# Low-Power Frequency Multiplier with One Cycle Lock-In Time and 100 ppm Frequency Resolution, for System Power-Management

Rafael Fried[1] and Ziv Azmanov[2]

1 - EPFL Swiss Federal Institute of Technology, Department of Electrical Engineering, DE-LEG, 1015 Lausanne, Switzerland

2 - National Semiconductor, 1090 Kifer Road, Sannyvale, CA-94086, USA

## Abstract

A low-power Frequency Multiplier (FMUL) with 100 ppm frequency resolution, +/-100 ps jitter, and one cycle frequency lock-in time is presented. It is used to generate clock frequencies up to 100 MHz using a reference frequency of 32,768 Hz, for advanced power management both at a device level and at a system level. The FMUL is implemented in a standard digital CMOS process and its area is 0.5 mm$^2$ @ 0.8$\mu m$.

## Introduction

Power management and low-power design have become a critical requirement, and a key differentiator, in a range of application. These include: portable Personal Computers (PCs), desktop “green” PCs, and other portable, battery operated, devices. The objective of power management in portable, battery operated, systems is to extend the battery life to the maximum, while retaining the required functionality. Moreover, since most portable PCs today use passive cooling techniques, power management and low-power design are essential for system thermal management. Furthermore, PCs are on a route to converging into a multi-functional home appliance, providing computing services, networking and communication services, entertainment services, and eventually, control and monitoring of other home appliances, such as, security systems, etc. This would require that the system would never be turned off. To enable such applications, advanced power management techniques, coupled with very low-power design would be a most critical requirement.

The conventional power management techniques include: removing the power supply from idle devices, and stopping the clock to idle devices or modules, based on activity monitoring timers. This is sometimes referred to as passive power management. The more advanced and complex approach, is the active power management, by which power consumed by the system, or device, is reduced while it is actively in use.

Active power management enables system OEMs and semiconductor suppliers to further reduce power consumption, beyond the level which is achieved using passive power management techniques, and use it as a major competitive advantage and a key differentiator.

Active power management requires a run-time, programmable, control of the clock frequency. This means that the clock frequency is reduced, or increased, according to the required performance and functionality of the system or the device. Moreover, the transitions in the clock frequency need to be done instantly and in a manner which is totally transparent to the user. The FMUL presented here, meets all of the above requirements while consuming very low power itself.

It uses a reference clock of 32,768 Hz to generate clock frequencies up to 100 MHz. In one typical application, the presented FMUL is employed in the implementation of a power management and system controller, which uses three main modes of operation: *(i)* Fully-on mode: In which the operating frequency ranges between 5-10 MHz *(ii)* Idle mode: In which the operating frequency is reduced to 1 MHz *(iii)* Power-Save mode: In which the operating frequency is 32,768 Hz. The controller’s transitions from one operation mode to the other is triggered by hardware events, but managed by the firmware which also changes the clock frequency by programming the FMUL registers according to the desired operating frequency. Some additional values of the presented circuit include: reducing system cost by eliminating high frequency crystals, which also reduces the Electro-Magnetic-Interference (EMI).

Typically, on-chip high frequency generators, using low frequency external reference clock, are implemented by a PLL [1]-[5]. However, in those applications where there is no need to correlate the phase between the high and low frequency clocks, like for instance embedded system controllers and PC I/O peripherals, it will be preferable to use the FMUL approach, rather than the PLL approach.

The FMUL presented here is based on [6]. It generates a high frequency clock by means of digitally controlled current-controlled-oscillator (CCO). This frequency is controlled directly and not by its phase leg/lead, as done in a PLL.

ISLPED 1996 Monterey CA USA
0-7803-3571-8/96/$5.00 © 1996

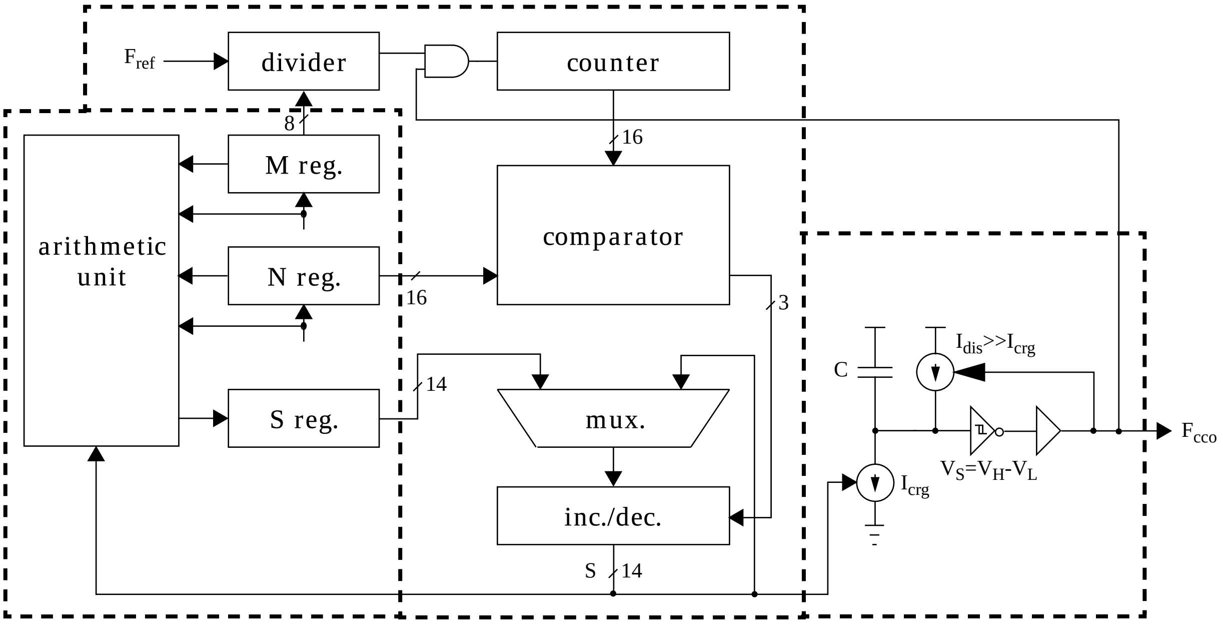


Fig. 1. FMUL principle block diagram.

Its main advantage over the conventional PLLs is the fact that theoretically unlimited frequency multiplication factors can be achieved while keeping very high frequency resolution, in the order of 100 ppm, and low current consumption. Trying to use PLL for the same task will cause instability problems, and consume very large area.

Additional advantages of the FMUL presented here, compared to a PLL, are the much shorter lock-in time, perfectly smooth clock transitions, and no need for additional external components for the loop filter, as needed when using a PLL to achieve high multiplication factors (low frequency reference clocks). Furthermore, the FMUL's performance is almost not affected by process parameters and supply voltage variations, and its stability is guarantied.

In this paper the basic principle of the FMUL is reviewed. Its accuracy and jitter causes are analyzed. Limitations to power consumption reduction are given. Solutions to achieve one cycle frequency lock-in time, and a solution to prevent discontinuity problems in the frequency curve, are presented. Results are shown.

## Digitally Controlled Frequency Multiplier

The FMUL's complete block diagram is presented in Fig.1. It consists of three modules: *(i)* the CCO, *(ii)* the CCO digital control logic, and *(iii)* the arithmetic module that is responsible for the one cycle frequency lock-in time.

Frequency locking is done in two phases, the first is coarse tuning, and the second is fine tuning. During the coarse tuning a binary search algorithm is implemented in a similar way to a successive approximation DAC, and during the fine tuning the frequency is incremented/decremented by small constant steps. The coarse tuning is done only once, during the chip's reset cycle.

The CCO configuration is a shmidt-trigger based current-driven oscillator, as shown in Fig. 2 [6]. The high frequency resolution is achieved by using 14 binary weighted current sources at the CCO, that operates similarly to a 14-bit DAC. However, in the FMUL presented here, as opposed to a DAC, neither linearity nor monotony of the high order bits is necessary.

The capacitor's charging current - $I_{crg}$, is digitally controlled by the value of the 14 bit control word - $S$ ($S_0$-$S_{13}$ in Fig. 2). The discharge current - $I_{dis}$, is constant, as given by (1), where $I_0$ is the value of the charge current LSB, $A$ is a constant and $A>>max\{S\}$.

$$I_{crg} = S \cdot I_0 \qquad I_{dis} = A \cdot I_0 \tag{1}$$

The CCO's output cycle time - $T_{cco}$ is defined in (2), where $V_S$ is the shmidt "voltage window", and $T_d$ is the discharge time including all the delays, which is approximately constant and much shorter than the charge time.

$$T_{cco} = T_d + \frac{C \cdot V_S}{S \cdot I_0} \tag{2}$$

The counter counts the number of $F_{cco}$ cycles during a time frame defined by $M$ cycles of the reference frequency - $F_{ref}$. This number is then compared to the value of $N$.

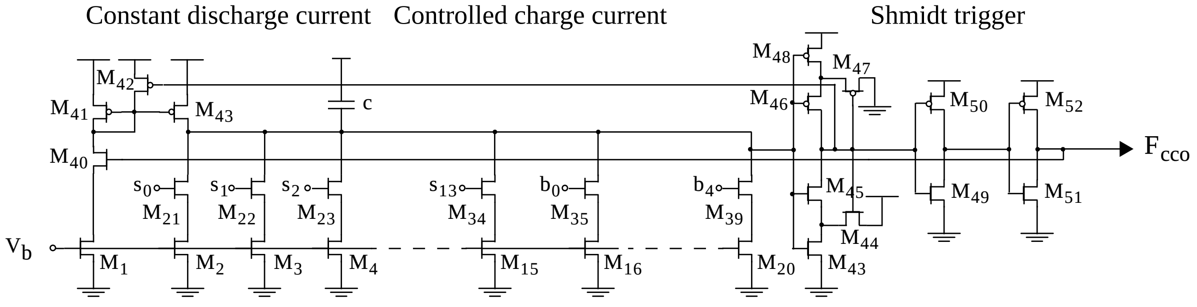


Fig. 2. Principle schematic of the Current Controlled Oscillator.

The desired relation between $F_{cco}$ and $F_{ref}$ is given by $F_{cco} = F_{ref} \cdot (N/M)$. The result of the comparison determines the next value of *S*. During coarse tuning this value is changed according to a binary search algorithm. During fine tuning it is either incremented or decremented by one LSB only.

The frequency resolution - $F_R$ is given by (3).

$$F_R \equiv \frac{\Delta T_{cco}}{T_{cco}} = \frac{1}{S} \cdot \frac{T_{cco} - T_d}{T_{cco}} \qquad (3)$$

Eq. (2) and (3) define the FMUL operating point as a function of the desired frequency resolution. Current consumption is inversely proportional to $F_R$, and its lower limit is defined by the minimum achievable $I_0$.

The detectable resolution - $D_R$ is given simply by $D_R = 1/N$ which is the detectable relative frequency error. It can be realized that *N* and *S* should be normally in the order of $1/F_R$.

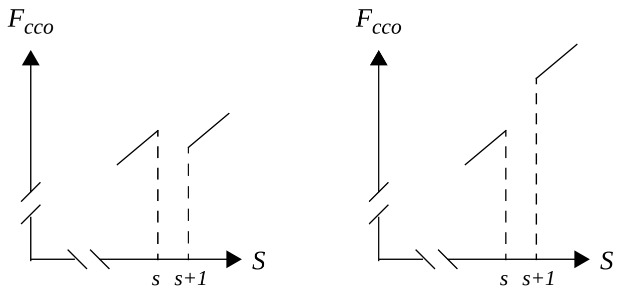
Jitter can be divided into two categories: *(i)* low frequency jitter due to the frequency correction every *M* reference cycles, and the phase error of the reference clock, *(ii)* high frequency jitter due to noises, where the most sensitive parameters are $V_S$ and $I_0$. The first category is inherently implemented in the FMUL and should be carefully examined for DSP applications such as voice compression. The second category, is design dependent and can be reduced by using proper circuit techniques.

Since frequency corrections are done only every *N* high-frequency cycles, it means that phase errors at the synthesized clock are averaged. On the other hand, due to the same reason, the corrections will be highly sensitive to reference clock phase errors. Thus, reference frequency should be generated by a crystal oscillator in order to minimize frequency drift problems, and instantaneous phase errors.

Frequency is controlled by discrete steps, and thus discontinuity may occur in the CCO (*S* to $F_{cco}$) curve, as shown in Fig. 3. The “gradient type” of discontinuity, shown in Fig 3a, will not affect the performance, on the other hand the “hole type” of discontinuity, shown in Fig 3b, might degrade frequency resolution, and cause jitter.

A special technique to escape from the possible “holes”, is implemented in the FMUL. Whenever more than 7 bits are simultaneously changed from “1” to “0” in the digital control word - *S*, (e.g., when the previous control was *S='01110011111111'* and the present one is *S='01110100000000'*), a bias charging current is incremented by one step - $a{*}I_0$ ($b_0$-$b_4$ in Fig. 2). This technique is a very simple way to move away from possible “holes” in the $F_{cco}$-*S* curve of the CCO, and impose the value of the control word - *S*, to be lower than the multiple simultaneous bits change value (assuming *a=8*, the control word, in the example above, will be *S='01110011111000'*).

.



a. “gradient” discontinuity.

b. “hole” discontinuity.

Fig. 3. Types of discontinuity in the frequency curve.

Many multi-clock chips and systems use three modes of operation: Fully-on, Idle, and Power-save. In order to achieve superior power management performance, both in the chip level and in the system level, active power management should be used. In this technique the system operating mode, and operating frequency are changed abruptly according to the system condition. The frequency lock-in time should be as short as possible and frequency independent. One cycle frequency lock-in time is achieved in the FMUL due to the CCO digital control, and the arithmetic unit. The CCO operating point is changed according to (4), where the index *i* is used to define the current operating point, and *f* the operating point after the transition.

$$\frac{S_f}{S_i} = \frac{I_{0i}}{I_{0f}} \cdot \frac{(M/N)_f \cdot T_{ref} - T_d}{(M/N)_i \cdot T_{ref} - T_d} \qquad (4)$$

The arithmetic unit is responsible for the implementation of (4). Whenever the system has to move from one frequency to another, the controller feeds the N and M registers with the new frequency multiplication factor, and the arithmetic unit outputs the new operating point - $S_f$.

Typically, in Idle mode, frequency resolution can be degraded compared to the Fully-on mode, thus $I_0$ can be kept constant and $S$ will be varied according to (4).

In case $F_R$ should be kept constant over all operation modes (Idle as well as Fully-on), $S$ has to be kept approximately constant, according to (3), and $I_0$ should be tuned to satisfy (4). This implies the need for a high-resolution DAC. In this case, it will be preferable to use a "self-trained" module, instead of the algorithmic one [7]. In this approach the system is being "moved" through all possible operation modes and frequencies, during a "learning" phase. This is done only once, at the system reset. All FMUL operating points are stored in registers in such a way that a one-cycle frequency lock-in time can be achieved, by recalling the proper operating point data from the registers.

Recovering from Power-save mode, where the FMUL is completely turned-off (zero current consumption) to Fully-on operation mode, is also performed in one cycle. This is achieved by waking the FMUL after Power-save mode, to the exact operating point that was acquired at Fully-on operation mode.

## Results

The FMUL was used in a PC I/O peripheral controller to generate a clock at the frequency of 48 MHz, using a reference clock of 32,768 Hz. To achieve 50% duty-cycle, $F_{cco}$ should be 96 MHz, implying *N/M=8789/3*. Current consumption in this configuration can be reduced down to 1.1 mA.

Fig. 4 presents an infinite scope samples (one on top of the other), and Fig. 5 shows spectrum analyzer measurement of the divided clock (24 MHz). Both a large scale and zoomed measurements are presented. The frequency 3 dB width is 6 KHz, implying a frequency standard deviation of 106 ppm, which can be translated to 4.4 ps.

## References

[1] R. Talmudi *et al.*, "A 100MIPS, 64b Superscalar Microprocessor with DSP Enhancements", ISSCC'91, pp. 100-101.

[2] D. Jeong *et al.*, "Design of PLL-Based Clock Generation Circuits", IEEE JSSC, vol. sc-22, pp. 255-261, Apr. 1987.

[3] J. Alvarez, H. Sanchez, G. Gerosa and R. Countryman, "A Wide-Bandwidth Low-Voltage PLL for PowerPC$^{TM}$ Microprocessors", IEEE JSSC, vol. 30, pp. 383-391, Apr. 1995.

[4] J. Dunning, G. Garcia, J. Lundberg and E. Nuckolls, "An All-Digital Phase-Locked Loop with 50-Cycle Lock Time Suitable for High-Performance Microprocessors", IEEE JSSC, vol. 30, pp. 412-422, Apr. 1995.

[5] Ilya I. Novof *et al.*, "Fully integrated CMOS Phase-Locked Loop with 15 to 240 MHz locking range and +/-50 ps jitter", IEEE JSSC, vol. 50, pp. 1259-1266, Nov. 1995.

[6] R. Fried and E. Rosin, "High Resolution Digitally Controlled Frequency Multiplier with Very Large Programmable Multiplication Factor", ESSCIRC'95.

[7] O. Falik and R. Fried, private correspondence, May 1995.

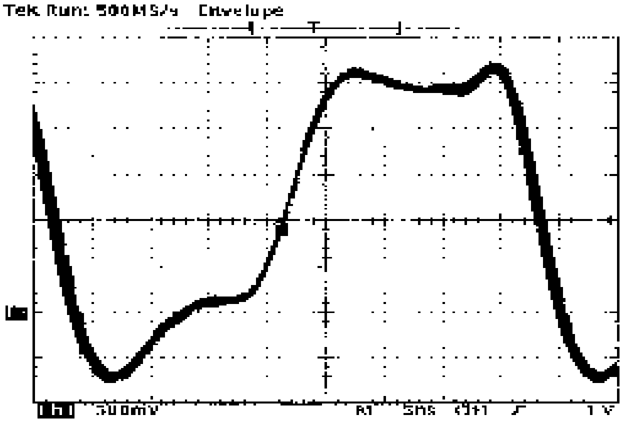
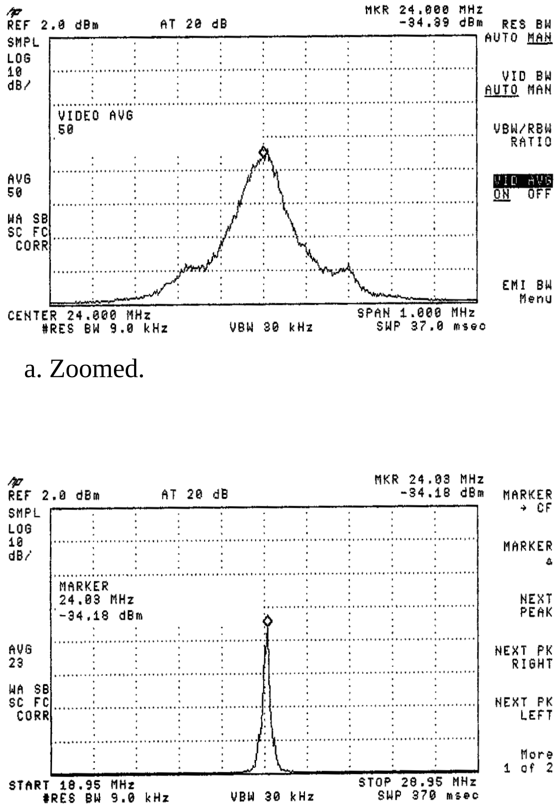


Fig. 4. Scope samples of the synthesized clock.



a. Zoomed.

b. Large scale.

Fig. 5. Spectrum of the synthesized clock.