Solution to Charge Pump Differential Equations

Silicon DSP Corporation
http://www.silicondsp.com

Author: Sasan Ardalan

Summary:

A charge pump "C" Code Block for Capsim simulation has been written based on the the formulas for the solution of the charge pump equations presented in Hanumolu, *et. al* [1]. The results show excellent correspondence with a Spice model. The differential equations were solved using State Space methods in [1] and accurately capture the initial conditions. A Capsim block has been developed called *CHPStateSpace* based on the approach in [1]. It has been successfully used to model a charge pump PLL. See separate documentation (http://www.silicondsp.com). Note the Capsim High Level Model of the PLL can be used prior to Spice simulations to speed up design and optimization.

Results:

Figure 1 shows the results for R=1kOhms, C_1=2pf and C_2=0.2pf. To show the impact of C_2 in smoothing the ripples Figure shows the case with C_2=2pf and C_1=2pf. Note this is for illustration. We need $C_1 >> C_2$.

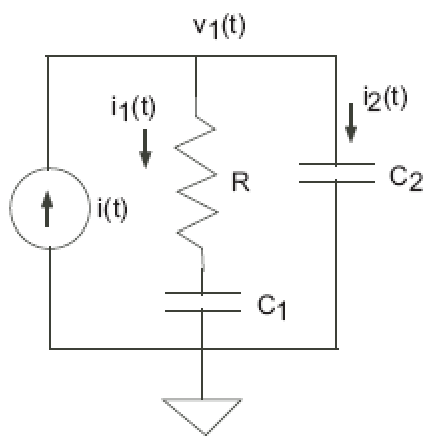


Figure 1 Charge Pump During Up Cycle

Appendix A shows the formulas from [1]. The C code is presented in Appendix B. Note **tp** is the duration of the "Up" pulse and **T_** is the duration for example of the period of the reference or next edge of VCO/N.   Ip is the current source value.

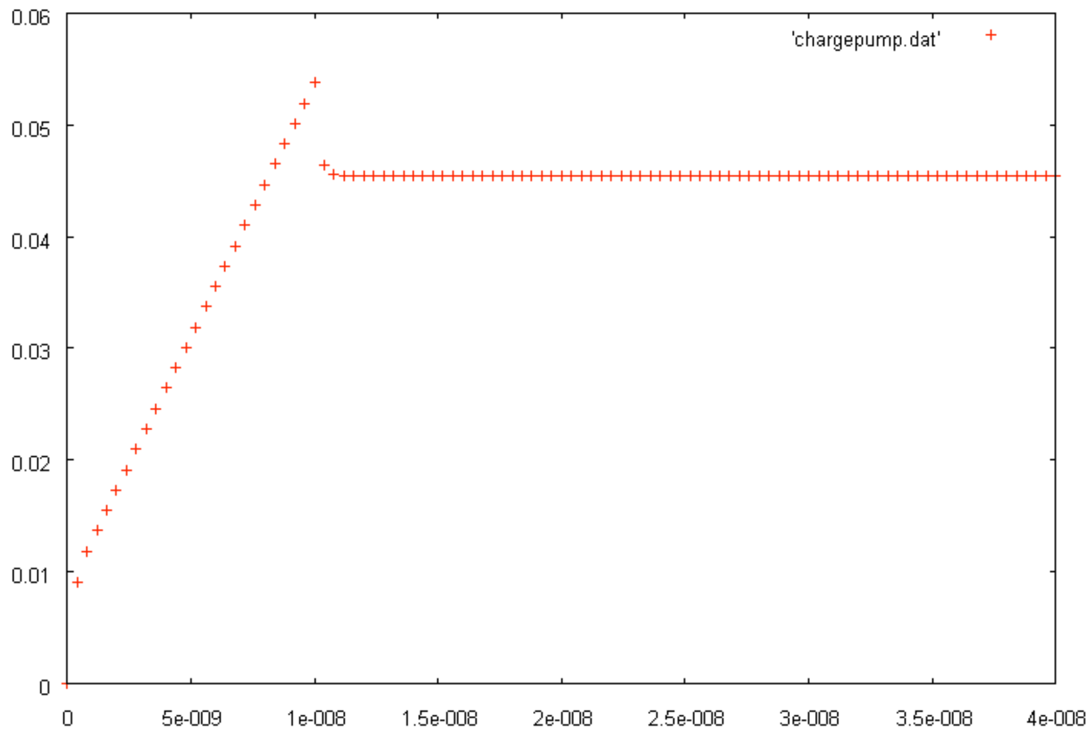


Figure 2 v(1) with C_1=2 pf and C_2=0.2 pf

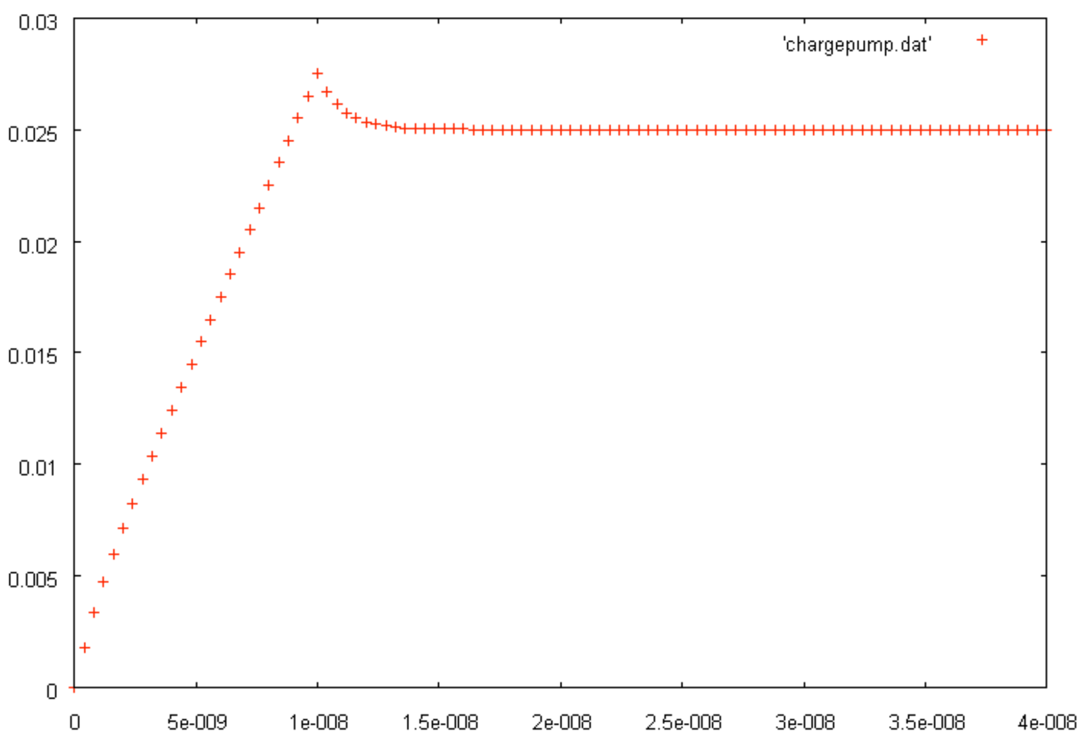


Figure 3 v(1) with C_1=2 pf and C_2=2 pf

References

[1] Hanumolu, P.K.; Brownlee, M.; Mayaram, K.; Un-Ku Moon,**"Analysis of charge-pump phase-locked loops"**, Circuits and Systems I: Regular Papers, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Volume 51, Issue 9, Date: Sept. 2004, Pages: 1665 – 1674

Appendix A

In the following from [1], v_{ctrl} is v_1. and v_c is v_2 across C_2. t_p is the on time of Up or Down pulse. T_ is related to the period of the reference or the next edge of VCO/N. That is the time $t_p<t<T\_$ is the off time of the Up or Down pulse. See Figure 3.

for $0 < t \le t_p$

$$v_{\text{ctrl}}(t) = v_{\text{ctrl}}(0)\left(g_1(t) + \omega_z g_2(t)\right) + v_c(0)\omega_2 g_2(t) + \frac{i_p}{C_2}\left(g_2(t) + \frac{\omega_z(g_1(t)-1)}{\omega_{p3}^2} + \frac{\omega_z t}{\omega_{p3}}\right)$$

$$v_c(t) = v_{\text{ctrl}}(0)\omega_z g_2(t) + v_c(0)\left(g_1(t) + \omega_2 g_2(t)\right) + \frac{i_p}{C_2}\left(\frac{\omega_z(g_1(t)-1)}{\omega_{p3}^2} + \frac{\omega_z t}{\omega_{p3}}\right)$$

for $t_p < t \le T_-$

$$v_{\text{ctrl}}(t) = v_{\text{ctrl}}(t_p)\left(g_1(t) + \omega_z g_2(t)\right) + v_c(t_p)\omega_2 g_2(t)$$

$$v_c(t) = v_{\text{ctrl}}(t_p)\omega_z g_2(t) + v_c(t_p)\left(g_1(t) + \omega_2 g_2(t)\right).$$

In the above, $\omega_2 = 1/RC_2$, $g_1(t) = \exp(-\omega_{p3}t)$, and $g_2(t) = (1/\omega_{p3})(1 - \exp(-\omega_{p3}t))$.

$$\omega_z = \frac{1}{RC_1}$$

$$\omega_{p3} = \frac{1}{R\left[\frac{C_1 C_2}{C_1 + C_2}\right]}$$

$$\omega_2 = 1/RC_2$$

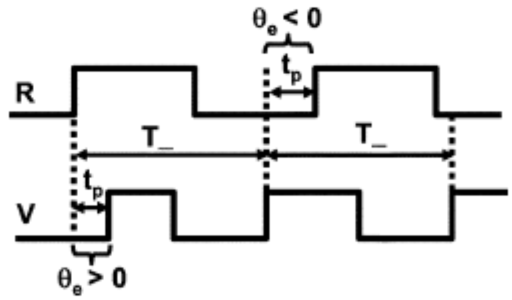


Figure 3 Definition of t_p ad T_ from [1].

In the above note that all the exponentials are based on e^{-at}, where, as we derived before:

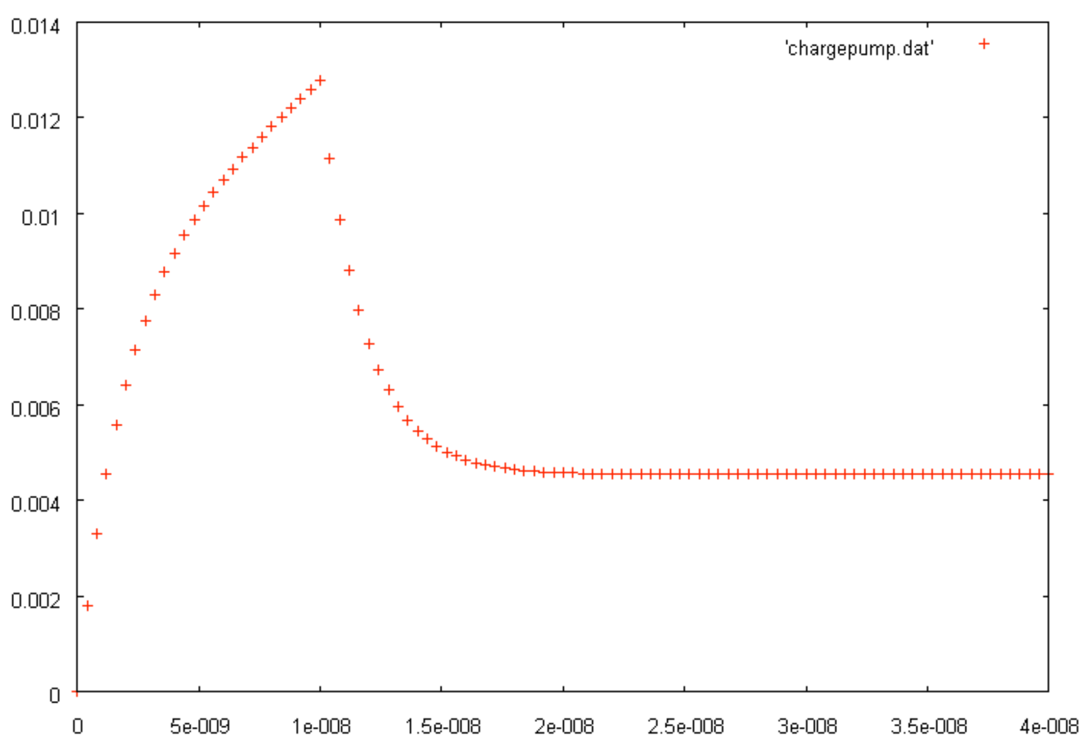
$$a = \frac{C_1 + C_2}{RC_1C_2}$$

Appendix B: C Code.

```
#include <math.h>
#include <stdio.h>


int main(int argc, char* argv[])
{

	double c1=2e-12;
	double c2=2e-12;
	double r=1000;
	double Ip= 0.00001;

	double dt=0.1e-9;
	double tmax=100e-9;
	double t;
	double tt;
	double tdelay=10e-9;
	double a;
	double v1;
	double invC2;
	double aInv;
	double coeff;
	double beta;

	double vctrl;
	double vctrlInit=0;

	double vc;
	double vcInit=0;
	double tp=10e-9;
	double T_;

	double g1;
	double g2;
```

```
	double w2;
	double wz;


	int i;
	int n=100;

	int flag=0;

	FILE *fp;

	fp=fopen("chargepump.dat","w");
	if(!fp) {
	 fprintf(stderr,"Could not open file to write:chargepump.dat
\n");
	   return 0;
	}

    tp=10e-9;
    T_=40e-9;
	tmax=T_;

    dt=tmax/(double)n;
	a=(c1+c2)/(r*c1*c2);
	aInv = 1/a;
	w2=1.0/(r*c2);
	wz=1.0/(r*c1);

	printf("a=%le\taInv=%le\tw2=%le\n",a,aInv,w2);

	vctrlInit=0;
    vcInit=0;
	flag=0;
	for(i=0; i<n+1; i++) {
		t=i*dt;


		if(t <= tp) {

		   g1=exp(-a*t);
		   g2=aInv*(1-exp(-a*t));


		    vctrl = vctrlInit*(g1+wz*g2)+
			   vcInit*w2*g2+
			   (Ip/c2)*(g2+wz*(g1-1)/(a*a)+wz*t/a);
		    vc=vctrlInit*wz*g2+vcInit*(g1+w2*g2)+
			   (Ip/c2)*(wz*(g1-1)/(a*a)+wz*t/a);
		}
		if ((t > tp) && (t<T_) && !flag) {

			     vctrlInit=vctrl;
			     vcInit=vc;
			flag=1;
		}
		if ((t > tp) && (t<T_) && flag) {
```

```
            tt=t-tp;
          g1=exp(-a*tt);
          g2=aInv*(1-exp(-a*tt));

            vctrl=vctrlInit*(g1+wz*g2)+vcInit*w2*g2;
            vc=vctrlInit*wz*g2+vcInit*(g1+w2*g2);

        }
        //printf("%le\t%le\n",t,vctrl);

  printf("%le\t%le\t%le\t%le\t%le\t%d\n",t,vctrl,vc,g1,g2,flag);
        fprintf(fp,"%le\t%le\n",t,vctrl);

    }




    fclose(fp);




  return 0;
}
```



Appendix C

The file: chargepump.dat

```
0.000000e+00 0.000000e+00
4.000000e-10  1.824200e-03
8.000000e-10  3.376678e-03
1.200000e-09  4.747014e-03
1.600000e-09  5.995259e-03
2.000000e-09  7.161662e-03
2.400000e-09  8.273205e-03
2.800000e-09  9.347975e-03
3.200000e-09  1.039809e-02
3.600000e-09  1.143169e-02
4.000000e-09  1.245421e-02
4.400000e-09  1.346931e-02
4.800000e-09  1.447943e-02
5.200000e-09  1.548621e-02
5.600000e-09  1.649076e-02
6.000000e-09  1.749380e-02
6.400000e-09  1.849585e-02
6.800000e-09  1.949722e-02
7.200000e-09  2.049813e-02
7.600000e-09  2.149875e-02
8.000000e-09  2.249916e-02
8.400000e-09  2.349944e-02
8.800000e-09  2.449962e-02
9.200000e-09  2.549975e-02
9.600000e-09  2.649983e-02
1.000000e-08  2.749989e-02
1.040000e-08  2.667572e-02
1.080000e-08  2.612327e-02
1.120000e-08  2.575295e-02
1.160000e-08  2.550472e-02
1.200000e-08  2.533832e-02
1.240000e-08  2.522678e-02
1.280000e-08  2.515202e-02
1.320000e-08  2.510190e-02
1.360000e-08  2.506831e-02
1.400000e-08  2.504579e-02
1.440000e-08  2.503069e-02
1.480000e-08  2.502057e-02
1.520000e-08  2.501379e-02
1.560000e-08  2.500924e-02
1.600000e-08  2.500620e-02
1.640000e-08  2.500415e-02
```

1.680000e-08	2.500278e-02
1.720000e-08	2.500187e-02
1.760000e-08	2.500125e-02
1.800000e-08	2.500084e-02
1.840000e-08	2.500056e-02
1.880000e-08	2.500038e-02
1.920000e-08	2.500025e-02
1.960000e-08	2.500017e-02
2.000000e-08	2.500011e-02
2.040000e-08	2.500008e-02
2.080000e-08	2.500005e-02
2.120000e-08	2.500003e-02
2.160000e-08	2.500002e-02
2.200000e-08	2.500002e-02
2.240000e-08	2.500001e-02
2.280000e-08	2.500001e-02
2.320000e-08	2.500000e-02
2.360000e-08	2.500000e-02
2.400000e-08	2.500000e-02
2.440000e-08	2.500000e-02
2.480000e-08	2.500000e-02
2.520000e-08	2.500000e-02
2.560000e-08	2.500000e-02
2.600000e-08	2.500000e-02
2.640000e-08	2.500000e-02
2.680000e-08	2.500000e-02
2.720000e-08	2.500000e-02
2.760000e-08	2.500000e-02
2.800000e-08	2.500000e-02
2.840000e-08	2.500000e-02
2.880000e-08	2.500000e-02
2.920000e-08	2.500000e-02
2.960000e-08	2.500000e-02
3.000000e-08	2.500000e-02
3.040000e-08	2.500000e-02
3.080000e-08	2.500000e-02
3.120000e-08	2.500000e-02
3.160000e-08	2.500000e-02
3.200000e-08	2.500000e-02
3.240000e-08	2.500000e-02
3.280000e-08	2.500000e-02
3.320000e-08	2.500000e-02
3.360000e-08	2.500000e-02
3.400000e-08	2.500000e-02
3.440000e-08	2.500000e-02
3.480000e-08	2.500000e-02

3.520000e-08 2.500000e-02
3.560000e-08 2.500000e-02
3.600000e-08 2.500000e-02
3.640000e-08 2.500000e-02
3.680000e-08 2.500000e-02
3.720000e-08 2.500000e-02
3.760000e-08 2.500000e-02
3.800000e-08 2.500000e-02
3.840000e-08 2.500000e-02
3.880000e-08 2.500000e-02
3.920000e-08 2.500000e-02
3.960000e-08 2.500000e-02
4.000000e-08 2.500000e-02