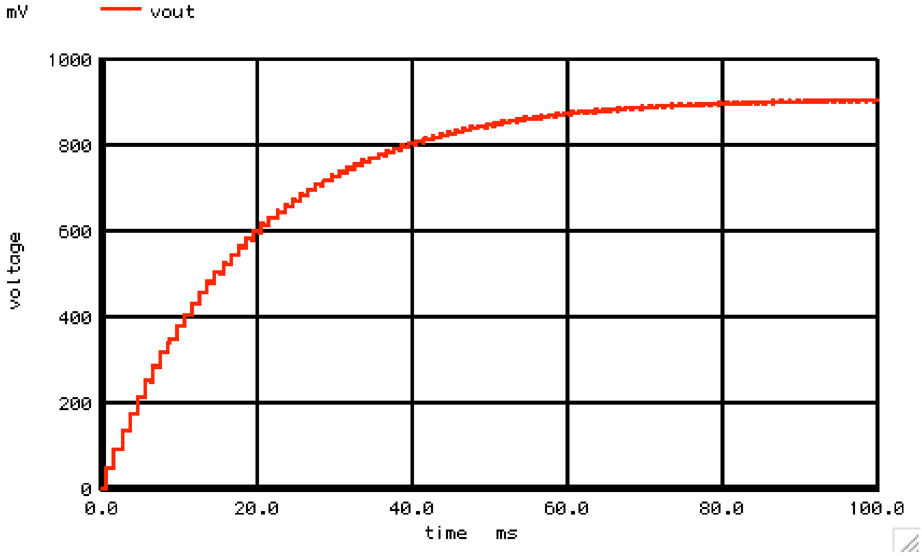
```
*=============SWITCH_CAPACITOR_POWER============
How energy efficient are switched capacitors?

=======================================================================
SWITCHING_POWER_LOSS
*
*
*              VR1             VR2         VOUT
*  VCC           S1 VC1    S2                     RL
*    _/\  /\  /\_/ ________/ ___/\  /\  /\_______/\  /\  /\_
*   _|_ \/  \/    :   _|_   :     \/  \/    _|_    \/  \/   |
*  /VCC\   R1   CNTL1 ___  CNTL2    R2       ___ COUT       _|_
*  \___/               |                     |             ///
*   _|_              _|_ C1                 _|_
*   ///              ///                   ///
*
*           VCNTL0    R1C1    20*10u  = 200us
*  VFREQ    _|_
*          /_  \      RLCOUT  10k*20u = 200msec
*   _|_   //\  \
*  /VFR\  \   \//     r=1/(freq*C) =1/1e-6*1e3 = 1kOhm
*  \___/   \___/
*   _|_     _|_
*   ///     ///
*
.OPTIONS  GMIN=1f       METHOD=trap   ABSTOL=1u     TEMP=27   srcsteps = 1  gminsteps = 1
.OPTIONS  RELTOL=.001   ABSTOL=1n     VNTOL=1u      ITL4=500  ITL1=400
*=========Create_Signal==================
VTime     VTime  0      DC     0      PWL(   0      0    1      1)
Vfreq     Vfreq  0      DC     1k
BVCNTL0   VCNTL0 0      V  =   sin( 6.283185*V(VFreq)*V(VTime))
BVCNTL1   VCNTL1 0      V  =   u( v(VCNTL0) -.3)
BVCNTL2   VCNTL2 0      V  =   u( -1*v(VCNTL0) -.3)
VCC       VCC    0      DC     1
R1        VCC    VR1    10
S1        VR1    VC1    VCNTL1 0      SWP
C1        VC1    0      1u
S2        VC1    VR2    VCNTL2 0      SWP
R2        VR2    VOUT   10
COUT      VOUT   0      20u
RL        VOUT   0      10K
.MODEL    SWP    SW(    VT=.5  VH=.2  RON=1m  ROFF=100MEG)
.control
*TRAN     TSTEP  TSTOP  TSTART TMAX   ?UIC?
tran      10u    100m   0      10u
set       pensize = 2
plot      vout
```

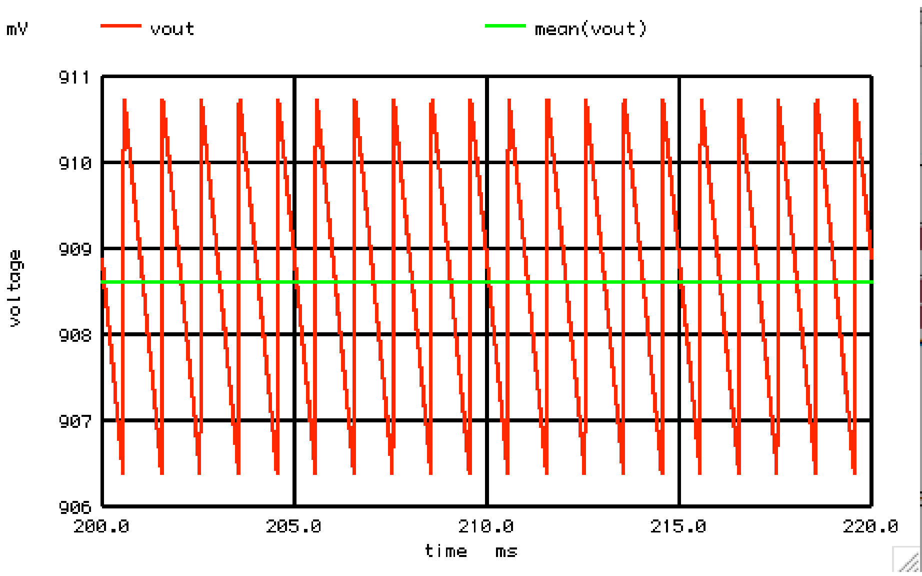



```
=====================================================================
The switch capacitor should appear as a 1k Ohm resistor.

       rout   = 1/(freq*C) =1/1e-6*1e3 = 1kOhm

With 20uF at the output, the RC should be 20msec.
So start the simulation after 200msec.

=====================================================================
tran        3u     220m      200m     3u
plot        vout   mean(vout)
```



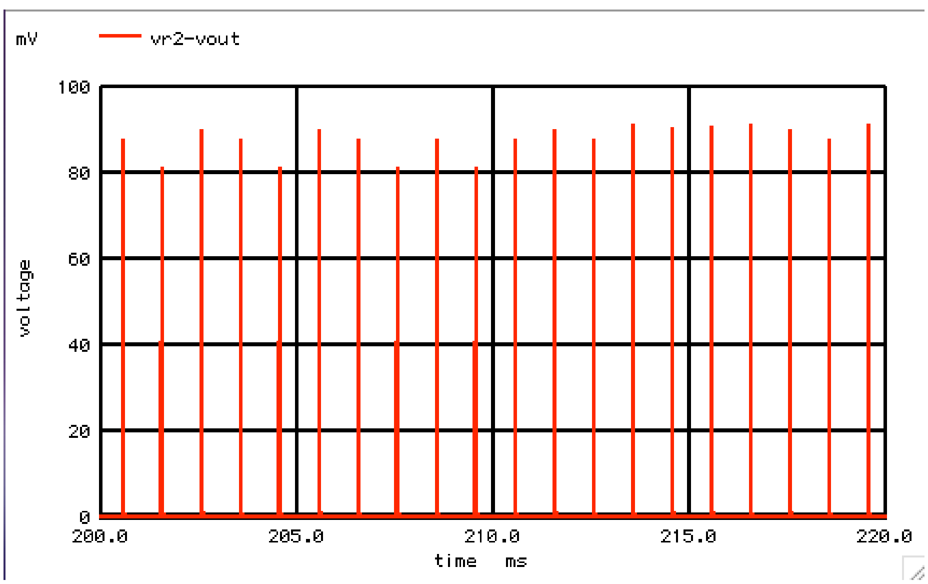

```
=====================================================================
Calculating power loss from DC values is easy.

=====================================================================
let          iout      =  mean(vout)/10K
let          pwrt      =  1*iout
let          pwrout    =  mean(vout)*iout
let          pwrloss   =  pwrt- pwrout
echo        "iout=$&iout pwrt =$&pwrt pwrout=$&pwrout  pwrloss =$&pwrloss  "
```

```
================================================================
iout    = 9.08614E-05   pwrt     = 9.08614E-05
pwrout = 8.25579E-05   pwrloss = 8.3035E-06
================================================================
```

But Calculating power loss in the switches is not so easy.

```
================================================================
SWITCHING_POWER_LOSS
*
*
*               VR1              VR2          VOUT
*  VCC              S1  VC1        S2                    RL
*     _/\  /\  /\_/ _________/ ___/\  /\  /\_______/\  /\  /\_
*    _|_  \/  \/    :   _|_      :    \/  \/      _|_   \/  \/   |
*   /VCC\     R1    CNTL1  ___   CNTL2      R2      ___ COUT      _|_
*   \___/                   |                        |            ///
*    _|_                   _|_  C1                  _|_
*    ///                   ///                      ///
*
*            VCNTLO      R1C1    20*10u  = 200us
*  VFREQ      _|_
*             /   \      RLCOUT  10k*20u = 200msec
*    _|_     //\   \
*   /VFR\    \   \//     r=1/(freq*C) =1/1e-6*1e3 = 1kOhm
*   \___/     \___/
*    _|_       _|_
*    ///       ///
================================================================
plot      vr2-vout
```




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

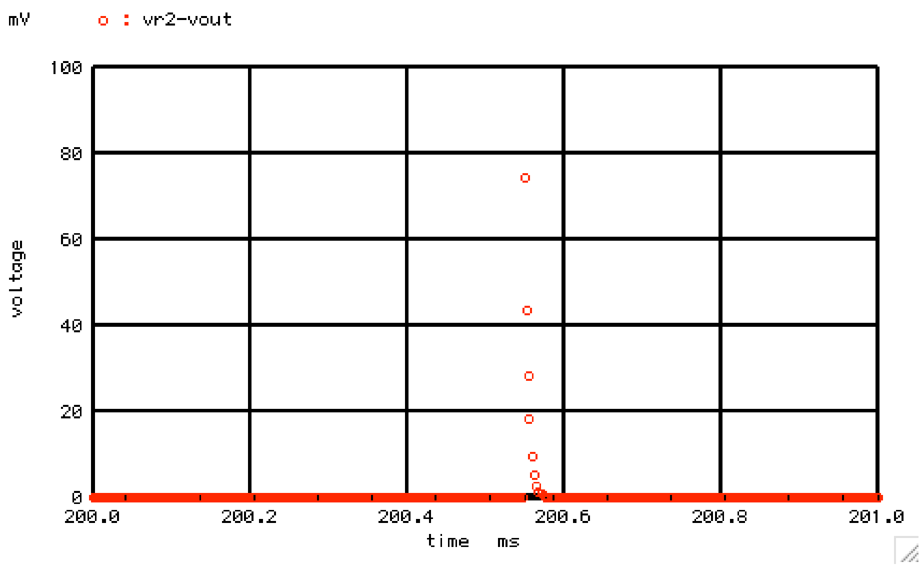
Switch energy is displayed as RC waveforms that are not uniformly spaced in time. The faster the waveform, the more time points.

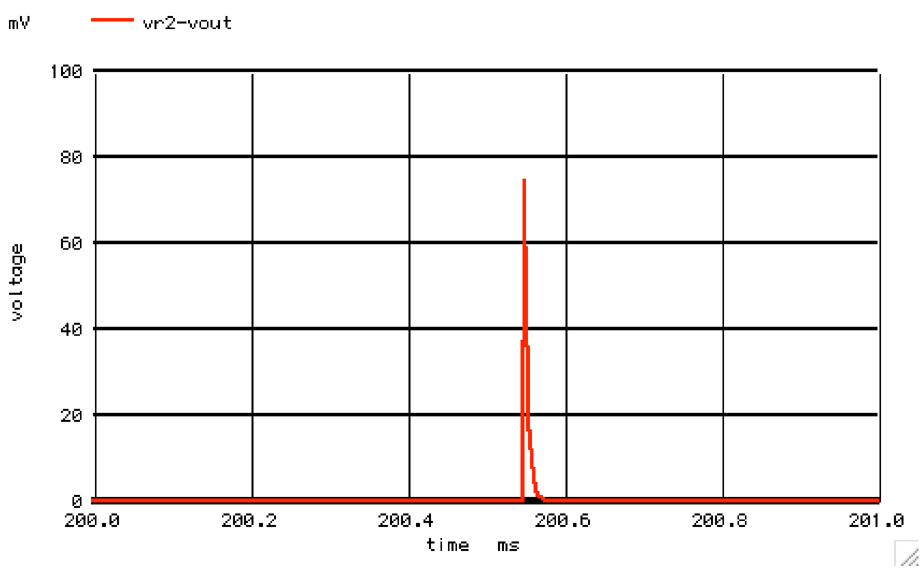
```
================================================================
let       numb   = length(vout)
let       numb2 = length(vout)-1

let       irr1 = (vcc-vr1)/10
let       irr1rms = sqrt(mean(irr1*irr1))
let       pwrr1 = irr1rms*irr1rms*10
```

```
let        irr2 = (vr2-vout)/10
let        irr2rms = sqrt(mean(irr2*irr2))
let        pwrr2 = irr2rms*irr2rms*10
let        pwrr1r2= pwrr1+ pwrr2
print      pwrr1r2

plot       vr2-vout pointplot xlimit 200m 201m
plot       vr2-vout           xlimit 200m 201m
```





pwrloss = 8.3035E-06

pwrr1r2 = 1.35964e-05

==

The non uniform timing of data points gives a calculated power in the switch resistors to be a factor of two too high.

Linearizing the time points will make things worse.

One solution is to scale each measurement by the difference

in time between all the points.

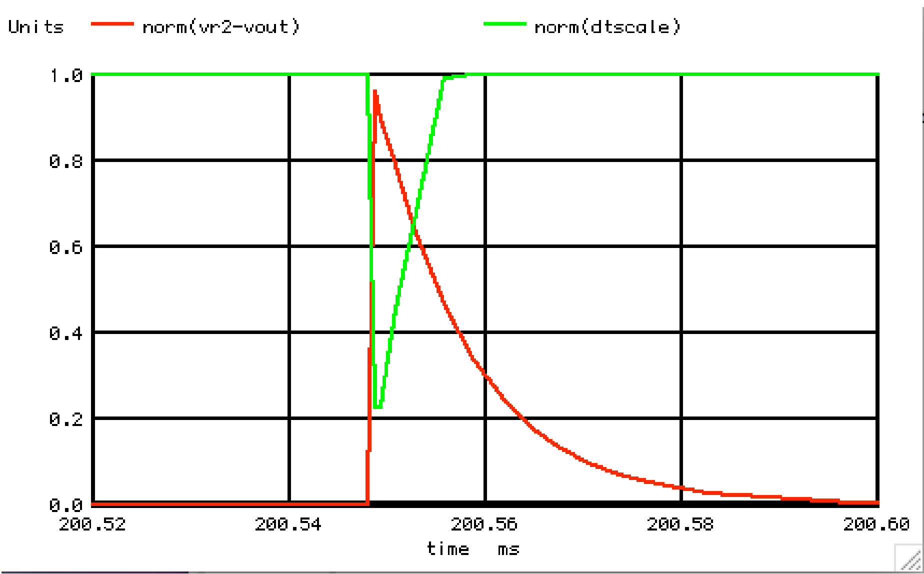
```
=======================================================================
set        numb            = length(vout)
let        numb2           = length(vout)-1
print      numb
let        dtscale         = vector($&numb)

let        indx            = 1
repeat     $&numb2
let        dtscale[indx]   = ($&numb)*(time[indx] -time[indx-1])/(time[$&numb2]-time[0])
let        indx            = indx +1
end

let        dtscale[0]      = dtscale[1]
let        irr1rms         = sqrt(mean(irr1*irr1*dtscale))
let        pwrr1           = irr1rms*irr1rms*vres[0]
let        irr2rms         = sqrt(mean(irr2*irr2*dtscale))
let        pwrr2           = irr2rms*irr2rms*vres[0]
let        pwrr1r2         = pwrr1+ pwrr2

print      pwrr1r2
print      vres[0]

plot       norm(vr2-vout) norm(dtscale) xlimit 200.52m 200.60m
```




```
numb     =   6.81100e+03
pwrr1r2  =   7.03901e-06
pwrloss  =   8.30716E-06
```

===

Now the power in the switch resistors come within 85% of the value calculated from the DC values.

A resistive voltage divider consisting of a 1K Ohm and a 10K Ohm resistor should produce a .9090V output with a 1 volt input. The simulated value is 99.946% of that. So powers loss is all due to the fact that the switch capacitor is looking like a simple 1K Ohm resistor.

```
     rout   = 1/(freq*C) =1/1e-6*1e3 = 1kOhm
```

But where is the power going? It is being lost as heat in R1 and R2.

```
=======================================================================
SWITCHING_POWER_LOSS
*
*
*              VR1             VR2         VOUT
*  VCC             S1 VC1    S2                     RL
*    _/\  /\  /\_/ ________/ ___/\  /\  /\_______/\  /\  /\_
*   _|_ \/  \/    :   _|_   :     \/  \/    _|_    \/  \/   |
*  /VCC\   R1   CNTL1 ___  CNTL2    R2      ___ COUT       _|_
*  \___/               |                     |             ///
*   _|_               _|_ C1                _|_
*   ///               ///                   ///
*
*           VCNTL0    R1C1    20*10u  = 200us
*  VFREQ    _|_
*          /_ \       RLCOUT  10k*20u = 200msec
*    _|_  //\  \
*   /VFR\ \   \//     r=1/(freq*C) =1/1e-6*1e3 = 1kOhm
*   \___/  \___/
*    _|_     _|_
*    ///     ///
=======================================================================
```

It is interesting to note that changing the values of R1 and R2 has no effect on the power loss. There is still a 1KOhm resistor from VCC to Vout. Decrease both R1 and R2 by half. Now the peak current will double. The RC power waveform of I*I*R will double in peak value. But the RC waveform will settle in half the time.

```
===========Full_Netlist_For_Copy_Paste=======================
SWITCHING_POWER_LOSS
*
*
*              VR1             VR2         VOUT
*  VCC             S1 VC1    S2                     RL
*    _/\  /\  /\_/ ________/ ___/\  /\  /\_______/\  /\  /\_
*   _|_ \/  \/    :   _|_   :     \/  \/    _|_    \/  \/   |
*  /VCC\   R1   CNTL1 ___  CNTL2    R2      ___ COUT       _|_
*  \___/               |                     |             ///
*   _|_               _|_ C1                _|_
*   ///               ///                   ///
*
*           VCNTL0    20*1u=20us
*  VFREQ    _|_
*          /_ \       Vin  10k*20u=200msec
*    _|_  //\  \
*   /VFR\ \   \//     r=1/(freq*C) =1/1e-6*1e3 = 1kOm
*   \___/  \___/
*    _|_     _|_
*    ///     ///
*
.OPTIONS  GMIN=1f       METHOD=trap   ABSTOL=1u     TEMP=27   srcsteps = 1  gminsteps = 1
.OPTIONS  RELTOL=.001   ABSTOL=1n     VNTOL=1u      ITL4=500  ITL1=400
*=========Create_Signal=================
VTime     VTime   0      DC     0      PWL(   0      0    1      1)
Vfreq     Vfreq   0      DC     1k
BVCNTL0   VCNTL0  0      V  =   sin( 6.283185*V(VFreq)*V(VTime))
BVCNTL1   VCNTL1  0      V  =   u( v(VCNTL0) -.3)
BVCNTL2   VCNTL2  0      V  =   u( -1*v(VCNTL0) -.3)
VCC       VCC     0      DC     1
R1        VCC     VR1    10
```

```
S1        VR1    VC1    VCNTL1   0      SWP
C1        VC1    0      1u
S2        VC1    VR2    VCNTL2   0      SWP
R2        VR2    VOUT   10
COUT      VOUT   0      20u
RL        VOUT   0      10K
.MODEL    SWP    SW(    VT=.5    VH=.2  RON=1m  ROFF=100MEG)

.control
*TRAN     TSTEP  TSTOP  TSTART TMAX   ?UIC?
tran      10u    100m    0       10u
set       pensize = 2
plot      vout

tran      3u    220m    200m    3u
plot      vout mean(vout)

let       iout = mean(vout)/10K
let       pwrt = 1*iout
let       pwrout =  mean(vout)*iout
let       pwrloss = pwrt- pwrout
echo      "iout=$&iout pwrt =$&pwrt pwrout=$&pwrout  pwrloss =$&pwrloss  "
plot      vr2-vout

*linearize
let       numb = length(vout)
let       numb2 = length(vout)-1
let       irr1 = (vcc-vr1)/10
let       irr1rms = sqrt(mean(irr1*irr1))
let       pwrr1 = irr1rms*irr1rms*10
let       irr2 = (vr2-vout)/10
let       irr2rms = sqrt(mean(irr2*irr2))
let       pwrr2 = irr2rms*irr2rms*10
let       pwrr1r2= pwrr1+ pwrr2
print     pwrr1r2

plot      vr2-vout pointplot xlimit 200m 201m
plot      vr2-vout xlimit 200m 201m
plot      vr2-vout

let       numb = length(vout)
let       numb2 = length(vout)-1
print     numb
let       dtscale  = vector($&numb)
let       indx = 1
repeat    $&numb2
let       dtscale[indx] = ($&numb)*(time[indx] -time[indx-1])/(time[$&numb2]-time[0])
let       indx = indx +1
end
let       dtscale[0]= dtscale[1]

let       irr1rms = sqrt(mean(irr1*irr1*dtscale))
let       pwrr1 = irr1rms*irr1rms*10
let       irr2rms = sqrt(mean(irr2*irr2*dtscale))
let       pwrr2 = irr2rms*irr2rms*10
let       pwrr1r2= pwrr1+ pwrr2
print     pwrr1r2

plot      norm(vr2-vout) norm(dtscale) xlimit 200.52m 200.60m


.endc

.end
```

9.16.10_12.57PM
dsauersanjose@aol.com
Don Sauer