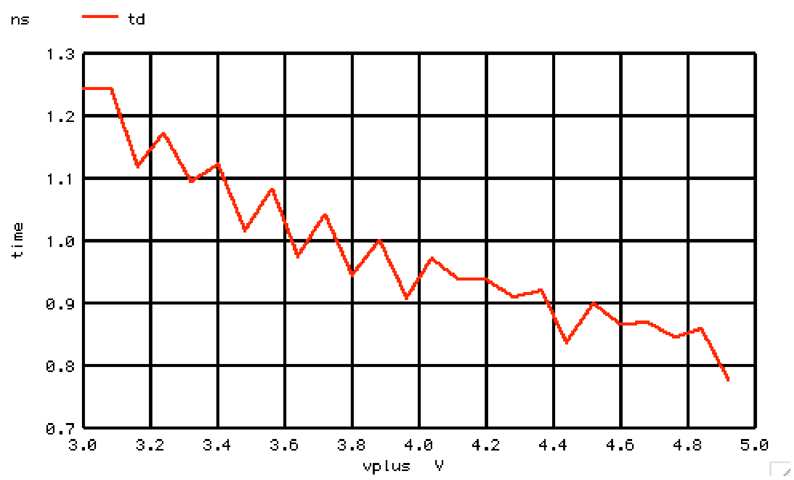
=============PropDelay_vs_VCC====================

WAVEFORMS IN SPICE ARE STORED AS ARRAYS.
ARRAYS CAN BE PROCESSED TO EXTRACT RELATIONSHIPS,
WHICH CAN FURTHER BE PLOTTED AS 2D FUNCTIONS.



Spice stores waveforms in arrays, and having access to that data can come in handy. Suppose one wants to know how the propagation delay of an inverter changes as a function of supply voltage.

```
*=========The_Inverter_SubCircuit=========================================
*
*                  ^
*   INV_R        /_\ VCC
*                 |
*               <-    1200ps at 6 stages at 3V
*              ||___   700ps at 6 stages at 5V
*           __ ||
*          |   ||_ MP1
*  |VIN|__ |     |  ____    2V overdrive at VCC = 3V
*  |___|   |     |__|VOUT   4V overdrive at VCC = 5V
*          |    _|  |____|
*          |   ||
*          |__ ||___ MN1
*              ||->
*                _|_
*                ///
*
.SUBCKT    INV_R  IN    OUT    VCC
MN1        OUT    IN    0      0      NMOSC    W=1u     L=1u
MP1        OUT    IN    VCC    VCC    PMOSC    W=3u     L=1u
C1         OUT    0     6f
.ENDS      INV_R
==========================================================================
```

An inverter can be modeled as a sub-circuit which uses standard MOS transistor models.

```
CMOS_Delay_vs_VCC
*          XINVR1        XINVR2                       XINVR3       XINVR4       XINVR5       XINVR6
*         |\            |\          ____             |\           |\           |\           |\
*  |VIN|_ | \/\_|VR2|_  | \/\_|VR3|_//  \ \_|R3B|_   | \/\_|VR4|_ | \/\_|VR5|_ | \/\_|VR6|_ | \/\_|VR7|
*  |___|  | /\/  |___|  | /\/ |___| \   \// |___|    | /\/ |___|  | /\/ |___|  | /\/ |___|  | /\/ |___|
*     |   |/            |/          \____/           |/           |/           |/           |/
*    _|_                              VN
*   / _ \
*  // \  \      _|_
*  \    \//    /VCC\     replace(OPT-SPACE)=>SPACE
*   \____/     \___/
*    _|_        _|_      1.18.10_6.42PM dsauersanjose@aol.com     www.idea2ic.com
*    ///        ///
==============================================================================
```

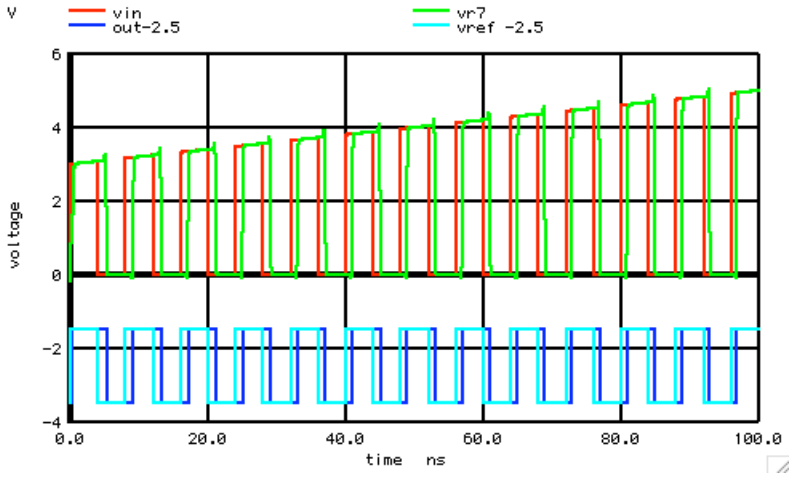
One can apply a piece wise linear ramp up waveform to

the supply and do a transient analysis on an array of inverters.

```
.OPTIONS   GMIN=1e-18     METHOD=traps
VCC        VCC     0      PWL      ( 0 3 100n 5 )     DC 2
VCLK       VCLK    0      DC 0     PULSE( 0  1 1p 6p  6p  4n 8n )
BIN        VIN     0      V =      V(VCC)*u( V(VCLK)-.5)
XINVR1     VIN     VR2    VCC      INV_R
XINVR2     VR2     VR3    VCC      INV_R
XINVR3     VR3     VR4    VCC      INV_R
XINVR4     VR4     VR5    VCC      INV_R
XINVR5     VR5     VR6    VCC      INV_R
XINVR6     VR6     VR7    VCC      INV_R
BOUT       OUT     0      V =      2*u( V(VR7)-V(VCC)/2) -1
BREF       VREF    0      V =      2*u( V(VCLK)-.5) -1
*TRAN      TSTEP   TSTOP  TSTART   TMAX         ?UIC?
.tran      10p     100n   0        10p         UIC
*=========Run_Sim==============================================
.control
run
set        pensize = 2
plot       vin     vr7 out-2.5    vref -2.5
```

==========

One might also want to scale the input signal to track VCC and perhaps to scale both input and output to unity to see the delay better.



Having access to the waveform data provides a better way to view the delay compared to looking at a plot.

```
*=========Create_AnySize_Arrays=================================================
compose    anysize_r      start = 0 stop = 99 step =1
compose    anysize_t      start = 0 stop = 99 step =1
compose    anysize_v      start = 0 stop = 99 step =1
let        num =          length(out)-5
let i  =   0
let t  =   0
let n  =   0
```

==========

How many transitions happen at the output waveform may not be known. But say one knows it is less than 100.

```
*=========Find_Edge_Timing======================================================
repeat     $&num
if         ( vref[i] < 0 & vref[i+1] > 0)
let t =    time[i]
let        anysize_r[n]= t
let v =    vcc[i]
let        anysize_v[n]= v
echo       n=   $&n   out_rise_ref= $&t  vcc = $&v
endif

if         ( vref[i] > 0 & vref[i+1] < 0)
let        t =   time[i]
let        anysize_r[n]= t
```

```
let v =   vcc[i]
let       anysize_v[n]= v
echo      n=   $&n   out_fall_ref= $&t vcc = $&v
endif

if        ( out[i] < 0 & out[i+1] > 0)
let t =   time[i]
let       anysize_t[n]= t
echo      n=   $&n   out_rise= $&t
let       n =   n +1
endif

if        ( out[i] > 0 & out[i+1] < 0)
let       t =   time[i]
let       anysize_t[n]= t
echo      n=   $&n   out_fall= $&t
let       n =   n +1
endif
let       i =   i +1
endrepeat

let       n3 =  n -1
```

==========

Some simple math can detect and printout each edge of the waveform. The following is the printout.

```
*=========Find_Edge_Timing==============================================
n = 0 out_rise_ref =  2.984E-12   vcc = 3.00006
n = 0 out_rise =      3.28366E-10
n = 1 out_fall_ref =  4.0088E-09  vcc = 3.08018
n = 1 out_fall =      5.25173E-09
n = 2 out_rise_ref =  8.004E-09   vcc = 3.16008
n = 2 out_rise =      9.121E-09
n = 3 out_fall_ref =  1.20088E-08 vcc = 3.24018
n = 3 out_fall =      1.31812E-08
n = 4 out_rise_ref =  1.60038E-08 vcc = 3.32008
n = 4 out_rise =      1.70992E-08
n = 5 out_fall_ref =  2.00088E-08 vcc = 3.40018
n = 5 out_fall =      2.11308E-08
n = 6 out_rise_ref =  2.4004E-08  vcc = 3.48008
n = 6 out_rise =      2.50205E-08
n = 7 out_fall_ref =  2.80088E-08 vcc = 3.56018
n = 7 out_fall =      2.90905E-08
n = 8 out_rise_ref =  3.2004E-08  vcc = 3.64008
n = 8 out_rise =      3.29786E-08
n = 9 out_fall_ref =  3.60088E-08 vcc = 3.72018
n = 9 out_fall =      3.70502E-08
n = 10 out_rise_ref = 4.0004E-08  vcc = 3.80008
n = 10 out_rise =     4.09482E-08
n = 11 out_fall_ref = 4.40088E-08 vcc = 3.88018
n = 11 out_fall =     4.50099E-08
n = 12 out_rise_ref = 4.8004E-08  vcc = 3.96008
n = 12 out_rise =     4.89125E-08
n = 13 out_fall_ref = 5.20088E-08 vcc = 4.04018
n = 13 out_fall =     5.29797E-08
n = 14 out_rise_ref = 5.60028E-08 vcc = 4.12006
n = 14 out_rise =     5.69421E-08
n = 15 out_fall_ref = 6.00088E-08 vcc = 4.20018
n = 15 out_fall =     6.09481E-08
n = 16 out_rise_ref = 6.40028E-08 vcc = 4.28006
n = 16 out_rise =     6.49121E-08
n = 17 out_fall_ref = 6.80088E-08 vcc = 4.36018
n = 17 out_fall =     6.89281E-08
n = 18 out_rise_ref = 7.2004E-08  vcc = 4.44008
n = 18 out_rise =     7.28411E-08
n = 19 out_fall_ref = 7.60088E-08 vcc = 4.52018
n = 19 out_fall =     7.69081E-08
n = 20 out_rise_ref = 8.00028E-08 vcc = 4.60006
n = 20 out_rise =     8.08688E-08
n = 21 out_fall_ref = 8.40088E-08 vcc = 4.68018
n = 21 out_fall =     8.48781E-08
n = 22 out_rise_ref = 8.80028E-08 vcc = 4.76006
n = 22 out_rise =     8.88483E-08
n = 23 out_fall_ref = 9.20088E-08 vcc = 4.84018
n = 23 out_fall =     9.28681E-08
n = 24 out_rise_ref = 9.6004E-08  vcc = 4.92008
n = 24 out_rise =     9.67813E-08

*=========Create_Edge_Time_Arrays=======================================
compose   vplus         start = 0 stop = $&n3 step =1
compose   td            start = 0 stop = $&n3 step =1
```

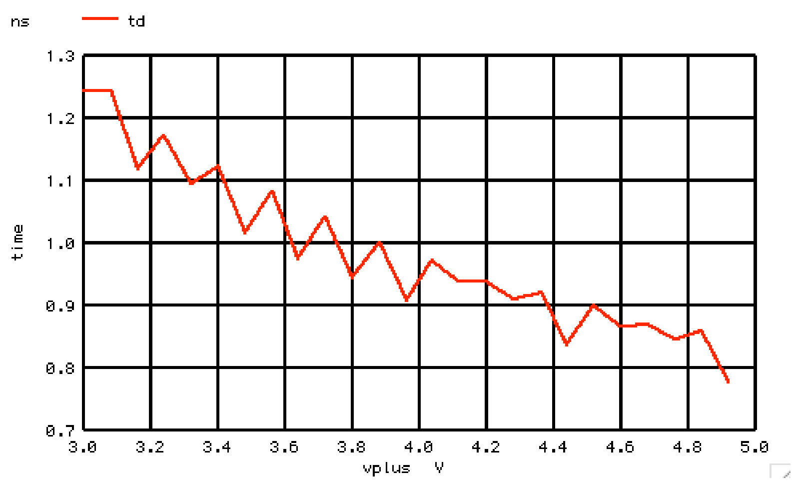
```
================================================================================
```

Keep track of the number of transitions, and now create two arrays. The first array is the supply voltage when a transition happened. The second is the time of each transitions.

```
*=========Transfer_Arrays===========================================
let        i  =  0
repeat     $&n
let        vplus[i] = anysize_v[i]
let        td[i] =    anysize_t[i]- anysize_r[i]
let        i =   i +1
endrepeat

let        td[0]  = td[1]
let        td[n2] = td[n2-1]
plot       td    vs   vplus
.endc
================================================================================
```

The delay array is composed of subtracting time points in the first array. The endpoints may needs some adjustment. The new vplus[] voltage array and new td[] time array will automatically be set to voltage units and time units. Spice's XY plot can plot the results.




```
===========Full_Netlist_For_Copy_Paste======================
CMOS_Delay_vs_VCC
*          XINVR1       XINVR2      ____                 XINVR3      XINVR4      XINVR5      XINVR6
*        |\            |\          /    \               |\           |\           |\           |\
*   |VIN|_| \/\_|VR2|_| \/\_|VR3|_//  \  \_|R3B|_| \/\_|VR4|_| \/\_|VR5|_| \/\_|VR6|_| \/\_|VR7|
*   |___| | /\/ |___| | /\/ |___| \    \// |___| | /\/ |___| | /\/ |___| | /\/ |___| | /\/ |___|
*     |   |/          |/           \____/        |/          |/          |/          |/
*     |                             VN
*   __|__         _|_
*  /  _  \       /VCC\
* //  \   \      \___/     replace(OPT-SPACE)=>SPACE
* \    \//         |
*  \____/         _|_
*    _|_          ///      1.18.10_6.42PM dsauersanjose@aol.com     www.idea2ic.com
*    ///

.OPTIONS  GMIN=1e-18    METHOD=traps
VCC       VCC     0      PWL      ( 0 3 100n 5 )     DC 2
VCLK      VCLK    0      DC 0     PULSE( 0  1 1p 6p  6p  4n 8n )
BIN       VIN     0      V =      V(VCC)*u( V(VCLK)-.5)
XINVR1    VIN     VR2    VCC      INV_R
XINVR2    VR2     VR3    VCC      INV_R
XINVR3    VR3     VR4    VCC      INV_R
XINVR4    VR4     VR5    VCC      INV_R
XINVR5    VR5     VR6    VCC      INV_R
XINVR6    VR6     VR7    VCC      INV_R
BOUT      OUT     0      V =      2*u( V(VR7)-V(VCC)/2) -1
BREF      VREF    0      V =      2*u( V(VCLK)-.5) -1
*TRAN     TSTEP   TSTOP  TSTART   TMAX        ?UIC?
.tran     10p     100n   0        10p         UIC
*=========Run_Sim=========================================
.control
```

```
run
set       pensize = 2
plot      vin     vr7 out-2.5   vref -2.5
*=========Create_AnySize_Arrays=============================================
compose   anysize_r     start = 0 stop = 99 step =1
compose   anysize_t     start = 0 stop = 99 step =1
compose   anysize_v     start = 0 stop = 99 step =1
let       num =       length(out)-5
let i  =  0
let t  =  0
let n  =  0
*=========Find_Edge_Timing==================================================
repeat    $&num
if        ( vref[i] < 0 & vref[i+1] > 0)
let t =   time[i]
let       anysize_r[n]= t
let v =   vcc[i]
let       anysize_v[n]= v
echo      n=   $&n   out_rise_ref= $&t  vcc = $&v
endif
if        ( vref[i] > 0 & vref[i+1] < 0)
let       t =   time[i]
let       anysize_r[n]= t
let v =   vcc[i]
let       anysize_v[n]= v
echo      n=   $&n   out_fall_ref= $&t vcc = $&v
endif
if        ( out[i] < 0 & out[i+1] > 0)
let t =   time[i]
let       anysize_t[n]= t
echo      n=    $&n   out_rise= $&t
let       n =   n +1
endif
if        ( out[i] > 0 & out[i+1] < 0)
let       t =   time[i]
let       anysize_t[n]= t
echo      n=   $&n   out_fall= $&t
let       n =   n +1
endif
let       i =   i +1
endrepeat
let       n3 =  n -1
*=========Create_Edge_Time_Arrays===========================================
compose   vplus        start = 0 stop = $&n3 step =1
compose   td           start = 0 stop = $&n3 step =1
*=========Transfer_Arrays===================================================
let       i  =  0
repeat    $&n
let       vplus[i] = anysize_v[i]
let       td[i] =    anysize_t[i]- anysize_r[i]
let       i =   i +1
endrepeat

let       td[0] = td[1]
let       td[n2] = td[n2-1]
plot      td    vs   vplus

.endc
*=========The_Inverter_SubCircuit============================================
*
*                ^
*  INV_R        /_\ VCC
*                 |
*              <-   1200ps at 6 stages at 3V
*            ||___   700ps at 6 stages at 5V
*          __||
*  ___    |  ||_ MP1
* |VIN|__|      |___    2V overdrive at VCC = 3V
* |___|  |      |__|VOUT  4V overdrive at VCC = 5V
*        |      |___|
*        |_
*          |__||
*             ||___ MN1
*             ||->
*              _|_
*             ///
*
.SUBCKT   INV_R  IN    OUT    VCC
MN1       OUT    IN    0      0      NMOSC   W=1u    L=1u
MP1       OUT    IN    VCC    VCC    PMOSC   W=3u    L=1u
C1        OUT    0     6f
.ENDS     INV_R
*==================The_CMOS_Model_Files====================================
.model            NMOSC             NMOS(
+ Level=  8       Tnom=27.0
```

```
*------------------Process-----------------------------------------------
+ tox=160e-10       xj=0.25e-06         nch=0.5e+17
*------------------V_threshold-------------------------------------------
+ vth0=0.72         nlx=0.12e-06
*------------------Bulk--------------------------------------------------
+ k1=1.04           k2=-1.209E-01
+ cdsc=-2.4E-4      cdscd=-1.506E-04    cdscb=-2.219E-04
*------------------mobility----------------------------------------------
+ u0=678            ua=8.964e-10
+ ub=1.472e-18      uc=-4.441E-17       vsat=86000
*------------------Subthresshold-----------------------------------------
+ nfactor=1.8
+ cit=-5.0E-04      voff=-7.862E-02
+ eta0=4.441e-16    etab=-2.E-01        dsub=0.7
*------------------Hot electrons-----------------------------------------
* alpha0=1.61e-05   beta0=36.68
*------------------VAF---------------------------------------------------
+ lint=.12e-06      pclm=.19            pscbe1=3.79e+08    pscbe2=9.4e-05
+ delta=0.01655     pvag=0.4484
*------------------Bulk_diode--------------------------------------------
+ js=5.858e-08
*------------------Resistance--------------------------------------------
+ rsh=70            rdsw=375
+ wr=0.7586         prwb=0              prwg=-4.441E-17
*------------------Capacitance-------------------------------------------
+ cj=0.0002424      cjsw=2.73e-10       mj=0.3551           mjsw=0.3873
+ cgso=9e-13        cgdo=9e-13          cgbo=7e-10
+ pb=0.5614         pbsw=0.8            xpart=0
+ dlc=5e-08         dwc=1.5e-07
*------------------BulkChargeEffect--------------------------------------
* a0=0.7            a1=0                a2=1                ags=0.05583
* b0=6.305e-08      b1=6.579e-08        keta=-1.531E-02
*------------------ShortChannel------------------------------------------
+ dvt0=2.2          dvt1=0.53           dvt2=-1.521E-01     drout=0.76
+ pdiblcb=.4        pdiblc1=0.00886     pdiblc2=0.00029
*------------------NarrowChannel-----------------------------------------
+ w0=2.6e-04        wint=0.16e-06
+ ww=-9.525E-14     wwn=1.0
+ dvt0w=0           dvt1w=5.3e6         dvt2w=-1.E-01
+ k3=2.53           k3b=-5              dwg=0               dwb=0
*------------------Noise-------------------------------------------------
* af=1              kf=1e-28            ef=0.95
*------------------Temperature-------------------------------------------
* pvsat=0           ute=-1.258E+00      kt1=-3.85E-01
* kt11=0            kt2=-3.098E-02      ua1=5.705e-09
* ub1=-1.147E-17    uc1=-1.302E-01      at=20380
* prt=-3.287E+02    lk1=0               lk2=0
* lvsat=0           la0=0               lags=0              lute=0
+ luc=0                                                                )


.model              PMOSC               PMOS(
+ Level=  8         Tnom=27.0
*------------------Process-----------------------------------------------
+ tox=1.725e-08     xj=2e-07            nch=1e+17
*------------------V_threshold-------------------------------------------
+ vth0=-0.90        nlx=1.84e-08
*------------------Bulk--------------------------------------------------
+ k1=0.3969         k2=0.03536
*------------------mobility----------------------------------------------
+ u0=268            ua=4.124e-09
* ub=-3.8E-19       uc=-1.512E-02       vsat=140000
*------------------Subthresshold-----------------------------------------
+ nfactor=.5016
* cit=0.0001        voff=-0.08          nfactor=1.0016    vth0=-0.94
* eta0=0.038        etab=-1.057E-02     dsub=0.3501
*------------------Hot electrons-----------------------------------------
+ alpha0=2.5e-08     beta0=28.92
*------------------VAF---------------------------------------------------
+ lint=.1e-06       pclm=3.2            pscbe1=5e+08        pscbe2=1e-05
* delta=0.009       pvag=4
*------------------Bulk_diode--------------------------------------------
+ js=3.95e-08
*------------------Resistance--------------------------------------------
+ rsh=135           rdsw=2500
* prwg=-7.2E-02     prwb=0.06           wr=0.8625
*------------------Capacitance-------------------------------------------
+ cj=0.0002424      cjsw=2.73e-10       mj=0.3551           mjsw=0.3873
+ cgso=9e-13        cgdo=9e-13          cgbo=7e-10
* cdsc=-2.4E-4      cdscd=-1.506E-01    cdscb=-2.219E-04
* pb=0.85           pbsw=0.88           xpart=0
* dlc=5e-08         dwc=1.5e-07
*------------------BulkChargeEffect--------------------------------------
* a0=1              a1=0                a2=2                ags=0.268
```

```
* b0=5e-07          b1=1e-07            keta=-1.0E-02
*------------------ShortChannel-----------------------------------------
* dvt0=2.9          dvt1=0.2            dvt2=-1.521E-01      drout=0.175
* pdiblcb=0         pdiblc1-            pdiblc2=0.001
*------------------NarrowChannel----------------------------------------
+ w0=2.6e-04        wint=0.16e-06
+ ww=-3.1E-19       wwn=1.9
* dvt0w=0.68        dvt1w=5.3e6         dvt2w=0.051
* k3=56             k3b=-3              dwg=0                dwb=1e-08
*------------------Noise------------------------------------------------
+ af=0.8            kf=1.50e-30         ef=0.95
*------------------Temperature------------------------------------------
* ldelta=0.02282    lpdiblc1=0.01877    ute=-1.500E+00
* cgsl=1.5e-10      cgdl=1.5e-10        ckappa=0.2463      cf=0
* kt1=-4.684E-01    kt1l=-2.0E-08       kt2=-2.818E-02
* ua1=-2.E-10       ub1=-4.5E-18        uc1=-2.000E-02
+ at=-1.5E+05       prt=1400                                          )

.end

4.11.10_4.54PM
dsauersanjose@aol.com
Don Sauer
http://www.idea2ic.com/
```