# Portable digital clock generator for digital signal processing applications

T. Olsson and P. Nilsson

A fully integrated clock generator with behaviour similar to a PLL is proposed. A free-running ring oscillator is used as internal clock and the output clock is generated using two counters. The clock generator is described in synthesisable VHDL-code and can therefore easily be made from standard cells found in any commercial standard CMOS cell library.

*Introduction:* We consider a digital system where a sample clock is available and where digital parts need a second stable system clock source to perform signal processing. The system clock can be generated outside the chip, which has the disadvantage of adding off-chip components. Using off-chip clock sources also increases the power consumption since the system clock, which is of higher frequency than other I/O signals must be driven through an input pad. Another possibility is to generate the system clock on-chip, using a clock generator.

Robust and easy implemented fully digital clock multipliers without phase locking are proposed in [1, 2]. These clock multipliers produce a fixed number of cycles for each period of an external reference clock signal followed by an idle margin. For many digital applications, using such simpler clock generators is a feasible solution. To simplify communication between clock domains and to avoid metastability, an analogue PLL clock generator is often chosen. However, integrating an analogue PLL in a digital noisy environment is difficult. In addition, the analogue PLL is also sensitive to process parameters and must therefore be redesigned for each new technology. Many PLLs and other clock generator circuits are aimed at generating a high frequency to enable high system throughput. However, one of the design goals in a digital design is to implement the circuit with as low a clock frequency as possible. This is to decrease power consumption and to limit clock skew problems. Accordingly, there is a need for clock generators at more moderate clock frequencies. Many of the methods used for producing high frequency clocks are less suitable for producing a lower frequency. As an example, using a typical 0.18 μm process, the delay for one inverter in a ring-oscillator is 27 ps [3]. If a ring oscillator is used to directly produce a 10 MHz clock in a digital clock generator, this would require more than 1800 inverters. Of course it is possible to produce a much higher frequency (for instance a 1 GHz clock) and then divide it. For such clock division, jitter in the high frequency clock is multiplied by the division factor. Also, the large clock multiplication factor from reference clock to high frequency clock makes the design of the clock generator more difficult. To produce a phase-locked or known high frequency, the period time of the ring oscillator must be controlled down to a few picoseconds. Controllable oscillators tend to be large, power hungry and hard to design.

For the proposed clock generator, a high frequency clock is generated on-chip and the frequency of this clock is left unknown and uncontrolled. To produce such a clock, a simple free-running ring-oscillator can be used. The free-running device consists of a few logic gates and does not consume much power. To produce the desired frequency (for instance 10 times the sample clock), the unknown high frequency clock is used both to measure the length of the sample clock and to generate the new clock. The clock generator to implement this is shown to be simple and small in this Letter. The simplicity of the clock generator and the fact that it is made from digital standard leaf cells makes it possible to describe it in an HDL language. In fact, the entire design is described in fully synthesisable VHDL-code [4]. The VHDL description makes the clock generator process independent and also enables system simulation including the clock generator in any digital VHDL simulator.

*Implementation:* Fig. 1 shows the block structure of the clock generator. The oscillator is used as a high frequency timing reference for the entire design. It is used both to measure the length of the reference period using a counter in the block 'Ref counter' and to generate the output 'clk' using the block 'output counter'. The block 'D' contains at least one D-flip-flop to synchronise the reference to the oscillator. A reason for using more than one flip-flop is to decrease the risk of metastability [5]. Although the clock generator is not a PLL, the clock output has the appearance of being in phase lock. Since the output clock is generated based on information from the previous reference period, the acquisition time is always one reference period.

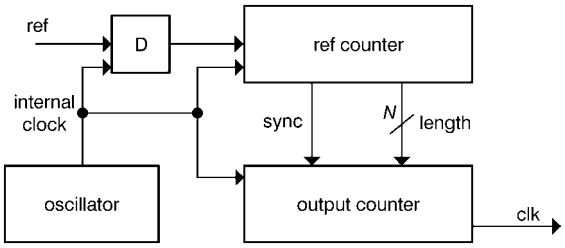


**Fig. 1** *Overview of digital clock generator*

*Oscillator:* The oscillator is implemented as a ring oscillator with one inverter replaced by a NAND-gate as shown in Fig. 2. The NAND-gate is used to enable power down by shutting down the oscillator. A frequency divider is inserted to give the ability to select between 4 clock rates $f$, $f/2$, $f/4$ and $f/8$, where $f$ is the output frequency from the ring oscillator.

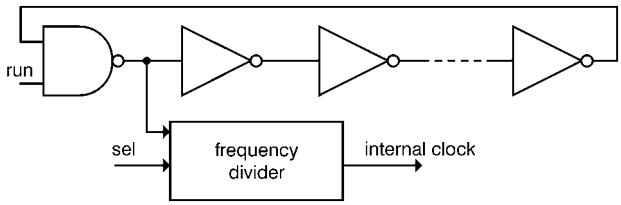


**Fig. 2** *Internal free-running clock generator made from ring oscillator*

In the VHDL description, an inverter in the ring oscillator is described as:

$$Q \Leftarrow \text{not(A) after invdel;}$$

where 'invdel' is an estimated delay for the inverter. This description using 'invdel' makes it possible to simulate the ring oscillator before synthesis. During synthesis, the clause 'after invdel' is automatically ignored and an inverter component with delay model from the cell library provider is inserted. Normally, the synthesis tool looks for possible Boolean simplifications and removes unnecessary gates, which means that a chain consisting of an even number of inverters is replaced by a wire. To obtain the synthesis tool to actually build a ring oscillator, a hierarchy protecting the ring oscillator from unwanted optimisations is created. Fig. 3 shows the hierarchy of the ring oscillator. The inverters are made as a component 'inverters' with one input and one output for each inverter. The connections between inverters are made on the next level.

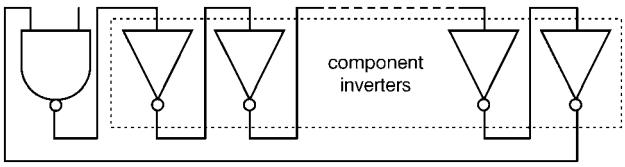


**Fig. 3** *Hierarchy of ring oscillator*

*Reference counter:* The reference counter uses the high rate clock from the oscillator to measure the length of one reference period. Output from the reference counter is a measurement ('length' in Fig. 1) of the previous reference period in number of oscillator periods. Also, a 1-bit signal ('sync' in Fig. 1) is generated at every transition from low to high of the reference. The signal 'sync' is used in the output generation counter to synchronise output clock flanks to the clock flanks of the reference.

*Output generation counter:* The output generation counter produces the output clock using the measured 'length' of the reference period

and the 'sync' signal. It counts up by two times the multiplication factor each oscillator period. When the output generation counter has reached the measured 'length' of the reference period, the output clock changes polarity and $2 \times MF$ is subtracted from the counter. Each time a sync pulse arrives, the counter is reset and the next output is set high. If another clock output of different multiplication factor is needed, another output generation counter just has to be added.

*Clock uncertainty:* The internal clock rate decides the uncertainty of the generated clock. For instance, if a 10 MHz clock is generated using an internal clock of 500 MHz, the period time for the output clock is 100 ns with an uncertainty of 2 ns. Using the input 'sel' in the internal clock generator, it is possible to increase the clock uncertainty to decrease power consumption.

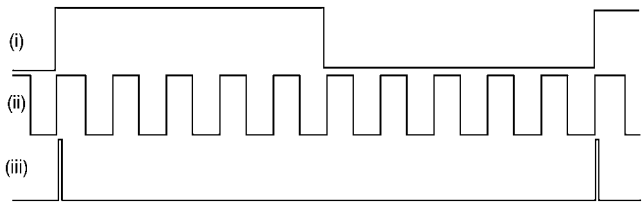


**Fig. 4** *Simulation plot from post-synthesis VHDL simulation*
(i) reference (ii) output clock (iii) 'sync'

*Synthesis and layout:* The clock generator was synthesised and verified using post-synthesis simulations using a multiplication factor of 10 for both a 0.35 μm 3.3 V and a 0.18 μm 1.8 V CMOS technology. For the 0.35 μm technology, the core area is 0.025 $mm^2$ with maximum internal clock rate 100 MHz. For the 0.18 μm technology the core area is 0.011 $mm^2$ with maximum internal clock rate 500 MHz. The synthesis is made with emphasis on small area and low power. It is possible to obtain a higher internal clock rate. The cost for increasing the internal clock rate is increased chip area and power consumption.

*Simulations:* The clock generator is verified using an analogue (0.35 μm only) and digital simulation environment. Fig. 4, which is from a digital post-synthesis (0.18 μm) simulation in VHDL, shows the input reference, the output clock and the 'sync' pulses for a multiplication factor of 10.

*Conclusions:* A robust and easy implemented clock generator is designed in fully synthesisable VHDL. The clock generator is extremely small (0.011 $mm^2$ in 0.18 μm CMOS) and it produces an accurate clock output after one period of the reference. Although it is not designed as a traditional PLL, its clock output is in phase with the reference. The synthesisable VHDL-code for the clock generator is found in [4].


*21 May 2003*
*Electronics Letters Online No: 20030910*
*DOI: 10.1049/el:20030910*

T. Olsson and P. Nilsson (*Department of Electroscience, Lund University, P.O. Box 118, SE-22100, Lund, Sweden*)

E-mail: thomas.olsson@es.lth.se

**References**

1 NILSSON, P., and TORKELSON, M.: 'A monolitic digital clock-generator for on-chip clocking of custom DSP's', *IEEE J. Solid-State Circuits*, 1996, **31**, (5)
2 OLSSON, T., *et al.*: 'A digitally controlled low-power clock multiplier for globally asynchronous locally synchronous designs'. Proc. of ISCAS'2000, May 2000
3 UMC 0.18: http://www.umc.com/english/process/d.asp
4 OLSSON, T.: 'A portable digital clock generator described in VHDL' (http://www.es.lth.se/home/ton/ckgen/ckgen.html)
5 WAKERLY, J.F.: 'Digital design principles and practices' (Prentice-Hall, Upper Saddle River, NJ, USA, 2000)