Qspice - Device Reference Guide by KSKelvin

KSKelvin Kelvin Leung

Created on 9-5-2023
Last Updated on 2-27-2024

B. Behavioral Source

B. Behavioral Source : Instance Parameters

Behavioral Source Instance Parameters

Name	Description	Units	Default
CPAR	Parallel capacitance	F	0.0
I[1]	Expression for current	A	(not set)
IC	Initial value	A or V	0.0
LAPLACE	Frequency dependence		(none)
NORTON	Use a Norton equivalent for a behavioral resistance		(not set)
R[1]	Expression for resistance	Ω	(not set)
RPAR	Parallel resistance Ignore in V	Ω	Infinite
RSER	Series resistance Ignore in I and R (Norton)	Ω	0.0
SYNTHESIZE	Use the synthesized Laplacian circuit even for .AC		(not set)
THEVENIN	Use the Thévenin equivalent for a behavioral resistance		(not set)
V[1]	Expression for voltage	V	(not set)

B. Instance Params : Rser, Rpar and Cpar

Qspice : B Source - Rpar Rser Cpar in V and I.qsch

- Rser, Rpar and Cpar in Behavioral Voltage and Current Source
  - Behavioral Voltage (V=<expression>) : Only Rser and Cpar are active
  - Behavioral Current (I=<expression>) : Only Rpar and Cpar are active
    - I don't recommend to use Rser and Rpar in Behavioral Resistance as it will be very confusing depends on Norton or Thevenin equivalent
  - In this example, IC initial value is used to allow 0V or 0A for transient simulation. As behavioral source not have effect in .ac (except for Laplace), .tran is used to reveal these parameters characteristic

B. Instance Params : Behavioral Resistor

Qspice : B Source - Behavioral R.qsch

- Behavioral Resistor
  - Syntax : Bnnn n001 n002 R=<expression>
    - Expression of resistance

B. Instance Params : Laplace

Qspice : B Source - Laplace (.tran).qsch | B Source - Laplace (.ac).qsch

- Laplace (.tran)
  - Laplace equation in B-source, by adding a new attribute Laplace=formula
  - In this simulation
    - $v_{ctrl} = \left(K_p + \frac{K_i}{s} + K_d s\right) v_{err}$
  - Laplace B-source support both transient (.tran) and ac (.ac) analysis

B-source with Laplace in .tran analysis

.tran 5/1K
.plot V(ctrl)
.plot V(err)

.param Kp=1
.param Ki=2*pi*1K
.param Kd=1/2/pi/1K

B-source with Laplace in .ac analysis

.ac dec 100 1 1Meg
.plot V(ctrl)
.plot V(err)

.param Kp=1
.param Ki=2*pi*1K
.param Kd=1/2/pi/1K

B. Function and Operators

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)	$dx/d\mathrm{time}$	Time derivitive x
delay(x,y)		x delayed by y
delay(x,y,z)		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)	$\int x\, d\mathrm{time} + y$	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

Available Variable or Constant not listed in HELP

- Time : Simulation Time
- Temp : Temperature
- PI : 3.14159265358979323846
- E : 2.7182818284590452354
- K : 1.380649e-23 J/°K
- Q : 1.602176487e-19 Coulomb

B. Function and Operators – delay(x,y) or delay(x,y,z)
Qspice : B Source - delay fix.qsch

- delay(x,y) or delay(x,y,z)
  - Delay(x,y,z) can reduce the amount of waveform memory that must be stored

delay(x,y)	x delayed by y
delay(x,y,z)[1]	x delayed by y, but store no more than z history

  - y can be parameter and z must be fixed constant
  - y must less than z or error will return

Syntax : delay(x,y) - x delayed by y
Syntax : delay(x,y,z) - x delayed by y, but store no more than z history

```
.param f = 100K
.param Td = 2µ
.param maxTd = 5µ
```

```
.tran 2/f
.plot V(sgl) V(delay)
.plot V(sgl) V(delay_w/maxTd)
```

B. Function and Operators – table(x,a,b,c,d,...)

Qspice : B Source - table.qsch | B Source - table(.func).qsch | B Source - table(attribute).qsch

- Table(x,a,b,c,d,...)
  - Interpolate x from the look-up table given as a set of pairs of constant values
- Setup a table
  - As table may contain a lot of data points, for visual purpose in schematic, user can consider
    - Each pair of (x,y) data uses a bracket
    - Break table with 2nd or more attribute
    - Use .func to define a table

Basic table setup in B-source

```
.dc V1 0 5 0.1
.plot V(sgl) V(table)
```

Define a table with 2nd or more attribute

```
.dc V1 0 5 0.1
.plot V(sgl) V(table)
```

Define a table with .func and with + line break

```
.dc V1 0 5 0.1
.plot V(sgl) V(table)

.func userTable(x) table(x,(1,1),(2,4),
+(3,9),(4,16))
```

B. Function and Operators – Integral idt(x,y,z)

Qspice : B Source - idt.qsch | B Source - idt (y-conditional).qsch

- idt(x,y,z)
  - Time Integral of x
  - $= \int x\,dt + y$
  - y is initial condition, which is the DC value during integral function is reset
  - z is to reset integral
  - Special technique for initial condition
    - If initial condition needs to be changed over time, conditional function can be used
    - User can also use voltage source to control initial voltage y over time

Syntax : idt(x,y,z)
Time integral of x with initial condtion of y reset when z > .5

Syntax : idt(x,y,z)
Time integral of x with initial condtion of y reset when z > .5

B. Function and Operators – Time Derivative ddt(x) and limit(x,y,z)

Qspice : B Source - ddt.qsch | B Source - limit.qsch

- ddt(x)
  - Time derivative
  - $= \frac{dx}{dt}$

Synatx : ddt(x)
Time derivative x

Formula
d(sin(omega*t))/dt = omega*cos(omega*t)

```
.param f=1/2/pi
.param omega=2*pi*f

.tran 5/f
.plot V(sin) V(dsin)
```

- limit(x,y,z)
  - Intermediate value of x, y, and z

```
.tran 2m
.plot V(sgl) V(limit)
```

B. Function and Operators – dlim(x,y,z) and ulim(x,y,z)

Qspice : B Source - dlim ulim.qsch

- dlim(x,y,z) and ulim(x,y,z)
  - dlim(x,y,z) : x bounded by y which it asymptotically starts to approach at y+z as a first inverse order Laurent series
  - ulim(x,y,z) : x bounded by y which it asymptotically starts to approach at y-z as a first inverse order Laurent series
  - These functions are used for soft limit

B. Function and Operators – random(x) [Basic Concept]

Qspice : B Source - random (seed).qsch

- Random(x)
  - Random(x) : Random number from 0 to 1 depending on the integer value of x
  - This is pseudo random function which takes a seed and output a number "appears random"
- Random()
  - Seed is determined by Qspice
  - If .option seed=<value> is used, user can assign an initial seed
  - If .option seedclock is used, seed is from 10MHz clock and process ID, which appear random in every run
- Random(x)
  - Seed is set by user
  - Assign a fixed value (e.g. .param), or varying value (from .step param)
- These two examples run same results every time in Qspice as initial seed not be changed in each run

B. Function and Operators – random(x) [.option seedclock]

Qspice : B Source - random (seed).qsch

- Random(x)
  - .option seedclock
  - Initialize the random number generater with a 10Mhz clock and the process ID number
  - If seedclock is used, initial seed are different in every run
  - random() give different random pattern in every run
  - But random(x) with user defined seed and not be impacted

.option seedclock

B. Function and Operators – random(x) [Random in .tran timestep]

Qspice : B Source – random (.tran).qsch

- Random(x)
  - Random number from 0 to 1 depending on the integer value of x
  - The one without arguments must be processed in the preprocessor (from Mike Engelhardt)
  - To create a random sequence in transient, x must change in each time step : i.e. random(1e8*time), multiple time with 1e8 is for x to be a large integer for random generation
  - For random run in each step, function can be
    - random(1e8*(time+random()))

With **random(1e8*(time+random()))**, it allows random voltage also at 0s

B. Function and Operators – AbortSim(x), variable/constant

Qspice : B Source - AbortSim.qsch ; B Source - Buildin Variable Constant.qsch

- AboutSim(x)
  - To stop the current simulation from a behavioral source
  - If x > 0.5 : *abort simulation*
    else : *return x*
  - The function works for .dc and .tran
  - Refer to Ø-Device, MaxExtStepSize() for same feature but in .dll

syntax : AbortSim(double arg)
if arg > 0.5 : stops simulation and goes to next step (if any)
else : returns the argument

.tran 4
.step param nix 0 1 0.2

- Build-in Variable / Constant
  - Time : Simulation Time
  - Temp : Temperature
  - PI : 3.14159265358979323846
  - E : 2.7182818284590452354
  - K : 1.380649e-23 J/°K
  - Q : 1.602176487e-19 Coulomb

.tran 1

C. Capacitor

C. Capacitor Instance Parameters in Qspice HELP

Capacitor Instance Parameters

Name	Description	Units	Default
CAPACITANCE	Capacitance(not usually labeled)	F	0.0
CSAT	Capacitance asymptotically approached in saturation	F	10% of CAPACITANCE
IC	Initial voltage on capacitor	V	0.0
L[1]	Length	m	0.0
M	Number of identical parallel devices		1.0
RPAR	Equivalent parallel resistance	Ω	Infinite
RSER	Equivalent series resistance	Ω	0.0
SATFRAC	Fractional drop in capacitance at VSAT		0.7
TC	Polynomial temperature coefficient list		0.0
TC1	1st order temperature coefficient	$°C^{-1}$	0.0
TC2	2nd order temperature coefficient	$°C^{-2}$	0.0
TEMP	Instance temperature	°C	Circuit temperature
TNOM	Temperature capacitance was measured(aka TREF)	°C	Circuit TNOM
VSAT	Voltage that drops capacitance to SATFRAC×CAPACITANCE	V	Infinite
W[1]	Width	m	0.0

Rpar and Rser in C model

C. Instance Params : IC Initial Condition and TC Temp Coefficient

Qspice : Capacitor - IC.qsch

- IC
  - IC : Initial Condition
  - Default IC=0V
  - Use in transient analysis for initial voltage of capacitor at time=0s

C. Instance Params : Vsat, Csat, Satfrac : For Nonlinear Capacitance

Qspice : C Capacitor - Vsat (.ac).qsch ; C Capacitor - Csat (.tran).qsch

- Vsat and Satfrac
  - Vsat : Voltage that drops capacitance to ***Satfrac x Capacitance***
  - Satfrac : Fractional drop in capacitance at Vsat
  - **Default Vsat=Infinite (V)**
  - **Default Satfrac=0.7**

```
.param freq=1K
.step param bias 1p 100 1        Sweep bias
.step param var list 10 20 30    Sweep Vsat

.tran 1/freq

Calculate equivalent capacitance in .tran analysis
.meas Ip max I(C1)
.meas Ceqv Ip/1/2/pi/freq
.meas plot Ceqv
```

- Csat
  - Csat : Capacitance asymptotically approached in saturation
    - ** Vsat must be set to see it effect
  - **Default Csat=10% of C**

```
.param freq=1K
.step param bias 1p 100 1        Sweep bias
.step param var list 0.4µ 0.2µ   Sweep Csat

.tran 1/freq

Calculate equivalent capacitance in .tran analysis
.meas Ip max I(C1)
.meas Ceqv Ip/1/2/pi/freq
.meas plot Ceqv
```

C. Instance Params : Replicate previous graph with .ac analysis

Qspice : C Capacitor - Vsat (.ac).qsch

- Vsat and Satfrac
  - Vsat : Voltage that drops capacitance to ***Satfrac x Capacitance***
  - Satfrac : Fractional drop in capacitance at Vsat
  - **Default Vsat=Infinite (V)**
  - **Default Satfrac=0.7**

- Capacitance equation
  - $X_C = \frac{1}{\omega C} = \frac{V_c}{I_c}$
  - $C = \frac{|I_c|}{|V_c|}\frac{1}{\omega} = \frac{|I_c|}{2\pi f|V_c|}$
  - This formula is used to calculate equivalent capacitance (small signal model) at different capacitor bias voltage

```
.param frq=1Meg
.step param bias 0 100 1          Sweep bias
.step param var list 10 20 30     Sweep Vsat

.ac list frq   ← in .ac analysis
.func Ceqv() I(V1)/1/2/pi/frq
.plot Ceqv()
```

Bias Voltage from 0V to 100V

C. Instance Params : Thermal Parameter

- Thermal Equation of Capacitor
  - $C = C_{nom} \times (1 + \sum \mathrm{TC}_n \, (T - T_{nom})^n)$
    - Where T is operating temperature TEMP, which uses as *.temp*, *.dc temp* or *.step param temp*
  - Thermal Instance Parameter and equation relationship
    - TC : Polynomial of Thermal Coefficient where TC=<TC1>,<TC2>,<TC3>,<TC4>,…
      - TCn = TC_n
    - TC1, TC2 : 1st and 2nd order Thermal Coefficient
      - TC1 = TC_1 and TC2 = TC_2
  - TEMP : Instance Temperature (Simulation Temperature of this capacitor)
  - TNOM : Temperature Capacitance was measured
    - C_{nom} : Capacitance C @ TNOM

C. Instance Params : TC, TC1 and TC2

Qspice : C Capacitor - TC.qsch | C Capacitor - TC1 TC2.qsch

- TC
  - TC : Temperature Coefficient list
  - **Default TC=0**
  - TC can be defined as TC=<value1>,<value2>,...
  - Formula
  - $C = C_{nom}(1 + \sum \mathrm{TC_n}(T - T_{nom})^n)$
  - In this example, Default Tnom is used, which is 27°C.

```
.step param TEMP -55 125 1
.param frq=1Meg
.ac list frq
.func imZ() V(C)/I(C1)
.func C() -1/2/pi/frq/imZ()
.plot C() Ccal()
.func Ccal() 1µ*(1+1e-3*(TEMP-27)+1e-7*(TEMP-27)^3)
```

- TC1 and TC2
  - TC1 : 1st order temp coeff
  - TC2 : 2nd order temp coeff
  - $C = C_{nom}(1 + \mathrm{TC_1}(T - T_{nom}) + \mathrm{TC_2}(T - T_{nom})^2)$

```
.step param TEMP -55 125 1
.param frq=1Meg
.ac list frq
.func imZ() V(C)/I(C1)
.func C() -1/2/pi/frq/imZ()
.plot C() Ccal()
.func Ccal() 1µ*(1+1e-3*(TEMP-27)+1e-5*(TEMP-27)^2)
```

C. Instance Params : Tnom and TEMP

Qspice : C Capacitor - Tnom.qsch | C Capacitor - Temp.qsch

- Tnom
  - Nominal temperature
  - **Default Tnom=27°C**
  - Capacitance Cnom in thermal equation is measured at Tnom (i.e. at Tnom, calculated capacitance equal model capacitance)

```
.step param TEMP 0 80 1
.step param Tnom list 20 60
.param frq=1Meg
.ac list frq
.func imZ() V(C)/I(C1)
.func C() -1/2/pi/frq/imZ()
.plot C()
```

- TEMP
  - Instance temperature
  - **Default TEMP=Circuit Temp**
  - With TEMP, capacitor is fixed at this temperature in regardless change of circuit temperature

```
.step param TEMP 20 80 1
.param frq=1Meg
.ac list frq
.func imZ1() V(C)/I(C1)
.func imZ2() V(C)/I(C2)
.func C1() -1/2/pi/frq/imZ1()
.func C2() -1/2/pi/frq/imZ2()
.plot C1() C2()
```

C. Capacitor – Behavioral Capacitance (C)

Qspice : C Capacitor - Behavioral C.qsch

- C
  - Behavioral Capacitance
  - **Format : C=<equation>**
  - Qspice supports behavioral capacitance for a given equation which can include any circuit node voltage or device current

Arbitrary Capacitance-Based Capacitance Device

QSPICE also supports a behavioral capacitance where an equation for charge is not available. To use it, give an equation for the capacitance is given. The equation can include any circuit node voltages or device currents of devices modeled as Thévenin equivalents(V-, L-, E-, or H-devices). An original numerical method is used to track the charge on the capacitance not before seen in SPICE programs.

Syntax: Cnnn N1 N2 C=<expression> [additional instance parameters]

Name	Description	Units	Default
C	Equation of charge	Coulomb	
IC	Initial voltage on capacitor	V	0.0
M	Number of identical parallel devices		1.0
RPAR	Equivalent parallel resistance	Ω	Infinite
NOISELESS	Ignore the noise contribution from RPAR		not set
TEMP	Instance temperature	°C	Circuit temperature

Behavioral Capacitor Model with C=<eqn>

```
.step param Vb 0.9 1.1 0.01
.param frq=1Meg
.ac list frq
.func C() -1/2/pi/frq/(im(V(C)/I(C1)))
.plot C()
```

C. Capacitor – Behavioral Capacitance (C)

Qspice : C Capacitor - Behavioral C (.tran).qsch

- C
  - Behavioral Capacitance
  - This schematic demonstrate using behavioral capacitance model in transient (.tran) analysis

Behavioral Capacitor model with C=<eqn>

```
.param frq=1K
.param Xc=1
.param Cinit=1/2/pi/frq/Xc
```

```
.tran 50m
.option maxstep=1/frq/100
.plot I(C1)
.plot V(C)
.plot V(bias)
```

C. Capacitor – Why Qspice not include parasitic inductance (ESL or Lser)

- Answer by Mike Engelhardt in Qspice forum
  - https://forum.qorvo.com/t/parasitic-inductance-of-passive-components-lser/15608/3
  - https://forum.qorvo.com/t/esl-in-capacitor-model/16944/2

Engelhardt Oct '23

I did not implement Lser for capacitors because it usually caused more harm than good.

A series inductance is not a great way to model the internal inductance of a capacitor. For foil capacitors, you really need a ladder of series R & L driving a distributed capacitance, though I've been able to match the complex impedance within a few degrees over several decades of frequency with just two lumps.

For ceramic capacitors, modeling their internal inductance is not really meaningful without modeling the parasitics of the PCB.

Anyway, when I watch people using Lser, it only increases simulation times without actually improving the integrity of the simulation.

Your mileage will vary,

–Mike

For any type of film capacitor, it is more of a distributed Rser/Rpar/Lser/Cpar network. Poorly described with a single lumped series L. For a ceramic capacitor, a series inductance can be a more viable representation, but usually adds no accuracy to the simulation since connecting inductance often dominates. It was a mistake to support Lser as a part of a capacitor in LTspice. It did much more harm than good. Anyway, below would be a viable equivalent schematic of a film capacitor in lumped constants between nodes A and B:

Notice there is nowhere a series inductance.

D. Diode

Model Parameters

Diode Model Parameters in Qspice HELP

Diode Instance Parameters

Name	Description	Units	Default
AREA	Relative area		1.0
M	Number of identical parallel devices		1.0
TEMP	Instance temperature	°C	Circuit temperature

Diode Model Parameters

Name	Description	Units	Default
AF	Flicker noise exponent	1.	1.0
BV	Reverse breakdown voltage	V	Infinite
CJO	Zero-bias junction capacitance	F	0.0
EG	Activation energy	eV	1.11
FC	Forward-bias depletion capacitance coefficient		0.5
GMAX	Maximum conductivity(straight-line extension)	℧	10000.
GP	Parallel conductivity added in lieu of global Gmin	℧	0.
IBV	Current at breakdown voltage	A	1e-10
IBVL	Low-level reverse breakdown knee current	A	0.0
IKF	High-injection knee current	A	1e308
IS	Saturation current	A	1e-14
ISR	Recombination current parameter	A	0.0
KF	Flicker noise coefficient		0.0
M	Grading coefficient		0.5
N	Emission coefficient		1.0
NBV	Reverse breakdown emission coefficient		1.0
NBVL	Low-level reverse breakdown emission coefficient		1.0
NR	ISR emission coefficient		2.0
RS	Series resistance	Ω	0.0
TBV1	1st order BV temperature coefficient	$°C^{-1}$	0.0
TBV2	2nd order BV temperature coefficient	$°C^{-2}$	0.0
TIKF	IKF temp coefficient	$°C^{-1}$	0.0
TNOM	Parameter measurement temperature(aka TREF)	°C	27
TG11	1st order GP temperature coefficient[1]	$°C^{-1}$	0.0
TGP2	2nd order GP temperature coefficient[1]	$°C^{-2}$	0.0
TRS1	1st order RS temperature coefficient[1]	$°C^{-1}$	0.0
TRS2	2nd order RS temperature coefficient[1]	$°C^{-2}$	0.0
TT	Transit-time[2]	sec	0.0
VJ	Junction potential	V	1.0
XTI	Saturation current temperature exponent		3.0

.model available parameters but not in HELP

- VP : Soft reverse recovery parameter (Default VP=0.01)

.model parameters only for information display but no electrical behavior

- MFG : manufacturer name
- Vrev : Peak reverse voltage
- Iave : Average current rating

D. Diode Model (I-V Characteristic) : IS

Qspice : Dmodel - IS.qsch

- IS
  - Is : Saturation current
    - Saturation current in reverse region
  - **Default IS = 1e-14A**
  - $I_D = I_s\left(e^{\frac{qV_D}{nkT}} - 1\right)$
    - $\frac{kT}{q}$ is called thermal voltage
      - q : electronic charge
        - 1.602176487e-19 Coulomb
      - k : Boltzmann constant
        - 1.380649e-23 J/°K
      - T : Temperature in Kelvin
        - $T = T_{celsius} + 273.15°C$
    - V_D : Diode voltage
    - n : emission coefficient

```
.param Vmin = -2
.param Vmax = 0.5
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)
.plot abs(I(D1)) ; for log scale
.step param Is list 1n 10n 100n
.model Dm D Is=Is
```

$|I_d|$
Magnitude only

D. Diode Model (I-V Characteristic) : N and Rs

Qspice : Dmodel - N.qsch / Dmodel - Rs.qsch

- N
  - n : emission coefficient
  - **Default N=1**
  - $I_D = I_s\left(e^{\frac{qV_D}{nkT}} - 1\right)$
  - N affects threshold voltage in forward bias

```
.param Vmin = 0
.param Vmax = 1
.step param n list 0.5 1 1.5
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)
.model Dm D Is=10n N=n
```

- RS
  - Rs : Series resistance
  - **Default RS=0Ω**
  - This is equivalent to resistor added in series to diode
  - ** V_D in $I_D = I_s\left(e^{\frac{qV_D}{nkT}} - 1\right)$ formula only represent diode model voltage drop. If RS is added, diode model anode to cathode voltage is $V_D + I_D R_S$

```
.param Vmin = -5
.param Vmax = 5
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D2)
.plot I(D1)
.model DRs D Rs=0.29
```

Forward Voltage (VF) or Threshold @ Specified Forward Current (IF)

** Condition : Recombination current is not used : ISR=0

- Background
  - In datasheet, it generally specify forward threshold by defining as forward voltage drop (VF) at a specified forward current level (IF)

- Diode model include RS
  - $V_{D,ext} = V_D + I_D R_S$ and $I_d = I_s\left(e^{\frac{qV_D}{nkT}} - 1\right)$
    - where $V_{D,ext}$ is diode voltage drop including series resistance
  - Re-arrange I_d formula : $V_D = n\,\frac{kT}{q}\,\ln\left(\frac{I_d}{I_S} + 1\right)$
  - Therefore,
    - $V_{D,ext} = n\,\frac{kT}{q}\,\ln\left(\frac{I_d}{I_S} + 1\right) + I_D R_S$
  - Substitute $V_{D,ext}$ by VF and I_D by IF and Qspice .model parameters
    - $\mathrm{V_F} = \mathrm{N}\,\frac{kT}{q}\ln\left(\frac{\mathrm{IF}}{\mathrm{IS}} + 1\right) + \mathrm{IF} \times \mathrm{Rs}$
      - where Qspice default temperature T is 27°C = 300.15K
      - k=1.380649e-23 J/°K and q=1.602176487e-19C → $\frac{kT}{q}$ = 2.5865e-2
      - Equation can be simplified with IF ≫ IS where $\left(\frac{\mathrm{IF}}{\mathrm{IS}} + 1\right) \rightarrow \frac{\mathrm{IF}}{\mathrm{IS}}$

- Simplified Results in Qspice
  - VF = 2.5865e-2***N***ln(**IF**/**IS**) + **IF** * **RS**
    - IS, N and RS are in diode .model
    - IF is specified forward current for diode threshold (e.g. IF=100mA)

Approximation formula to calculate Vd threshold at IF, @27oC
.param VF=2.5865e-2*N*ln(IF/IS)+IF*RS

Model Parameters
.param IS=5n
.param N=2
.param Rs=.58

IF forward Current
.param IF=100m

.dc V1 0 1 0.01
.plot I(D1)
.options listparam

.model 1N4148 D Is=5n Rs=.58 n=2
+Cjo=.87p m=.025 tt=7.9n Nbv=2000
+BV=100 IBV=150n XTI=6
+mfg="ON Semiconductor"
+Vrev=100 Iave=200m

Output Window

```
TEMP      = 27        "CKTTEMP"
IS        = 5N        "5N"
N         = 2         "2"
RS        = 580M      ".58"
VF        = 927.646M  "2.5865E-2*N*LN(IF/IS)+IF*RS"
IF        = 100M      "100M"
```

D. Diode Model (I-V Characteristic : Breakdown) : BV, IBV and NBV

Qspice : Dmodel - BV IBV.qsch / Dmodel - NBV.qsch

- BV and IBV
  - BV : Breakdown Voltage
  - IBV : Current at breakdown voltage
  - **Default BV=Infinite**
  - **Default IBV=1e-10**
  - Breakdown region eqn
    - $I_D = -\mathrm{IBV}\, e^{-\frac{q(\mathrm{BV}+V_D)}{kT}}$

```
.param Vmin = -1.2
.param Vmax = 0
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)
.model Dm D BV=1 IBV=1
```

- NBV
  - NBV : Reverse breakdown emission coefficient
  - **Default NBV=1**
  - Change the sharpness of breakdown reverse current

```
.param Vmin = -1.2
.param Vmax = 0
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)
.model Dm D BV=1 IBV=1 NBV=nbv
.step param nbv list 0.5 1 2
```

D. Diode Model (I-V Characteristic : Breakdown) : IBVL and NBVL

Qspice : Dmodel - IBVL.qsch / Dmodel - NBVL.qsch

- IBVL
  - IBVL : low-level reverse breakdown knee current
  - **Default IBVL=1**

```
.param Vmin = -1.2
.param Vmax = -0.6
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)
.step param ibvl list 0 4
.model Dm D BV=1 IBV=1 IBVL=ibvl
```

- NBVL
  - NBVL : Low-level reverse breakdown emission coefficient
  - **Default NBVL=1**
  - Change the sharpness of breakdown reverse current when IBVL is used

```
.param Vmin = -1.2
.param Vmax = -0.6
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)
.step param nbvl list 0.5 1 5
.model Dm D BV=1 IBV=1 IBVL=4
+NBVL=nbvl
```

D. Diode Model (Capacitance) : CJO and M

Qspice : Dmodel - CJO.qsch / Dmodel - M.qsch

- CJO
  - Cjo : Zero-bias junction capacitance
  - **Default CJO=0F**
  - $C_j = \frac{C_{jo}}{\left(1-\frac{V_D}{\Phi_0}\right)^m}$
    - Φ_0 : Junction potential (VJ)
    - m : Grading coefficient
      - M = 0.33 : linearly garded
      - M = 0.5 : abrupt junction

```
.param Vmin = -0.2
.param Vmax = 0.345
.param f = 1K
```

```
.model Dm D Cjo=100p
.func Zim() imag(V(anode)/I(D1))
.func Ceqv() -1/2/pi/f/Zim()

.ac list f
.step param var Vmin Vmax 0.01
.plot Ceqv()
```

- M
  - m : grading coefficient
  - **Default M=0.5**
  - $C_j = \frac{C_{jo}}{\left(1-\frac{V_D}{\Phi_0}\right)^m}$
    - m : Grading coefficient
      - M = 0.33 : linearly garded
      - M = 0.5 : abrupt junction

```
.param Vmin = -0.2
.param Vmax = 0.345
.param f = 1K
```

```
.model Dm D Cjo=100p m=m
.func Zim() imag(V(anode)/I(D1))
.func Ceqv() -1/2/pi/f/Zim()

.ac list f
.step param var Vmin Vmax 0.01
.plot Ceqv()
.step param m list 0.3 0.5 0.7
```

D. Diode Model (Capacitance) : VJ and FC

Qspice : Dmodel - VJ.qsch / Dmodel - FC.qsch

- VJ
  - Vj : Junction potential
  - **Default VJ=1V**
  - $C_j = \frac{C_{jo}}{\left(1-\frac{V_D}{\Phi_0}\right)^m}$
    - Φ_0 : Junction potential (VJ)
      - may range from 0.2 to 1V

```
.param Vmin = -0.2
.param Vmax = 0.345
.param f = 1K
```

```
.model Dm D Cjo=100p VJ=vj
.func Zim() imag(V(anode)/I(D1))
.func Ceqv() -1/2/pi/f/Zim()

.ac list f
.step param var Vmin Vmax 0.01
.plot Ceqv()
.step param vj list 0.2 0.5 1
```

- FC
  - Fc : Forward-bias depletion capacitance coefficient
  - **Default FC=0.5**
  - A factor between 0 and 0.95 (limit by 0.95 in Qspice), which determines how the junction capacitance is calculated when the junction is forward-biased

```
.param Vmin = -0.2
.param Vmax = 0.345
.param f = 1K
```

```
.model Dm D Cjo=100p FC=fc
.func Zim() imag(V(anode)/I(D1))
.func Ceqv() -1/2/pi/f/Zim()

.ac list f
.step param var Vmin Vmax 0.01
.plot Ceqv()
.step param fc list 0.1 0.2 0.95
```

D. Diode Model (Capacitance) : Parameters can affect Cj in frequency

Qspice : Dmodel - Cj Effect – IS.qsch

- Params can affect Cj in frequency
  - IS : Saturation current

```
.param Vmin = -0.2
.param Vmax = 0.345
.param f = 1K
```

```
.ac list f
.step dec param f 1 1G 10
.plot Ceqv()
.model Dm D Cjo=100p Is=Is
.step param Is list 1e-14 1e-9 1e-7
```

D. Diode Model (Reverse Recovery) : TT

Qspice : Dmodel - TT.qsch

- TT : Transit-time
  - TT is not the time it takes a diode to turn off. It's a measure of the amount of charge that needs to be pulled out to turn the diode off. Specifically, the stored charge is TT times the forward current. For the product to be Coulomb, the dimensions of TT must be time
  - $Q_{store} = I_{fwd} \times tt$, where tt is transit-time in Qspice
  - **Default TT=0s** [equivalent to disable reverse recovery from TT]
    - ** If TT is used, Rs must be non-zero (it can affect forward I-V characteristic if Rs=0 with a finite TT

```
.param vfwd=0.834
.param vrev=-10
.param td=0.1n
.param p=1p
```

```
.tran 10*td
.option maxstep=p/50
.plot V(Qrev)
.plot I(D1)
.plot V(anode)
```

```
.param tt=10n
.model Dm D Rs=1e-12 Tt=tt
```

Mike's comment : If TT isn't zero, Rs can not be zero, because of the way QSPICE handles dQ/dT soft recovery.

```
pwl 0 vfwd td vfwd td+p vrev
```

Integral negative diode current : Q=I*t

```
.meas Ifwd max -I(V1)
.meas Qstore Ifwd*tt
```

```
V=idt(uramp(I(V1)),0)
```

D. Diode Model (Reverse Recovery) : VP

Qspice : Dmodel - VP.qsch

- VP : Soft reverse recovery parameter
  - No description in Qspice HELP
  - **Default VP=1e-2 (i.e. Default VP=0.01)**
  - It affect current amplitude at reverse recovery
    - Increase VP can reduce current amplitude but increase reverse recovery time

```
.param vfwd=0.834
.param vrev=-10
.param td=0.1n
.param p=1p
```

```
pwl 0 vfwd td vfwd td+p vrev
```

```
.tran 20*td
.option maxstep=p/50

.plot V(Qrev)
.plot I(D1)

.step param vp list 1e-2 2e-2 4e-2

.param tt=10n
.model Dm D Rs=1e-12 Tt=tt Vp=vp
```

Integral negative diode current : Q=I*t

```
V=idt(uramp(I(V1)),0)
```

```
.meas Ifwd max -I(V1)
.meas Qstore Ifwd*tt
```

D. Diode Model (Reverse Current) : CJ

Qspice : Dmodel - CJO (RR).qsch

- CJ : Junction capacitance
  - Junction capacitance leads to reverse current but ***NOT*** a reverse recovery
    - For example, SiC has negligible reverse recovery and it model may not have TT but with CJO
  - Reverse charge depends on junction capacitance and reverse voltage
  - $\frac{dV_d}{dt}$ or $\frac{dI_d}{dt}$ are factor can affect peak reverse current in reverse recovery

```
.param vfwd=0.834
.param vrev=-10
.param td=0.1n
.param p=duration
```

```
pwl 0 vfwd td vfwd td+p vrev
```

```
.tran 5*td
.option maxstep=p/50
.plot V(Qrev)
.plot I(D1)
.plot V(anode)

.step param duration list 20p 50p 100p

.model Dm D CJO=100p
```

Integral negative diode current : Q=I*t

V=idt(uramp(I(V1)),0)

D. Diode Model (Reverse Current) : CJ

Qspice : Dmodel - CJ in Reverse Voltage.qsch / Dmodel - CJ in Reverse Current.qsch

- Cj Explanation
  - Junction capacitance is in parallel to diode and its capacitance is a function of diode reverse voltage
  - Simulation #1
    - Cj is nonlinear to diode reverse voltage, to model Cj, it requires use non-linear capacitor with Csat, Satfrac and Vsat
  - Simulation #2
    - Apply reverse voltage can simulate capacitor charging current, which act like reverse recovery but actually is not
    - Reverse recovery rely on forward current as reverse recovery is to remove this charge to turn off the diode
    - However, Cj is simply a mechanism that current charging a nonlinear capacitor

Explanation of Reverse Recovery and Junction Capacitance Charging

Qspice : Dmodel - Reverse Recovery and Capacitor Charging.qsch

- Explanation of Reverse Recovery and Junction Capacitance Charging Current
  - Reverse Recovery is only preset if forward current is applied
  - If forward current is 0A, reverse current is by charging junction capacitance (Cj), and this is a nonlinear capacitance (capacitance decrease with cathode voltage) and therefore, reverse current reduce over time even ramp of cathode voltage is constant, where $I_d = C_j \frac{dV_d}{dt}$, where C_j is a function of V_d
  - If forward current is preset, reverse recovery is applied (TT and Vp). A extra potion of reverse current is generated and this extra charge equal $I_{fwd} \times TT$

PSPICE Static Model Params (Recombination) : IKF, ISR and NR

** Modeling of Recombination Current

- Semiconductor Device Modeling with SPICE (Section 1.9.1)
  - $I_D = \left(K_{hli} I_F + K_{gen} I_R\right) - I_{Breakdown}$
    - $I_F = I_S\left(e^{\frac{qV_D}{nkT}} - 1\right)$
      - I_S is IS : Saturation current (default = 1e-14A)
      - n is N : Emission coefficient (default = 1)
    - $K_{hli} = \sqrt{\frac{I_{KF}}{I_{KF}+I_F}}$ for $I_{KF} > 0$: If $I_{KF} \to \infty$, $K_{hli} = 1$
      - I_{KF} is IKF : High injection knee current (default = 1e308)
    - $I_R = I_{SR}\left(e^{\frac{qV_D}{n_R kT}} - 1\right)$
      - I_{SR} is ISR : Recombination current parameter (default = 0A)
      - n_R is NR : ISR emission coefficient (default = 2)
    - $K_{gen} = \sqrt{\left[\left(1 - \frac{V_D}{\Phi_0}\right)^2 + 0.005\right]^m}$
      - m is M : Grading coefficient (default = 0.5)
      - Φ_0 is VJ : Junction potential (default = 1V)
    - $I_{Breakdown} = IBV\, e^{-\frac{q(BV+V_D)}{kT}}$
      - IBV is IBV : Current at breakdown voltage (default = 1e-10A)
      - BV is BV : Reverse breakdown voltage (default = Infinite V)
      - This is equation to describe breakdown region. Reverse leakage is described in $\left(K_{hli} I_F + K_{gen} I_R\right)$

D. Diode Model (Recombination) : IKF

Dmodel - IKF.qsch

- IKF
  - IKF : High injection knee current
  - **Default IKF=1e308**
  - $I_D = (K_{hli} I_F + K_{gen} I_R)$
  - $I_D = K_{hli} I_F$
    - $K_{hli} = \sqrt{\frac{I_{KF}}{I_{KF}+I_F}}$ for $I_{KF} > 0$
  - The modification effect may only be easily observed with ID in log scale

```
.param Vmin = 0.1
.param Vmax = 0.6
```

```
.dc V1 {Vmin} {Vmax} 0.01
.plot I(D1)

.step param ikf list 0.1 1 10
.model Dm D Is=100n N=1 IKF={ikf}
```

D. Diode Model (Recombination) : ISR

Qspice : Dmodel - ISR.qsch

- ISR
  - Isr : Recombination current parameter
  - **Default ISR=0A**
  - $I_D = \left(K_{hli} I_F + K_{gen} I_R\right)$
  - In following examples, I_F is forced to be negligible as compare to $K_{gen} I_R$ in calculating I_D
  - $I_D = K_{gen}\, I_{SR} \left(e^{\frac{qV_D}{n_R kT}} - 1\right)$
    - $K_{gen} = \sqrt{\left[\left(1 - \frac{V_D}{\Phi_0}\right)^2 + 0.005\right]^m}$
    - n_R is NR : default = 2V
    - m is M : default = 0.5
    - Φ_0 is VJ : default = 1V

```
.param Vmin = -5
.param Vmax = 0.4
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)

.step param isr list 0 1e-6 2e-6
.model Dm D Is=1e-10 N=2 ISR=isr NR=1
```

Is and N are set to equivalent with negligible If as compare to Ir to demonstrate Ir effect

- Remark
  - As IS is set to 1e-12, in reverse and ISR=0, leakage current can still be observed through equation I_F
  - $@V_D = -5\text{V}$

    $K_{gen} = \sqrt{\left[\left(1 - \frac{-5}{1}\right)^2 + 0.005\right]^{0.5}}$ =2.4495
  - Therefore
  - ID for ISR=2e-6, VD=-5 is 4.8990uA
  - ID for ISR=1e-6, VD=-5 is 2.4495uA

D. Diode Model (Recombination) : NR and M

Qspice : Dmodel - NR.qsch / Dmodel - M (Recombination).qsch

- NR
  - Nr : ISR emission coefficient
  - **Default NR=2**
  - $I_D = K_{gen}\, I_{SR} \left(e^{\frac{qV_D}{n_R kT}} - 1 \right)$

```
.param Vmin = 0
.param Vmax = 2
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)

.step param nr list 1 2 3
.model Dm D Is=1e-12 N=5 ISR=1e-9 NR=nr
```

Is and N are set to equivalent with negligible If as compare to Ir to demonstrate Ir effect

- M
  - M : Grading coefficient
  - **Default M=0.5**
  - $K_{gen} = \sqrt{\left[\left(1 - \frac{V_D}{\Phi_0}\right)^2 + 0.005\right]^m}$
  - The modification effect may only be easily observed with ID in log scale

```
.param Vmin = 0.5
.param Vmax = 1.4
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)

.step param m list 1m 0.5 1.3 2
.model Dm D Is=1e-12 N=5 ISR=1e-9 NR=2 M=m
```

Is and N are set to equivalent with negligible If as compare to Ir to demonstrate Ir effect

Textbook Diode Modeling

Qspice : Dmodel (Textbook).qsch

- Textbook Diode Modeling
  - IS=1e-12 to minimize leakage current when diode is OFF
  - RS=$\frac{\Delta V_D}{\Delta I_D}$ in the region that diode is ON
  - N=$\frac{\mathrm{VF}-\mathrm{IF}\times\mathrm{RS}}{0.025865\times\ln\left(\frac{\mathrm{IF}}{\mathrm{IS}}\right)}$: IF is current at VF

- Assume textbook requirement is
  - Threshold = 0.7V and $\frac{\Delta I_D}{\Delta V_D} = 100\mathrm{A/V}$
  - By $\mathrm{VF} = \frac{\mathrm{IF}}{\frac{\Delta I_D}{\Delta V_D}} + \mathrm{Threshold}$
    - We use a higher IF to determine the curve,
    - @ IF=20A → VF=$\frac{20\mathrm{A}}{100\mathrm{A/V}} + 0.7\mathrm{V}$=0.9V
  - Therefore
    - IS=1e-12
    - RS=1/100=0.01
    - N=$\frac{0.9-20\times0.01}{0.025865\times\ln\left(\frac{20}{1e-12}\right)}$=0.883

```
.param Vmin = 0.4
.param Vmax = 1.2
```

```
.dc V1 Vmin Vmax 0.01
.plot I(D1)
.model Dm D Is=1e-12 N=0.883 RS=0.01
```

D. Diode

Behavioral Diode Model Parameters

Behavioral Diode Model Parameters in Qspice HELP

QSPICE includes a simplified, behavioral, diode. To use those device equations, specify a non-zero RON.

Behavioral Diode Model Parameters

Name	Description	Units	Default
RON	Forward on resistance	Ω	0.0[3]
ROFF	Off resistance	Ω	1./GMIN
RZEN	Breakdown resistance(aka RREV)	Ω	RON
VFWD	Forward voltage drop	V	0.0
VREV	Reverse voltage drop	V	0.0
EPSILON	Width of quadratic region splining off and on regions	V	0.0
REVEPSILON	Width of quadratic region splining breakdown and on regions	V	0.0
CJO	Shunt capacitance	F	0.0

[3] The value of zero means don't use these equations, but the conventional SPICE semiconductor diode equations.

Behavioral Diode Model Params : Ron, Roff, Vfwd, Vrev

Qspice : Behavioral D Model - Ron Roff Vfwd Vrev.qsch

```
.model BeD D Ron=1 Roff=10
+ Vfwd=0 Vrev=5
+ Epsilon=0  Revepsilon=0

.dc Vfwd -7 2 0.1
.plot I(D1)
```

** Important Note : Refer to Qspice help, Ron must include and set to non-zero to activate behavioral diode model

Behavioral Diode Model Params : Rzen, Epsilon, Revepsilon

Qspice : Behavioral D Model - Epsilon Revepsilon Rzen.qsch

```
.model BeD D Ron=1 Roff=10Meg
+ Vfwd=0 Vrev=5
+ Epsilon={val} Revepsilon={val}
+ Rzen=0.5

.step param val 0 2 1
.dc Vfwd -10 4 0.1
.plot I(D1)
```

* Roff change to 10Meg in this study

Behavioral Diode Model Params : Cjo

Qspice : Behavioral D Model - Cjo.qsch

```
.param bias = 0
.model BeD_Cjo D Ron=1m Roff=1Meg
+ Vfwd=0.7 Cjo=100n
```

Cjo is equivalent to a capacitor parallel the behavioral diode

```
.ac dec 100 1 1Meg
.step param bias list 0 0.6 0.8

.plot V(cathode)@3
.plot V(cathode)@2
.plot V(cathode)@1
```

D. Diode

Ideal Diode .model

Ideal Diode Model

Qspice : Special - IdealD (.dc).qsch / Special - IdealD (.tran).qsch

- Ideal Diode Characteristic
  - No breakdown current
  - No reverse leakage
  - No capacitance effect
  - No reverse recovery
- Ideal Model with Diode .model
  - .model Idlm D N=0.01 RS=10µ ; Vf@1A=0
  - .model Idlm D N=0.36 RS=10µ ; Vf@1A=0.3
  - .model Idlm D N=0.85 RS=10µ ; Vf@1A=0.7
  - ** RS is added to prevent gmin stepping without series resistance
- Ideal Model with Behavioral Diode .model
  - .model IdealD D Vfwd=0 Ron=1m Roff=100Meg ; Vf,th=0
  - .model IdealD D Vfwd=0.3 Ron=1m Roff=100Meg ; Vf,th=0.3
  - .model IdealD D Vfwd=0.7 Ron=1m Roff=100Meg ; Vf,th=0.7

Diode Type

- Germanium
  - Bandgap : 0.67eV
- Silicon
  - Bandgap : 1.1eV
- Ultra-fast Recovery
- Silicon-Schottky
  - Bandgap : 0.69eV
- SiC-Schottky
  - Bandgap : 3.2eV
- GaN
  - Bandgap : 3.4eV
- Zener

E. F. G. H.
Dependent Sources

E. F. G. H. Dependent Sources

Qspice : Dependent Source - Basic.qsch

- Dependent Sources
  - [E] VCVS : Voltage Dependent Voltage Source
  - [G] VCCS : Voltage Dependent Current Source
  - [H] CCVS : Current Dependent Voltage Source
  - [F] CCCS : Current Dependent Current Source

- ** Current Dependent Src
  - For current dependent source, the current sense has to come from voltage source instead of current source
  - A 0V voltage source can be used for current sensing
  - (Explanation from Mike Engelhardt) Sensing the current in a current is complicated because it might have VSAT or VSAT2, meaning it's not a current source

E-Source Alternative Syntax

- E-source
  - Syntax: Exxx n+ n- value={<expression>}
  - This is an alternative syntax of the behavioral voltage source (B-source)
  - This syntax only available in using text, no Qspice symbol support this format, for Qspice to backward compatible to legacy spice syntax

```
.tran 1
.plot V(idtE2)
.plot V(idtE1)
.plot V(idtB1)
.plot V(sgl)
```

Syntax: Exxx n+ n- value={<expression>}
alternative syntax of the behavioral source with E-source formula must have curly bracket {}

```
E1 idtE1 0 Value {idt(V(sgl))}
E2 idtE2 0 Value={idt(V(sgl))}
```

Can accept without "="

```
* C:\QspiceKSKelvin\01 User Gu
V1 sgl 0 1
B1 idtB1 0 V=idt(V(sgl))
E1 idtE1 0 Value {idt(V(sgl))}
E2 idtE2 0 Value={idt(V(sgl))}
.plot V(idtE2)
.plot V(idtE1)
.plot V(idtB1)
.plot V(sgl)
.tran 1
.end
```

I. Current Source

I. Current Source : Instance Parameters

Current Source Instance Parameters

Name	Description	Units	Default
AC	AC magnitude, optionally followed by phase angle	A, °	0.
ACMAG	AC magnitude	A	0.
ACPHASE	AC phase	°	0.
DC	DC value of source	A	0.
EXP	Exponential source description		
LOG	Interpolate between PWL and CHIRP points		(not set)
PULSE	Pulse description		
PWL	Piecewise linear description		
SFFM	Single frequency FM description		
SINE	Sinusoidal source description(aka SIN)		
TIMECTRL	Time step control, one of NONE, LIMITS[1], BREAKS[2], or BOTH	String	LIMITS
VSAT[3]	Don't source power(example)	V	0.
VSAT2	Don't source power and don't conduct in reverse	V	0.
XTRAP	Extrapolate beyond PWL and CHIRP points		(not set)

I. Instance Params : VSAT

Qspice : I Source - VSAT.qsch

- VSAT
  - VSAT : Does **NOT** source power
  - **Default VSAT=0V**
  - Characteristic
    - Only source current when V(-ve,+ve) > VSAT
    - Current source doen't source current at V(-ve,+ve)=0
      - Equivalent to Open
    - Not to apply negative voltage between V(-ve,+ve) as it will have current flow out from -ve
  - Usage
    - Useful for modeling an active load or a bias current in a macromodel

```
.dc V1 0 10 0.1
.step param VSAT list 0 2 4 6 10
.func ZI1() V(a)/-I(V1)
.plot ZI1()
.plot -I(V1)
```

I. Instance Params : VSAT2

Qspice : I Source - VSAT2.qsch

- VSAT2
  - VSAT2 : Does **NOT** source power and **NOT** conduct in reverse
  - **Default VSAT2=0V**
  - If VSAT2 is used, VSAT will be overridden

I. Instance Params : XTRAP

Qspice : I Source - XTRAP.qsch

- XTRAP
  - Xtrap : Extrapolate beyond PWL and CHIRP points
  - **Default XTRAP is not set**
  - Without XTRAP, the current stays constant with the value specified with the last time point

I. Instance Params : Timectrl

Qspice : I Source - TIMECTRL.qsch

- TIMECTRL
  - Timectrl : Time step control
    - None
    - Limits
    - Breaks
    - Both
  - **Default Timectrl=Limits**

```
pulse 0 1 4n 0 0 10n 20n
Timectrl=none
Timectrl=limits
Timectrl=breaks
Timectrl=both
.tran 1000n
.plot I(I1) I(I2)
```

J. JFET Transistor

J. JFET Model Parameters

JFET Model Parameters

Name	Description	Units	Default
AF	Flicker noise exponent		1.0
ALPHA	Impact ionization coefficient	V^{-1}	0.0
B	Doping tail parameter		0.0
BETA	Transconductance parameter	A/V^2	0.0001
BETATCE	Exponential transconductance parameter temperature coefficient	%/°C	0.0
CGD	Zero-bias G-D junction capacitance	F	0.0
CGDO	G-D overlap(i.e. constant) capacitance	F	0.0
CGS	Zero-bias G-S junction capacitance	F	0.0
CGSO	G-S overlap(i.e. constant) capacitance	F	0.0
ETA	Philips, et al.-style subthreshold conduction parameter		0.0
ETATC1	ETA first order tempco	$°C^{-1}$	0.0
ETATC2	ETA second order tempco	$°C^{-2}$	0.0
FC	Forward bias junction fit parameter		0.5
GDSNOI	Shot noise coefficient for nlev = 3	A	2.0
GMAX	Maximum junction conductivity(straight line extension)	℧	10K
IS	Junction saturation current	A	1e-14
ISR	Recombination current parameter	A	0.0
KF	Flicker Noise Coefficient		0.0
LAMBDA	Channel length modulation parameter	V^{-1}	1.0
M	Grading coefficient(aka MJ)		0.5
N	Emission coefficient		1.0
NLEV	Noise level(equation selector)		0.0
NR	ISR emission coefficient		2.0
PB	Gate junction potential	V	1.0
RD	Drain resistance	Ω	0.0
RDTC1	RD first order tempco	$°C^{-1}$	0.0
RDTC2	RD second order tempco	$°C^{-2}$	0.0
RDS	Additional Drain-Source leakage resistance	Ω	infinite
RDSTC1	RDS first order tempco	$°C^{-1}$	0.0
RDSTC2	RDS second order tempco	$°C^{-2}$	0.0
RG	Gate resistance	Ω	0.0
RGTC1	RG first order tempco	$°C^{-1}$	0.0
RGTC2	RG second order tempco	$°C^{-2}$	0.0
RONX	Channel conductivity multiplier in linear region[1]		1.0
RS	Source resistance	Ω	0.0
RSTC1	RS first order tempco	$°C^{-1}$	0.0
RSTC2	RS second order tempco	$°C^{-2}$	0.0
TNOM	Parameter measurement temperature(aka TREF)	V	0.0
VK	Impact ionization knee current	V	0.0
VT0	Threshold Voltage(aka VTO or VT)	V	-2.0
VTOTC	Threshold Voltage temperature coefficient	$°C^{-1}$	1.0
XTI	Saturation current temperature exponent		3.0

J. JFET Basic Equation

- JFET equation in Semiconductor Device Modeling with SPICE Section 3.1.3)
  - For channel pinched off : $V_{GS} - V_{T0} \leq 0$
    - $I_D = 0$
  - For saturated region : $0 \leq V_{GS} - V_{T0} \leq V_{DS}$
    - $I_D = \beta\,(V_{GS} - V_{T0})^2\,(1 + \lambda V_{DS})$
  - For linear region : $0 < V_{DS} < V_{GS} - V_{T0}$
    - $I_D = \beta\,[2\,(V_{GS} - V_{T0}) - V_{DS}]\,(1 + \lambda V_{DS})$

J. JFET Instance Params (I_D) : BETA and VTO

Qspice : JFET - Beta.qsch / JFET - VTO.qsch

- BETA
  - Beta : Transconductance parameter
  - **Default BETA=0.0001A/V²**
  - $I_D = \beta\,(V_{GS} - V_{T0})^2\,(1 + \lambda V_{DS})$

- VTO (aka VT0 or VT)
  - VTO : Threshold Voltage
  - **Default VTO=-2V**
  - $I_D = \beta\,(V_{GS} - V_{T0})^2\,(1 + \lambda V_{DS})$

J. JFET Instance Params (I_D) : Lambda and Betatce

Qspice : JFET - LAMBDA.qsch / JFET - BETATCE.qsch

- LAMBDA
  - Lambda : Channel length modulation parameter
  - **Default LAMBDA=0V^{-1}**
  - $I_D = \beta\,(V_{GS} - V_{T0})^2\,(1 + \lambda V_{DS})$

- BETATCE
  - Betatce : Exponential transconductance parameter temperature coefficient
  - **Default BETATCE=0 (%/°C)**

J. JFET Instance Params (Diode GD and GS) : IS, N, ISR, NR, M and PB

Qspice : JFET - IS N.qsch / JFET - ISR NR M PB.qsch

- Two diodes in JFET
  - DGD and DGS
    - These two diodes are identical model, with recombination current included
  - $I_{GS} = (I_F + K_{gen} I_R) + I_I$
    - $I_F = I_S\left(e^{\frac{qV_D}{nkT}} - 1\right)$
    - $I_R = I_{SR}\left(e^{\frac{qV_D}{n_R kT}} - 1\right)$
    - $K_{gen} = \sqrt{\left[\left(1 - \frac{V_D}{\Phi_0}\right)^2 + 0.005\right]^m}$
    - I_S : IS Junction saturation current (Default IS=1e-14A)
    - n : N emission coefficient (Default N=1)
    - I_{SR} : ISR Recombination current parameter (Default ISR=0A)
    - n_R : ISR emission coefficient (Default NR=1)
    - m : M Grading coefficient (Default M=0.5)
    - Φ_0 : PB Gate junction potential (Default PB=1V)
  - Diode modeling refer to D. Diode section

Vds set is 0V
test both Dgd and Dgs
with same Vgs sweep

```
.dc Vgs -1 1 0.01
.plot Ig(J1)
.step param vto list -2 0 2
.model Jmdl NJF IS=1e-16 N=0.8
```

Vds set is 0V
test both Dgd and Dgs
with same Vgs sweep

```
.dc Vgs -1 1 0.01
.plot Ig(J1)
.step param vto list -2 0 2
.model Jmdl NJF IS=1e-16 N=0.8
+ ISR=1e-12 NR=0.5 M=1 PB=1
```

J. JFET Instance Params (Diode GS) : Alpha and VK

Qspice : JFET - ALPHA VK.qsch

- Gate Source Diode in Pspice model
  - $I_{GS} = (I_F + K_{gen} I_R) + I_I$
    - $I_I = \alpha\, I_D [V_{DS} - (V_{GS} - V_{T0})]\, e^{\frac{V_k}{V_{DS} - (V_{GS} - V_{T0})}}$ for $0 < G_{GS} - V_{TO} < V_{DS}$
    - $I_I = 0$ otherwise
    - α is ALPHA Impact ionization coefficient (Default ALPHA=0V^{-1})
    - V_k is VK Impact ionization knee current (Default VK=0V)

J. JFET Instance Params (Capacitance) : CGS, CGD, CGSO, CDSO

Qspice : JFET - CGS CGD.qsch / JFET - CGSO CGDO.qsch

- CGS, CGD
  - CGS : Zero-bias G-D junction capacitance
  - CGD : Zero-bias G-S junction capacitance
  - **Default CGS=0F**
  - **Default CGD=0F**
  - This example simulate with both CGS and CGD

```
.ac list f        .param f=1Meg
.step param bias -3 0.5 0.1
.func imZ() imag(V(G)/Ig(J1))
.func Ceqv() -1/2/pi/f/imZ()
.plot Ceqv()
.model Jmdl NJF CGS=100p CGD=100p
```

- CGSO, CDSO
  - CGSO : G-S overlap capacitance
  - CDSO : D-S overlap capacitance
  - **Default CGSO=0F**
  - **Default CDSO=0F**
  - This example only simulate CGS with CGSO. GDSO has same effect but to affect CDSO
  - ** M, PB, FC has effect in junction capacitance, refer to D. diode

```
.ac list f        .param f=1Meg
.step param bias -5 0 0.1
.func imZ() imag(V(G)/Ig(J1))
.func Ceqv() -1/2/pi/f/imZ()
.plot Ceqv()
.step param cgso list 10p 20p 40p
.model Jmdl NJF M=3 CGS=100p CGSO=cgso
```

K. Mutual Inductance

K. Mutual Inductance

Qspice : K Mutual - Two Winding.qsch / K Mutual - Three Winding.qsch

- K Mutual Inductance
  - K : coupling coefficient of coupled inductors
    - Ideal coupling : 1
    - Range : -1 to 1
  - Two or more coupled inductors are required
    - Press L two times to get symbol with a dot notation (not necessary, but recommend for current direction)

- Inductors as Transformer
  - $\frac{L_p}{N_p^2} = \frac{L_s}{N_s^2}$ and $n = \frac{N_p}{N_s}$
    - N is number of turns
  - $L_p = n^2 L_s$ or $L_s = \frac{1}{n^2} L_p$
  - In practice, Lp is measured from primary with secondary open

Demo of two windings

```
.tran 10/1K
.plot V(pri)
.plot V(sec)
.param n=2
.param Lp=100µ
.param Ls=Lp/n**2
```

Demo of three windings

```
.tran 10/1K
.plot V(pri)
.plot V(sec1)
.plot V(sec2)
.param n=2
.param Lp=100µ
.param Ls=Lp/n**2
```

L. Inductor

Inductor L Instance Parameters in Qspice HELP

Inductor Instance Parameters

Name	Description	Units	Default
AU	Wire or stripline is made of gold		
AL	Wire or stripline is made of aluminum		see below
AG	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[1]	Equivalent parallel resistance	Ω	INDUCTANCE÷(15.91×GMIN)
RSER	Equivalent series resistance	Ω	0.0
SATFRAC	Fractional drop in inductance at ISAT		0.7
TC[2]	Temperature coefficient polynomial		(none)
TC1[2]	1st order temperature coefficient		0.0
TC2[2]	2nd order temperature coefficient		0.0
THICK	Thickness of stripline on top of a PCB	m	0.0
TNOM	Temperature inductance was measured(aka TREF)	°C	Circuit TNOM
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	

Arbitrary Inductance Device

QSPICE also supports an arbitrary inductance device. To use it, give an equation for the flux due to the inductor. The equation can include any circuit node voltages or device currents of devices modeled as Thévenin equivalents(V-, L-, E-, or H- devices). In the interest of completeness, the device supports transinductance.

Syntax: Cnnn N1 N2 Q=<expression> [additional instance parameters]

Name	Description	Units	Default
FLUX	Equation of flux(aka F)	Weber	
IC	Initial flux due to the inductor	Weber	
M	Number of identical parallel devices		1.0
RPAR	Equivalent parallel resistance	Ω	Infinite
RSER	Equivalent series resistance	Ω	0.0
NOISELESS	Ignore the noise contribution from RPAR and RSER		not set
TEMP	Instance temperature	°C	Circuit temperature

L. Instance Params : Rser, Frequency and Q

Qspice : L - Rser Frequency Q.qsch

- Rser
  - Equivalent series resistance
  - Default Rser=0
  - Rser is a constant over entire frequency

- Frequency and Q
  - Frequency : Frequency at Q, also used to compute Rser due to skin effect
  - Q : Quality factor at FREQUENCY
  - Formula
    - $Q = \frac{X_L}{R} = \frac{2\pi f L}{R_{@f}}$
    - Therefore, series resistance can changes over frequency to simulate skin effect if Frequency and Q are used

L. Instance Params : Rpar – Equivalent Parallel Resistance

Qspice : L - Rpar Prove.qsch | L - Rpar (.option LFT).qsch

- Rpar
  - Equivalent parallel resistance
  - Default Rpar=Inductance/(15.91*Gmin)
    - 15.91 is only an approximate, exact formula is $\frac{10^2}{2\pi}$
    - Gmin is minimum conductance in .option, default gmin=1e-12
  - To calculate Rpar, .ac analysis is used, with Lp-Rp model (Rpar=Rp)
    - Impedance : $Z = \frac{V}{I}$
    - Admittance : $Y = \frac{1}{Z} = G + jB$
    - $Rp = \frac{1}{G}$, where G is conductance

- .option LFT for Rpar
  - If circuit option LFT is sent, then the resistance is chosen to be equal to the reactance at frequency LFT
  - Rpar formula changed to Rpar = 2*pi*LFT*Inductance

Default Rpar=Inductance/(15.91*Gmin) in HELP
**15.91 is approximation from 1/2/pi*1e2

** Inductor always with a default Rpar!

```
.ac dec 100 1 100Meg
Default gmin in Qspice is 1e-12
.option gmin=1e-12
```

```
Formula to calculate Rp from Lp-Rp model
.func Z() V(L)/I(L1)
.func Y() 1/Z()
.func Rp() 1/re(Y())
.plot Rp()
```

Default Rpar=Inductance/(15.91*Gmin) in HELP
**15.91 is approximation from 1/2/pi*1e2

.option LFT=100K

** If .option LFT is used, Rpar formula is changed

```
.ac dec 100 1 100Meg
Default gmin in Qspice is 1e-12
.option gmin=1e-12
```

```
Formula to calculate Rp from Lp-Rp model
.func Z() V(L)/I(L1)
.func Y() 1/Z()
.func Rp() 1/re(Y())
.plot Rp()
```

L. Instance Params : Rpar – Equivalent Parallel Resistance

Qspice : L - Rpar=0 Gmin=1e-308 LFT=1e308.qsch

- Force Rpar to Infinite
  - If simulation requires to set Rpar to infinite, three method can be considered
  - #1 : Set Rpar=0
    - This method is mentioned by Mike Engelhardt in Qspice forum and not documented in HELP
  - #2 : Set gmin=1e-308
    - As 1/gmin is infinite, Rpar becomes infinite
    - This is useful if multiple inductors in schematic, but be caution it may introduce other problem if non-linear device in schematic when gmin is required
  - #3 : Set .option LFT=1e308
    - As LFT is infinite, Rpar becomes infinite
    - I suggest to use this method
  - In this demonstration, a R1 is added in parallel for visual aids, as Rpar=infinite will return NaN in Rp() and nothing is shown in waveform viewer.

Engelhardt 17d

I think it's hard to find an inductor with less parallel loss than the minimum added by default in QSPICE.

But if you don't like QSPICE'S help on this, you can add an attribute "Rpar=0" to the inductor.

You might want to do the same for the capacitors.

L. Instance Params : Rpar – Equivalent Parallel Resistance

- Quick Summary of Rpar
- Rpar formula priority
  - 1st priority : If **Rpar=<value>**, Rpar is set to equal <value>
  - 2nd priority : If **.option LFT** is set and **without Rpar**, Rpar = 2*pi*LFT*Inductance
  - 3rd priority : If **no .option LFT** and **without Rpar**, Rpar = Inductance/(15.91*gmin)
- Force Rpar to Infinite
  - #1 : Set Instance parameter Rpar=0 individually
  - #2 : Set .option LFT=1e308
  - #3 : Set .option gmin=1e-308 (not recommend)

L. Instance Params : Isat, Lsat, Satfrac - For Nonlinear Inductance

Qspice : L Inductor - Isat (.ac).qsch

- Isat and Satfrac
  - Isat : Current that drops inductance to ***Satfrac x Inductance***
  - Satfrac : Fractional drop in inductance at Isat
  - **Default ISAT=Infinite (A)**
  - **Default SATFRAC=0.7**
  - Nonlinear effect is valid in +ve and –ve current direction

```
.param frq=1Meg
.step param bias -10 10 0.1      Sweep bias
.step param var list 1 2 3       Sweep Isat

.ac list frq
.func Leqv() V(VL)/1/2/pi/frq
.plot Leqv()
```

Bias Current from -10V to 10V

- Inductance equation
  - $X_L = \omega L = \frac{V_L}{I_L}$
  - $L = \frac{|V_L|}{|I_L|}\frac{1}{\omega} = \frac{|V_L|}{2\pi f |I_L|}$
  - This formula is used to calculate equivalent inductance (small signal model) at different inductor bias current

L. Instance Params : Isat, Lsat, Satfrac - For Nonlinear Inductance

Qspice : L Inductor - Isat (.tran).qsch / L Inductor - Lsat (.tran).qsch

- Isat and Satfrac
  - **Isat** : Current that drops inductance to ***Satfrac x Inductance***
  - **Satfrac** : Fractional drop in inductance at Isat
  - **Default ISAT=Infinite (A)**
  - **Default SATFRAC=0.7**

- Lsat
  - **Lsat** : Inductance asymptotically approached in saturation
    - ** Isat must be set to see it effect
  - **Default LSAT=10% of L**

Demonstration of Nonlinear Inductor (model B-H saturation)

Qspice : L Inductor Nonlinear (Tran) - Current and BH.qsch

- Nonlinear Inductor in Qspice
  - This simulation demonstrate transient of non-linear inductor with a sinusoidal voltage source, which can expect a non-linear current

- Reference : B and H for inductor
  - Magnetic field intensity $H = \frac{B}{\mu} = \frac{N}{l_e} i_L$ (unit : A/m)
  - Magnetic field strength $B = \frac{\phi_B}{A_e} = \frac{1}{N\,A_e} \int v_L\, dt$ (unit : Tesla)
    - B also called magnetic flux density

Demonstration of Nonlinear Inductor (model B-H hysteresis : lossy)

Qspice : L Inductor Nonlinear (Tran) - Current and BH.qsch

Plot equivalent L

L Inductor (AC) - Current and BH.qsch

Comment about Saturation and Hysteresis of B-H curve

Isat, Lsat and Satfrac are key parameters in Qspice for non-linear inductance modeling. General interest is just a saturating include. This represents inductor saturating when current reach certain level (i.e. inductance drop according to current, which can model by Isat, Lsat and Satfrac). This characteristic is saturation in B-H curve.

For hysteresis of B-H curve, it can generally model with a parallel resistor Rpar to represent loss.

May be some people prefer to model non-linear inductor with other way, but I found Qspice definition generally good enough as L vs bias current can be easily measure practically.

L. Arbitrary Inductance : Inductor modeling with FLUX

- Arbitrary Inductance Device
  - To use arbitrary inductance device, give an equation for the flux due to the inductor

Arbitrary Inductance Device

QSPICE also supports an arbitrary inductance device. To use it, give an equation for the flux due to the inductor. The equation can include any circuit node voltages or device currents of devices modeled as Thévenin equivalents(V-, L-, E-, or H- devices). In the interest of completeness, the device supports transinductance.

Syntax: Cnnn N1 N2 Q=<expression> [additional instance parameters]

Name	Description	Units	Default
FLUX	Equation of flux(aka F)	Weber	
IC	Initial flux due to the inductor	Weber	
M	Number of identical parallel devices		1.0
RPAR	Equivalent parallel resistance	Ω	Infinite
RSER	Equivalent series resistance	Ω	0.0
NOISELESS	Ignore the noise contribution from RPAR and RSER		not set
TEMP	Instance temperature	°C	Circuit temperature

- Basic Formula between Inductance and Flux
  - [equation 1] : Inductor Voltage v_L and Flux ϕ relationship : $v_L = \frac{d\phi}{dt}$
  - [equation 2] : Inductor Voltage v_L and Inductor Current i_L relationship : $v_L = L\frac{di_L}{dt}$
  - By $\frac{d\phi}{dt} = L\frac{di_L}{dt}$ → Inductance $L = \frac{d\phi}{di_L}$

L. Arbitrary Inductance : Inductor modeling with FLUX

Qspice : Flux formula.qsch

- Qspice inductor modeling with Flux ϕ
  - For linear inductor, if $\frac{d\phi}{di_L}$ is constant → $\phi = L\, i_L$
    - Therefore, $L = \frac{\phi}{i_L}$
    - *In Qspice* : flux = L * I(Lnnn)
      - where Lnnn is the name attribute of inductor symbol (e.g. L1, L2)
  - For nonlinear inductor, ϕ can be expressed by

$$\phi = a \tanh(b\, i_L)$$

    - Therefore, $L = \frac{d\phi}{di_L} = \frac{d\, a \tanh(b\, i_L)}{di_L} = a\, b\, (1 - \tanh^2(b\, i_L))$
    - *In Qspice* : flux = a*tanh(b*I(Lnnn))
      - L0A @ 0A = a * b : L0A is inductance at 0A
      - b determines the spread
      - If $b\, i_L$ is large, $(1 - \tanh^2(b\, i_L)) = 0$, therefore, this formula will give 0H inductance when i_L is large
  - Nonlinear and linear flux formula can be combined to handle 0H inductance when i_L is large
    - *In Qspice* : flux = a*tanh(b*I(Lnnn)) + Lmin * I(Lnnn)
      - Lmin @ $i_L = \infty$: Lmin : Lmin is inductance at current → ∞
      - L0A @ $i_L = 0$: a*b – Lmin
      - b determines the spread

Assume flux formula is a*tanh(b*IL)+Lmin*IL

```
.param b = 0.7
.param L0A = 1µ
.param a = (L0A-Lmin)/b
.param Lmin=58n
.func flux() a*tanh(b*IL)+Lmin*IL
```

Calculate Inductance by dflux/dIL

```
.func dflux() D(flux())
```

Sweep IL from -10A to 10A

```
.step param IL -10 10 0.2
.op
.plot dflux()
.plot flux()
```

$L = \frac{d\phi}{di_L} = \text{dflux}()$

determined by Lmin

L. Arbitrary Inductance : Inductor modeling with FLUX

Qspice : L Inductor (AC) - Flux eqn 1.qsch ; L Inductor Nonlinear (Tran) - Flux eqn 1.qsch

- Flux model
  - .ac is used to calculate inductance and bias current relationship
  - .tran is used to demonstrate non-linear inductor current and proportional B-H curve
  - Flux equation uses format
    - Flux = a*tanh(b*I(L1))

```
.param b = 0.59
.param L0A = 1µ
.param a = L0A/b
flux=a*tanh(b*I(L1))
.param frq=1Meg
.step param bias -10 10 0.1      Sweep bias
.step param var list 1 2 3       Sweep Isat
.ac list frq
.func Leqv() V(VL)/1/2/pi/frq
.plot Leqv()
```

Flux model with .ac

Flux model with .tran

L. Arbitrary Inductance : Inductor modeling with FLUX – Nonlinear with Lmin

Qspice : L Inductor (AC) - Flux eqn 2.qsch ; L Inductor Nonlinear (Tran) - Flux eqn 2.qsch

- Flux Model
  - This is nonlinear flux model include Lmin at large inductor current
  - Flux equation uses format
    - Flux = a*tanh(b*I(L1))+Lmin*I(L1)

L. Behavioral Inductance with Flux (Arbitrary Inductance)

Qspice : L - Behavioral L with Flux.qsch

- Behavioral L with Flux
  - For linear inductor
    - $\phi = L \times I_L$
  - In Qspice
    - Flux = <eqn> * I(Ln)
      - <eqn> is inductance equation supports voltage node or device current
      - Ln is name attribute of inductor

Linear Inductor flux formula : Flux = L * IL
Therefore, behavioral inductance model
can be derived using Arbitrary Inductance Device

```
.step param Vctrl 0.9 1.1 0.01
.param frq=1Meg
.ac list frq
.func imZ() V(L)/I(L1)
.func L() imZ()/2/pi/frq
.plot L()
```

L. Behavioral Inductance with Flux (Arbitrary Inductance)

Qspice : L - Behavioral L with Flux (.tran).qsch

- Flux
  - Behavioral Inductance with flux
  - This schematic demonstrate using behavioral inductance (with flux) model in transient (.tran) analysis

Linear Inductor flux formula : Flux = L * IL
Therefore, behavioral inductance model
can be derived using Arbitrary Inductance Device

```
.param frq=1K
.param XL=1
.param Linit=XL/2/pi/frq
```

```
.tran 50m uic
.option maxstep=1/frq/100
.plot I(L1)
.plot V(L)
.plot V(ctrl)
```

M. MOSFET

(VDMOS)

** VDMOS with different default as monolithic MOSFET models, it popularly used in board level SMPS

M. MOSFET NMOS and PMOS

Ohmic / Saturation
$V_D > V_S$
V_{GS} is +ve ($V_G > V_S$)

Ohmic / Saturation
$V_S > V_D$
V_{GS} is –ve ($V_G < V_S$)

Voltage Relations	NMOS
$V_{GS} > V_{th}$ $V_{DS} < V_{GS} - V_{th}$	Ohmic (Triode/Linear)
$V_{GS} > V_{th}$ $V_{DS} \leq V_{GS} - V_{th}$	Saturation (Active)
$V_{GS} < V_{th}$	Cutoff

M. MOSFET NMOS and PMOS Ohmic, Saturation and Cutoff

Qspice : NMOS PMOS (Basic) - Region NMOS.qsch | NMOS PMOS (Basic) - Region PMOS.qsch

- NMOS
  - N-channel MOSFET

- PMOS
  - P-channel MOSFET

MOSFET Level 1 Model Parameters

Refer D. Diode .model parameters

MOSFET Level 1 Model Parameters

Name	Description	Units	Default
A	Additional non-linear Gate-drain capacitance abruptness		1.0
AD	Default drain area	m^2	0.0
AF	Flicker noise exponent		1.0
AS	Default source area	m^2	0.0
BV	Body diode breakdown voltage	V	Infinite
CAPOP [9]	Charge(Capacitance) model		1
CBD	Zero-bias B-D junction capacitance	F	0.0
CBS	B-S junction capacitance	F	0.0
CGBO	Gate-bulk overlap capacitance	F/m	0.0
CGDMAX [8]	Maximum additional non-linear Gate-Drain capacitance	F	0.0
CGDMIN [8]	Minimum additional non-linear Gate-Drain capacitance	F	0.0
CGDO	Gate-drain overlap capacitance	F/m	0.0
CGS	Gate-source capacitance that doesn't scale with width	F	0.0
CGSO	Gate-source overlap capacitance	F/m	0.0
CJ	Bottom junction cap per area	F/m^2	0.0
CJO	Body diode zero-bias capacitance	F	0.0
CJSW	Side junction cap per length	F/m	0.0
EG	Body diode activation energy	V	1.11
ETA [8]	Philips, et al.-style subthreshold conduction parameter		0.0
ETA2	Inverse sharpness of Ids limit for non-square law variation		0.0
FC	Forward bias junction fit parameter		0.5
GAMMA	Bulk threshold parameter	$V^{1/2}$	0.0
GDSNOI	Channel shot noise coefficient(nlev=3)		1.0
GM	Ideal transconductance for non-square law variation	℧	none
GMAX	Maximum bulk PN conductivity	℧	1000.
IBV	Body diode current at breakdown voltage	A	1e-10
IDS	Maximum drain current for non-square law variation	A	Infinite
IGSS	Gate-Source leakage current	A	0.0
IS	Bulk junction saturation current	A	1e-14
JS	Bulk junction saturation current density	A/m^2	0.0
K1 [10]	Threshold voltage sensitivity to bulk node	$V^{1/2}$	0.25
KF	Flicker noise coefficient		0.0
KP	Transconductance parameter	A/V^2	2e-5
L	Default channel length	m	1e-5
LAMBDA	Channel length modulation	V^{-1}	0.0
LD	Lateral diffusion	m	0.0
MJ	Bulk grading coefficient		0.5
MJSW	Side grading coefficient		0.33
N	Bulk(or Body) PN junction emission coefficient		1.0
NBV	Body diode breakdown emission coefficient		1.0
NLEV	Noise level(equation selector)		0
NRD	Default drain squares		0.0
NRB	Default bulk squares		0.0
NRG	Default gate squares		0.0
NRS	Default source squares		0.0
NSS	Surface state density	cm^{-2}	0.0
NSUB	Substrate doping	cm^{-2}	0.0
PB	Bulk junction potential	V	1.0
PD	Default drain perimeter	m	0.0
PHI	Surface potential	V	1.0
PCHAN	Flag to specify vertical PMOS geometry		not set
PS	Default source perimeter	m	0.0
RB	Bulk resistance	Ω	0.0
RD	Drain resistance	Ω	0.0
RDS	Additional Drain-Source shunt resistance	Ω	Infinite
RG	Gate resistance	Ω	0.0
RON	Ron at zero drain current for non-square law variation	Ω	none
RONX [8]	Channel conductivity multiplier in linear region		1.0
RS	Source resistance	Ω	0.0
RSH	Sheet resistance	Ω/□	0.0
TNOM	Parameter measurement temperature(aka TREF)	°C	27.0
TOX	Oxide thickness	m	1.5e-8
TPG	Gate type		1
TT	Bulk PN junction Transit-time	sec	0.0
U0	Surface mobility(aka UO)	$cm^2/V\times sec$	600
VDMOS [8]	Flag to specify vertical geometry		not set
VFB [10]	Flat band voltage	V	-1.0
VIGSS	Voltage at which IGSS was measured	V	5.0
VJ	Body diode Junction potential	V	1.0
VT0	Threshold voltage(aka VTO)	V	0.0
W	Default channel width	m	1e-5
XPART [10]	Channel charge partitioning		1
XTI	Body diode Saturation current temperature exponent		3

Display purpose only

- Vds : VDS rating
- Qq : Gate charge
- Mfg : manufacturer

Linear and Saturation Region : VTO, KP, W, L, LD, LAMBDA

- From Semiconductor Device Modeling with Spice (Section 4.2.3)
  - Linear region : $V_{GS} > V_{TH}$ and $V_{DS} < V_{GS} - V_{TH}$
    - $I_{DS} = KP \frac{W}{L-2X_{jl}} \left(V_{GS} - V_{TH} - \frac{V_{DS}}{2}\right) V_{DS}(1 + \lambda V_{DS})$
  - Saturation region : $V_{GS} > V_{TH}$ and $V_{DS} > V_{GS} - V_{TH}$
    - $I_{DS} = \frac{KP}{2} \frac{W}{L-2X_{jl}} (V_{GS} - V_{TH})^2 (1 + \lambda V_{DS})$
  - where
    - V_{TH} : VTO , zero-bias threshold voltage (Default VTO=0 V)
    - KP : KP , transconductance (Default KP=1 A/V^2)
    - W : W, channel width (Default W=1e-4 m)
    - L : L, channel length (Default L=1e-4 m)
    - X_{jl} : LD, lateral diffusion (Default LD=0 m)
    - λ : LAMBDA channel-length modulation (Default LAMBDA=0 V^{-1})

Figure 4-1 Structure of the MOST.

M. MOSFET Params : VTO, KP, W and L

Qspice : VDMOS - VTO.qsch / VDMOS - KP.qsch / VDMOS - W.qsch / VDMOS - L.qsch

- VTO
  - VTO : Zero-bias threshold voltage
  - **Default VTO=0V**

.param Vdrain=10
.dc Vgate -4 6 0.1 .plot Id(M1)

.step param vto list -2 0 2
.model Mmdl VDMOS VTO=vto

- KP
  - KP : transconductance
  - **Default KP=1 A/V²**

- W and L
  - W : Channel Width (Default 1e-4)
  - L : Channel Length (Default 1e-4)
  - Effect is similar to KP as a gain
    - $\frac{KP}{2}\frac{W}{L-2X_{jl}}$ (for saturation region)

.param Vdrain=10
.dc Vgate 3 9 0.1 .plot Id(M1)

.step param kp list 0.5 1 2
.model Mmdl VDMOS VTO=5 KP=kp

M. MOSFET Params : LD and ETA

Qspice : VDMOS - LD.qsch / VDMOS - ETA.qsch

- LD
  - LD : lateral diffusion
  - **Default LD=0 m**
  - $L_{eff} = L - 2LD > 0$
    - Therefore, $0 < LD < \frac{L}{2}$

.param Vdrain=10
.dc Vgate 3 9 0.1 .plot Id(M1)

.step param ld list 0 0.1e-4 0.2e-4
.model Mmdl VDMOS VTO=5 LD=ld

- ETA (aka ksubthres)
  - ETA : Vds dependence of threshold voltage
  - Static feedback on threshold voltage
  - **Default ETA=0**
  - Qspice accepts parameter name as ETA or ksubthres

.param Vdrain=10
.dc Vgate -10 20 0.1.plot Id(M1)

.step param eta list 0 2 3 4
.model Mmdl VDMOS VTO=5 ETA=eta

M. MOSFET Params : Lambda

Qspice : VDMOS -LAMBDA.qsch

- LAMBDA
  - Labmda : channel-length modulation
  - **Default LAMBDA=0 V^{-1}**

.param Vdrain=10 .plot Id(M1)
.dc Vdrain 0 10 0.1 Vgate 5 10 1

.model Mmdl VDMOS VTO=5

.param Vdrain=10 .plot Id(M1)
.dc Vdrain 0 10 0.1 Vgate 5 10 1

.model Mmdl VDMOS VTO=5 Lambda=0.02

M. MOSFET Params (Gate) : VIGSS and IGSS, RG and CGS

Qspice : VDMOS - VIGSS IGSS.qsch / VDMOS - Rg Cgs.qsch

- VIGSS and IGSS
  - Vigss : Voltage at which IGSS was measured
  - Igss : Gate-Source leakage current
  - **Default VIGSS=5V**
  - **Default IGSS=0A**
  - Parameters to determine Igs in relates to Vgs

- RG and CGS
  - Rg : Gate Resistance
  - Cgs : Gate-source capacitance that doesn't scale with width
  - **Default RG=0Ω**
  - **Default CGS=0F**
  - Parameters to determine gate resistance and capacitoance

M. MOSFET Params (Cgd) : CGDMAX, CGDMIN and A

Qspice : VDMOS - CGDMIN CGDMAX.qsch / VDMOS - A.qsch

- CGDMAX and CGDMIN
  - Cgdmax : Maximum additional non-linear Gate-Drain capacitance
  - Cgdmin : Minimum additional non-linear Gate-Drain capacitance
  - **Default Cgdmax=0F**
  - **Default Cgdmin=0F**
  - Model non-linear of Cgd capacitance

```
.param f=10Meg
.ac list f
.step param Vdrain -10 50 0.5
```

Calculate Capacitance

```
.func imZ() imag(V(d)/Ig(M1))
.func Ceqv() -1/2/pi/f/imZ()
.plot Ceqv()
```

```
.model Mmdl VDMOS VTO=5 Cgdmin=10p Cgdmax=100p
```

- A
  - A : Additional non-linear Gate-drain capacitance abruptness
  - **Default A=1**
  - Modify Cgd vs drain voltage profile

```
.param f=10Meg
.ac list f
.step param Vdrain -10 50 0.5
.step param a list 0.2 0.4 1
```

Calculate Capacitance

```
.func imZ() imag(V(d)/Ig(M1))
.func Ceqv() -1/2/pi/f/imZ()
.plot Ceqv()
```

```
.model Mmdl VDMOS VTO=5 Cgdmin=10p Cgdmax=100p A=a
```

M. MOSFET Params :

Qspice : VDMOS - BodeDiode.qsch / VDMOS - Rb Rd Rds Rg Rs.qsch

- Resistance
  - Rb : Bulk resistance
  - Rd : Drain resistance
  - Rds : Additional Drain-Source shunt resistance
  - Rg : Gate resistance
  - Rs : Source resistance
  - **Default Rb=Rd=Rg=Rs=0Ω**
  - **Default Rds=Infinite**

- Body Diode
  - Parameters same as diode model parameters
  - Refer to section : D. Diode Diode Model Parameters
  - This idea diode also provide Cds capacitance with its junction capacitance Cj

.dc Vdrain -1 4 0.01
.plot Id(M1)

.model Mmdl VDMOS VTO=5 N=0.3

M. MOSFET : Drain-Source Turn ON Resistance : Rds,on

- Rds,on
  - This is not a direct parameters, but as to design SMPS, you may want to know what parameters in MOSFET model affect Rds,on
  - In SMPS, Rds,on is generally considered as MOSFET fully turn ON, where
    - $V_{GS} \gg V_{TH}$ and V_{DS} is low, MOSFET in linear region, and $R_{DS,on} = \frac{V_{DS}}{I_{DS}}$
  - Linear region formula : $V_{GS} > V_{TH}$ and $V_{DS} < V_{GS} - V_{TH}$
    - $I_{DS} = KP \frac{W}{L-2X_{jl}} \left(V_{GS} - V_{TH} - \frac{V_{DS}}{2}\right) V_{DS}(1 + \lambda V_{DS})$
  - Therefore
    - $R_{DS,on} = \frac{V_{DS}}{I_{DS}} = \frac{V_{DS}}{KP\frac{W}{L-2X_{jl}}\left(V_{GS}-V_{TH}-\frac{V_{DS}}{2}\right)V_{DS}(1+\lambda V_{DS})} = \frac{1}{KP\frac{W}{L-2X_{jl}}\left(V_{GS}-V_{TH}-\frac{V_{DS}}{2}\right)(1+\lambda V_{DS})}$
    - $\lim_{V_{DS}\to 0} R_{DS,on} = \frac{1}{KP\frac{W}{L-2X_{jl}}\left(V_{GS}-V_{TH}-\frac{0}{2}\right)(1+\lambda\times 0)} = \frac{1}{KP\frac{W}{L-2X_{jl}}(V_{GS}-V_{TH})} = \frac{L-2X_{jl}}{KP\cdot W\cdot(V_{GS}-V_{TH})}$
  - With Qspice model parameters, also consider drain and source resistance in model
    - $@V_{DS} \to 0\text{V} : R_{DS,on} = \frac{\text{L}-2\,\text{LD}}{\text{KP}\cdot\text{W}\cdot(V_{GS}-\text{VTO})} + \text{RD} + \text{RS}$
      - Where in default, VTO=0, KP=1, L=1e-4, W=1e-4, LD=0, Rd=0 and Rs=0

M. MOSFET : Drain-Source Turn ON Resistance : Rds,on

Qspice : VDMOS - (Calculate) Rdson.qsch

- Example #1
  - Mmdl model
  - $@V_{DS} \rightarrow 0V$
  - $R_{DS,on} = \frac{L - 2\,LD}{KP \cdot W \cdot (V_{GS} - VTO)} + RD + RS$
    - RD=0
    - RS=0
  - $R_{DS,on}@V_{DS} \rightarrow 0V = 141.67m\Omega$

Calculate Rds,on

.dc Vdrain 1p 5 0.01 .plot V(d)/Id(M1)

.model Mmdl VDMOS VTO=4 KP=2 L=4e-4 W=2e-4 LD=3e-5

.model IRF530 VDMOS Rg=3 Vto=4 Rd=50m Rs=12m
+Rb=60m Kp=5 lambda=.01 Cgdmax=1n Cgdmin=.26n
+Cgs=.2n Cjo=.4n Is=52p ksubthres=.1
+mfg=International_Rectifier Vds=100 Ron=160m Qg=26n

- Example #2
  - Mmdl model
  - $@V_{DS} \rightarrow 0V$
  - $R_{DS,on} = \frac{L - 2\,LD}{KP \cdot W \cdot (V_{GS} - VTO)} + RD + RS$
    - VTO=4, KP=5
    - L=W=1e-4, LD=0
    - RD=50m
    - RS=12m
  - $R_{DS,on}@V_{DS} \rightarrow 0V = 95.33m\Omega$

Calculate Rds,on

.dc Vdrain 1p 5 0.01 .plot V(d)/Id(M1)

.model Mmdl VDMOS VTO=4 KP=2 L=4e-4 W=2e-4 LD=3e-5

.model IRF530 VDMOS Rg=3 Vto=4 Rd=50m Rs=12m
+Rb=60m Kp=5 lambda=.01 Cgdmax=1n Cgdmin=.26n
+Cgs=.2n Cjo=.4n Is=52p ksubthres=.1
+mfg=International_Rectifier Vds=100 Ron=160m Qg=26n

M. MOSFET
Level 2010 (SiC FET)
Model

M. MOSFET Level 2010 (SiC FET) Model

Name	Description	Units	Default
A	Sharpness of transistion from CGDMIN to CGDMAX		1.0
AF	MOSFET Flicker noise exponent		1.0
BV	Body diode breakdown voltage	V	Infinite
CGDMAX	Gate-Drain maximum capacitance	F	0.0
CGDMIN	Gate-Drain minimun capacitance	F	0.0
CGS	Gate-source overlap capacitance	F	0.0
CJO	Body diode zero bias cap	F	0.0
EG	Body diode activation energy	V	1.11
ETA	Philips, et al.-style subthreshold conduction	V	0.0
ETATC1	ETA first order tempco	$V/°C^{-1}$	0.0
ETATC2	ETA second order tempco	$V/°C^{-2}$	0.0
GDSNOI	Noise equation selector		0
GMAX	Maximum conductivity of any PN junction	℧	1000.
IBV	Body diode current at breakdown voltage	A	1e-10
IGSS	Gate-Source leakage	A	0.0
IS	Body diode saturation current	A	1e-14
JAF	JFET Flicker Noise Exponent		1.0
JALPHA	JFET impact ionization coefficient	V^{-1}	0.0
JB	JFET doping tail parameter		0.0
JBETA	JFET transconductance parameter	A/V^2	1.0
JBETATCE	JFET exponential temperature coefficient	%/°C	0.0
JCGD	JFET G-D junction cap	F	0.0
JCGDO	JFET G-D junction cap	F	0.0
JCGS	JFET G-S junction capactance	F	0.0
JCGSO	JFET G-S junction capactance	F	0.0
JFC	JFET forward bias junction fit parm.		0.5
JIS	JFET gate junction saturation current	A	1e-14
JISR	JFET recombination current parameter	A	0.0
JKF	JFET Flicker Noise Coefficient		0.0
JLAMBDA	JFET Channel length modulation parameter	V^{-1}	0.0
JM	JFET grading coeff.		0.5
JN	JFET emission coefficient		1.0
JNR	JFET Isr emission coefficient		2.0
JPB	JFET gate junction potential	V	1.0
JRD	JFET drain ohmic resistance	Ω	0.0
JRDS	JFET additional Drain-Source leakage resistance	Ω	Infinite
JRDSTC1	JFET RDS first order tempco	$°C^{-1}$	0.0
JRDSTC2	JFET RDS second order tempco	$°C^{-2}$	0.0
JRDTC1	JFET RD first order tempco	$°C^{-1}$	0.0
JRDTC2	JFET RD second order tempco	$°C^{-2}$	0.0
JRG	JFET gate ohmic resistance	Ω	0.0
JRGTC1	JFET RG first order tempco	$°C^{-1}$	0.0
JRGTC2	JFET RG second order tempco	$°C^{-2}$	0.0
JRONX	JEFT channel conductivity multiplier in linear region		1.0
JRS	JFET source ohmic resistance	Ω	0.0
JRSTC1	JFET RS first order tempco	$°C^{-1}$	0.0
JRSTC2	JFET RS second order tempco	$°C^{-2}$	0.0
JVK	JFET ionization knee current	V	0.0
JVTO	JFET threshold voltage	V	-2.0
JVTOTC	JFET Vto's temperature coefficient	$°C^{-1}$	0.0
JXTI	JFET IS temperature coefficient		3.0
KF	MOSFET flicker noise coefficient		0.0
KP	MOSFET transconductance parameter	A/V^2	1.0
LAMBDA	MOSFET channel length modulation	V^{-1}	0.0
LG	JFET gate inductance	H	0.0
LGRPAR	JFET gate inductance parallel loss	Ω	Infinite
LMID	Inductance making cascode conection	H	0.0
LS	MOSFET Source Inductance	H	0.0
LSRPAR	MOSFET Source inductance parallel loss	Ω	Infinite
LMIDRPAR	Cascode inductance parallel loss	Ω	Infinite
M	Body diode grading coefficient(MJ)		0.5
N	Bulk diode emission coefficient		1.0
NBV	Body diode breakdown emission coeff		1.0
NLEV	MOSFET noise equation selecctor		0
NOISELESS	The device does not contribute to a .noise		
RB	Body diode seris resistance	Ω	0.0
RBTC1	MOSFET RB first order tempco	$°C^{-1}$	0.0
RBTC2	MOSFET RB second order tempco	$°C^{-2}$	0.0
RD	MOSFET drain ohmic resistance	Ω	
RDS	MOSFET drain-source shunt resistance	Ω	Infinite
RDTC1	MOSFET RD first order tempco	$°C^{-1}$	0.0
RDTC2	MOSFET RD second order order	$°C^{-2}$	0.0
RG	MOSFET gate resistance	Ω	0.0
RONX	MOSFET conductivity multiplier in linear region		1.0
RS	MOSFET Source ohmic resistance	Ω	0.0
RSTC1	MOSFET RS first order tempco	$°C^{-1}$	0.0
RSTC2	MOSFET RS second order tempco	$°C^{-1}$	0.0
THETA	MOSFET Vgs dependence on mobility ala mos3	V^{-1}	0.0
TNOM	Parameter measurement temperature(TNOM)	°C	27.0
TRB1	MOSFET RB first order tempco	$°C^{-1}$	0.0
TRB2	MOSFET RB second order tempco	$°C^{-2}$	0.0
TRD1	MOSFET RD first order tempco	$°C^{-1}$	0.0
TRD2	MOSFET RD second order tempco	$°C^{-2}$	0.0
TRS1	MOSFET RS first order tempco	$°C^{-1}$	0.0
TRS2	MOSFET RS second order tempco	$°C^{-2}$	0.0
TT	Bulk diode transit time	s	0.0
VJ	Body diode Junction potential	V	1.0
VTO	MOSFET threshold voltage	V	0.0
VTOTC	MOSFET's VTO tempco	$°C^{-1}$	0.0
XTI	Body diode Saturation current temperature exp.		3.0

M. MOSFET Level 2010 (SiC FET) Cascode Configuration

- MOSFET Level 2010 (SiC FET) Model Parameters
  - SiC Cascode Configuration for SiC FET, which contains a JFET and MOSFET
    - SiC JFET
    - Si MOSFET
  - Reference
    - "Cascode Configuration Eases Challenges of Applying SiC JFETs", United SiC, App Note USCi_AN0004
    - "Origins of SiC FETs and their evolution towards the perfect switch", United SiC, White Paper

Normally Off UnitedSiC Cascode		Normally Off Typical SiC MOSFET
Die Size	(Smaller) R_{DSA}~ 1.75mΩ-cm²	(Larger) R_{DSA}~ 3.1- 4.5 mΩ-cm²
Gate Drive	(Standard) V_{GS} = 0V to 12V OR (SIC) V_{GS} = -10V to 20V	V_{GS} = -5V to 20V
Threshold	$V_{GS(TH)}$ = 5V Typical	$V_{GS(TH)}$ = 2.2V Typical
Intrinsic Diode	Low Qrr, +10% Over Temperature	High Qrr, High VF 3X Over Temperature
Avalanche	Yes	Yes
Short Circuit	Yes	Low

M. MOSFET Level 2010 : Reverse Current

Qspice : UF3C065080T3S Reverse Current.qsch

- Reverse Current
  - TT (transit-time) is set to 0 in Qorvo SiC FET, reverse recovery of bulk diode is not be used
  - Reverse current is simulated by parasitic capacitance in SiC FET
  - From Qorvo SiC FET model, reverse current is mainly contributed from Cgs and Cgd of JFET
    - jCgs, jCgso
    - jCgd, jCgdo

Disable JFE Cgs and Cgd

zero : Cgs, Cgdmin, Cgdmax, Cjo, jCgs, jCgd, jCgso, jCgdo
jCgs, jCgd, jCgso, jCgdo can almost reassemble reverse I profile

```
.model UF3C065080T3Srr NMOS level=2010 Rs=1.36m Rd=1.36m Rg=4.5 Vto=5.77 Kp=95.1
+ Lambda=0.306 RonX=2.13 eta=.1 Is=404f N=1.00935 Rb=2.02m XTI=1.94703 Vtotc=-4m
+ trd1=4m trs1=4m Cgs=1.5n*0 Cgdmin=300p*0 Cgdmax=750p*0 Cjo=500p*0 m=.9 trb1=-1.64m
+ NBV=5 IBV=1m BV=20 Lg=4n LgRpar=62.8 Ls=1.5n LsRpar=23.6 Lmid=2n LmidRpar=31.4
+ jIs=1p jBeta=1 jVto=-9 jRonX=2 jRs=35m jRd=15m jRds=1G jCgs=3000p*n jCgd=1800p*n
+ jCgso=250p*n jCgdo=50p*n jM=.8 jLambda=.2 jRsTC1=10m jRdTC1=10m jRg=5 Vds=650
+ Ids=31 Ron=80m Qg=51n mfg=Qorvo tt=0
```

Enable JFET Cgs and Cgd

zero : Cgs, Cgdmin, Cgdmax, Cjo, jCgs, jCgd, jCgso, jCgdo
jCgs, jCgd, jCgso, jCgdo can almost reassemble reverse I profile

```
.model UF3C065080T3Srr NMOS level=2010 Rs=1.36m Rd=1.36m Rg=4.5 Vto=5.77 Kp=95.1
+ Lambda=0.306 RonX=2.13 eta=.1 Is=404f N=1.00935 Rb=2.02m XTI=1.94703 Vtotc=-4m
+ trd1=4m trs1=4m Cgs=1.5n*0 Cgdmin=300p*0 Cgdmax=750p*0 Cjo=500p*0 m=.9 trb1=-1.64m
+ NBV=5 IBV=1m BV=20 Lg=4n LgRpar=62.8 Ls=1.5n LsRpar=23.6 Lmid=2n LmidRpar=31.4
+ jIs=1p jBeta=1 jVto=-9 jRonX=2 jRs=35m jRd=15m jRds=1G jCgs=3000p*n jCgd=1800p*n
+ jCgso=250p*n jCgdo=50p*n jM=.8 jLambda=.2 jRsTC1=10m jRdTC1=10m jRg=5 Vds=650
+ Ids=31 Ron=80m Qg=51n mfg=Qorvo tt=0
```

Q. Bipolar Transistor (Level 1, Extended SPICE Gummel-Poon)

Q. Bipolar Transistor : NPN and PNP

Voltage Relations	NPN
$V_E < V_B < V_C$	Active
$V_E < V_B > V_C$	Saturation
$V_E > V_B < V_C$	Cutoff
$V_E > V_B > V_C$	Reverse

Voltage Relations	PNP
$V_E < V_B < V_C$	Reverse
$V_E < V_B > V_C$	Cutoff
$V_E > V_B < V_C$	Saturation
$V_E > V_B > V_C$	Active

Q. Bipolar Transistor : NPN and PNP : Active, Saturation and Cutoff

Q. Bipolar Transistor Model Parameters

Bipolar Transistor Model Parameters

Name	Description	Units	Default
AF	Flicker Noise Exponent		1.0
BF	Ideal forward beta		100.
BR	Ideal reverse beta		1.0
BRS	Substrate parasitic transistor beta		.0
CJC	Zero bias B-C capacitance	F	0.0
CJE	Zero bias B-E capacitance	F	0.0
CJS	Zero bias C-S capacitance(aka CCS)	F	0.0
CN[1]	Tempco for hole mobility	$°C^{-1}$	NPN: 2.42 PNP: 2.20
D[1]	Tempco for scattering-limited hole carrier velocity	V^{-1}	NPN: 0.87 PNP: 0.52
EG	Energy gap for IS temp. dependency	V	1.11
FC	Forward bias junction fit parameter		0.5
GAMMA[1]	Epitaxial region doping factor		1e-11
GMAX	Maximum PN conductivity(straight-line extension)	℧	1000.
IBC	B-C saturation current	A	0.0
IBE	B-E saturation current	A	0.0
IKF	Forward beta roll-off corner current(aka IK)	A	Infinite
IKR	Reverse beta roll-off corner current	A	Infinite
IRB	Current for base resistance=(rb+rbm)/2	A	Infinite
IS	Saturation Current	A	1e-16
ISC	B-C leakage saturation current(aka C4)	A	IS
ISE	B-E leakage saturation current(aka C2)	A	IS
ISS	Substrate junction saturation current	A	0.0[2]
ITF	High current dependence of TF	A	0.0
KF	Flicker Noise Coefficient		0.33
MJC	B-C junction grading coefficient(aka MC)		0.33
MJE	B-E junction grading coefficient(aka ME)		0.0
MJS	Substrate junction grading coefficient(aka MS)		0.0
NC	B-C leakage emission coefficient		2.0
NE	B-E leakage emission coefficient		1.5
NEPI[1]	Epitaxial region emission coefficient		1.0
NF	Forward emission coefficient		1.0
NK	High-current roll-off coefficient		0.5
NR	Reverse emission coefficient		1.0
NS	Substrate junction emission coefficient		1.0
PTF	Excess phase	°	0.0
QCO[1]	Epitaxial region charge factor	C	0.0
QUASIMOD[1]	Temperature dependence equation selector		0
RB	Zero bias base resistance	Ω	0.0
RBM	Minimum base resistance	Ω	0.0
RC	Collector resistance	Ω	0.0
RCO[1]	Epitaxial region resistance	Ω	0.0
RE	Emitter resistance	Ω	0.0
SUBS	Set to 2 to specify lateral geometry.		1
TF	Ideal forward transit time	sec	0.0
TNOM	Parameter measurement temperature(aka TREF)	°C	27.0
TR	Ideal reverse transit time	sec	0.0
TRB1	Rb 1st order temperature coefficient	$°C^{-1}$	0.0
TRB2	Rb 2nd order temperature coefficient	$°C^{-2}$	0.0
TRBM1	Rbm 1st order temperature coefficient(aka TRM1)	$°C^{-1}$	0.0
TRBM2	Rbm 2nd order temperature coefficient(aka TRM2)	$°C^{-2}$	0.0
TRC1	RC 1st order temperature coefficient	$°C^{-1}$	0.0
TRC2	RC 2nd order temperature coefficient	$°C^{-2}$	0.0
TRE1	RE 1st order temperature coefficient	$°C^{-1}$	0.0
TRE2	RE 2nd order temperature coefficient	$°C^{-2}$	0.0
TVAF1	1st order tempco of forward Early voltage	$V/°C^{1}$	0.0
TVAF2	2nd order tempco of forward Early voltage	$V/°C^{2}$	0.0
TVAR1	1st order tempco of reverse Early voltage	$V/°C^{1}$	0.0
TVAR2	2nd order tempco of reverse Early voltage	$V/°C^{2}$	0.0
VAF	Forward Early voltage(aka VA)	V	Infinite
VAR	Reverse Early voltage(aka VB)	V	Infinite
VG[1]	Extrapolated band gap voltage at 0°K	V	1.206
VJC	B-C junction built in potential(aka PC)	V	0.75
VJE	B-E junction built in potential(aka PE)	V	0.75
VJS	Substrate junction built in potential(aka PS)	V	0.75
VO[1]	Carrier mobility knee voltage	V	10.0
VTF	Voltage giving VBC dependence of TF	V	Infinite
XCJC	Fraction of CJC connected to intrinsic base[3]		1.0
XCJC2	Fraction of CJC connected to intrinsic base[4]		1.0
XCJS	Fraction of CJS connected internally to Rc		1.0
XTB	Forward and reverse beta temp. exp.		0.0
XTF	Coefficient for bias dependence of TF		0.0
XTI	Temperature exponent for IS(aka PT)		3.0

Q. Bipolar Transistor Basic Equation

- Semiconductor Device Modeling with SPICE (Section 2.2.1)
  - Basic equation from BJT Ebers-Moll static model
    - $I_C = I_{CT} - \frac{I_{EC}}{\beta_R}$
      - $I_{CT} = I_{CC} - I_{EC}$
    - $I_E = -\frac{I_{CC}}{\beta_F} - I_{CT}$
    - $I_B = \frac{I_{CC}}{\beta_F} + \frac{I_{EC}}{\beta_R}$
  - Therefore
    - $I_C = I_{CC} - I_{EC} - \frac{I_{EC}}{\beta_R} = \beta_F\left(I_B - \frac{I_{EC}}{\beta_R}\right) - I_{EC} - \frac{I_{EC}}{\beta_R}$
    - $I_C = \beta_F I_B - \left(\frac{\beta_F}{\beta_R} + \frac{1}{\beta_R} + 1\right) I_{EC}$

- Understanding (Approximation approach)
  - If $V_{CE} \gg V_{BE}$, diode D_{BC} is reverse bias, where $I_{EC} \to 0$
    - $I_C = \beta_F I_B$
  - If $V_{BE} \gg V_{CE}$, diode D_{BC} is forward bias, and with $\beta_F \gg \beta_R$
    - $I_C = \beta_F I_B - \frac{\beta_F}{\beta_R} I_{EC}$

Q. Bipolar Params ($\beta_F|\beta_R$) : BF and BR

Qspice : NPN - BF.qsch / NPN - BR.qsch

- BF
  - BF : Ideal forward beta
  - **Default BF=100**
  - $I_C = \beta_F I_B$ for $V_{CE} \gg V_{BE}$

- BR
  - BR : Ideal reverse beta
  - **Default BR=1**
  - $I_C = \beta_F I_B - \frac{\beta_F}{\beta_R} I_{EC}$
    - This effect is mainly observed when both BE and BC diode are forward bias, where collector-emitter voltage is close to 0V
    - Smaller β_R, larger $\frac{\beta_F}{\beta_R}$ and less I_C and less $\frac{I_C}{I_B}$

Sweep Parameters
.param Vcollector = 10
.param Ibmax=10µ

Plot Output
.plot Ic(Q1)/Ib(Q1)
.plot Ic(Q1)

.dc Ibsrc 10n {Ibmax} {Ibmax/100}

.step param bf list 100 200 300
.model Qmdl NPN BF={bf}

Sweep Parameters
.param Vcollector = 0.2
.param Ibmin=10p

Plot Output
.plot Ic(Q1)/Ib(Q1)
.plot V(B)

.dc Vbscr 0.2 0.6 0.001

.step param br list 0.5 1 2
.model Qmdl NPN BR={br} BF=100

Q. Bipolar Params ($\beta_F|\beta_R$) : IKF

Qspice : NPN - IKF.qsch / NPN - IKR.qsch

- IKF
  - IKF : Forward beta roll-off corner current
  - **Default IKF=infinite (A)**
  - IKF models β_F drop when I_C increase
  - IKF is I_C where effective BF (β_F) reduces multiple approximate by (1-NK)
  - This example Default NK=0.5, effective BF is reduced to 50 @ Ic=IKF

Sweep Parameters
.param Vcollector = 10
.param Ibmax=20µ

Plot Output
.plot Ic(Q1)/Ib(Q1)
.plot Ic(Q1)

.dc Ibsrc 10n {Ibmax} {Ibmax/100}

$$\beta_{F,\text{eff}} = \frac{I_C}{I_B}$$

.step param ikf list 0.2m 0.6m 1e12
.model Qmdl NPN BF=100 IKF=ikf

- IKR
  - IKR : Reverse beta roll-off corner current
  - **Default IKR=infinite (A)**
  - IKR models β_F drop when I_C increase
  - IKR is $\frac{I_C}{BF}$ where effective BR (β_R) reduces to 50%

Sweep Parameters
.param Vcollector = 0.2
.param Ibmin=10p

Plot Output
.plot Ic(Q1)/Ib(Q1)
.plot Ic(Q1)/bf

.dc Vbscr 0.2 0.6 0.001

.param bf=100
.step param ikr list 1n 4n 1e12
.model Qmdl NPN BR=1 BF=bf IKR=ikr

Q. Bipolar Params ($\beta_F|\beta_R$) : NK

Qspice : NPN - NK.qsch

- NK
  - NK : High-current roll-off coefficient
  - **Default NK=0.5**
  - This parameter is only be used if IKF not infinite

Sweep Parameters
.param Vcollector = 10
.param Ibmax=20µ

Plot Output
.plot Ic(Q1)/Ib(Q1)
.plot Ic(Q1)

.dc Ibsrc 10n {Ibmax} {Ibmax/100}

.step param nk list 0.2 0.5 0.7
.model Qmdl NPN BF=100 IKF=200µ NK=nk

Q. Bipolar Params (B-E and B-C Diode) : IS, NF, BF, NR, BR

Qspice : NPN - IS NF BF.qsch / NPN - IS NR BR.qsch

- Diode of Base-Emitter
  - IS : Saturation Current
  - NF : Forward emission coefficient
  - **Default IS = 1e-16A**
  - **Default NF = 1**
  - In this case
    - $I_B = I_{BE} = \frac{I_{CC}}{\beta_F}$
    - $= \frac{1}{\beta_F} I_S \left(e^{\frac{qV_{BE}}{n_F kT}} - 1\right)$
    - where β_F is BF

Sweep Parameters

.param Vcollector = 10

.param Vbmax=0.8

.dc Vbsrc 0n {Vbmax} {Vbmax/100}

Plot Output

.plot Ib(Q1)

** Reverse Bias D_{BC}

.model Qmdl NPN IS=1e-16 NF=0.7 BF=100

- Diode of Base-Collector
  - IS : Saturation Current
  - NR : Reverse emission coefficient
  - **Default IS = 1e-16A**
  - **Default NR = 1**
  - In this case
    - $I_B = I_{BC} = \frac{I_{EC}}{\beta_R}$
    - $= \frac{1}{\beta_R} I_S \left(e^{\frac{qV_{BE}}{n_R kT}} - 1\right)$
    - where β_R is BR

Sweep Parameters

.param Vcollector = 10

.param Vbmax=0.8

.dc Vbsrc 0n {Vbmax} {Vbmax/100}

Plot Output

.plot Ib(Q1)

** Reverse Bias D_{BE}

.model Qmdl NPN IS=1e-16 NR=0.7 BR=1

** IS and N explanation refer to Diode .model section

Q. Bipolar Params (Active Region) : VAF

Qspice : NPN - VAF.qsch

- VAF (aka VA)
  - VAF : Forward Early voltage
  - **Default VAF=Infinite (V)**

Q. Bipolar Params (Resistance): RB, RC, RE, RBM and IRB

Qspice : NPN - RBM IRB.qsch

- RB, RC, RE
  - RB : Zero bias base resistance
    - If RBM and IRB are used, base resistor is current dependent
  - RC : Collector resistance
  - RE : Emitter resistance

- RBM, IRB
  - RBM : Minimum base resistance
  - IRB : Current for base resistance
    - $R_{B,effective}@\mathrm{IRB} = \frac{1}{2}(\mathrm{RB} + \mathrm{RBM})$
  - Default RBM=0Ω
  - Default IRB=Infinite(A)
  - If these 2 parameters are defined, base resistance is function of base current

Sweep Parameters
.param Vcollector = 10
.param Vbmax=1

Plot Output
.plot Ib(Q1) Ib(Q2)

.dc Vbsrc 0n {Vbmax} {Vbmax/100}

.model Qmdl1 NPN IS=1e-16 NF=1μ BF=1 RB=1

.model Qmdl2 NPN IS=1e-16 NF=1μ BF=1 RB=1 RBM=0.5 IRB=0.5

Q. Bipolar Params (Capacitance) : CJC, MJC, VJC, CJE, MJE, VJE

- B-C, B-E and C-S nonlinear capacitance
  - Model nonlinear capacitor between B-C, B-E and C-S
  - Parameters definition follows diode CJO, M and VJ
  - $C_j = \frac{\text{CJO}}{\left(1-\frac{V_D}{\text{VJ}}\right)^{\text{M}}}$
    - CJO : Zero-bias junction capacitance
    - M : Grading coefficient
    - VJ : Junction potential

Diode	CJO	M	VJ
B-C	CJC	MJC	VJC
B-E	CJE	MJE	VJE
C-S	CJS	MJS	VJS

- Refer to D. Diode CJO, M and VJ
- C-S normally not modeled

```
.param f=1Meg
.ac list f
.step param bias -2 0.6 0.1
.step param VCE list 0 2 10 50
```

Calculate Equivalent C

```
.func Zim() imag(V(B)/Ib(Q1))
.func Ceqv() -1/2/pi/f/Zim()
.plot Ceqv()
```

```
.model Qmdl NPN CJE=100p MJE=0.5 VJE=1
+CJC=100p MJC=0.5 VJC=1
```

Q. Bipolar Params (Capacitance) : CJC, MJC, VJC, CJE, MJE, VJE

Qspice : NPN - CJC CJE.qsch

- B-C, B-E and C-S nonlinear capacitance
  - Model nonlinear capacitor between B-C, B-E and C-S
  - Parameters definition follows diode CJO, M and VJ
  - $C_j = \frac{CJO}{\left(1-\frac{V_D}{VJ}\right)^M}$
    - CJO : Zero-bias junction capacitance
    - M : Grading coefficient
    - VJ : Junction potential

- Simulation
  - This simulation modeled both capacitance between B-C and B-E. Change VCE to observe overall capacitance effect in different bias voltage

```
.param f=1Meg
.ac list f
.step param bias -2 0.6 0.1

.step param VCE list 0 2 10 50
```

```
.model Qmdl NPN CJE=100p MJE=0.5 VJE=1
+CJC=100p MJC=0.5 VJC=1
```

Calculate Equivalent C

```
.func Zim() imag(V(B)/Ib(Q1))
.func Ceqv() -1/2/pi/f/Zim()
.plot Ceqv()
```

Diode	CJO	M	VJ
B-C	CJC	MJC	VJC
B-E	CJE	MJE	VJE
C-S	CJS	MJS	VJS

- Refer to D. Diode CJO, M and VJ
- C-S normally not modeled

- Simulation Explanation
  - CBC and CBE are nonlinear capacitance of junction diode from B to C and B to E
  - Higher reverse voltage, lower capacitance
  - Therefore, increase VCE, increase reverse voltage of B-C and affects B-C capacitance. However, increase VCE doesn't affect VBE, and no effects in B-E capacitance
  - @VB=0V, when VCE=0, no reverse in both B-C and B-E and capacitance equals CJC+CJE

Q. Bipolar Params (Capacitance) : CJC, MJC, VJC, CJE, MJE, VJE

Qspice : NPN - CJC MJC VJC.qsch / NPN - CJE MJE VJE.qsch

- CJC, MJC and VJC
  - CJC : Zero bias B-C capacitance
  - MJC : B-C junction grading coefficient
  - VJC : B-C junction built in potential
  - **Default CJC=0F**
  - **Default MJC=0.5**
  - **Default VJC=1**

.param f=1Meg
.ac list f
.step param bias -10 0.6 0.1

Calculate Equivalent C
.func Zim() imag(V(B)/Ib(Q1))
.func Ceqv() -1/2/pi/f/Zim()
.plot Ceqv()

.model Qmdl NPN CJC=100p MJC=0.5 VJC=1

- CJE, MJE and VJE
  - CJE : Zero bias B-E capacitance
  - MJE : B-E junction grading coefficient
  - VJE : B-E junction built in potential
  - **Default CJE=0F**
  - **Default MJE=0.5**
  - **Default VJE=1**

.param f=1Meg
.ac list f
.step param bias -10 0.6 0.1

Calculate Equivalent C
.func Zim() imag(V(B)/Ib(Q1))
.func Ceqv() -1/2/pi/f/Zim()
.plot Ceqv()

.model Qmdl NPN CJE=100p MJE=0.5 VJE=1

Q. Bipolar Params (B-E Diode) : BVBE, IBVBE, NBVBE

Qspice : NPN - BVBE IBVBE NBVBE.qsch

- BVBE, IBVBE, NBVBE
  - Qspice Revision History
    - 12/30/2023 Implemented bipolar B-E breakdown via model parameters BVBE, IBVBE, AND NBVBE.
  - Parameters to model breakdown characteristic of Base-Emitter Diode
  - BVBE : Reverse breakdown voltage
  - IBVBE : Current at breakdown voltage
  - NBVBE : Emission coefficient
  - **Default BVBE=0V**
  - **Default IBVBE=1e-10A**
  - **Default NBVBE=2**

Default Value
BVBE=0
IBVBE=1e-10
NBVBE=2

```
.model Qmdl NPN BVBE=2 IBVBE=0.1 NBVBE={NBVBE}
.step param NBVBE list 2 4 6
.dc V1 -2.2 1 0.01
.plot Ib(Q1)
```

Q. Bipolar Transistor (MEXTRAM 504, Level 504 by NXP)

Q. Transistor (MEXTRAM 504)

- MEXTRAM 504
  - Release history
    - Qspice Revision History : 11/29/2023 Released a MEXTRAM 504(rev 12) implementation
    - https://forum.qorvo.com/t/models-of-transistors-with-thermal-parameters/15809/5
  - Comment from Mike Engelhardt
    - It's the current minor revision level, 12
    - It should be able to properly run that model without the collector current going to zero
    - If no self heating node is specified, it will make one called Q1.DT. Otherwise you can make a five terminal transistor and connect a thermal network the temperature node

```
.dc V1 0 12 1m I1 10µ 50µ 10µ
.plot Ic(Q1)
.plot Q1.DT

.model KSC1845F_504 npn level=504 is=1.075431e-013 bf=362.1
+ bri=1.3565 vef=267 ver=20.6691 ik=0.25 ibf=1e-012 ibr=1e-012 mlf=2 rbv=144.908
+ rbc=12.092 re=1.5 rcc=1 rcv=150 scrcv=225 ihc=1.5m axi=.1 cje=7p cjc=4.525739e-012
+ pe=0.36 pc=0.3659045 vde=0.6 vdc=0.5 taue=250p taub=2.1e-011 af=1 kf=0 vgb=1.1809
+ vgc=1.1809 vgs=1.1809 vgj=1.1809 exavl=0 wavl=1K Xcje=0.25 Cbeo=2p
+ Rth=100 Cth=100n
```

R. Resistor

R. Resistor Instance and Model Parameters

- Resistor Syntax
  - Rnnn N1 N2 [<model>] <resistance> [instance parameters]
  - .model <model> RES [model parameters]

Resistor Instance Parameters

Name	Description	Units	Default
AC	Value to use for .ac analysis	Ω	RESISTANCE
B[1]	NTC coefficient		0.0
CTH[2]	Thermal capacitance	F	Model value
RESISTANCE	Resistance	Ω	none
RTH[2]	Thermal resistance	Ω	Model value
L[3]	Length	m	1.0
W[3]	Width	m	1.0
M	Number of identical parallel instances		1.0
TC[1]	Polynomial temperature coefficient list		0.0
TC1[1]	1nd order temperature coefficient	$°C^{-1}$	0.0
TC2[1]	2nd order temperature coefficient	$°C^{-2}$	0.0
TEMP	Instance temperature	°C	Circuit temperature
TNOM	Temperature resistance was measured(aka TREF)	°C	Circuit TNOM

Resistor Model Parameters

Name	Description	Units	Default
CTH[2]	Thermal capacitance	F	0.0
DEFW	Default device width	m	1.0
NARROW[3]	Narrowing of resistor	m	1.0
R	Resistor multiplier		1.0
RTH[2]	Thermal resistance	Ω	0.0
RSH[3]	Sheet resistance(for instance L and W)	Ω/□	0.0
TC[1]	Polynomial temperature coefficient list		0.0
TC1[1]	1nd order temperature coefficient	$°C^{-1}$	0.0
TC2[1]	2nd order temperature coefficient	$°C^{-2}$	0.0
TCE[1]	Exponential order temperature coefficient	%/°C	0.0
TNOM	Parameter measurement temperature(aka TREF)	°C	Circuit TNOM

R. Resistor – Resistance and AC

Qspice : R - Resistance.qsch | R - AC.qsch

- Resistance
  - Resistance can be specified in three way
    - #1 : Resistance value at 1st attribute
    - #2 : <model> at 1st attribute and resistance value at 2nd attribute
    - #3 : <model> at 1st attribute, resistance value after 2nd attribute with Resistance=<value>

- AC
  - Value to use for .ac analysis
  - **Default = RESISTANCE**
  - In this example, .op or .tran set R1=1Kohms but .ac set R1=500ohms

R. Resistor – RSH, L and W (Sheet Resistance)

- RSH, L and W
  - RSH : Sheet Resistance
  - L : Length
  - W : Width
    - RSH is model param and L, W are instance params
  - If RESISTANCE is defined, RSH, L and W will be ignored
  - Sheet resistance is the resistance of a square of the conductive thin film with uniform thickness
  - $R = R_{SH} \times \frac{L}{W}$

R. Resistor – Thermal Related Parameters

- Resistor Thermal Equation
  - Formula with temperature coefficient list TC : $R_T = R_{nom}(1 + \sum T_{Cn}\Delta T^n)$
    - TC supports a list of temperature coefficient in format TC=<value1>,<value2>,<value3>,…
  - Formula with temperature coefficient TC1 and TC2 : $R_T = R_{nom}(1 + T_{C1}\Delta T + T_{C2}\Delta T^2)$
  - Formula with temperature coefficient TCE : $R_T = R_{nom}\, e^{\frac{T_{CE}}{100}\Delta T}$
  - Formula with B (or beta) for thermistor (NTC) : $R_T = R_{nom}\, e^{B\times\left(\frac{1}{T}-\frac{1}{T_{nom}}\right)}$
  - where Temperature in above formula is in Kelvin, $\Delta T = T - T_{nom}$

R. Resistor Params : Tnom and TEMP (Instance param), TC

Qspice : R - TNOM TEMP.qsch | R - TC.qsch

- Tnom and TEMP
  - Tnom : Parameter measurement temperature
  - TEMP : Instance temperature in Instance param
  - **Default Tnom=27oC**
  - **Default TEMP=Circuit** Temperature
  - In this example, instance temperature TEMP is assigned for R2, which forced R2 at TEMP instead of circuit temperature

```
.model Rmdl R Tnom=50 TC1=0.01
.plot V(R)/I(R1) V(R)/I(R2)
.dc Temp -25 105 1
.op
```

- TC
  - TC : Polynomial temperature coefficient list
  - **Default TC=0**
  - *TC=<value1>,<value2>,<value3>,...*
  - $R_T = R_{nom}(1 + \sum T_{Cn} \Delta T^n)$
  - where Tnom is R defined measurement temperature

```
.model Rmdl R TC=0.01,0.0001,0.000005
.plot V(R)/I(R1)
.dc Temp -25 105 0.001
.op
```

R. Resistor Params : TC1 and TC2

Qspice : R - TC1.qsch / R - TC2.qsch

- TC1
  - TC1 : 1nd order temperature coefficient
  - **Default TC1=0°C^{-1}**
  - RT = R*(1+TC1*(T-Tnom))
  - Where Tnom is R defined measurement temperature

```
.model Rmdl R TC1=0.01
.plot V(R)/I(R1)
.dc Temp -25 105 0.001
.op
```

- TC2
  - TC2 : 2nd order temperature coefficient
  - **Default TC2=0°C^{-2}**
  - RT = R*(1+TC2*(T-Tnom)^2)
  - TC2 and TC1 can be used together

```
.model Rmdl R TC2=0.0001
.plot V(R)/I(R1)
.dc Temp -25 105 1
.op
```

R. Resistor Instance Param : TCE and B for Thermistor (NTC)

Qspice : R - TCE.qsch | R - B.qsch

- TCE
  - TCE : Exponential order temperature coefficient
  - **Default TCE=0%/°C**
  - RT = R*exp((T-Tnom)/100*TCE)
  - If TCE is specified non-zero, TC1 and TC2 are ignored

```
.model Rmdl R TCE=1
.plot V(R)/I(R1) Rcal()
.dc Temp -25 105 1
.step param Temp -25 105 1
.op
.func Rcal() 1K*exp((Temp-27)/100*1)
```

- B (for thermistor/NTC)
  - NTC Coefficient
  - **Default B=0**
  - $R_T = R_{nom}\, e^{B\times\left(\frac{1}{T}-\frac{1}{T_{nom}}\right)}$
  - Refer to TDK NTC thermistor general information, B value of common NTC materials range is from 3000 to 5000

```
.op
.step param TEMP -80 240 1
.step param B 2000 5000 1000

.func R() V(R)/I(R1)
.plot R()
```

R. Resistor Params : Rth and Cth (Thermal Resistance and Capacitance)

Qspice : R - Rth Cth (Thermal Circuit).qsch

- RTH and CTH
  - RTH : Thermal Resistance
  - CTH : Thermal Capacitance
  - **Default RTH=0**
  - **Default CTH=0**
  - Cth and Cth supports both instance (e.g. R1) and .model parameters (e.g. R2)
  - DT (Delta Temperature ΔT)
    - DT is rise in temperature above ambient of resistor
    - 3-terminal resistor (3rd node in square braces) : ΔT is returned at 3rd node
    - 2-terminal resistor : ΔT is returned as Rnnn.DT

R. Resistor Params : Rth and Cth (Thermal Resistance and Capacitance)

Qspice : R - Rth Cth (Self Heating).qsch

- Self Heating Resistor
  - This is an example for resistor with positive temperature coefficient
  - Resistance and temperature relationship is controlled by temperature coefficient TC. Rth and Cth determine thermal response to achieve self heating simulation

- Thermistor Self Heating
  - This example use NTC coefficient B to setup a thermistor for self heating simulation
  - A loading resistor is required to limit maximum current for simulation to reach steady state with using NTC

R. Resistor : Behavioral Resistor (R)

Qspice : R - Behavioral R Demo 01.qsch | R - Behavioral R Demo 02.qsch

- Behavioral Resistor
  - To use formula or voltage/current node for resistance, type R=… in <val> of resistor
  - Also support to use Behavioral source (B) for behavioral resistor
    - Replace V=expression with R=expression

S. Voltage Controlled Switch

S. Voltage Controlled Switch Model Parameters

- S. Voltage Controlled Switch Model Parameters
  - It is common to define Switch Model for usage as multiple switches in a schematic
  - Syntax: Snnn N1 N2 NC+ NC- <model> [instance parameters]
  - .model <model> SW [model parameters]

Voltage Controlled Switch Model Parameters

Name	Description	Units	Default
ETA[1]	Sub-/over-Threshold conduction	V	0.0
ROFF	Off resistance	Ω	1e6
RON	On resistance	Ω	1.0
TTOL	Temporal tolerance	sec	1e308
VH	Hysteresis voltage	V	0.0
VOFF	Voltage when open	V	0.0
VON	Voltage when closed	V	0.0
VT	Threshold voltage	V	0.0

Input Control Voltage

S. Switch Model Params : Ron, Roff, VT, Von and Voff

Qspice : Switch - Ron Roff VT.qsch

- RON, ROFF, VT
  - Ron : On resistance
  - Roff : Off resistance
  - VT : Threshold voltage
  - **Default RON=1Ω**
  - **Default ROFF=1MegΩ**
  - **Default VT=0V**

```
.model mSW SW Ron=2 Roff=10 Vt=0.5 Vh=0
```

```
pulse 0 1 0 1 1 1 4
.tran 4
.func Rsw() V(sw)/I(I1)
.plot Rsw()
.plot V(ctrl)
```

- VON, VOFF
  - Von : Voltage when closed
  - Voff : Voltage when open
  - **Default VON=0V**
  - **Default VOFF=0V**
  - switch smoothly transitions between on and off
  - If VT and VH are specified, VON and VOFF are ignored

```
.model mSW SW Ron=2 Roff=10 Von=0.7 Voff=0.3
```

```
pulse 0 1 0 1 1 1 4
.tran 4
.func Rsw() V(sw)/I(I1)
.plot Rsw()
.plot V(ctrl)
```

S. Switch Model Params : VH

Qspice : Switch - VH.qsch

- VH
  - Vh : Hysteresis voltage
  - **Default VH=0V**
  - Switch is ON
    - V(+) - V(-) > Vt + Vh
  - Switch is OFF
    - V(+) - V(-) < Vt - Vh
  - Switch maintain its state
    - Vt - Vh < V(+) - V(-) < Vt + Vh

```
.model mSW SW Ron=2 Roff=10 Vt=0.5 Vh=0.2
```

+ve Vh

```
pulse 0 1 0 1 1 1 4
.tran 4
.func Rsw() V(sw)/I(I1)
.plot Rsw()
.plot V(ctrl)
```

- VH (-ve explained)
  - Negative Vh
  - If VH is negative the switch smoothly transitions between on and off

```
.model mSW SW Ron=2 Roff=10 Vt=0.5 Vh=-0.2
```

-ve Vh

```
pulse 0 1 0 1 1 1 4
.tran 4
.func Rsw() V(sw)/I(I1)
.plot Rsw()
.plot V(ctrl)
```

S. Switch Model Params : (VON, VOFF) vs (VT, VH)

- Change between [VON, VOFF] and [VT, VH]
  - Important Note
    - VH must be negative for smooth transition
    - VON and VOFF is Pspice syntax which must be a smooth transition
  - For VON > VOFF
    - VT = (VON + VOFF)/2
    - VH = |VON – VOFF|/2 * (-1) : negative for smooth transition
      - For discrete switching action, remove the negative sign, but this is not an exact equivalent change.
  - For VON < VOFF
    - VT and VH : Same as above
    - Swap ROFF and RON

S. Switch Model Params : (VON, VOFF) vs (VT, VH)

Qspice : Switch - VH neg formula.qsch

- Formula in Smooth Transition Region
  - $L_m = ln\left(\sqrt{\mathrm{RON} \times \mathrm{ROFF}}\right)$, $L_r = ln\left(\frac{\mathrm{RON}}{\mathrm{ROFF}}\right)$, $V_m = \frac{\mathrm{VON+VOFF}}{2}$, $V_d = \mathrm{VON} - \mathrm{VOFF}$
  - For VON > VOFF : $R_s = e^{L_m + \frac{3\,L_r\,(V_c - V_m)}{2\,V_d} - 2\,L_r\left(\frac{V_c - V_m}{V_d}\right)^3}$
  - For VON < VOFF : $R_s = e^{L_m - \frac{3\,L_r\,(V_c - V_m)}{2\,V_d} + 2\,L_r\left(\frac{V_c - V_m}{V_d}\right)^3}$

```
.plot V(in)/I(VS2) Rs()*1V/1A
.plot V(in)/I(VS1)

.param Roff=10
.param Ron=1
.param Voff=-10m
.param Von=10m
.param Vt=(Von+Voff)/2
.param Vh=abs(Von-Voff)/2*-1

.model SW1 SW ROFF=10 RON=1 Vt=Vt Vh=Vh
.model SW2 SW ROFF=Roff RON=Ron VOFF=Voff VON=Von

.dc V2 -20m 20m 0.1m

.func Lm() ln(sqrt(Ron*Roff))
.func Lr() ln(Ron/Roff)
.func Vm() (Von+Voff)/2
.func Vd() (Von-Voff)
.func Rs() exp(Lm() + 3*Lr()*(V(ctrl)-Vm())/(2*Vd()) - 2*Lr()*((V(ctrl)-Vm())/Vd())^3)
```

S. Switch Model Params : (VON, VOFF) vs (VT, VH)

Qspice : Switch - Von Voff VT VH.qsch

```
.model SW1 SW ROFF=10 RON=1 Vt=5m Vh=-5m
.model SW2 SW ROFF=10 RON=1 VOFF=0 VON=10m
```

```
.model SW1 SW ROFF=1E12 RON=10m Vt=5m Vh=-5m
.model SW2 SW ROFF=1E12 RON=10m VOFF=0 VON=10m
```

```
.dc V2 -20m 20m 0.1m
```

- Two examples

```
.model SW1 SW ROFF=10 RON=1 Vt=5m Vh=-5m
.model SW2 SW ROFF=10 RON=1 VOFF=0 VON=10m
```

```
.model SW1 SW ROFF=1E12 RON=10m Vt=5m Vh=-5m
.model SW2 SW ROFF=1E12 RON=10m VOFF=0 VON=10m
```

ROFF and RON is extreme

```
.dc V2 -20m 20m 0.1m
```

LOG Scale

S. Switch Model Params : ETA

Qspice : Switch - ETA.qsch

- ETA
  - ETA : Sub-/over- Threshold conduction
  - **Default ETA=0**
  - Prevents the control voltage Jacobian from vanishing
  - Only effective in smoothly transitions cases
    - Vt is negative
    - Von / Voff are used without Vt and Vh

```
.model mSW SW Ron=2 Roff=10 Vt=0.5 Vh=-0.2 ETA=ETA
```

```
pulse 0 1 0 1 1 1 4
.step param ETA list 0 0.05 0.1 0.15
.tran 4
.func Rsw() V(sw)/I(I1)
.plot Rsw()
.plot V(ctrl)
```

S. Switch Model Params : TTOL

Qspice : Switch - TTOL.qsch

- TTOL
  - Ttol : Temporal tolerance
  - **Default TTOL=1e308s**

- Usage
  - TTOL allows one to determine how accurately the switch time should be found
  - It only affect timestep at switch time, which maintain simulation speed but improve accuracy at switching or logic action

S. Switch Model Params : TTOL (with VH hysteresis)

Qspice : Switch - TTOL with VH.qsch

- TTOL
  - Ttol : Temporal tolerance
  - **Default TTOL=1e308s**

- With Vh hysteresis
  - Extra timestep occurs according to switching action with hysteresis
  - ** TTOL may not response to negative Vh (-Vh) as switch is defined as smoothly transit instead of discrete state

T. Lossless Transmission Line

T. Lossless Transmission Line Instance Parameters

- T. Lossless Transmission Line
  - Syntax: Tnnn L+ L- R+ R- Zo=<value> Td=<value>

Lossless Transmission Line Instance Parameters

Name	Description	Units	Default
F [1]	Frequency	Hertz	1GHz
NL[1]	Normalized length at frequency given	wavelengths	¼
TD	Transmission delay	sec	
Z0	Characteristic impedance(aka ZO)	Ω	50Ω

T. Instance Params : TD, F and NL

Qspice : Tline - Td.qsch ; Tline - F NL.qsch

- TD and Z0
  - Td : Transmission delay
  - Z0 : Characteristic Impedance
  - **Default TD is not set**
  - **Default Z0=50Ω**
  - This simulation terminate transmission line with characteristic impedance Zo, no reflection of signal
  - If TD is given, F and NL are ignored

- F and NL
  - F : Frequency
  - NL : Normalized length at frequency given
  - **Default F=1GHz**
  - **Default NL=1/4**
  - Equation
    - $NL = \frac{Length}{Wavelength} = \frac{Length}{\lambda}$
    - Period per λ : $T = \frac{1}{F}$
    - Delay $T_d = T \times \frac{Length}{\lambda} = \frac{NL}{F}$

V. Voltage Source

V. Voltage Source - Instance Params

- V. Voltage Source
  - Syntax: Vnnn N+ N- <voltage> [Additional Instance Parameters]

Voltage Source Instance Parameters

Name	Description	Units	Default
AC	AC magnitude, optionally followed by phase angle	V, °	0.
ACMAG	AC Magnitude	V	0.
ACPHASE	AC Phase	°	0.
DC	DC source value	V	0.
EXP	Exponential source description		
LOG	Interpolate between PWL and CHIRP points		(not set)
PULSE	Pulse description		
PWL	Piecewise linear description		
RSER	Internal series resistance		
SFFM	Single frequency FM description		
SINE	Sinusoidal source description(aka SIN)		
TIMECTRL	Time step control, one of NONE, LIMITS[1], BREAKS[2], BOTH	String	LIMITS
TIME_SCALE_FACTOR	Multiplies times, divides frequencies		1.
VALUE_SCALE_FACTOR	Multiplies voltages		1.
XTRAP	Extrapolate beyond PWL and CHIRP points		(not set)

V. Voltage Source (Instance Params) : DC

- DC
  - DC source value
  - This DC value is used in .dc analysis
  - Special Usage
    - For pulse in .tran, DC value can define voltage before Tdelay or at 0s
    - Without DC, this voltage equals <value1> in pulse

```
.tran 3
.param Td=1
.plot V(pulse-WithDC)
.plot V(pulse-WithoutDC)
```

pulse-WithoutDC Special Usage in Pulse

pulse -1 1 Td 0 0 0.5 1

pulse-WithDC

pulse -1 1 Td 0 0 0.5 1

DC 0

V. Voltage Source (Instance Params) : Time/Value_Scale_Factor

Qspice : Vsource - Time_Scale_Factor.qsch / Vsource - Value_Scale_Factor.qsch

- TIME_SCALE_FACTOR
  - Time_Scale_Factor : multiples times, divides frequency
  - Default Time_Scale_Factor=1

- VALUE_SCALE_FACTOR
  - Value_Scale_Factor : multiples voltages
  - Default Value_Scale_Factor=1

V. Voltage Source : PULSE (Pulse)

Qspice : Vsource PULSE.qsch

Syntax : PULSE V1 V2 Td Trise Tfall Ton Tperiod

pulse_woDC

PULSE -1 1 0.4 0.2 0.2 0.2 1

pulse_wDC

PULSE -1 1 0.4 0.2 0.2 0.2 1
DC 0

```
.tran 2.5
.plot V(pulse_woDC) V(pulse_wDC)
```

V. Voltage Source : PULSE (Pulse) : Usage

Qspice : Vsource PULSE-Step.qsch / Vsource PULSE-SinglePulse.qsch

- Step with PULSE
  - To generate a step, only input value to ***Trise***

- Single Pulse with PULSE
  - To generate a single non-repeating pulse, does not specify ***Tperiod*** in pulse statement

V. Voltage Source : SIN (Sine Wave)

Qspice : Vsource SIN.qsch

Phi : level = Voffset + sin(Phi)
e.g.: =1+sin(30°) = 1.5

Syntax : SIN Voffset Vamp Freq Td Theta Phi

sin .tran 3

sin 1 1 1 0.5 0 30

sin_theta

sin 0 1 1 0 1 90

Amplitude drops by e^{-1}~0.378 in 1/Theta seconds

V. Voltage Source : EXP (Exponential)

Qspice : Vsource EXP.qsch

.plot V(expTperiod) V(exp)

Syntax : EXP V1 V2 Td1 Tau1 Td2 Tau2 Td3 N Tperiod

.tran 200

.param V1=0
.param V2=1
.param Td1=2
.param Tau1=0.7
.param Td2=4
.param Tau2=0.5
.param Td3=9
.param N=3
.param Tperiod=50

exp

EXP V1 V2 Td1 Tau1 Td2 Tau2 Td3 N

expTperiod

EXP V1 V2 Td1 Tau1 Td2 Tau2 Td3 N Tperiod

V. Voltage Source : SFFM (Single Frequency FM)

Qspice Vsource SFFM.qsch

- SFFM
  - Single Frequency FM
  - It is hard to observe FM signal in time domain waveform
  - FFT is used to explain its nature

Syntax : SFFM Voff Vamp Fcar MDI Fsig

SFFM

SFFM 0 1 100K 0.5 10K

.tran 10/10K

.plot V(SFFM)

FFT magnitude is RMS

- Transient formula
  - Voff+Vamp*sin((2p*Fcar*time)+MDI*sin(2p*Fsig*time))

V. Voltage Source : GAUSS (Gaussian Pulse)

Qspice Vsource GAUSS.qsch

Syntax : GAUSS Td Amp Sigma [Period]

gauss

GAUSS 2 1 2

gauss_period

GAUSS 2 1 2 50

```
.plot V(gauss_period) V(gauss)
.tran 100
```

V. Voltage Source : IGAUSS (Imaginary Gaussian Pulse)

Qspice Vsource IGAUSS.qsch

Syntax : IGAUSS Td Period Sigma Amp

.step param Sigma list 1 5 10

IGAUSS 10 50 Sigma 1

.plot V(igauss)
.tran 100

V. Voltage Source : CHIRP (Piece-wise Linear Chirp)

Qspice Vsource Chirp.qsch

Syntax : CHIRP(AMP T1 FREQ1 T2 FREQ2 [...]) [LOG] [XTRAP]

```
.param AMP=1
.param T1=5
.param FREQ1=1
.param T2=15
.param FREQ2=2
.param T3=30
.param FREQ3=10
```

```
CHIRP AMP T1 FREQ1 T2 FREQ2 T3 FREQ3

.plot V(chirp)
.tran 35                .options MAXSTEP=1/1000
```

XTRAP : extrapolate beyond PWL and CHIRP points

V. Voltage Source : PWL (Piece-wise Linear) – XTRAP and LOG

Qspice : Vsource PWL-Xtrap.qsch/ Vsource PWL-Log.qsch

- Basic Usage and XTRAP
  - Standard
    - last value will hold
  - Xtrap
    - Extrapolate beyond PWL and CHIRP points

Syntax : PWL(t1 v1 t2 v2 t3 v3...) [LOG] [XTRAP]

- LOG
  - Log
    - Interpolate between PWL and CHIRP points with log function

Syntax : PWL(t1 v1 t2 v2 t3 v3...) [LOG] [XTRAP]

V. Voltage Source : PWL (Piece-wise Linear) – REPEAT

Qspice : Vsource PWL-Repeat.qsch

- PWL Repeat Syntax
  - Vnnn N+ N- PWL **REPEAT FOREVER** t1 v1 t2 v2 t3 v3… **ENDREPEAT**
  - Vnnn N+ N- PWL **REPEAT FOR N** t1 v1 t2 v2 t3 v3… **ENDREPEAT**
    - N is number of times
    - ** last voltage value of a repeated PWL will be ignored and forced to be equal to the first value

Syntax : PWL FOR[EVER] [times] (t1 v1 t2 v2 t3 v3…) Endrepeat

pwl1

PWL REPEAT FOREVER (0 0 1 1 2 4 3 9 3+1p 0) Endrepeat

pwl2

PWL REPEAT FOR 3 (0 0 1 1 2 4 3 9 3+1p 0) Endrepeat

.tran 20

.plot V(pwl2)
.plot V(pwl1)

V. Voltage Source : PWL (Piece-wise Linear) – Load from file

Qspice : Vsource PWL file.qsch / Vsource PWL-Repeat-File.qsch

- PWL : load from file
  - Filename is quoted with single ('filename') or double ("filename") quotation
  - Delimiter can be space [ ] or comma [,]
  - To load from file, syntax is
    - FILE filaname
  - Support repeat syntax, but remember last value in repeated PWL is ignored and forced to be equal to first value

Syntax : PWL FILE file.txt [LOG] [XTRAP]

Vsource PWL file format1.txt

1	1	1
2	2	4
3	3	9
4	4	16
5	5	25
6	6	36
7	7	49
8	8	64
9	9	81
10	10	100

Vsource PWL file format2.txt

1	1,1
2	2,4
3	3,9
4	4,16
5	5,25
6	6,36
7	7,49
8	8,64
9	9,81
10	10,100

Syntax : PWL FOR[EVER] [times] FILE 'filename' Endrepeat

V. Voltage Source : PWL (Piece-wise Linear) – Usage

Qspice : Vsource PWL-SinglePulse.qsch

- Step and Single Pulse
  - To create step or single pulse with PWL, a datapoint to maintain voltage level before change is required
  - For example, if change occurs at 2s, define an extra datapoint at 1.999s or (2-1p)

W. Wire and BUS

W. Wire and Bus : Introduction

[1] Place a voltage source and draw wire
[2] Rename voltage source to V1[0:2]
[3] Place net name as A[0:2]
[4] Now, the wire becomes a bus, in this example, A[m] is voltage of V1[m]

[5] Right click and select “Place a Bus Tap”
[6] Place net name to bus tap wire

W. Wire and Bus : Bus for V/I-source

- Identical I-source
  - File : BUS - Isrc.qsch
  - Bus can replicate current or voltage source for multiple nodes usage
  - In netlist
    - I1[0] 0 A[0] sin 0 1 1
    - I1[1] 0 A[1] sin 0 1 1
    - I1[2] 0 A[2] sin 0 1 1
  - ** .dc for I1 won't work as three I-sources I1[0], I1[1] and I1[2] in this format

W. Wire and Bus : Bus for Ø-Device

- Bus for Ø-Device
  - Files
    - BUS_DLL.qsch
    - bus_dll.cpp
    - bus_dll.dll
  - Procedure
    - For Ø-Device, defines an output port with label name with net[0:7] array
    - To output bit, data type should be boolean
    - In template generation, it creates variable name as **net_0_** for bit 0 and **net_7_** for bit 7

```
extern "C" __declspec(dllexport) void bus_dll
{
   unsigned int in   = data[0].ui; // input
   bool         &d_0_ = data[1].b ; // output
   bool         &d_1_ = data[2].b ; // output
   bool         &d_2_ = data[3].b ; // output
   bool         &d_3_ = data[4].b ; // output
   bool         &d_4_ = data[5].b ; // output
   bool         &d_5_ = data[6].b ; // output
   bool         &d_6_ = data[7].b ; // output
   bool         &d_7_ = data[8].b ; // output

// Implement module evaluation code here:
   d_0_ = in & 0x01;
   d_1_ = in & 0x02;
   d_2_ = in & 0x04;
   d_3_ = in & 0x08;
   d_4_ = in & 0x10;
   d_5_ = in & 0x20;
   d_6_ = in & 0x40;
   d_7_ = in & 0x80;
}
```

Y. Piezoelectric Crystal

Y. Piezoelectric Crystal : Instance Parameters

- Y. Piezoelectric Crystal
  - Syntax: Ynnn N+ N- <frequency1> dF=<value> Ctot=<value> [Q=<value>]

Piezoelectric Crystal Instance Parameters

Name	Description	Units	Default
CTOT	Total capacitance as measured at DC	F	
DF[1]	Difference between series and parallel resonant frequencies	Hz	
FREQUENCY[1]	Resonant frequency	Hz	
IC	Initial internal current(Used with UIC on the .tran)	A	0.0
M	Number of parallel devices		1.0
RSER	Series resistance	Ω	Determined from Q
Q	Quality factor		1.6e13÷FREQUENCY[2]

$Q = \min(1e6, \frac{1e13}{\text{FREQUENCY}^2})$
If default equation is used

- If DF > 0 : ***Series Resonant Freq = Frequency*** and ***Parallel Resonant Freq = Frequency + DF***
- If DF < 0 : ***Series Resonant Freq = Frequency + DF*** and ***Parallel Resonant Freq = Frequency***

Y. Piezoelectric Crystal : Formula and Equivalent Circuit

Qspice : Y Crystal - Equivalent Circuit.qsch

- Piezoelectric Crystal Equation
  - Series Resonant Frequency : $f_s = \frac{1}{2\pi\sqrt{L_s C_s}}$
  - Parallel Resonant Frequency : $f_p = \frac{1}{2\pi\sqrt{L_s \frac{C_p C_s}{C_p + C_s}}}$
  - Crystal Oscillators Q-factor : $Q = \frac{X_L}{R} = \frac{2\pi f L_s}{R_s}$
  - Total Capacitance : CTOT = $C_p + C_s$

- Calculated Value
  - $C_p = \left(\frac{f_s}{f_p}\right)^2 \text{CTOT}$
  - $C_S = \text{CTOT} - C_p$
  - $L_s = \frac{1}{(2\pi f_s)^2 C_s}$
  - $R = \frac{2\pi f L}{Q}$

Ã-Device MULTGMAMP

Ã-Device

- Ã-Device
  - Syntax

    Ãnnn VDD VSS OUT IN- IN+ MULT+ MULT- IN-- IN++ EN ¥ ¥ ¥ ¥ ¥ ¥ <TYPE> [INSTANCE PARAMETERS]

  - Ã-device models a highly configurable Gm block than can mimic the behavior of a complementary MOSFET output, there are two types MULTGMAMP and RROPAMP

Ã-Device Types

TYPE	Behavior
MULTGMAMP	Multiplying Gm Amplifier
RROPAMP	Rail-to-Rail Output OpAmp

MULTGMAMP is Operational Transconductance Amplifier (OTA)

To Identify what <TYPE> a symbol is

- In Symbol Properties > 1st attribute
- View > Netlist : from device syntax

```
* C:\Users\kelvinleung\Documents\QSPICE\Untitled.qsch
Ã1 ¥0 ¥1 ¥2 ¥3 ¥4 ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ MultGmAmp Gm=val
.end
```

Ã-Device [MULTGMAMP]

- Ã-Device
  - Syntax: **Ãnnn VDD VSS OUT IN- IN+ MULT+ MULT- IN-- IN++ EN ¥ ¥ ¥ ¥ ¥ ¥ <TYPE> [INSTANCE PARAMETERS]**
  - Formula in Linear Region : Iout = GM * V(MULT+,MULT-) * V(MIN(IN+,IN++),MAX(IN-,IN--))

Ã-Device Instance Parameters in MULTGMAMP

Ã-Device Instance Parameters

Name	Description
CAPINCM	Common eigenmode input capacitance
CAPINNM	Normal eigenmode input capacitance
CAPVDD	Capacitance from output to Vdd
CAPVSS	Capacitance from output to Vss
EN	Equivalent input voltage noise density
ENK	EN corner frequency
FT	3dB bandwidth of transconductance with no voltages slewing
GM	Ideal transconductance
IC	Initial condition of Vin × Vmult
IN	Equivalent input current noise density
INF	Common mode input current noise density proportional to frequency
INK	IN corner frequency
IOUT	Maximum sourcing current
ISNK	Maximum sinking current
ISNKKNEE	Sharpness of Maximum sinking current limit
ISRC	Maximum sourcing current
ISRCKNEE	Sharpness of max sourcing current limit
M	Number of parallel devices
REF	Logic threshold for enable(from Vss)
ROUT	Additional impedance added to output(2*R to Vdd, 2*R to Vss)
TEMP	Instance temperature
TTOL	Temporal tolerance for enable & UVLO
UVLO	Minimum supply voltage
VCROSS	Cross conduction voltage range
VDSAT	Voltage where gm starts to switch over to a resistance
VDSAT1	Voltage where gm starts to switch over to a resistance(top FET)
VDSAT2	Voltage where gm starts to switch over to a resistance(bottom FET)
VINHIGH	Output range measured from positive rail
VINHIGHKNEE	Sharpness of positive input range limit
VINLOW	Input range measured from negative rail
VINLOWKNEE	Sharpness of negative input range limit
VOS1	Offset voltage for input
VOS2	Offset voltage for multiplying input
VOUTMAX	Maximum output voltage measured from negative rail
VOUTMIN	Minimum output voltage measured from negative rail

MULTGMAMP : Basic

Qspice : Multgmamp - Basic.qsch | Multgmamp - Basic (Default Rout).qsch

- Basic
  - MULTGMAMP is Operational Transconductance Amplifier (OTA) whose differential input voltage produces an output current
  - By simplify its input only with IN- and IN+

    $I_{out} = Gm \times (V_{IN+} - V_{IN-})$
  - As OTA output a current, its output voltage depends on loading impedance, where

    $V_{out} = Z_{out} \times I_{out}$

    $V_{out} = Z_{out} Gm \times (V_{IN+} - V_{IN-})$
  - Zout can be defined in two ways
    - External Resistor
    - Instance parameter Rout
  - If Rout is not defined by user and no output impedance, Default Rout = 500TΩ, where voltage gain is 500T @ Gm=1

MULTGMAMP : GM (Ideal transconductance)

Qspice : Multgmamp – Gm with mult.qsch

- Gm : Ideal transconductance
  - Operational Transconductance Amplifier (OTA) is an amplifier whose differential input voltage produces an output current
  - Without IN++ and IN--, formula is **Iout = GM * V(MULT+,MULT-) * V(IN+,IN-)**
  - Gm can be positive or negative, but must be non-zero

MULTGMAMP : GM (Ideal transconductance) and FT (3dB Bandwidth)

Qspice : Multgmamp - Gm.qsch | Multgmamp - FT.qsch

- Gm Ideal transconductance
  - No default value of Gm
    - Must be user provided
  - MULTGMAMP has ideal frequency response if other instance parameters are in default

- FT (3dB Bandwidth)
  - FT : 3dB bandwidth of transconductance with no voltages slewing
  - **Default FT=0** (i.e. infinite)
  - FT is frequency

MULTGMAMP : CAPINNM (Normal eigenmode input capacitance)

Qspice : Multgmamp - Capinnm.qsch

- CAPINNM : Normal eigenmode input capacitance
  - **Default CAPINNM=0**
  - This capacitance is between IN- and IN+

MULTGMAMP : CAPINCM (Common eigenmode input capacitance)

Qspice : Multgmamp - Capincm.qsch

- CAPINCM : Common eigenmode input capacitance
  - **Default CAPINCM=0**
  - This capacitance is between IN- to GND and IN+ to GND

MULTGMAMP : Vdd, Vss and VDSAT – Output Voltage

Qspice : Multgmamp - Vdd Vss.qsch | Multgmamp - VDSAT.qsch

- Vdd and Vss Nets
  - This is not parameters but Net 1 and Net 2 in Ã-Device
  - Output voltage swing is limited by Vdd and Vss
  - ** In this example, Vout cannot reach Vdd and Vss because Default VDSAT is set in Qspice

- VDSAT
  - Vdsat : Voltage where gm starts to switch over to a resistance
  - **Default VDSAT = 0.5**
  - **VDSAT=0** forced this value to default!
  - **Set VDSAT to small value (e.g. 1p)** for no saturation

MULTGMAMP : VDSAT1, VDSAT2, VOUTMIN, VOUTMAX – Output Voltage

Qspice : Multgmamp - VDSAT1 VDSAT2.qsch ; Multgmamp | VOUTMAX VOUTMIN.qsch

- VDSAT1 and VDSAT2
  - VDSAT1 : Voltage where gm starts to switch over to a resistance(top FET)
  - VDSAT2 : Voltage where gm starts to switch over to a resistance(bottom FET)
  - **Default VDSAT1=VDSAT**
  - **Default VDSAT2=VDSAT1**

- VOUTMIN and VOUTMAX
  - Voutmin : Minimum output voltage measured from negative rail
  - Voutmax : Maximum output voltage measured from negative rail

MULTGMAMP : Iout/Isrc, Isnk – Output Sink and Src Current

Qspice : Multgmamp - ISRC ISNK.qsch ; Multgmamp - ISRCKNEE ISNKKNEE.qsch

- IOUT/ISRC and ISNK
  - Iout/Isrc : Maximum sourcing current (toward load)
  - Isnk : Maximum sourcing current (toward device)
  - **Default ISRC = Infinite**
  - **Default ISNK = IOUT or ISRC**
    - i.e. if ISNK not specified, ISNK equals IOUT or ISRC

- ISRCKNEE and ISNKKNEE
  - Isrcknee : Sharpness of max sourcing current limit
  - Isnkknee : Sharpness of Maximum sinking current limit
  - **Default Isrckness = 0.1**
  - **Default Isnkkness = Isrcknee**

- Important Note
  - To have effect, must defines ISRC or ISNK

MULTGMAMP : ROUT, CAPVDD and CAPVSS

Qspice : Multgmamp - Rout CAPVDD CAPVSS.qsch

- ROUT, CAPVDD and CAPVSS
  - ROUT : Additional impedance added to output (2*R to Vdd, 2*R to Vss) (**Default ROUT=500Tohms**)
  - CAPVDD : Capacitance from output to Vdd (**Default CAPVDD=0**)
  - CAPVSS : Capacitance from output to Vss (**Default CAPVSS=0**)

MULTGMAMP : ROUT (Default)

Qspice : Multgmamp - Rout Test Circuit (.ac).qsch

- Rout (Default)
  - AC test circuit for Rout
    - By $I_{out} = Gm \times (V_{IN+} - V_{IN-})$
    - With no external resistance,
      $V_{out} = R_{out} \times I_{out}$
      $V_{out} = R_{out} Gm \times (V_{IN+} - V_{IN-})$
    - Set $Gm = 1$ and $V_{IN+} - V_{IN-} = 0$
    - $V_{out} = R_{out}$
  - Results : **Default Rout=500T**

MULTGMAMP : ROUT [In Question!!!]

Qspice : Multgmamp - Rout Default.qsch

- ROUT Default
  - This setup calculate Rout based on having V(out)=0V and measure Rtotal between Vdd and Vss by $R_{total} = 4\,R_{out} = \frac{V_{dd} - V_{ss}}{I_{source}}$
  - Default $R_{out} = \frac{2000}{Gm}$
  - If Rout is assigned, depends on Gm, actual Rout is set with min(Rout,$\frac{2000}{Gm}$)

MULTGMAMP : Input Range VINLOW, VINHIGH, VINHIGHKNEE, VINLOWKNEE

Qspice : Multgmamp - VINHIGH VINLOW.qsch ; Multgmamp - VINHIGHKNEE VINLOWKNEE.qsch

- VINLOW and VINHIGH
  - VINLOW : Input range measured from negative rail
  - VINHIGH : Input range measured from positive rail
  - **Default VINLOW=0**
  - **Default VINHIGH=0**
  - **Value is negative**, which limit input range. For example, VINHIGH=-0.5, Input HIGH is limited to Vdd+VINHIGH = 2-0.5=1.5 in this example

- VINLOWKNEE VINHIGHKNEE
  - Vinlowknee : Sharpness of negative input range limit
  - Vinhighknee : Sharpness of positive input range limit
  - **Default VINHIGHKNEE=0**
  - **Default VINLOWKNEE=0**
  - Increase KNEE soften the sharpness of input range, more output signal to come

MULTGMAMP : UVLO and VOS1

Qspice : Multgmamp - UVLO.qsch ; Multgmamp - VOS1.qsch

- UVLO
  - UVLO : Under Voltage Lock Out - Minimum supply voltage
  - Default UVLO is Infinite
  - UVLO is compare to supply voltage = Vdd-Vss

- VOS1
  - VOS1 : Offset voltage for input
  - Iout = GM * (*VOS1*+V(IN+,IN-))
    - This parameter is not in HELP formula, but it actually implemented
  - Default VOS1 = 0

Ã-Device RROPAMP

Ã-Device RROPAMP Amplifier

- Ã-Device RROPAMP Amplifier
  - Syntax: Ãnnn VDD VSS OUT IN- IN+ MULT+ MULT- IN-- IN++ EN ¥ ¥ ¥ ¥ ¥ ¥ RROPAMP [INSTANCE PARAMETERS]
  - A generic rail-to-rail output opamp specified by Avol, GBW, slew rate and phase margin. It is implemented with two Gm amps as above with an internal node for a Miller capacitor connected to the output.

Basics		
Symbol Type		Ã
Description		Generic Rail-to-Rail Output OpAmp
Allow Shorted Pins		False
Library File		
String Attributes		
Text Order	Invis.	Content
Name:		Ã1
1st attribute	✔	RRopAmp
Avol		100K
GBW		5Meg
Slew		5Meg
Rload		2K
Phi		60

Ã-Device Instance Parameters

Name	Description
AVOL	Open loop voltage gain when loaded with RLOAD
CAPINCM	Common eigenmode input capacitance
CAPINNM	Normal eigenmode input capacitance
CAPVDD	Capacitance from output to Vdd
CAPVSS	Capacitance from output to Vss
EN	Equivalent input voltage noise density
ENK	EN corner frequency
FT	3dB bandwidth of transconductance with no voltages slewing
GBW	Dominate pole gain-bandwidth
IC	Initial condition of Vin × Vmult
IN	Equivalent input current noise density
INF	Common mode input current noise density proportional to frequency
INK	IN corner frequency
IOUT	Maximum sourcing current
ISNK	Maximum sinking current
ISNKKNEE	Sharpness of max sinking current limit
ISRC	Max sourcing current
ISRCKNEE	Sharpness of max sourcing current limit
M	Number of parallel devices
PHI	Nominal phase margin
REF	Logic threshold for enable(from Vss)
RLOAD	Load resistance used in AVOL, GBW specification
ROUT	Open loop output impedance
SLEW	Slew rate in Volts per second
TEMP	Instance temperature
TTOL	Temporal tolerance for enable & UVLO
UVLO	Minimum supply voltage
VCROSS	Cross conduction voltage range
VDSAT	Voltage where gm starts to switch over to a resistance
VDSAT1	Voltage where gm starts to switch over to a resistance(top FET)
VDSAT2	Voltage where gm starts to switch over to a resistance(bottom FET)
VINHIGH	Output range measured from positive rail
VINHIGHKNEE	sharpness of positive input range limit
VINLOW	Input range measured from negative rail
VINLOWKNEE	Sharpness of negative input range limit
VOS1	Offset voltage for input
VOS2	Offset voltage for multiplying input
VOUTMAX	Maximum output voltage measured from negative rail
VOUTMIN	Minimum output voltage measured from negative rail

RROPAMP : RLOAD, AVOL, GBW and PHI

Qspice : RROPAMP - Second-Order TF.qsch

- RROPAMP models $G_{opamp}(s)$ with second-order transfer function
  - Opamp Open-Loop Transfer Function : $v_{out} = G_{opamp}(s)\,(v_+ - v_-)$
  - RROPAMP model this transfer function with two poles and one zero
  - AVOL (open loop gain), GBW (gain bandwidth) and PHI (phase margin) are defined with RROPAMP loaded with a load resistor as RLOAD

- Pole 1
  - Determine by **AVOL** and **GBW**
- Pole 2 and Zero
  - Determine by **PHI**
  - pole 2 and zero are apart by two decades which seems to be fixed by Qspice
  - Therefore, PHI cannot change how far they apart from each other, but move it through frequency to set phase margin

RROPAMP : RLOAD, AVOL, GBW and PHI

Qspice : RROPAMP - AVOL.qsch | RROPAMP - GBW.qsch

- RLOAD
  - Load resistance used in AVOL, GBW and PHI specification
  - **Default Rload=2K**
  - Output resistor with R=Rload is required to obtain frequency response as according to these parameters specified
- AVOL
  - Open loop voltage gain when loaded with RLOAD
  - **Default AVOL=1000k (120dB)**
  - $\text{Gain} = \frac{V_{out}}{V_{+} - V_{-}}$
  - AVOL is gain @ 0Hz
- GBW
  - Dominate pole gain-bandwidth
    - GBW means frequency where gain of opamp falls to unity (0dB or 1)
  - **Default GBW=5Meg**

RROPAMP : RLOAD, AVOL, GBW and PHI

Qspice : RROPAMP - PHI.qsch | RROPAMP - AVOL GBW PHI RLOAD.qsch

- PHI
  - Nominal phase margin
  - **Default PHI=60°**
  - Phase margin is measured of phase to -180° at 0dB (or at GBW frequency)
  - PHI should limit in the range of [0° , 90°]
    - PHI=0 or negative will force PHI to default at 60°
    - PHI>90° will limit at 90°

- AVOL, GBW and PHI
  - AVOL, GBW, PHI are three parameters to determine open-loop frequency response of RROPAMP
  - RLOAD
    - These three parameters are specified at this Rload condition

RROPAMP : Poles and Zero Location

Qspice : RROPAMP - PHI - Pole Zero Pair.qsch

RROPAMP contains two poles and one zero

```
.ac dec 20 1e-3 1e12
.step param phi list 1p 30 70 80 81 90
.plot V(OUT)
```

** Be careful, if PHI=0,
it force PHI to default which is 60°

This matlab is not to replicate exact Qspice result but to demonstrates this profile is created with two –ve poles and one +ve zero

RROPAMP : SLEW

Qspice : RROPAMP - SLEW.qsch

- SLEW
  - Slew rate in Volts per second (maximum change rate)
    - $SLEW = \frac{\Delta V_{out}}{\Delta t}$
  - **Default SLEW=5Meg**

¥-Device

¥-Device

- ¥-Device
  - Syntax:

    ¥nnn N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13 N14 N15 N16 <TYPE> [INSTANCE PARAMETERS]

  - The ¥-device supplies gate, flop, and a few other types of functional behaviors. The device expects exactly 16 pins, but not all are used. Unused pins must be specified connected to "¥".

¥-Device Types

TYPE	Behavior
AND	AND gate
CLOCKSYNC	Selects between a Sync and Clock inputs
D-FLOP	D-type flip-flop
EXTOSC	Oscillator programmed with an external resistor
HMITT	Schmitt trigger
JK-FLOP	JK-type flip-flop
MONOSTABLE	Retriggerable monostable
OR	OR gate
PS-FLOP	SMPS flip-flop
RS-FLOP	RS-type flip-flop
T-FLOP	Toggle flip-flop
XOR	XOR gate
Φ-DET	Phase/Frequency detector

To Identify what <TYPE> a symbol is

- In Symbol Properties > 1st attribute
- View > Netlist : from device syntax

```
* C:\Users\kelvinleung\Documents\QSPICE\Unt
¥1 ¥0 ¥1 ¥2 ¥3 ¥4 ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ OR
.end
```

¥-Device : How EN pin works

Qspice - Logic - EN.qsch

- How EN works
  - If EN is preset in symbol, normally, a signal is used to control its status. Here discuss two situations
  - #1 No connect to EN pin
    - Netlist assign ¥n as net name. EN pin is active to monitor EN signal. In this case, no output will generate
  - #2 Assign EN pin with net ¥
    - ¥ net name is equivalent to unassigned in ¥-device
    - Unassigned EN pin will be considered as Default ENABLE
    - Therefore, a symbol can be created with pin assigned to ¥ for their default

Netlist
¥1 Vdd 0 Q ¥0 + 0 ¥1 ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ HMITT Vt=0.5 Vh=0.1

Netlist
¥1 Vdd 0 Q ¥0 + 0 ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ HMITT Vt=0.5 Vh=0.1

¥-Device : Net ¥

- Net ¥
  - Unconnected Pin and Pin connected to Net ¥ is different
  - Net ¥ is a special net name, in HELP of ¥-Device, it specified that unused pins must be specified connected to "¥"
    - Two methods to force a pin to ¥ if
  - This is an example of AND gate
    - Device ¥1 : NC pin is equivalent to 0V and AND gate output 0
    - Device ¥2 : ¥ is assigned as Net in schematic and this input specified unused for this AND gate
    - Device ¥3 : ¥ is assigned but in System Properties, this input specified unused and become invisible in symbol

¥-Device : HELP > Simulator > Device Reference > ¥-Device Common Instance Parameters (example from AND Gate Instance Param)

Name	Description	Units	Default
CAPVDD	Capacitance from an output to Vdd	F	0.
CAPVSS	Capacitance from an output to Vss	F	0.
IC	Initial condition(needed, e.g., for a ring osc)		
M	Number of parallel devices		1.
REF	Logic reference voltage	V	(Vdd + Vss) ÷ 2
RSINK	Resistance to Vss when output high	Ω	RSRC
RSRC	Resistance to Vdd when output low(aka ROUT)	Ω	100.
TD	Delay(aka TD1)	s	0.
TD2	Asymmetrical delay	s	TD
TEMP	Instance temperature	°C	27.
TFALL	Fall time	s	0.
TRISE	Rise time	s	0.
TTOL	Temporal tolerance	s	1μs
UVLO	Minimum Vdd-Vss voltage to operate	V	0.
ZMULT	Impedance multiplier when biased half way		1.

¥ Instance Params : REF Logic Reference Voltage

Qspice : Logic - REF.qsch

- REF : Logic reference voltage
  - In default, ¥-Device logic threshold is $\frac{Vdd+Vss}{2}$

REF	Logic reference voltage	V	(Vdd + Vss) ÷ 2

  - User can change logic threshold by [right click device] > [Add New Attribute] to add REF=<value>
    - This example show logic output follows Vdd/0 with logic input threshold equal 0.5V to interface with 1V input logic

¥ Instance Params : REF Logic Reference Voltage

Qspice : Logic - REF (Dual Supply).qsch

- REF : Logic Reference Voltage
  - REF is not absolute voltage to node 0, but reference to Vss node of ¥-Device
  - In this example, ¥1 and ¥2 are with same REF value, but as ¥1 Vss=0V and ¥2 Vss=-1V, their logic reference is different reference to node 0 (GND). Therefore, even their input is same, their output can be different

¥ Instance Params : Delay (TD) and Asymmetrical Delay (TD2)

Qspice : Logic - Td.qsch ; Logic - Td Td2.qsch

- TD (Delay)
  - Td : Delay
  - **Default TD=0s**
  - ** If TD2 is not set, both rising and falling delay times are same
  - ** If TD > logic H duration or delay TD2 > logic L duration, output always LOW or HIGH

- TD2 (Asymmetrical delay)
  - TD2 : Asymmetrical delay
  - **Default TD2=TD**
  - If different delay times for rising and falling edge, set TD2 for falling edge delay time (i.e. TD only control rising edge delay time)

¥ Instance Params : Rise time (Trise), Fail time (Tfall) and UVLO

Qspice : Logic - Trise Tfall.qsch

- TRISE and TFALL
  - Trise : Rise time
  - Tfall : Fall time
  - **Default TRISE=0s**
  - **Default TFALL=0s**

- UVLO
  - UVLO : Under Voltage Lock Out
  - **Default UVLO=0V**
  - UVLO : Minimum Vdd-Vss voltage to operate
    - Output only enabled when Vdd-Vss > UVLO
    - Output equals Vss when gate disable

¥ Instance Params : Output R and C (Rsink, Rsrc, Capvdd, CapVss)

Qspice : Logic - RSINK RSRC.qsch ; Logic - Capvdd Capvss.qsch

- RSINK and RSRC
  - RSINK : Res to Vss when o/p high
  - RSRC : Res to Vdd when o/p low
  - **Default RSINK=100Ω**
  - **Default RSRC=100Ω**

- Example
  - This simulation shows loading effect when output to a 100 ohms, where o/p level is reduced to half of Vdd

- CAPVDD, CAPVSS
  - Capvdd : Capacitance from an output to Vdd
  - Capvss : Capacitance from an output to Vss

- Explanation
  - Capacitance Capvdd/Capvss equivalent Cdd/Css this example
  - This demo can have frequency response because Rsink=Rsrc=100 internally

¥ Instance Params : TTOL Temporal Tolerance

Qspice : Logic - TTOL.qsch

- TTOL
  - Ttol : Temporal tolerance
  - **Default TTOL=1u**
  - TTOL allows one to determine how accurately the switch time should be found

¥ [HMITT] Instance Params : Threshold (VT) and Half Hysteresis (VH)

Qspice : HMITT - Vt Vh.qsch

- VT and VH
  - Vt : Threshold voltage
  - Vh : Half hysteresis voltage
  - HMITT is schmitt trigger logic input
  - low trip point is Vt-Vh and the high trippoint is Vt+Vh

¥ Truth Table : AND, OR, XOR

Qspice : Truth Table of AND OR XOR.qsch

- AND

¥-Device AND			
A	B	Q	_Q
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

- OR

¥-Device OR			
A	B	Q	_Q
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

- XOR

¥-Device XOR			
A	B	Q	_Q
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

¥ Truth Table : D-FLOP and T-FLOP

Qspice : Truth Table of D-FLOP.qsch ; Truth Table of T-FLOP.qsch

- D-FLOP
  - PRE : Active High (=SET)
  - CLR : Active High (=RESET)

¥-Device D-flop

CLK	D	PRE	CLR	Q	_Q
↑	0	0	0	0	1
↑	1	0	0	1	0
x	x	0	0	Q	_Q
x	x	1	0	1	0
x	x	x	1	0	1

- T-FLOP

¥-Device T-flop

CLK	T	PRE	CLR	Qn+1
↑	0	0	0	0
↑	1	0	0	_Qn
x	x	0	0	Q
x	x	1	0	1
x	x	x	1	0

¥ Truth Table : SR-FLOP and JK-FLOP

Qspice : Truth Table of SR-FLOP.qsch ; Truth Table of JK-FLOP.qsch

- SR-FLOP

¥-Device SR-flop			
R	S	Q	_Q
0	0	0	1
0	1	1	0
1	x	0	1

- JK-FLOP

¥-Device JK-flop					
CLK	J	K	PRE	CLR	Qn+1
↑	0	0	0	0	Qn
↑	0	1	0	0	0
↑	1	0	0	0	1
↑	1	1	0	0	_Qn
x	x	x	0	0	Q
x	x	x	1	0	1
x	x	x	x	1	0

¥-Device : Monostable and Latch (Sample and Hold S&H)

Qspice : Usage - Monostable.qsch | Usage - Latch Sample and Hold.qsch

- Monostable
  - TON : Monostable on time
  - **Default TON=0s**
  - Need to add this attribute when Monostable Symbol is called to use in Qspice
  - TON must be specified to use Monostable
  - CLK trigger a minimum on-time pulse

- Latch
  - Edge-sensitive Sample and Hold

¥ Truth Table : Φ-DET (Phi-Det) [Symbol : PhaseDetector in analog]

Qspice : Truth Table of Phi-Det.qsch

- Φ-DET (Phi-Det)
  - Phase/Frequency Detector

¥-Device Phi-Det		
REF	**OSC**	**OUT**
↑	x	Up
x	↑	Down
↑	↑	Unchange

- Important
  - Output is current, not voltage
    - Suggest to set a higher Igain for OUT can reach Vdd/Vss with load (Default Igain only 10uA)
  - This device is a [Vdd, 0, Vss] 3-state counter, therefore, must operate with dual supply
    - REF edge counts up
    - OSC edge counts down

- Phase/Frequency Detection
  - Next slide is simulation results in phase change and frequency change
  - Mean of PWM output pattern can differentiate it is OSC signal phase or frequency deviate from REF signal
  - This is a key component in Phase Lock Loop (PLL)

¥-Device Φ-DET (Phi-Det) : OSC and REF Relationship with OUT

Qspice : Usage - Phi-Det (Freq and Phase).qsch

¥-Device EXTOSC : Oscillator programmed with an external resistor

Qspice : Usage - Extosc - Rtime .tran.qsch | Usage - Extosc - Rtime .step.qsch

- EXTOSC
  - Oscillator programmed with an external resistor
  - Instance Parameters
    - DUTY : Oscillated V(Q) duty [Default DUTY = 0.9]
    - FREQ : Oscillated Frequency @ RT [Default FREQ=100kHz]
    - RT : Timing Resistance (Rtime) value at FREQ [Default RT=100kohms]
  - Symbol : ***ExtOsc4*** in analog
    - Oscillator frequency is controlled by external resistor with this symbol as no Vres1 and Vres2 pins

- V(Q) frequency vs Rtime
  - This simulation uses .step param Rtime and .meas to plot frequency of V(Q) vs Rtime relationship
  - It can observe that V(Q) frequency has inverse relationship with Rt
    - Freq of V(Q) = $\frac{\mathrm{RT}}{\mathrm{Rtime}} \times \mathrm{FREQ}$

¥-Device EXTOSC : Oscillator programmed with an external resistor

Qspice : Usage - Extosc - VRES1 .step.qsch | Usage - Extosc - VRES2 .step.qsch

- V(Q) frequency vs Vres1/2
  - Symbol : ***ExtOsc2*** in analog
    - Oscillator frequency is controlled by voltage at Vres1 or Vres2 pins
  - This model with Vres1 and Vres2
    - Vres2 is not mentioned in Qspice HELP, but indeed there
  - Oscillator uses minimum value of Vres1 and Vres2 to determine output frequency
    - Instance Parameter VRT (voltage on timing resistor) can affect Frequency
    - Freq of V(Q)
      $= \frac{\mathrm{RT}}{\mathrm{Rtime}} \times \mathrm{FREQ} \times \frac{\min(\mathrm{Vres1},\mathrm{Vres2})}{\mathrm{VRT}}$
  - Typically, the spare one would be used for frequency foldback when the output is shorted to ground
  - ** Fmult should connect to ¥ in using ExtOsc2 symbol
    - Method #1 : Connect to a Net ¥
    - Method #2 : Right Click symbol, System Properties, find Pin Name Fmult and assign Net as ¥, Fmult will be invisible and connect to ¥

¥-Device I2C

Qspice : Usage – I2C.qsch

- I2C
  - Master I²C
    - Not in Qspice HELP
  - Write information into slave registers to instruct its to perform task
  - Rising edge from **WRITE** will trigger I2C ¥-Device to output I2C data which defined in **addr=** and **data=**
  - SCL : Serial Clock
  - SDA : Serial Data

- Syntax
  - ¥nnn VDD VSS SDA SCL EN WRITE ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ ¥ I2C addr=<addr> data=<reg> <data> …

¥-Device I2C

Qspice : Usage – I2C.qsch

- ¥-Device I2C Timing Diagram
  - I2C Timing
    - [Address 7-bits][Read/Write][ACK][REG][ACK][DATA][ACT]
  - To Register 0C , Data 0x0F (0b00001111)
    - 0011111101000011001000011111
    - In HEX : [0x1f]01[0x0C]1[0b00001111]1
  - To Register 08, Data 0x01 (0b00000001)
    - 0011111101000010001000000011
    - In HEX : [0x1f]01[0x08]1[0b00000001]1

¥1
addr=0x1f
data=0x0C 0b00001111
0x0D 0b11111111
0x08 0b00000001

Address
0x1F=0b0011111 (7bits)

¥-Device I2C : Reference of I2C Timing Diagram

- I2C Bus : Write
  - Reference
    - Understanding the I2C Bus
    - Application Report SLVA704
    - https://www.ti.com/lit/an/slva704/slva704.pdf

Writing to a Slave On The I²C Bus

To write on the I²C bus, the master will send a start condition on the bus with the slave's address, as well as the last bit (the R/$\overline{W}$ bit) set to 0, which signifies a write. After the slave sends the acknowledge bit, the master will then send the register address of the register it wishes to write to. The slave will acknowledge again, letting the master know it is ready. After this, the master will start sending the register data to the slave, until the master has sent all the data it needs to (sometimes this is only a single byte), and the master will terminate the transmission with a STOP condition.

Figure 8 shows an example of writing a single byte to a slave register.

Master Controls SDA Line

Slave Controls SDA Line

Write to One Register in a Device

Figure 8. Example I²C Write to Slave Device's Register

Read=1
Write=0

€-Device

€-Device DAC : Build-in Symbol DAC8.qsym

Qspice : DAC - Basic.qsch

- DAC
  - DAC : Digital to Analog Converter
- Important Pins
  - Ref : the maximum voltage value that the DAC can reach
  - A7 : MSB of input word A
  - A0 : LSB of input word A
  - EN : Enable
- Instance Params
  - Common instance params can refer to ¥-Device section (e.g. REF)

€-Device DAC : Build-in Symbol DAC8strobe.qsym

Qspice : DAC - Strobe.qsch

- Strobe
  - In practice, it named as load DAC strobe (LDAC)
  - This pin transfers all input register data to the DAC registers
  - It resolve jitter in V(out) in previous slide
- Symbol
  - Build-in symbol with strobe is
  - DAC8strobe.qsym

€-Device DAC Instance Params : RDAC

Qspice : DAC - RDAC.qsch ; DAC - CODE.qsch

- RDAC
  - Rdac : R of R-2R DAC
  - R-2R resistor ladder network is inexpensive solution for digital to analog conversion
  - Reference
    - https://en.wikipedia.org/wiki/Resistor_ladder
  - Therefore, to prevent loading effect, a voltage buffer should be used in DAC output

- CODE
  - Code : Non-zero maps from -n to n-1 to 0 to 2n-1
  - **Default CODE=0**
  - CODE=0 : Unsigned
  - CODE=1 : Signed - two's complement

£-Device

£-Device : Dual Gate Driver

Qspice : Dual Gate Driver.qsch

- £-Device Dual Gate Driver
  - Syntax: £nnn Vdd Vss BOOST TOPGATE SW BOTGATE TCTRL BCTRL EN ¥ ¥ ¥ … GATEDRIVER [INSTANCE PARAMETERS]
  - This behavioral device expects Top Input and Bottom Input has deadtime in between

£ [Dual Gate Driver] Instance Params

Gate Driver Instance Parameters

Name	Description	Units	Default
BOOST	Add logic such that the top FET is off if the bottom is controled to be on		(not set)
BUCK	Add logic such that the bottom FET is off if the top is controlled to on		(not set)
CAPVDD	Capacitance from a logic output to Vdd(or BOOST)	F	0.
CAPVSS	Capacitance from a logic output to Vss(or SW)	F	0.
M	Number of parallel devices		1.
REF	Logic reference for enable input	V	(Vdd + Vss) ÷2
ROFF1	Resistance used to pull bottom FET to VSS	Ω	RON1
RON1	Resistance used to pull bottom FET to VDD	Ω	1.
ROFF2	Resistance used to pull top FET to SW	Ω	ROFF
RON2	Resistance used to pull top FET to BOOST	Ω	RON1
RPASSIVE	Resistance used to pull both FETs' gates low when the driver is not enabled	Ω	1Meg
TEMP	Instance temperature	°C	27.
TTOL	Temporal tolerance	s	
UVLO	Minimum Vdd-Vss voltage to operate	V	2.

£ [Dual Gate Driver] Instance Params - Buck and Boost

Qspice : Dual Gate Driver (Buck).qsch; Dual Gate Driver (Boost).qsch

- BUCK
  - Add logic such that the bottom FET is off if the top is controlled to on
  - **Tin** is inverted to drive Bottom AND gate
    - Bin connects to LOW
  - Switch of buck converter is generally at High Side

- BOOST
  - Add logic such that the top FET is off if the bottom is controlled to be on
  - **Bin** is inverted to drive Top AND gate
    - Tin connects to HIGH
  - Switch of boost converter is generally at Low Side

£-Device : Tri-state Buffer

- Tri-state Buffer
  - Syntax: £nnn Vdd Vss Q Q̄ IN EN ¥ ¥ ¥ … 3STATE [INSTANCE PARAMETERS]
  - A tri-state output allows multiple circuits to share the same output line(s)
  - Output into High-Z when disable

£-Device Tri-state Buffer			
EN	IN	Q	_Q
0	0	Hi-Z	Hi-Z
0	1	Hi-Z	Hi-Z
1	0	0	1
1	1	1	0

Ø-Device
.DLL

Workflow to Create .DLL 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

Workflow to Create .DLL 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 [10] F5
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Select All Ctrl+A

[8]

[9]

"cpp_demo_x1.dll" created successfully [10]

CAP OVR

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

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

// Implement module evaluation code here

[10] Right click > Compile DLL

- If success, a successful statement in status bar

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

Ø-Device Instance Parameters

Ø-Device Instance Parameters

Name	Description	Units	Default
COUT	Output capacitance	C	0.
MAXTIMESTEP	Maximum time between evaluations	s	Infinite
REF	Logic threshold for inputs declared as boolean	V	.5
ROUT	Output impedance	Ω	1000.
VHIGH	Logic high level for outputs declared as boolean	V	1.
VLSB	ADC and DAC incremental voltage for an LSB change for multi-bit binary data types	V	1.

Ø-Device Instance Params : Rout / Cout and Vhigh

Qspice : DLLlink - Rout Cout.qsch / DLLlink - VHIGH.qsch

- ROUT and COUT
  - Rout : Output Impedance
  - Cout : Output capacitance
  - **Default ROUT=1000Ω**
  - **Default COUT=0F**
  - Example
    - In this example, it shown circuit arrangement of Rout and Cout in .dll device output

- VHIGH
  - Vhigh : Logic high level for outputs declared as Boolean
  - **Default VHIGH=1V**

Ø-Device (Template)

- Ø-Device Advanced C++ Template Options
  - [1] Struct template
    - Declare the structure variable which can be use between function
    - Pointer variable, e.g. inst->[variable]
  - [2] MaxExtStepSize()
    - Decide maximum time step in simulation (maxstep)
    - Special usage : Can return -1e308 to abort simulation and goes to the next step (if any)
  - [3] Trunc()
    - Limit Temporal TOLerance timestep (TTOL)
  - [4] Display()
    - For diagnostic print statements
    - To display a string in output windows during simulation

Ø-Device (Template)

Qspice : DLLex.qsch

```c
// Automatically generated C++ file on Mon Nov 13 22:18:25 2023
//
// To build with Digital Mars C++ Compiler:
//
//    dmc -mn -WD dllex_x1.cpp kernel32.lib

#include <stdio.h>
#include <malloc.h>
#include <stdarg.h>
#include <time.h>

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 .DLL
// See https://docs.microsoft.com/en-us/windows/win32/dlls/dllmain for more information.
int __stdcall DllMain(void *module, unsigned int reason, void *reserved) { return 1; }

void display(const char *fmt, ...)
{ // for diagnostic print statements
   msleep(30);
   fflush(stdout);
   va_list args = { 0 };
   va_start(args, fmt);
   vprintf(fmt, args);
   va_end(args);
   fflush(stdout);
   msleep(30);
}

void bzero(void *ptr, unsigned int count)
{
   unsigned char *first = (unsigned char *) ptr;
   unsigned char *last  = first + count;
   while(first < last)
      *first++ = '\0';
}
```

union uData
Define all data types

display()
To print in console
As msleep(30), can slow down simulation each time display() is called

bzero()
Reset all value in struct to zero

```c
// #undef pin names lest they collide with names in any header file(s) you might include.
#undef in
#undef out

struct sDLLEX_X1
{
  // declare the structure here
};

extern "C" __declspec(dllexport) void dllex_x1(struct sDLLEX_X1 **opaque, double t, union uData *data)
{
   double  in  = data[0].d; // input
   double  K   = data[1].d; // input parameter
   double &out = data[2].d; // output

   if(!*opaque)
   {
      *opaque = (struct sDLLEX_X1 *) malloc(sizeof(struct sDLLEX_X1));
      bzero(*opaque, sizeof(struct sDLLEX_X1));
   }
   struct sDLLEX_X1 *inst = *opaque;

// Implement module evaluation code here:

}

extern "C" __declspec(dllexport) double MaxExtStepSize(struct sDLLEX_X1 *inst)
{
   return 1e308; // implement a good choice of max timestep size that depends on struct sDLLEX_X1
}

extern "C" __declspec(dllexport) void Trunc(struct sDLLEX_X1 *inst, double t, union uData *data, double *timestep)
{ // limit the timestep to a tolerance if the circuit causes a change in struct sDLLEX_X1
   const double ttol = 1e-9;
   if(*timestep > ttol)
   {
      double &out = data[2].d; // output

      // Save output vector
      const double _out = out;

      struct sDLLEX_X1 tmp = *inst;
      dllex_x1(&(&tmp), t, data);
   // if(tmp != *inst) // implement a meaningful way to detect if the state has changed
   //    *timestep = ttol;

      // Restore output vector
      out = _out;
   }
}

extern "C" __declspec(dllexport) void Destroy(struct sDLLEX_X1 *inst)
{
   free(inst);
}
```

[1] Struct template
Define pointer data used throughout the code

if(!*opaque) only run at beginning

main code

Main

MaxExtStepSize() [2]
To determine maximum time step

Trunc() [3]
To determine if any next timestep

destroy()
Call this function upon termination of the simulation

Another good reference from @physicboy
https://github.com/physicboy/QSPICE/blob/main/PWM_example_SRbuck/Three%20Complexity%20Levels%20of%20PWM%20Implementation%20for%20Digital%20Controller%20Simulation.pdf

Ø-Device (Template) : [2] MaxExtStepSize()

Qspice : DLLlink_MaxExtStepSize_Abort.qsch | dlllink_mest.cpp

- MaxExtStepSize()
  - Return value in MaxExtStepSize() determines maxstep
  - Return 1e308 equivalent maxstep is infinite (default)
  - This example demonstrates change of maxstep from infinite to 0.02 during simulation

```
struct sDLLLINK_MEST
{
  // declare the structure here
  float t;
};

extern "C" __declspec(dllexport) void dlllink_mest(stru
{
   double  in  = data[0].d; // input
   double &out = data[1].d; // output

   if(!*opaque)
   {
      *opaque = (struct sDLLLINK_MEST *) malloc(sizeof(
      bzero(*opaque, sizeof(struct sDLLLINK_MEST));
   }
   struct sDLLLINK_MEST *inst = *opaque;

// Implement module evaluation code here:
   out = in;
   inst->t = t;
}

extern "C" __declspec(dllexport) double MaxExtStepSize(
{
   //return 1e308; // implement a good choice of max ti
   if (inst->t > 49.9)
      return 0.2e-1;
   else
      return 1e308;
}
```

assign an inst->t

inst->t = t as MaxExtStepSize() will not accept t but only inst->t

If condition
- Before t<49.9s, maxstep set to max
- After t>49.9s set maxstep=0.02

Ø-Device (Template) : [2] MaxExtStepSize()

Qspice : DLLlink_MaxExtStepSize_Abort.qsch | dlllink_maxextstepsize.cpp

- MaxExtStepSize()
  - A special usage (abort simulation)
    - Have MaxExtStepSize() to return -1e308 can abort simulation and goes to the next step (if any)
    - ** Specifically -1e308, no other number

```
struct sDLLLINK_MAXEXTSTEPSIZE
{
  // declare the structure here
  double x;
};
```

Declare pointer inst->x

```
extern "C" __declspec(dllexport) void dlllink_maxextste
{
   double in = data[0].d; // input

   if(!*opaque)
   {
      *opaque = (struct sDLLLINK_MAXEXTSTEPSIZE *) mall
      bzero(*opaque, sizeof(struct sDLLLINK_MAXEXTSTEPS
   }
   struct sDLLLINK_MAXEXTSTEPSIZE *inst = *opaque;

// Implement module evaluation code here:
   inst->x = in;
}
```

Pointer is required to pass value of "in" to other function, in this case, to EaxExtStepSize()

```
extern "C" __declspec(dllexport) double MaxExtStepSize(
{
   if (inst->x > 0.5) {return -1e308;}
   return 1e308; // implement a good choice of max time
}
```

Ø-Device (Template) : [3] Trunc(), Example#1 - Rising Edge Detection

Qspice : DLL_CLK.qsch | dll_clk.dll

- Advanced C++ Template Options
  - Include a void **Trunc() template**
  - Include a **struct** template

- Purpose of this example
  - Limit timestep with Trunc() if the circuit causes a change
  - Struct to store instance variable which only used within C++ code (pointer to pass variable between functions)
  - Demonstrate how to implement rising edge detection and sampling time calculation

```
struct sDLL_CLK
{
  // declare the structure here
  bool last_clk;
  double lastT;
};

extern "C" __declspec(dllexport) void dll_clk(struct sDLL_CLK
{
   bool    clk = data[0].b; // input
   double  in  = data[1].d; // input
   double &out = data[2].d; // output

   if(!*opaque)
   {
      *opaque = (struct sDLL_CLK *) malloc(sizeof(struct sDLL_
      bzero(*opaque, sizeof(struct sDLL_CLK));
   }
   struct sDLL_CLK *inst = *opaque;

// Implement module evaluation code here:
   if (clk & !inst->last_clk){
      double T = t - inst->lastT;   // T : sampling period ca
      inst->lastT = t;
      out = in;
   }
   inst->last_clk = clk;
}

extern "C" __declspec(dllexport) void Trunc(struct sDLL_C
{ // limit the timestep to a tolerance if the circuit cau
   const double ttol = 1e-9;
   if(*timestep > ttol)
   {
      double &out = data[2].d; // output

      // Save output vector
      const double _out = out;

      struct sDLL_CLK tmp = *inst;
      dll_clk(&(&tmp), t, data);
   // if(tmp != *inst) // implement a meaningful way to detec
   //    *timestep = ttol;
      if((tmp.last_clk != inst->last_clk) & !inst->last_clk)
         *timestep = ttol;

      // Restore output vector
      out = _out;
   }
}
```

User code

[1] Declare structure variable
inst->last_clk : to store last clock
inst->last : to store last rising edge time

[2] detect rising edge (last_clk=0, clk=1)

[3] Calculate time between two rising edge (T)

Line#74, next slide

[4] if the circuit causes a change, limit the timestep to *timestep=ttol
User needs to define what the change (if condition) to force *timestep=ttol

Ø-Device (Template) : [3] Trunc() and ttol

Qspice : DLL_CLK.qsch | dll_clk.dll

Line#74 : const double ttol=1e-9;

Line#74 : const double ttol=1e308;

Ø-Device (Template) : [3] Trunc() and ttol

Qspice : DLL_CLK.qsch | dll_clk.dll

Line#74 : const double ttol=1e-9;

Comment of Trunc() template

- Most critical in Trunc() for user to modify are these 2 lines

```
const double ttol = 1e-9;
// if(tmp != *inst) // implement a meaningful way to detect if the state has changed
//    *timestep = ttol;
```

  - ttol : limit Temporal tolerance timestep
  - if (condition) : condition where timestep temporary force to ttol
    - In Trunc(), struct tmp is created for the purpose to compare *inst
    - User has to utilize tmp and *inst to detect if the state has changed

Ø-Device (Template) : [3] Trunc() and ttol – Example#2 Zero-crossing

Qspice : DLLlink_trunc.qsch | dlllink_trunc.cpp

- Trunc() example
  - This example is to reduce simulation timestep at signal zero-crossing
  - Code keynote
    - Use inst->x = in to pass data to Trunc()
    - In Trunc(), use if condition to force ttol when tmp.x within ±0.1

```
struct sDLLLINK_TRUNC
{
  // declare the structure here
  double x;
};

extern "C" __declspec(dllexport) void dll
{
   double in = data[0].d; // input

   if(!*opaque)
   {
      *opaque = (struct sDLLLINK_TRUNC *)
      bzero(*opaque, sizeof(struct sDLLLI
   }
   struct sDLLLINK_TRUNC *inst = *opaque;

// Implement module evaluation code here:
   inst->x = in;
}
```

```
const double ttol = 2e-8;
if(*timestep > ttol)
{

   // Save output vector

   struct sDLLLINK_TRUNC tmp = *inst;
   dlllink_trunc(&(&tmp), t, data);
// if(tmp != *inst) // implement a mear
//    *timestep = ttol;
   if (tmp.x < 0.1 & tmp.x > -0.1)
      *timestep = ttol;
```

** If use inst->x to replace tmp.x, couldn't achieve this pattern

Ø-Device (Template) : [4] Display()

Qspice : DLLlink_msg.qsch | dlllink_msg.cpp

- Display() example
  - Include display() can return a string in output window during simulation
  - This example detect zero crossing of input x and output its time (simulation time)
  - Display() is used to return this zero crossing time with a string

Include void display()

```
struct sDLLLINK_MSG
{
  // declare the structure here
  double last_x;
};

extern "C" __declspec(dllexport) void dlllink_msg(struct sDLLLINK
{
   double  x  = data[0].d; // input
   double &t0 = data[1].d; // output

   if(!*opaque)
   {
      *opaque = (struct sDLLLINK_MSG *) malloc(sizeof(struct sDLL
      bzero(*opaque, sizeof(struct sDLLLINK_MSG));
   }
   struct sDLLLINK_MSG *inst = *opaque;

// Implement module evaluation code here:
   if (x>=0 & inst->last_x<0)
   {
      t0 = t;
      display("x zero crossing time (rising) = %fms\n",t0/1e-3);
   }
   inst->last_x = x;
}
```

Display a string

Ø-Device (.DLL) Supplementary

Ø-Device : Simulation time and call from directory

Qspice : cpp_time.cpp

- Time
  - **t** is simulation time in C++ code

```
// #undef pin names lest they collide witl
#undef time

extern "C" __declspec(dllexport) void cpp_
{
    double &time = data[0].d; // output

// Implement module evaluation code here:
    time = t;
}
```

- Dll from other directory
  - 1st attribute can accept absolute or relative path name
  - If space in directory path, add "" for string format
    - e.g. "C:\Qspice Cpp\cpp_time"
  - To call for example cpp_time.dll, not to include .dll
  - Don't attempt to modify .cpp when path included

Ø-Device : Simulation time – Analog and DLL Time

Qspice : Analog and DLL Time.qsch

- Analog and DLL Time
  - Exact time "Time" is analog Newton-Raphson iteration which can be called from Behavioral source V=Time
  - .DLL "t" is dll time from very last timestep

- In this example
  - out0 output .DLL time "t", which is a different by a timestep to analog time "Tnewton" : Timestep = V(Tnewton)-V(Tc++)
  - In the same way, for out1=in1, .DLL output is last timestep data sample and therefore, out1 not exactly equal in1 at same time!

```
extern "C" __declspec(dllexport) void myn
{
   double  in0  = data[0].d; // input
   double  in1  = data[1].d; // input
   double  in2  = data[2].d; // input
   double  in3  = data[3].d; // input
   double &out0 = data[4].d; // output
   double &out1 = data[5].d; // output
   double &out2 = data[6].d; // output
   double &out3 = data[7].d; // output

// Implement module evaluation code here:

   out0=t;
   out1=in1;
}
```

out0 = t ;

out1 = in1 ;

According to Mike Engelhardt

1. Tnewton is the time in the analog Newton-Raphson iteration
2. The .DLL gets the time from the very last timestep

.tran 100µ
.plot V(PWL)-V(PWLc++)
.plot V(Tnewton)-V(Tc++)
.plot V(x)

For verification

.option maxstep=10n

Ø-Device : Simulation time – Analog and DLL Time

Qspice : Analog and DLL Time.qsch

- Change of Timestep #1
  - Different between DLL t and analog time can be affected by assigning max time step with .option maxstep
  - In this example, time different is clamped to equal maxstep setting

- Change of Timestep #2
  - Qspice is adaptive time step and timestep reduces with TTOL in SW, ¥-Device, DLL etc…
  - An example is PULSE Voltage Source, where time step is reduced at rise or fall with default TIMECTRL=LIMITS

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>
  - The reason V(out) looks delayed is because output of .DLL is always one time step delay

```
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
```

×-Device Transformer

×-Device: Transformer

Qspice : Transformer - Basic.qsch

- ×-Device: Transformer
  - Syntax: ×nnn «PRI+ PRI- SEC1+ SEC1- SEC2+ SEC2 [...]» <TURNS=N1 N2 N3 ...> [Additional Instance Parameters]
  - Ideal Transformer Equation (Inductance → ∞)
    - $n = \frac{N_1}{N_2} = \frac{v_1}{v_2}$ and $Z_{N1} = n^2 Z_{N2}$

.ac dec 100 1 1G

Turn Ratio = 2:1

.func Zin() V(N1)/-(I(V1))
.plot Zin()
.plot V(N1)/V(N2)

×-Device: Transformer Instance Params

Transformer Instance Parameters

Name	Description	Units	Default
ISAT[1]	Current that drops inductance to SATFRAC of zero-current value	A	Infinite
K[1]	Alternate means of specifying LEAK		1
L	Primary zero-current inductance	H	Infinite
LEAK	Leakage inductance	H	0
LSAT[1]	Inductance asymptotically approached in saturation	H	10% of L
RPAR[1]	Primary parallel resistance	Ω	Infinite
RSER[1]	Primary series resistance	Ω	0.0
SATFRAC[1]	Fractional drop in L at ISAT		0.7
TURNS	List of relative number of turns		

× Instance Params : L, Leak, Rser, Rpar, Isat, Lsat, Satfrac

Qspice : Transformer - L Leak Rser Rpar.qsch ; Transformer - Isat Lsat Satfrac.qsch

- L, Leak, Rser, Rpar
  - L : Primary Inductance
  - Leak : Leakage Inductance
  - Rser : Series Resistance
  - Rpar : Parallel Resistance
  - Equivalent model is shown in this simulation

- ISAT, LSAT, SATFRAC
  - Isat : Current that drops inductance to SATFRAC*L
  - Satfrac : Fractional drop in L at Isat
  - Lsat : Inductance asymptotically approached in saturation
  - These parameters are ignored unless L is given