pairs of constant values.
tan(x)	$\tan x$	Tangent of x.
tanh(x)		Hyperbolic tangent of x
ulim(x,y,z)		x bounded by y which it asymptotically starts to approach at y-z as a first inverse order Laurent series

Operators grouped in reverse order of precedence of evaluation

Operand	Description
&	Boolean AND
|	Boolean OR
>	True if expression on the left is greater than the expression on the right.
<	True if expression on the left is less than the expression on the right.
>=	True if expression on the left is greater than or equal the expression on the right.
<=	True if expression on the left is less than or equal the expression on the right.
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	** / ^ Raise left hand side to power of right hand side. Same as '^'.
!	Boolean not the following expression.

Available Function in B source not listed

- Trunc(x) ; floor(x) ; int(x) : rounded down integer
- Rint(x) ; round(x) : rounded to nearest integer
- Ceil(x) : rounded up integer
- Ustep(x) : x > 0 ? 1 : 0
- Uramp(x) : x > 0 ? x : 0

Hierarchical Block

Hierarchical Block

Qspice : filter_circuit.qsch ; filter_circuit_app.qsch

[1] Create a child schematic (.qsch) with circuit and net label
[2] Right click on net label and select “This is a Port”

[3] Create a new schematic which will call to use hierarchical
[4] Edit → Draw Hierarchical Entry

Hierarchical Block

Qspice : filter_circuit.qsch ; filter_circuit_app.qsch

[5] Change component text to match child schematic name
[6] Right click hierarchy component and “Add Port”
[7] Name ports as port name defined in child schematic
[8] Right click hierarchy component and “Enter Schematic” should open child schematic

A completed example

Hierarchical Block – Get subckt with Hierarchical Block method

[1] If a hierarchical block is created, in top-level schematic View > Netlist

Engelhardt

> Could we hear some detail how can we do it?

1. In the higher-level schematic, do top-level menu(not right click) command View=>Netlist.
2. In the netlist, identity the subcircuit of the hierarchal block you wish to library as widely usable IP.
3. Select that block of text and copy it to the clipboard with Ctrl-C.
4. Close the netlist and paste(Ctrl-V) the text into a schematic(or a blank symbol). That will invoke the 3rd party import routine.
5. You'll now have a symbol that contains the circuitry that you can use in any schematic in any directory.
6. Enjoy.

–Mike

[2] In the netlist, identify the subcircuit of the hierarchal block
[3] Select that block of text and copy it to the clipboard with Ctrl-C
[4] Close the netlist and paste (Ctrl-V) the text into a schematic (or a blank symbol) to invoke the 3rd party import routine
[5] You'll now have a symbol that contains the circuitry that you can use in any schematic in any directory.

Symbol for Subckt [Embedded Subckt]

Symbol for Subckt [Embedded Subckt]

[1] Assume user has a sub-circuit .subckt in text format library file
[2] Use a text editor to open library file, copy text for sub-circuit
[3] In Qspice schematic, paste the text (Ctrl-V)

[4] Rename pins by referring to description in model file

** If sub-circuit consist of other .subckt, click this block to include entire file

Symbol for Subckt [Embedded Subckt]

[5] File > New > New Symbol to open a Symbol Window
[6] In schematic, Ctrl-C to copy Component X1
[7] Goto Symbol Window, Ctrl-V to paste component

[8] Delete the box, rearrange pins location, and draw the symbol
[9] Save into a .qsym symbol format

Symbol for Subckt [Embedded Subckt]

[10] In Schematic, Symbol & IP Browser, Right Click to “Add a symbol directory”
[11] Drag created component to schematic
[12] ** text library is no longer required as .subckt is integrated into symbol

A completed example

Symbol for Subckt [Embedded Subckt] : Label with Text

Qspice : irf530 with text.qsym

- Text can be used in label
  - For example, instead of changing net name, you can
    - Right click > Text
    - Right click on text > Select “Comment”
      - Text not comment will become valid item in netlist
      - Can change font size and color
    - Be careful 1st attribute is device name (e.g. IRF530 in example), and doesn’t comment it

** Technique to Create embedded SUBCKT Script from Schematic

Qspice : RC_Params.qsym ; RC_sch.qsch

[1] Draw a schematic

This example has

- Two ports : in and out
- Two input params : R and Fcutoff
- One calculated parameter : C

[2] View > Netlist, copy this netlist

Subcircuit Definition

Syntax: .subckt NAME N1 N2 N3 ...
...
...
...
.ends NAME

[3 : Method#1] Convert netlist to subckt

- First line add .subckt
- RC is NAME of subckt determined by user
- Follow with ports (net) : in out
- Follow with params : R=1K Fcutoff=1K
- Remove .param R=1K and .param Fcutoff=1K
- Last line add .ends RC

This is the finished version

```
.subckt RC in out params: R=1K Fcutoff=1K
R1 in out R
C1 out 0 C
.param C=1/2/pi/R/Fcutoff
.ends RC
```

Copy and paste .subckt script to schematic, then follow standard symbol creation procedure for embedded SUBCKT symbol creation

** Technique to Create embedded SUBCKT Script from Schematic

Qspice : RC_noParams.qsym

[3 : Method#2] Convert netlist to subckt

- First line add .subckt
- RC is NAME of subckt determined by user
- Follow with ports (net) : in out
- Remove .param R=1K and .param Fcutoff=1K
- Last line add .ends RC

This is the finished version

```
.subckt RC in out
R1 in out R
C1 out 0 C
.param C=1/2/pi/R/Fcutoff
.ends RC
```

Copy and paste .subckt script to schematic, then follow standard symbol creation procedure for embedded SUBCKT symbol creation

[4 : Method#2] Add input param into symbol

- Two input params : R=1K and Fcutoff=1K
- Type T or Right Click > Text, input
  - R=1K
  - Fcutoff=1K
- This will create 2nd and 3rd attribute in String Attributes, where you can select to visible or not in symbol

Autogenerate Symbol Pin Assignment

Qspice : Autogenerate Symbol Pin Assignment.qsch

```
.subckt Block01 1
.ends Block01

.subckt Block02 1 2
.ends Block02

.subckt Block03 1 2 3
.ends Block03

.subckt Block04 1 2 3 4
.ends Block04

.subckt Block05 1 2 3 4 5
.ends Block05

.subckt Block06 1 2 3 4 5 6
.ends Block06

.subckt Block07 1 2 3 4 5 6 7
.ends Block07

.subckt Block08 1 2 3 4 5 6 7 8
.ends Block08

.subckt Block14 1 2 3 4 5 6 7 8 9 10 11 12 13 14
.ends Block14
```

Symbol for Subckt [Create Symbol and Link to Library]

Symbol for Subckt [Create Symbol and Link to Library]

Example to create subckt symbol for irf530

[1] File → New → New Symbol

[2] Right Click → Pin (to add 3 pins with order D, G, S)

[3] Right Click at center of Pin to review PinOrder and PinLabel

irf530.lib

```
*Feb 16, 2010
*Doc. ID: 90181, Rev. A
*File Name: part irf530_PS.txt and part irf
*This document is intended as a SPICE model
*constitute a commercial product data sheet
*appropriate data sheet of the same number
*limits.
.SUBCKT irf530 1 2 3
*************************************
*      Model Generated by MODPEX    *
*Copyright(c) Symmetry Design Systems*
*         All Rights Reserved       *
*    UNPUBLISHED LICENSED SOFTWARE  *
*   Contains Proprietary Information *
*      Which is The Property of     *
*     SYMMETRY OR ITS LICENSORS     *
*Commercial Use or Resale Restricted *
*   by Symmetry License Agreement   *
*************************************
* Model generated on Apr 24, 96
* Model format: SPICE3
* Symmetry POWER MOS Model (Version 1.0)
* External Node Designations
* Node 1 -> Drain
* Node 2 -> Gate
* Node 3 -> Source
M1 9 7 8 8 MM L=100u W=100u
* Default values used in MM:
```

Symbol for Subckt [Create Symbol and Link to Library]

[4] Draw a box for outline
[5] Put an “X” in Symbol Type in Symbol Properties
[6] Right Click → Text → Put an “X”

[7] To justify Pin label, double click center of Pin
[8] Click red dot other than its centered justification
[9] Save symbol file as .qsym

Symbol for Subckt [Create Symbol and Link to Library]

[10] In Schematic, Symbol & IP Browser, Right Click to "Add a symbol directory"
[11] Drag created component to schematic

[12] Right Click on symbol, "Add New Attribute" as irf530
[13] In Symbol Properties, add "Library File" as irf530.lib
** library file is required to be put in schematic directory

Symbol to call Schematic .qsch (Hierarchical)

Symbol to call Schematic .qsch : Method #1

Qspice : RC_sch.qsym ; RC_sch.qsch

[1] Draw a schematic
This example has
- Two ports : in and out

[2] Create a symbol
- Pin name needs to match schematic ports (order not important)
- Use Text to assign
  - Name : X1
  - 1st attribute : [schematic name]
- Symbol Type : Blank (nothing)
  - Don't assign a X (X for subckt), hierarchical entry no symbol type

Remark :
Major Different for Symbol to call schematic (hierarchical entry) and subckt
- To call schematic (hierarchical entry)
  - Symbol Type : Blank
  - Name : X1
  - 1st attribute : schematic name
- To call subckt
  - Symbol Type : X
  - Name : X1
  - 1st attribute : subckt name

Symbol to call Schematic .qsch : Method #2

Qspice : RC_sch.qsym ; RC_sch.qsch

[1] Draw a schematic
This example has

- Two ports : in and out
- Right click these nets and select "This is a port" (only these ports will auto generate hierarchical entry)

[2] Right Click > Open Parent Schematic
It will ask to automatically generate a parent schematic

[3] Copy hierarchical block in parent with Ctrl-C

[4] File > New > New Symbol, paste with Ctrl-V
** Symbol Type : Blank (nothing)

- Don't assign a X (X for subckt), hierarchical no symbol type. If you re-open a hierarchical symbol, please pay attention in here as it may auto assign an X into Symbol Type

[5] Save as a .qsym symbol

Remark :
Major Different for Symbol to call schematic (hierarchical entry) and subckt

- To call schematic (hierarchical)
  - Symbol Type : Blank
  - Name : X1
  - 1st attribute : schematic name
- To call subckt
  - Symbol Type : X
  - Name : X1
  - 1st attribute : subckt name

Qspice HELP Reference
Help > Schematic Capture > Schematic Hierarchy

Convert MOSFET M to subckt Symbol

Convert MOSFET M to subckt Symbol

Qspice : Call Lib from M.qsch

It is possible to convert MOSFET symbol M into a sub-circuit symbol by

1. Change Symbol Type from MN to X
2. Library File as subckt library file
3. 1st attribute as subckt name
4. [Optional] Change Symbol Name to X?

** this alternative symbol is 4 pins (+ base), which cannot support 3 pin subckt

Pin Order in Symbol MN and MP

Pin Order for Symbol MN (NMOS)

Application

Basics	
Symbol Type	MN
Description	N-Channel MOSF...
Allow Shorted ...	False
Library File	NMOS.txt
String Attributes	
Text Order Invis.	Content
Name:	M1
1st attribute	NMOS
Pin Nets	
Pin Name Invis.	Net
D	1
G	2
S	3

Pin Order for Symbol MP (PMOS)

Application

Basics	
Symbol Type	MP
Description	P-Channel MOSF...
Allow Shorted ...	False
Library File	PMOS.txt
String Attributes	
Text Order Invis.	Content
Name:	M2
1st attribute	PMOS
Pin Nets	
Pin Name Invis.	Net
D	1
G	2
S	3

Ø-Device (.DLL)

Simulation with C++ (Ø-Device)

Qspice : Cpp_Demo.qsch

[1] In schematic, Right click > Draw Hierarchical Entry
[2] Right click hierarchical block and Show Symbol Properties
[3] In Symbol Type, change to Ø(.dll)

[4] Right click hierarchical block > Add Port
[5] Right click each port, select corresponding Port Type
- For In, Port Type : Input
- For Out, Port Type : Output

[6] Right click each port, select corresponding Data Type
- For In, Data Type : float (64 bit double)
- For Out, Data Type : float (64 bit double)

[7] Right click hierarchical block > C++ Interface
- Create C++ Template > OK

Simulation with C++ (Ø-Device)

Qspice : Cpp_Demo.qsch

QSPICE™ - cpp_demo_x1.cpp

File Edit View Help

```
// Automatically generated C++ file on Tue Sep 12 23:27:26 2023
//
// To build with Digital Mars C++ Compiler:
//
//    dmc -mn -WD cpp_demo_x1.cpp kernel32.lib

union uData
{
   bool b;
   char c;
   unsigned char uc;
   short s;
   unsigned short us;
   int i;
   unsigned int ui;
   float f;
   double d;
   long long int i64;
   unsigned long long int ui64;
   char *str;
   unsigned char *bytes;
};

// int DllMain() must exist and return 1 for a process to load the
// See https://docs.microsoft.com/en-us/windows/win32/dlls/dllmain
int __stdcall DllMain(void *module, unsigned int reason, void *res

// #undef pin names lest they collide with names in any header fil
#undef In
#undef Out
#include <cmath>

extern "C" __declspec(dllexport) void cpp_demo_x1(void **opaque, d
{
   double  In  = data[0].d; // input
   double &Out = data[1].d; // output

// Implement module evaluation code here:
   Out = pow(In,5);
}
```

Compile DLL F5
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Select All Ctrl+A

[7]

[8]

"cpp_demo_x1.dll" created successfully [9]

CAP OVR

[7] add #include<cmath> if math function is needed

[8] Implement the function of the device below the comment

// Implement module evaluation code here

[9] Right click > Compile DLL

- If success, a successful statement in status bar

[10] Run SPICE simulation that call the C++ device

Ø-Device with Input Parameter

Qspice : CodeBox_with_Param.qsch

[1] For Input parameter, in creating hierarchical block

1. Right click the block > select Add attribute
2. [data type] [Input Port name] = [parameter] or <val>

As GND is needed if hierarchical doesn't have input port

- Add Port
- Right click Port > Data Type > DLL's GND

** Important note!! Whenever you change input port, parameter or output port, you should recreate and copy code to a new C++ template, as Qspice requires particular order for index in data[]

- Right click the block > C++ Interface > Create C++ Template
- In this example, float IN=var will auto generate as input parameter double IN = data[0].d;

DLL's GND in Ø-Device

Qspice : Floating_CodeBox.qsch

- Two Purposes of DLL's Gnd
  - Floating operation where Ø-Device reference is not 0
  - Only output but no input port is defined (previous slide)

Example of floating ground operation

Engelhardt

The inputs and outputs of a .DLL go through a converter. If there's no DLL GND specified, the converters operate ground referenced. If you give a .DLL a GND, then inputs and output are referred to that .DLL GND port. It let's you run your logic hot decked as one might in an offline converter.

Use of MAXSTEP in C++ block (Ø-Device)

Qspice : Cpp_MAXSTEP.qsch

- If delay in response is observed when C++ block is used, consider to limit maximum time step by adding
  - .options MAXSTEP=<value>

```
pulse 0 1 0 1/f/10 1/f/10 1/f*3/10 1/f
.param f = 100K
.tran 20/f
.plot V(in) V(out)
.options MAXSTEP = 1/f/1000
```

```
pulse 0 1 0 1/f/10 1/f/10 1/f*3/10 1/f
.param f = 100K
.tran 20/f
.plot V(in) V(out)
.options MAXSTEP = 1/f/1000
```

Bus and Hierarchical Block

Bus and Hierarchical Block

- Bus and Hierarchical Block
  - With Bus, data is defined as Data[n:m]
  - In Qspice, this net name format create a series of net names from Data[n] to Data[m]
    - If n<m, net name are Data[n], Data[n+1], Data[n+2], … , Data[m]
    - If n>m, net name are Data[n], Data[n-1], Data[n-2], … , Data[m]
  - For hierarchical block, subckt bus net names are assigned according to index sequence
  - To use data bus, it is recommending bus, hierarchical block and subckt with same data bus index, which can prevent unexpected behavior in net assignment

Bus and Hierarchical Block

Qspice : Parent-BUS.qsch / BUS_Hierarchical.qsch / bus_dll.cpp

[1] BUS is equivalent to create multiple net names with []
A[n:m] creates A[n], A[n+1], … A[m]

```
* C:\QspiceKSKelvin\01 User Guide and Script\01 Qspice Reference (
Ø†X1 «in´ui» «A[0]´b A[1]´b A[2]´b A[3]´b» «»  BUS_DLL
V1 in 0 pwl 0 0 1 15
X2 A[0] A[1] A[2] A[3] o0 o1 o2 o3 BUS_Hierarchical

.subckt BUS_Hierarchical d[0] d[1] d[2] d[3] o0 o1 o2 o3
V1 d[0] o0 0
V2 d[1] o1 0
V3 d[2] o2 0
V4 d[3] o3 0
.ends BUS_Hierarchical

.tran 1
.plot V(o0) V(o1)+1.1 V(o2)+2.2 V(o3)+3.3
.plot V(A[0]) V(A[1])+1.1 V(A[2])+2.2 V(A[3])+3.3
.plot V(in)
.end
```

[2] Subckt X1, hierarchical X2 are all feed in same name and order

Bus and Hierarchical Block : Change BUS name order

Qspice : Parent-BUS.qsch / BUS_Hierarchical.qsch / bus_dll.cpp

[1] Reverse BUS index is equivalent to create multiple net names with []
A[m:n] creates A[m], A[m-1], … A[n]

```
* C:\QspiceKSKelvin\01 User Guide and Script\01 Qspice Reference (
Ø†X1 «in´ui» «A[3]´b A[2]´b A[1]´b A[0]´b» «» BUS_DLL
V1 in 0 pwl 0 0 1 15
X2 A[3] A[2] A[1] A[0] o0 o1 o2 o3 BUS_Hierarchical

.subckt BUS_Hierarchical d[0] d[1] d[2] d[3] o0 o1 o2 o3
V1 d[0] o0 0
V2 d[1] o1 0
V3 d[2] o2 0
V4 d[3] o3 0
.ends BUS_Hierarchical

.tran 1
.plot V(o0) V(o1)+1.1 V(o2)+2.2 V(o3)+3.3
.plot V(A[0]) V(A[1])+1.1 V(A[2])+2.2 V(A[3])+3.3
.plot V(in)
.end
```

[2] Subckt X1-d[0] is connected to A[3]
[3] This order is feed into hierarchical block X2, e.g. A[3] is feed into X2-d[0]

Bus and Hierarchical Block : Change Hierarchical net label order

Qspice : Parent-BUS.qsch / BUS_Hierarchical.qsch / bus_dll.cpp

```
* C:\QspiceKSKelvin\01 User Guide and Script\01 Qspice Reference (
Ø†X1 «in´ui» «A[0]´b A[1]´b A[2]´b A[3]´b» «»  BUS_DLL
V1 in 0 pwl 0 0 1 15
X2 A[0] A[1] A[2] A[3] o0 o1 o2 o3 BUS_Hierarchical

.subckt BUS_Hierarchical d[3] d[2] d[1] d[0] o0 o1 o2 o3
V1 d[0] o0 0
V2 d[1] o1 0
V3 d[2] o2 0
V4 d[3] o3 0
.ends BUS_Hierarchical

.tran 1
.plot V(o0) V(o1)+1.1 V(o2)+2.2 V(o3)+3.3
.plot V(A[0]) V(A[1])+1.1 V(A[2])+2.2 V(A[3])+3.3
.plot V(in)
.end
```

[2] Hierarchical block X2 name is reversed, but Hierarchical / Subckt net assignment is based on order, therefore, A[0] is feed to hierarachical subckt d[3] in this case

General Technique

Max Time Step in .tran (and .bode) : Two methods

Qspice : MaxTimeStep.qsch

sin 0 1 1G .plot V(V1G)

Method 1 : Traditional Berkeley Syntax

.tran IGNORED TSTOP [TSTART [MAXSTEP]] [UIC]

.tran 0 10n 0 10p

Max Time Step

Method 2 : MAXSTEP in Simulator Option

.tran 10n

.options MAXSTEP=10p

Max Time Step
(this method apply to .tran and .bode)

Add [additional instance parameters]

1. Right Click on Component
2. Select "Add New Attribute"
3. Type parameter name and value [refer to help for full list of instance parameters]

This is an example to assign 0.2 ohms series resistance to inductor L1

Inductor Instance Parameters

Name	Description	Units	Default
AG	Wire or stripline is made of gold		
AL	Wire or stripline is made of aluminum		see below
AU	Wire or stripline is made of silver		
BEND	Fractional inductance correction for wire bend or proximity effects		1.
CPAR	Parallel capacitance	F	0.
CU	Wire or stripline is made of copper		see below
DIAMETER	Diameter of wire or air coil	m	
FREQUENCY	Frequency at Q. Also used to compute Rser due to skin effect		
HEIGHT	Height of PCB stripline above ground plane	m	
IC	Initial current if uic is specified on .tran statement	A	none
INDUCTANCE	Inductance of inductor	H	0.0
ISAT	Current causing inductance to drop to SATFRAC×INDUCTANCE	A	Infinite
LENGTH	Length of wire, stripline, or air coil	m	
LSAT	Inductance asymptotically approached in saturation	H	10% of INDUCTANCE
M	Number of parallel inductors		1.0
NI	Wire is made of nickel		
Q	Quality factor at FREQUENCY		
RPAR	Equivalent parallel resistance	Ω	Infinite
RSER	Equivalent series resistance	Ω	0.0
SATFRAC	Fractional drop in inductance at ISAT		0.7
THICK	Thickness of stripline on top of a PCB	m	0.0
TURNS	Number of turns of an air coil		
VERBOSE	Print wire L, Rser, Rpar results on the console		(not set)
WIDTH	Width of stripline on top of a PCB	m	

AC and DC Attribute in Source

Qspice : AC with Transient Source.qsch ; AC with Bias.qsch

Technique to perform AC analysis with a transient source

1. Right Click on Voltage/Current source
2. Select “Add New Attribute”
3. Type “AC 1” to define a 1V source for AC sweep
4. Add a .ac analysis statement, and comment transient analysis

Technique to perform AC analysis with DC in source

1. Right Click on Voltage/Current Sourec, Add New Attribute
2. To add DC source, type “DC …”
   - If without DC, simulator may not interpret the DC voltage during simulation. Best practice is to add DC

Laplace Time and Frequency Domain Simulation

Qspice : Laplace Simulation - Fdomain.qsch ; Laplace Simulation - Tdomain.qsch

2nd-order system step response
transfer function : omega^2/(s^2+2*zeta*omega+omega^2)

.param zeta = 0.7
.param omega = 1

.tran {eT}
.param eT=25

pulse 0 1 0 0 0 {eT/2} {eT}

Time Domain

Frequency Domain

Laplace = (s+100)/((s+1)*(s+10))

.ac dec 100 {1e-2/2/pi} {1e4/2/pi}

Laplace Time and Frequency Domain Simulation

Qspice : Laplace Close Loop.qsch

- Close Loop System Time and Bode
  - A technique to get Gp(s) and GH(s) is to add a perturbing source between output and feedback and perform ac analysis
  - In this example, Laplace function can collect in series for both .tran and .ac directive

AC (.ac) and Frequency Response Analysis (.bode)

Qspice : ACmethod.qsch ; BODEmethod.qsch

Technique to Probe Drain Current / General Current Probe

Qspice : Current_Probe_Method.qsch

3 common current probe method

1. Add a series resistor and probe R current
2. Add 0V voltage source and probe current of this voltage source. +ve represent current flow from + to – direction within symbol (i.e. current flow downward in above example)
3. Ctrl-A (Add Plot) in waveform viewer and select Id(M?)

Monte Carlo

Qspice : Monte Carlo.qsch

Random number from 0 to 1 depending on the seed

1K*(1+.1*{random()})

This example .1*{random()}
equivalent +0% to +10% change

```
.op
.step param dummy 0 100 1
.plot I(R1)

.options seedclock
.options seed=5
```

This enable random seed to be generated

This assign manual seed

Engelhardt

9-4-2023

OK, I just implemented

.options seedclock

It convolutes a 10MHz system clock with the simulation process ID to generate a physically random integer to seed the Mersenne Twister.

Include File (.inc) : HELP > Schematic Capture > Simulator > Include File (.inc)

Qspice : INC demo.qsch

- Include File
  - To include a file to execute by simulator
  - This allow to simplify schematic for directive or reuse purpose
- Example
  - This example use .inc to include a file called QspiceScript.txt
  - This script can
    - Calculate .func
    - Define .plot
    - Calculate .meas

Output Window

```
.meas vo_pk max v(out):
        9.99976 (at Time=0.05)
.meas tau find time when v(out)=0.63*vo_pk:
        0.004674    0.004674
```

QspiceScript.txt

```
.func dVOUT() d(V(out))
.plot dVOUT()
.plot V(out) V(in)
.meas Vo_pk max V(out)
.meas tau find time when V(out)=0.63*Vo_pk
```

Selection Guide option for Circuit Elements with 3rd Party Library

- Purpose
  - Use Q transistor as an example of how to have selection guide from 3rd party library
- Procedure
  - In C:\Program Files\Qspice, create a .txt file
    - e.g. My_NPN.txt
    - May require admin access
  - Copy and paste .model context into .txt and save
    - https://ltwiki.org/index.php?title=Standard.bjt
    - This link contains a list of BJT model
  - In Qspice schematic, add a NPN transistor with shortcut Q
  - Right click transistor, open symbol properties and change the library file from NPN.txt to My_NPN.txt
  - Right click transistor and Selection Guide is available now
- Reference
  - https://forum.qorvo.com/t/adding-model-files-to-qspice/14963/7

stevenbennett 5h

That's very helpful thanks. For anyone wanting more detail, this is what worked for me:

1: Create a custom .model containing text file in C:\Program Files\QSPICE e.g. My_NPN.txt

2: Paste in single, or multiple, .model statements e.g. from Standard.bjt - LTwiki-Wiki for LTspice 1 and save.

3: Add an NPN transistor from the "Q" folder in the Symbols & IP folder list in QSPICE.

4: Open the symbol properties for the NPN transistor by double clicking and change the Library File from NPN.txt to My_NPN.txt

5: Right click the NPN symbol and choose Selection Guide, which will now display all the added models.

6: The file My_NPN.txt will survive any of the frequent QSPICE updates.

B-Source as Comparator

Qspice : B-Source as Comparator.qsch

- Concept of Ideal Comparator with Behavioral Voltage Source
  - Formula of B-source is : if(V(pos)>V(neg),V(Vdd),V(Vss))
  - Practical comparator output normally is open-drain configuration, this is just for simulation purpose

Flat(x) and MC(x,y) functions equivalent to Ltspice

Qspice : Flat and MC Function.qsch

- Uniform random distribution
  - LTspice offers flat(x) and mc(x,y) functions, but not in Qspice (last check 10-3-2023)

- Function for flat(x) and mc(x,y)
  - .func flat(x) x*((random()*2)-1) ← Generate random [-x, x]
  - .func mc(x,y) x*(1+y*(random()*2-1)) ← Generate random [x*(1-y), x*(1+y)]

.step param x 1 200 1 Dummy For Loop
.op

.plot V(mc)
.plot V(flat)
.plot V(random)

flat(x) : Random number between -x and x with uniform distribution
.func flat(x) x*((random()*2)-1)

mc(x,y) : A random number between
x*(1+y) and x*(1-y) with uniform distribution
.func mc(x,y) x*(1+y*(random()*2-1))

Time in .tran and Logic Diagram in Waveform Viewer with .plot

Qspice : Time in .tran.qsch ; Logic Signal Plot.qsch

- Time in .tran
  - In .tran, run time is stored as a default parameter name Time
  - Therefore, use a B-source can convert Time into a voltage
  - Time can also be used in function

```
.tran 1    .func runtime() Time
.plot V(etime)
.plot runtime()
```

- Logic Diagram
  - A simple idea to plot logic signal into logic diagram format
  - Idea is to add an offset for each logic in .plot

```
.param f=1
.tran 1/f*16
.plot V(Bit3)+3.3 V(Bit2)+2.2 V(Bit1)+1.1 V(Bit0)
```

Dummy TTOL device to help in adaptive timestep

Qspice : TTOL - Dummy TTOL element - Enhance Timestep.qsch

- Dummy TTOL device
  - Qspice uses adaptive timestep
  - If a circuit uses a B-source, if(x,y,z) as a comparator, without TTOL device, its simulation timestep can far from compare instance and output looks weird
    - Example on Top Row
    - Precise time instance at compare action, but as no extra timestep after compare action, output looks like ramping as next timestep is far away (interpolation)
  - To resolve this without using MAXSTEP to limit timestep, a dummy TTOL device can be used (e.g. Switch), with TTOL instance parameters included
    - Example on Bottom Row
    - Extra time steps are added after V(compare) flip the switch, with additional time steps, output looks reasonable
    - Smaller TTOL value can yield a better results but with longer elapsed time

TTOL device to help in adaptive timestep

Qspice : TTOL - TTOL device to Interface Analog and Digital.qsch

- TTOL device interface
  - Qspice uses adaptive timestep
  - If a circuit uses a B-source, if(x,y,z) as a comparator, without TTOL device, its simulation timestep can far from compare instance and output looks weird
    - Example in Top Figure
    - Precise time instance at compare action, but as no extra timestep at compare action, output looks like trapezoidal as next timestep is far away
  - To resolve this without using MAXSTEP to limit timestep, a TTOL device can be used (e.g. buffer, with default TTOL=1u)
    - Example in Bottom Figure
    - Extra time steps are added after V(cmpA) flip the buffer, with additional time steps, output looks square waveform
    - Smaller TTOL value can yield a better results but with longer elapsed time

.meas
Measure Statements

Available Syntax for .meas

HELP > Simulator > Command Reference > Measure(.meas)

- Syntax: .meas NAME find EXPRESSION1 at EXPRESSION2
- Syntax: .meas NAME find EXPRESSION1 when EXPRESSION2=EXPRESSION3
- Syntax: .meas NAME find EXPRESSION1 when EXPRESSION2=EXPRESSION3 td=5n cross=10
- Syntax: .meas NAME find EXPRESSION1 when EXPRESSION2=EXPRESSION3 cross=last
- Syntax: .meas NAME deriv EXPRESSION1 at EXPRESSION2
- Syntax: .meas NAME trig EXPRESSION1=EXPRESSION2
- Syntax: .meas NAME targ EXPRESSION1=EXPRESSION2
- Syntax: .meas NAME trig EXPRESSION1=EXPRESSION2 targ EXPRESSION3=EXPRESSION4
- Syntax: .meas NAME trig EXPRESSION1=EXPRESSION2 rise=1 targ EXPRESSION1=EXPRESSION2 rise=11
- Syntax: .meas NAME avg|max|min|pp|rms|integ EXPRESSION1
- Syntax: .meas NAME avg|max|min|pp|rms|integ EXPRESSION1 from EXPRESSION2 to EXPRESSION3
- Syntax: .meas NAME avg|max|min|pp|rms|integ EXPRESSION1 trig EXPRESSION2=EXPRESSION3 targ EXPRESSION4=EXPRESSION5
- Syntax: .meas NAME four FREQ EXPRESSION [...]
- Syntax: .meas NAME fra FREQ INPUT OUTPUT [...]

Example of .meas in .ac analysis

Qspice : meas ac demo 01.qsch

```
.param fn = 1K
.param wn = 2*pi*fn
.param z = 0.2
Laplace=wn^2/(s^2+2*z*wn*s+wn^2)
```

```
.ac dec 100 100 1e4
.plot V(out)
```

Support 6 types of measure
avg/ave : average
max : maximum
min : minimum
pp : peak to peak
rms : root mean square
integ : integral

```
.meas gainpeak max dB(V(out))
.meas freq@+5dB find frequency WHEN dB(V(out))=5 cross=2
.meas phase@2kHz find phase(V(out)) WHEN frequency=2K
```

```
.meas gainpeak max db(v(out)):
     (   8.13428, -0.350754) (at Frequency=954.993)
.meas phase@2khz find phase(v(out)) when frequency=2k:
     (  -165.064,          0)        2000
.meas freq@+5db find frequency when db(v(out))=5 cross=2:
     (   1150.32,          0)     1150.32
```

gainpeak find maximum of V(out) in dB

phase@2kHz find phase
at frequency is 2kHz

Example of .meas in .ac analysis for Q-factor

Qspice : meas - Q of LCR Resonant.qsch

```
.param L=10µ
.param C=1µ
.param R=0.2

.ac dec 1000 1 1G
.options LISTPARAM
.plot V(out)
```

[1] Add this option to display parameter evaluations result in output window

Series RLC:

$$Q = \frac{1}{R}\sqrt{\frac{L}{C}}$$

Q formula of Series RLC

.param Qcal 1/R*(L/C)**0.5

Q calculation from Bandwidth (BW) and Center Frequency (fo) : Q=fo/BW

```
.meas Vmax max mag(V(out))
.meas fo FIND frequency WHEN mag(V(out))=Vmax
.meas fL FIND frequency WHEN mag(V(out))=Vmax/sqrt(2) rise=1
.meas fH FIND frequency WHEN mag(V(out))=Vmax/sqrt(2) fall=last
.meas BW fH-fL
.meas Q fo/BW
```

Output Window

```
  --- Parameter Evaluations ---
TEMP        = 27        "CKTTEMP"
L           = 10µ       "10µ"
C           = 1µ        "1µ"
R           = 200M      "0.2"
QCAL        = 15.8114   "1/R*(L/C)**0.5"

C:\QspiceKSKelvin\01 User Guide and Script\01 Qspice Refer

Total elapsed time: 0.129028 seconds.
```

In simulation, it has .param calculation results

Simulation | Post Process

Output Window

```
.meas vmax max mag(v(out)):
     (  0.999914,          0)  (at Frequency=50350.1)
.meas fo find frequency when mag(v(out))=vmax:
     (   50350.1,          0)     50350.1
.meas fl find frequency when mag(v(out))=vmax/sqrt(2) rise=1:
     (   48762.4,          0)     48762.4
.meas fh find frequency when mag(v(out))=vmax/sqrt(2) fall=last:
     (   51946.8,          0)     51946.8
.meas bw fh-fl:
     (   3184.36,          0)       1e+09
.meas q fo/bw:
     (   15.8117,          0)       1e+09
```

In post process, it has .meas calculation results

Simulation | Post Process

Example of .meas in .ac analysis

Qspice : meas tran demo 01.qsch

pulse 0 10 0 10n 10n 25µ 50µ

.tran 20m
.options MAXSTEP=0.1µ

.mease out@1ms V(out) at=1m

.mease out_steady avg V(out) from 19m to 20m
.mease Tr find Time when V(out)=0.9*out_steady

.meas Pin avg V(in)*-I(V1) from 19m to 20m
.meas Pout avg V(out)*I(R1) from 19m to 20m
.meas %error (Pin-Out)/Pout*100

```
.mease out@1ms v(out) at=1m:
        6.74785       0.001
.mease out_steady avg v(out) from 19m to 20m:
        5.00264
.mease tr find time when v(out)=0.9*out_steady:
     0.000562296 0.000562296
.meas pin avg v(in)*-i(v1) from 19m to 20m:
        5.00497
.meas pout avg v(out)*i(r1) from 19m to 20m:
        5.00528
.meas %error (pin-pout)/pout*100:
     -0.0061396        0.02
```

.meas with trig and targ for time different calculation

Qspice : meas tran demo 02.qsch

.meas a2 trig V(signal)=0.8 cross=2
[time different from 2nd crossing V(signal)=0.8 to end of simulation]

.meas a1 trig V(signal)=0.5
[time different from V(signal)=0.5 to end of simulation]

.meas b1 targ V(signal)=-0.2
[time different from V(signal)=-0.2 to start of simulation]

.meas c1 trig V(signal)=-0.4 rise=1 targ V(signal)=-0.8 fall=2
[time different from 1st rising of V(signal)=-0.4 to 2nd falling of V(signal)=-0.8]

.meas with four (fourier component)

Qspice : meas fourier demo 01.qsch

```
.func rms2pk(in) in*sqrt(2)
```

```
.tran 5/1K
.plot V(sgl)
```

THD Total Harmonic Distortion

```
.four 1K V(sgl)
```

Fourier component with .meas

```
.meas xx four 1K V(sgl)
.meas |xx| abs(xx)
```

THD (.four)

```
.four 1k v(sgl):
 Magnitude of Fundamental(RMS): 0.999922
 Harmonic Frequency     Magnitude    Phase
    1     1.000e+03     1.000e+00     0.00°
    2     2.000e+03     3.498e-08   105.41°
    3     3.000e+03     5.996e-01     0.00°
    4     4.000e+03     1.713e-08    64.78°
    5     5.000e+03     1.996e-01    -0.00°
    6     6.000e+03     1.341e-08    35.02°
    7     7.000e+03     1.292e-06    -8.50°
    8     8.000e+03     2.028e-08   -30.72°
    9     9.000e+03     2.366e-07    -5.73°
 THD = 63.1981%(63.1981%)
```

.meas with four

```
.meas xx four 1k v(sgl):
     (1.54886e-07, -0.999999)
.meas |xx| abs(xx):
       0.999999        0.005
```

Fourier component is a complex number (re+j*im)

Magnitude (rms) can be calculated with abs()

.meas with four (fourier component) [also with .step]

Qspice : meas fourier demo 02.qsch

.step dec param Vamp 100m 10 2
.tran 2m
.plot V(X)

THD Total Harmonic Distortion
.four 1K V(X)
Fourier component with .meas
.meas xx four 1K V(x) format : complex number
.meas |xx| abs(xx) convert complex to magnitude

```
.meas xx four 1k v(x):
   0 (-7.73763e-08,-0.0707086)
   1 (-2.48947e-07, -0.223511)
   2 (-3.44697e-06, -0.551818)
   3 (-7.78738e-06, -0.698485)
   4 ( 2.254e-06, -0.797766)
.meas |xx| abs(xx):
   0  0.0707086      0.002
   1   0.223511      0.002
   2   0.551818      0.002
   3   0.698485      0.002
   4   0.797766      0.002
```

Plot .meas data in Waveform Viewer

Qspice : meas waveform viewer.qsch

Method #1 :

[1] Add ".meas plot [Name]"

Method #2 :

[1] Run Simulation

[2] In Output Window

Right click in Post Process > Plot these Measurements

[3] X-axis default is .step parameter

[4] If you want to display X-axis parameter name

Right click x-axis > add bracket (or curly bracket) to parameter

.meas – FRA : fourier component between OUTPUT and INPUT

Qspice : meas - fra demo 01.qsch

- Syntax : .meas NAME **fra** FREQ INPUT OUTPUT [… range limits …]
  - FRA : Fourier component of OUT at FREQ divided by the Fourier component of IN at FREQ
  - Range limits can be set with from/to or trig/targ syntax
  - Normalization is to the time domain RMS

.meas – FRA : fourier component between OUTPUT and INPUT

Qspice : meas - fra demo 02.qsch

- Frequency response (bode plot) from time domain with use of FRA
  - In this example, .step is used to sweep FRA frequency
  - Time domain simulation is performed at each FRA frequency

fmeas sweep from 100Hz to 10K with 50 points per decade

After .meas is ready in output window
Right click → Plot these Measurements

.option / .options
Set Simulator Options

Set Simulator Options

Set Simulator Options

Syntax: .option NAME1=VALUE1 [NAME2=VALUE2 [...]]

Recognized Options

Name	Description	Default
ABSTOL	Absolute error tolerance	1e-12A
ACCT	Print accounting information	(not set)
ASCII	ASCII .qraw file	(not set)
BINARY	Override command line switch to use ASCII .qraw file	(not set)
BODEAMPFREQ	Frequency with the minimum perturbation amplitude. Set to 0. for constant amplitude.	(not set)
BODEINPUT[1]	Override input node for transfer function computation(AKA BODEIN)	auto
BODEPERIODS	Maximum number of periods to include in deconvolution	20
BODEREF	Reference node to use for Frequency Response Analysis	Node 0 (global ground)
BODEOUTPUT[1]	Override output node for transfer function computation((AKA BODEOUT)	auto
BODETOL	A Frequency Response Analysis relative tolerance	10.
CAPOP	0: Use model value 1: Use Meyer, >1 Use BSIM1	0
CHGTOL	Charge error tolerance	1e-14C
CSHUNT	Capacitance added from every node to ground(aka CMIN)	0F
DEFAD	Default MOSFET area of drain	$0m^2$
DEFAS	Default MOSFET area of source	$0m^2$
DEFL	Default MOSFET length	10µm
DEFW	Default MOSFET width	10µm
FEATHER	Trap integration damping factor	0
GMIN	Minimum conductance	1e-12℧
GMINSTEPS[2]	Number of Gmin steps	10
GSHUNT	Conductance added from every node to ground	0℧
ITL1	DC iteration limit	100
ITL2	DC transfer curve iteration limit	50
ITL4	Transient analysis iteration limit	10
KEEPOPINFO	Record operating point for small-signal analysis	(not set)
KEEPOPINFO	Record operating point for small-signal analysis	(not set)
LAUNCHQUX[3]	Open the .qraw file in the waveform viewer after the simulation	(not set)
LIST[4]	Print an expanded netlist	(not set)
LISTPARAM	Print a list of the evaluated parameters	(not set)
MAXORD	Maximum integration order	2
MAXSTEP	Maximum timestep size for .bode and .tran	infinite
METHOD	Integration method(trap or Gear)	trapezoidal
MINBREAK[5]	Minimum time between breakpoints	0s
NOOPITER	Go directly to Gmin stepping	(not set)
NUMDGT	Number of significant digits in an ASCII .qraw file	15
PIVREL	Minimum relative matrix pivot	1e-3
PIVTOL	Minimum absolute matrix pivot	1e-13
RELTOL	Relative error tolerance	0.1%
RIC[6]	Impedance of source asserting initial conditions	1mΩ
SAVEPOWERS[7]	Compute and save the dissipation of components	(not set)
SEED[8]	Initialize the random number generater used in .param statements	
SEEDCLOCK	Initialize the random number generater with a 10Mhz clock and the process ID number(aka SEEDCLK).	(not set)
SRCSTEPS[2]	Number of source steps(aka ITL6)	10
TEMP	Operating temperature	27°C
TNOM	Nominal temperature(aka TREF)	27°C
TRTOL	Truncation error overestimation factor	2.5
TRTOL2	Another dimensionless truncation error guidance	1e-8
TRYTOCOMPACT	Try compaction for LTRA lines	(not set)
VNTOL	Voltage error tolerance	1µV

1] If a resistor is used to indicate where to insert the perturbation, the resistive divider's contribution is excluded.

2] Since an adaptive step size algorithms are used, the value of GMINSTEPS or SRCSTEPS is irrelevant unless set to zero, which means don't try the stepping algorithm.

3] Useful when running simulations from the command line. Don't use it if QSPICE64.exe or QSPICE80.exe are launched from the GUI.

4] Solely for internal diagnostic purposes. Probably not what you're looking for.

5] MINBREAK is automatically computed if left zero.

6] Inductor currents are asserted with the compliance of 1e9 * RIC.

7] Computes the true power dissipation while ignoring displacement currents. Implemented for BJTs, Capacitors, Diodes, Inductors, JFETs, MOSFET level 1, MOSFET level 2010 and VDMOS.

8] Used in .param functions Random() and Gauss(double sigma).

Simulator Options : CSHUNT and GSHUNT

Qspice : option - CSHUNT.qsch ; option - GSHUNT.qsch

- CSHUNT
  - Capacitance added from every node to ground(aka CMIN)
  - **Default CSHUNT=0F**

- Example to explain
  - Cshunt is equivalent to add Cshunt1/2/3 in node a/b/c

- GSHUNT
  - Conductance added from every node to ground
  - **Default GSHUNT=0℧**

- Example to explain
  - Gshunt is equivalent to add Rshunt1/2/3 = $\frac{1}{\text{GSHUNT}}$ in node a/b/c

Simulator Options : Seed and Seedclock

Qspice : option - Seed Seedclock.qsch

- Seed
  - Initialize the random number generator used in .param statements
  - Same random pattern is generated between Simulation Run

```
.option seedclock
.option seed=719749
```

```
.op
.step param n 0 10 1
.plot V(random)
```

- Seedclock (aka Seedclk)
  - Initialize the random number generator with a 10Mhz clock and the process ID number(aka SEEDCLK)
  - Different random pattern is generated between Simulation Run

```
.option seedclock
.option seed=719749
```

```
.op
.step param n 0 10 1
.plot V(random)
```

Simulator Options : Gmin

Qspice : option - Gmin Diode.qsch

- Gmin
  - Minimum conductance
    - LTspice : Conductivity added to every PNjunction to aid convergence
  - Default Gmin=1e-12℧
- Explanation
  - Upper simulation use Gmin=1e-3 and measure I(V1) profile of D1a/D2a
  - Lower simulation force Gmin=0 (no effect of Gmin) and added R2a/R2b = $\frac{1}{1e-3}$, and measure I(V2) profile of D2a/D2b
  - This example demonstrate Gmin is equivalent to add shunt conductance for every PNjunction

Simulator Options : Gmin

Qspice : option - Gmin L (.dc).qsch ; option - Gmin L (.tran).qsch

- Gmin
  - Minimum conductance
  - In Qspice, Gmin also applied to inductor in .op and .tran initial inductor current calculation
    - Unlike PN junctions, gmin is only applied in inductor for its initial current calculation, but not added during transient analysis

.options Gmin=val

.step dec param val 1e-12 1e-4 10
.plot I(L1)
.op ← DC Operation Point Analysis

.options Gmin=val

.step dec param val 1e-12 1e-4 10
.plot I(L1)
.tran 1µ ← Transient Analysis I_L @ 0s
.meas Iinitial find I(L1) at 0

Simulator Options : RIC

Qspice : option - RIC L.qsch

- RIC
  - Impedance of source asserting initial conditions
  - Inductor currents are asserted with the compliance of 1e9 * RIC
  - Default RIC=1mΩ

- Important note
  - RIC only affect inductor current if .ic is used to define inductor initial current
  - In this simulation example, initial inductor current is plotted with Gmin=1e-12 and Gmin=1e3 with .ic I(L1)=1
    - When RIC=1e-3 (default), initial current is always equal .ic defined value

```
.options Gmin=1e-12
.options RIC=RIC
.step dec param RIC 1e-12 1e-3 10
```

```
.ic I(L1)=1
.plot I(L1)
.tran 1n
.meas Iinitial find I(L1) at 0
```

```
.options Gmin=1e3
.options RIC=RIC
.step dec param RIC 1e-12 1e-3 10
```

```
.ic I(L1)=1
.plot I(L1)
.tran 1n
.meas Iinitial find I(L1) at 0
```

Waveform Viewer

Waveform Viewer Plot Config File (*.pfg) and .plot directive

- Waveform Viewer Config File (*.pfg) and .plot
  - In waveform viewer, plot config can be saved with File > Save Config : [qschname].pfg
    - This config file save windows, traces and axis setting
    - Press spacebar in waveform viewer can re-load config file [qschname].pfg
  - Two unique feature [qschname].pfg can provide but not support by .plot
    - Pre-define x-axis Quantity
    - Pre-define x and y-axis range

Waveform Viewer	Plot Config File [1] [qschname].pfg	.plot command in schematic	Outcome
Closed before Simulation	No	No	A blank waveform viewer
	No	Yes	Plot according to .plot command
	Yes	[ignore]	Plot according to [qschname].pfg config
Opened before Simulation	[ignore]	[ignore]	Keep windows and traces setting from last plot, reset x and y-axis
[1] Save plot config in Waveform Viewer : File > Save Config			
** In Waveform Viewer, press Spacebar to reload [qschname].pfg plot config file			
** to use .plot, delete [qschname].pfg and close waveform viewer before run of simulation			

Data Export in Waveform Viewer – with @ in expression for .step

Qspice : waveform - with @ for step.qsch

```
sin 0 1 fsin

.step param fsin list 40G 80G
.tran 1/fsin
.plot V(sin)
```

- Data Export
- Setup Data Export
  - File > Export Data
  - Number Points : All
  - Expression(s) : V(sin),FSIN

- Data Export with @
- Setup Data Export
  - File > Export Data
  - Number Points : All
  - Expression(s) : V(sin)@1,V(sin)@2

```
Time,V(sin),FSIN
0,0,40000000000
2.001953125e-12,0.482183772079123,40000000000
4.00390624999999e-12,0.844853565249706,40000000000
6.00585937500001e-12,0.998118112900149,40000000000
8.00781250000003e-12,0.903989293123441,40000000000
1.0009765625e-11,0.58579785745643,40000000000
1.20117187500001e-11,0.122410675199201,40000000000
1.40136718750001e-11,-0.371317193951856,40000000000
1.6015625e-11,-0.773010453362737,40000000000
1.80175781249999e-11,-0.983105487431211,40000000000
2.00195312499998e-11,-0.949528180593055,40000000000
2.20214843749997e-11,-0.680600997795516,40000000000
2.40234374999995e-11,-0.242980179903377,40000000000
0,0,80000000000
2.001953125e-12,0.844853565249706,80000000000
4.00390625000001e-12,0.903989293123441,80000000000
6.00585937500003e-12,0.122410675199201,80000000000
8.0078125e-12,-0.773010453362736,80000000000
1.00097656249999e-11,-0.949528180593055,80000000000
1.20117187499998e-11,-0.242980179903377,80000000000
```

```
Time,V(sin)@1,V(sin)@2
0,0,0
2.001953125e-12,0.482183772079123,0.844853565249706
4.00390624999999e-12,0.844853565249706,0.903989293123441
6.00585937500001e-12,0.998118112900149,0.122410675199201
8.00781250000003e-12,0.903989293123441,-0.773010453362736
1.0009765625e-11,0.58579785745643,-0.949528180593
1.20117187500001e-11,0.122410675199201,-0.596254180248215
1.40136718750001e-11,-0.371317193951856,-0.596254180248215
1.6015625e-11,-0.773010453362737,-0.596254180248215
1.80175781249999e-11,-0.983105487431211,-0.596254180248215
2.00195312499998e-11,-0.949528180593055,-0.596254180248215
2.20214843749997e-11,-0.680600997795516,-0.596254180248215
2.40234374999995e-11,-0.242980179903377,-0.596254180248215
0,0,0
2.001953125e-12,0.482183772079122,0.844853565249706
4.00390625000001e-12,0.844853565249707,0.903989293123441
6.00585937500003e-12,0.998118112900148,0.122410675199201
8.0078125e-12,0.903989293123442,-0.773010453362736
1.00097656249999e-11,0.585797857456455,-0.949528180593055
1.20117187499998e-11,0.122410675199269,-0.242980179903377
```

Batch mode

Qspice Execution Files

- Qspice execution files
  - Directory (default installation) : C:\Program Files\QSPICE
  - Schematic Capture and Waveform Viewer Program (HELP > Waveform Viewer)
    - Execution file : QUX.exe
    - Function #1 : Convert .qsch schematic to .cir
    - Function #2 : Export data from data file .qraw
  - QSPICE Simulator (HELP > Simulator)
    - Execution file : QSPICE64.exe [Enable Fast (less accurate) Math]
    - Execution file : QSPICE80.exe
    - Function : Run simulation from .cir
  - Post Processor (HELP > Post Processor)
    - Execution file : QPOST.exe
    - Function : Execute .meas and .four from .qraw

Batch command basic workflow

Qspice : Qspice_Batch_Command_LPFexample.bat / LPF Circuit.qsch

- Batch command workflow
  - Run CMD in Windows, in Command Prompt
  - Set path for Qspice program
    - path C:\Program Files\QSPICE\
  - Set variable name for working folder
    - set Qname=LPF Circuit
    - set Qpath=C:\QspiceKSKelvin\Qspice Batch
  - Goto schematic .qsch directory
    - cd %Qpath%
  - Convert .qsch to .cir (netlist)
    - QUX -Netlist "%Qname%.qsch"
  - Run Qspice simulation for .qraw
    - QSPICE64 -binary "%Qname%.cir"
    - QSPICE64 -ascii "%Qname%.cir" -r "%Qname%-ascii.qraw"
  - Export data from .qraw to .csv
    - QUX -Export "%Qname%.qraw" V(mid),V(out)
  - Post Process .meas and .four
    - QPOST "C:\QspiceKSKelvin\Qspice Batch\%Qname%.cir"

Command Prompt

```
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kelvinleung>path C:\Program Files\QSPICE\

C:\Users\kelvinleung>set Qname=LPF Circuit

C:\Users\kelvinleung>set Qpath=C:\QspiceKSKelvin\Qspice Batch

C:\Users\kelvinleung>cd %Qpath%

C:\QspiceKSKelvin\Qspice Batch>QUX -Netlist "%Qname%.qsch"

C:\QspiceKSKelvin\Qspice Batch>LPF Circuit.cir
C:\QspiceKSKelvin\Qspice Batch>QSPICE64 -binary "%Qname%.cir"
C:\QspiceKSKelvin\Qspice Batch\LPF Circuit.cir

Total elapsed time: 0.0049635 seconds.

C:\QspiceKSKelvin\Qspice Batch>QUX -Export "%Qname%.qraw" V(mid),V(out)

C:\QspiceKSKelvin\Qspice Batch>LPF Circuit.csv
C:\QspiceKSKelvin\Qspice Batch>QPOST "C:\QspiceKSKelvin\Qspice Batch\%Qname%.cir"
.meas fc find frequency when db(mag(v(out)))=-3:
     (    242610,          0)     242610

C:\QspiceKSKelvin\Qspice Batch>
```

This PC > OS (C:) > QspiceKSKelvin > Qspice Batch

Name

- LPF Circuit.cir
- LPF Circuit.csv
- LPF Circuit.qraw
- LPF Circuit.qsch
- Qspice_Batch_Command_LPFexample.bat

QUX.exe : Netlist a Schematic (.qsch)

- Syntax for QUX buildtimestamp
  - QUX.exe -buildtimestamp

```
C:\Program Files\QSPICE>QUX.exe -buildtimestamp

C:\Program Files\QSPICE>Build Nov  3 2023 09:11:08
```

- Syntax for -Netlist
  - QUX.exe -Netlist <schematicfile> [-stdout]
    - <schematicfile> : name (+path) of a .qsch schematic, adds " " quotation for filename
    - If "-stdout" is not specified, the name of the netlist(.cir) file is computed from the name of the input .qsch file
    - [-stdout] : the netlist is printed on the console instead of to a file (not recommended since QSPICE employs a character set that most terminals can't handle

```
C:\QspiceKSKelvin\Qspice Batch>QUX -Netlist "%Qname%.qsch" -stdout

C:\QspiceKSKelvin\Qspice Batch>* LPF Circuit.qsch
L1 in mid 1
C1 mid 0 1
R1 out 0 1
V1 in 0 AC 1
L2 mid out 1
.ac dec 100 10K 1Meg
.plot V(mid) V(out)
.MEAS fc FIND frequency WHEN db(mag(V(out)))=-3
.end
```

QUX.exe : Export Datafile (.qraw)

- Syntax for -Export
  - QUX.exe -Export <datafile> <expr1[,expr2[,...]]> [Npoints] [CSV|SPICE|ASCII] [-stdout]
    - <datafile> : name of a .qraw file
    - <expr1[,expr2[,...]]> : expressions of data to extract
      - No space are allowed in the expression
      - Comma-separated expressions
    - [Npoints] : number of equally-spaced data points to extract
      - Default Npoints=1000
      - Npoints=1e308 or Npoints="all" : all datapoints are extracted, waveform is not interpolated
    - [CSV|SPICE|ASCII]
      - CSV : Comma-Separated Value file
      - SPICE : .qraw in binary
      - ASCII : .qraw in ASCII
    - [-stdout] : extracted data is printed on the console instead of to a file

QSPICE64.exe and QSPICE80.exe : QSPICE Simulator

- Syntax for output data .qraw name same as netlist .cir name
  - QSPICE64.exe -binary <netlistname> : Binary file format for output data .qraw
  - QSPICE64.exe -ascii <netlistname> : Ascii file format for output data .qraw

- Synatx for specify output data .qraw name
  - QSPICE64.exe -[ascii/binary] <netlistname> -r <path> : specify the name of output data file
  - Example
    - set Qname=LPF Circuit
    - QSPICE64 -ascii "%Qname%.cir" -r "%Qname%-ascii.qraw"

.qraw Binary Data format

Binary-binary.qraw

```
Title: * Binary.qsch
Date: Sun Nov  5 21:41:33 2023
Plotname: DC Transfer Characteristic
Flags: real
Abscissa: 1.000000000000000e+00 5.000000000000000e+00 lin
No. Variables: 4
No. Points: 3
Command: QSPICE64, Build Nov  3 2023 09:29:29
.param temp=27
Variables:
    0   V1  voltage
    1   V(a)    voltage
    2   I(V1)   current
    3   P(V1)   power
Binary:
NUL NUL NUL NUL NUL NUL ð? NUL NUL NUL NUL NUL NUL ð? NUL NUL NUL NUL NUL
```

Binary vs Ascii

Binary-binary.qraw | Binary-ascii.qraw

```
Title: * Binary.qsch
Date: Sun Nov  5 21:41:38 2023
Plotname: DC Transfer Characteristic
Flags: real
Abscissa: 1.000000000000000e+00 5.000000000000000e+00 lin
No. Variables: 4
No. Points: 3
Command: QSPICE64, Build Nov  3 2023 09:29:29
.param temp=27
Variables:
    0   V1  voltage
    1   V(a)    voltage
    2   I(V1)   current
    3   P(V1)   power
Values:
0       1.000000000000000e+00
    1.000000000000000e+00
    0.000000000000000e+00
    0.000000000000000e+00
1       3.000000000000000e+00
    3.000000000000000e+00
    0.000000000000000e+00
    0.000000000000000e+00
```

Newline </n>

```
00000130 76 6f 6c 74 61 67 65 0a 09 32 09 49 28 56 31 29 voltage..2.I(V1)
00000140 09 63 75 72 72 65 6e 74 0a 09 33 09 50 28 56 31 .current..3.P(V1
00000150 29 09 70 6f 77 65 72 0a 42 69 6e 61 72 79 3a 0a ).power.Binary:.
00000160 00 00 00 00 00 00 f0 3f 00 00 00 00 00 00 f0 3f ......ð?......ð?
00000170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000180 00 00 00 00 00 00 08 40 00 00 00 00 00 00 08 40 .......@.......@
00000190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000001a0 00 00 00 00 00 00 14 40 00 00 00 00 00 00 14 40 .......@.......@
000001b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

Binary Format : Float 64