

AN ALL-DIGITAL CLOCK GENERATOR FIRM-CORE BASED ON DIFFERENTIAL FINE-TUNED DELAY FOR REUSABLE MICROPROCESSOR CORES

Mauro Olivieri and Alessandro Trifiletti
Univ. of Rome "La Sapienza, Rome, Italy

ABSTRACT:

Clock generator cores play an increasingly important role in the VLSI design of embedded microprocessors supporting specialized power management modes. We present a fully digital, standard-cell-based design of a specialized PLL architecture that can be recompiled on different cell libraries. On a 0.45 μm CMOS implementation, the circuit features a 16 ps jitter, 19.5-to-72 MHz frequency range with a 32KHz input, and less than 50 clock cycles wakeup time.

1. INTRODUCTION

Semi-custom VLSI designs are increasingly based on re-usable cores, in the form of soft-cores (i.e. synthesizable descriptions), firm-cores (i.e. gate level netlists), or hard-cores (i.e. predefined layout blocks) [12].

In VLSI embedded microprocessor cores supporting specialized power management modes, the clock generation sub-system is essential for guaranteeing a precise timing reference as well as fast stop/wakeup phases, to take advantage of idle periods and reduce power consumption. Specialized designs can use internally generated clock whose frequency and phase cannot be exactly predicted [6][11][2]. General-purpose precise clock generators usually multiply an external clock source by an integer factor. The most popular approaches in this field can be divided into phase-locked-loop (PLL) and delay-locked-loop (DLL) architecture schemes [4]. From the implementation point of view, all-digital designs must be distinguished from partially analog design, the former allowing a potentially higher precision at the expense of full-custom design and less easy portability.

DLL schemes are gaining increasing favor because of their improved phase noise, but a fully digital implementation can hardly be adapted to support fine tuning of the frequency multiplication factor, and therefore they are often devoted to exact frequency replication (synchronization)[3]. In fact, to implement precise frequency synthesis, partially analog delay control is implemented in the delay lines by means of specialized cells [7]. Existing fully digital PLL implementations are characterized by the presence of jitter due to the delay quantization error caused by the digital cells [5]. Lower jitter is introduced by means of special full-custom designed delay cells that allow a finer tuning of the delay, but prevent the possibility of semi-custom clock generator design [10]. The same approach has been presented for low jitter DLLs with fixed multiplication factor [3]. A generally important performance index is the efficient recovery from idle state (wakeup time) and efficient change of frequency multiplication factor. Different programmable clock frequencies can be useful in power-critical systems for adapting the power consumption to variable computational loads. This is more efficient than

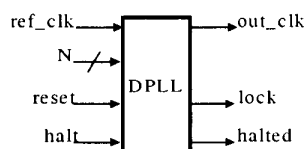


Fig.1 – Top level view of the clock generator core

repetitively entering idle states, because it reduces input data buffering and avoids frequent interrupt handling routines [9].

We present the design of an all-digital standard-cell-based firm-core realizing a PLL clock generator core with programmable multiplication factor. The scheme employs a novel PLL architecture based on standard logic gates, allowing a jitter time one order of magnitude lower than the minimal gate delay. The design can be considered technology independent as it can be recompiled on different cell libraries without affecting correct operation. Other features include wide frequency range and fast wake-up time from idle mode. We illustrate the architecture model, its functional simulation, the logic design and the detailed Spice-simulated performance of a 0.45 μm CMOS implementation.

2. FUNCTIONAL BLOCK DESCRIPTION

Fig. 1 illustrates the top-level view of the clock generator core. The IO lines have been chosen in order to simplify the use of the core and to increase its re-usability. The main lines are *ref_clock*, *out_clock* and *N*, respectively input (reference clock), output clock and selected harmonic: *reset* (In) is used at power-on reset and the couple *halt* (In), *halted* (Out) is used to drive the clock generator in idle state without any internal clock activity. The *lock* (Out) line flags the end of a frequency acquisition phase and it can be useful to enable timing critical microprocessor activities (i.e. synchronous peripherals, UART, RAM accesses) when the output clock has reached its stable state¹.

3. ARCHITECTURE IMPLEMENTATION

For a reliable reference clock signal, the Digital PLL (DPLL) must achieve fast lock-in transient, zero-jitter performance and unconditional stability. The error signal which has to be reduced down to zero is mainly due to a possible change in the selected output frequency and, for a given frequency, is due to the jitter inherent to the DCO. To make the lock-in time as fast as possible we estimate the difference between the output clock frequency

¹ For the proposed implementation the stable state condition is $\langle \text{output clock frequency} \rangle = N \cdot \langle \text{reference clock frequency} \rangle$

and the desired frequency (reference clock frequency multiplied by the selected harmonic) by measuring the number of cycles of out_clock in a cycle of ref_clock. Such approach allows to correct the instantaneous output frequency by means of a LUT (Look-Up Table) overcoming the limit of the linear feedback approach which could lead to a cumbersome design optimisation to get fast transient behaviour and reliable stability margins. Once the frequency acquisition has been achieved, the DCO is then controlled by a BangBang phase-detector, which provides small changes to the output frequency around the reference one, and takes advantage of a fine-grain delay-line in the DCO. In Fig. 2 a general block scheme of the clock generator is shown, the *phase detectors* block implements both frequency and phase discriminators and provides, on separate lines, the output errors. The *offset control & LUT* block plays the role of control unit and gain factor. The offset signal coming from the LUT is low-pass filtered during frequency acquisition phase by the *filter* block, whereas during tracking phase the output frequency is updated with a fixed fine-grain step. This approach has been found to be effective both for the speed of the PLL scheme implemented and for the jitter suppression capability.

4. LOGIC DESIGN

The gate level synthesis of the proposed architecture relies on very basic logic components available in any standard cell library. The user level re-usable firm-core consists a gate-level netlist, to be compiled on the target cell library.

The top hierarchical view of the logic level implementation of the clock generator is depicted in Fig. 3. The Digital Controlled Oscillator (DCO) block is driven by a new control word every two

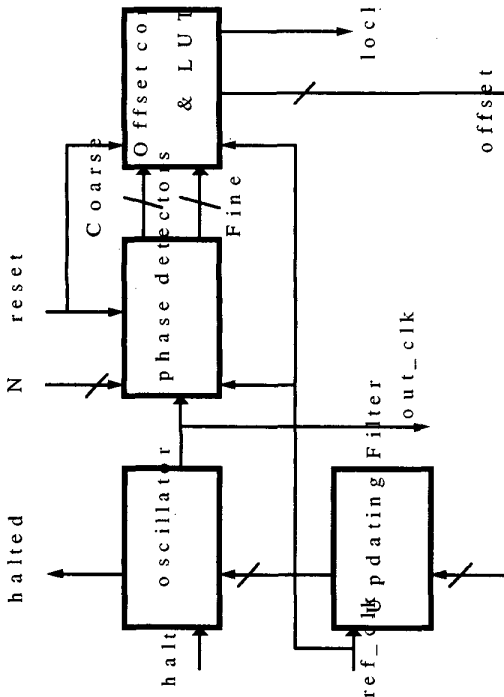


Fig.2 – Functional block view

periods of *ref_clk*. A simple accumulating adder is in charge of updating the control word value with a numerical offset defined by the multiplexing network in the bottom/left part of the illustration. The adder corresponds to the *filter* block in Fig. 2 (actually a first order digital filter). The *Range Detector* block, through the Enable BangBang (EBB) signal, selects whether the offset should be decided by the Phase Detector BangBang (PDBB) or by the Counter Reminder Register (CRR) value. Negative offset values are encoded in 2's complement. The BangBang operation is enabled when the frequency error detected by the CRR enters a sufficiently narrow window. In the proposed implementation, if the CRR's ten most significant bits are all zeroes or all ones, then the CRR absolute value is between -8 and +7, and the BangBang operation is enabled. In the other case, to allow a faster adjustment of the DCO frequency, the CRR value – multiplied by the LUT gain factor – sums to the control word. The LUT gain factor could be run-time sized to an optimal value, dependent of *N* and of the current number of active delay elements in the DCO, to have a single cycle coarse frequency acquisition. In order to allow an efficient hardware implementation maintaining the loop stability, such optimal gain factor is approximated with a power of two, updated every *ref_clock* cycle.

Gate level implementations of the macro-blocks of Fig. 3 are detailed in Fig. 4. In particular, the DCO is composed of a ring oscillator structure with two variable delay lines. The fine grain delay line exploits the difference between two- and three-input NAND cell delay to allow a finer grain the clock period synthesis. The order of magnitude of such delay difference is about 10 times shorter than the delay of a basic gate (NAND cell with Standard Load in the target library), so that the proposed clock generator core can reach a time discretization of 1/10 the delay of a NAND gate in the target technology. The absolute value of the delay difference is not essential for the proper operation of the feedback control loop, so that chip-level slow variations of the temperature or process variations do not affect correct operation.

The user can predict a first-order performance estimation of the clock generator core in the target technology in terms of frequency range, as a function of the basic gate delay in the target technology. The minimum and maximum nominal frequency values are the following:

$$T_{\min} = NF \cdot T_{pdF} + NG_{\min} \cdot T_{pdG} + T_{mux}$$

$$T_{\max} = NF \cdot (T_{pdF} + \delta_{pdF}) + NG_{\max} \cdot T_{pdG} + T_{mux}$$

where

- T_{pdF} is the delay of a minimal NAND gate driving the standard load capacitance in the given technology,
- T_{pdG} is the delay of a two cascaded minimal inverters driving a standard load capacitance in the given technology,
- δ_{pdF} is the delay difference between a 3-input NAND and a 2-input NAND driving a standard load,
- NG_{\min} and NG_{\max} are the minimum and maximum number of active inverters pairs in the coarse grain delay line (Fig. 4),
- NF is the number of NAND stages in the fine grain delay line (Fig. 4).

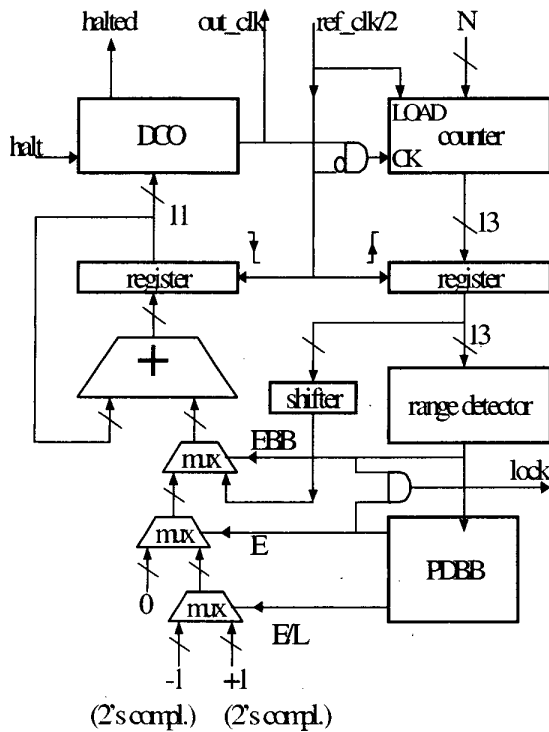


Fig. 3 – Logic design of the clock generator core

Accurate layout level performance estimation have to be carried out from back-annotated circuit simulation after the place&route phase, as for any semi-custom design.

5. PERFORMANCE EVALUATION

We performed functional, gate-level and circuit level simulation of the proposed clock generator scheme in order to verify its correct operation and evaluate its performance effectiveness. Circuit level simulation refers to a specific implementation on a target 0.45 μm CMOS process.

In Fig. 5 a Matlab simulation of the phase-frequency acquisition process is shown: the overall acquisition time is below 50 *out_clock* cycles, which corresponds to the expected wake-up time of the lock generator. Such result is considerable with respect to existing embedded microprocessor implementations where the wakeup time can take up to 1024 cycles [8]. In Fig. 6 the acquisition of a new harmonic is shown, corresponding to a change in the programmable frequency multiplication factor. Further simulations have been carried out to check the effectiveness of jitter suppression under DCO frequency drift: the scheme has proven to be robust to white noise and to sinusoidal interfering signals.

Circuit level simulation shows that the PLL control circuitry can sustain the maximum frequency of the oscillator. Table 1 reports a summary of the performance data referring to the circuit level Spice simulation of the specific implementation of the core.

Table 1 – Performance data for a 0.45 μm standard cell implementation of the clock generator core

std cell count	379
equiv gate count	1474
ring osc equiv gates / whole core equiv gates	0.81
ring osc power / whole core power	0.86
whole core power (max freq, $V_{dd} = 3.3\text{V}$)	1.6mW
max frequency	72 MHz
min frequency	19 MHz

6. CONCLUSIONS

We presented a novel programmable clock generator architecture and its implementation as a gate-level firm-core which can be mapped on different standard cell libraries. With respect to existing designs, the following features are supported:

- very precise frequency and phase synthesis, thanks to a specialized fine grain delay line
- very low jitter, limited to much less than the basic gate delay in the target digital technology
- fast wake-up response and fast frequency change
- fully digital standard cell based implementation
- technology independency, meaning that the core can be remapped on different cell libraries without affecting the correct operation.

7. REFERENCES

- [1] Burd, T.W. and Brodersen, R., Processor design for portable systems, *Journal of VLSI Sig. Proc.*, 1996, avail. at <http://infopad.eecs.berkeley.edu/infopad-ftp/papers/>
- [2] Chen, D.L. Designing on-chip clock generators, *IEEE Circuits Devices Magazine*, vol. 8 pp. 32-36 July 1992
- [3] Kim B.S. and Kim L.S., A low power 100 MHz All Digital Delay Locked Loop, *Proceedings of 1997 IEEE International Symposium on Circuits and Systems. ISCAS '97. IEEE, New York, NY, USA; 1997*, pp. p.1820-3 vol.3.
- [4] Kim, B., Weigandt, T.C, Gray, P.R., PLL/DLL System noise analysis for low jitter clock synthesizer design, *ISCAS 94, 1994, IEEE*.
- [5] MSP430 Mixed Signal Microcontrollers datasheet, Texas Instruments, Dallas, Texas, 1999. <http://www.ti.com>
- [6] Nilsson, P. and Torkelson, M., A monolithic digital clock generator for on-chip clocking of custom DSP's, *IEEE Journal of Solid State Circuits*, 31(5), May 1996
- [7] Park, J., Koo, Y., Kim, W., A semi-digital Delay locked loop for clock skew minimization, *Proc. of the International Conference on VLSI design, 1998*.
- [8] PIC16C57 Programmable Mixed Signal Microcontroller datasheet, Microchip Technology Inc., 1996. <http://www.microchip.com>
- [9] M. Olivieri, A. Trifiletti and A. De Gloria, A Low-Power Microcontroller with On-Chip Self-Tuning Digital Clock-Generator for Variable-Load Applications, *ICCD99, Austin, TX, Oct. 1999*.

- [10] Rahkonen, T., Eksyma, H., A 3-V programmable clock generator with a built-in phase interpolator, 1998 Midwest Symposium on Circuits and Systems. IEEE, Los Alamitos, CA, USA; 1999; pp. p.488-91.
- [11] Santoro and Horowitz, SPIM, a Pipelined 64x64 bit

iterative multiplier, Jour. of Solid state circuits, 24(2), pp. 487-493, Feb. 1989

- [12] Zorian, Y. and Gupta, R. K., Design and test of core-based systems on chips IEEE Des. & Test of Computers, Dec.97, pp. 14-25.

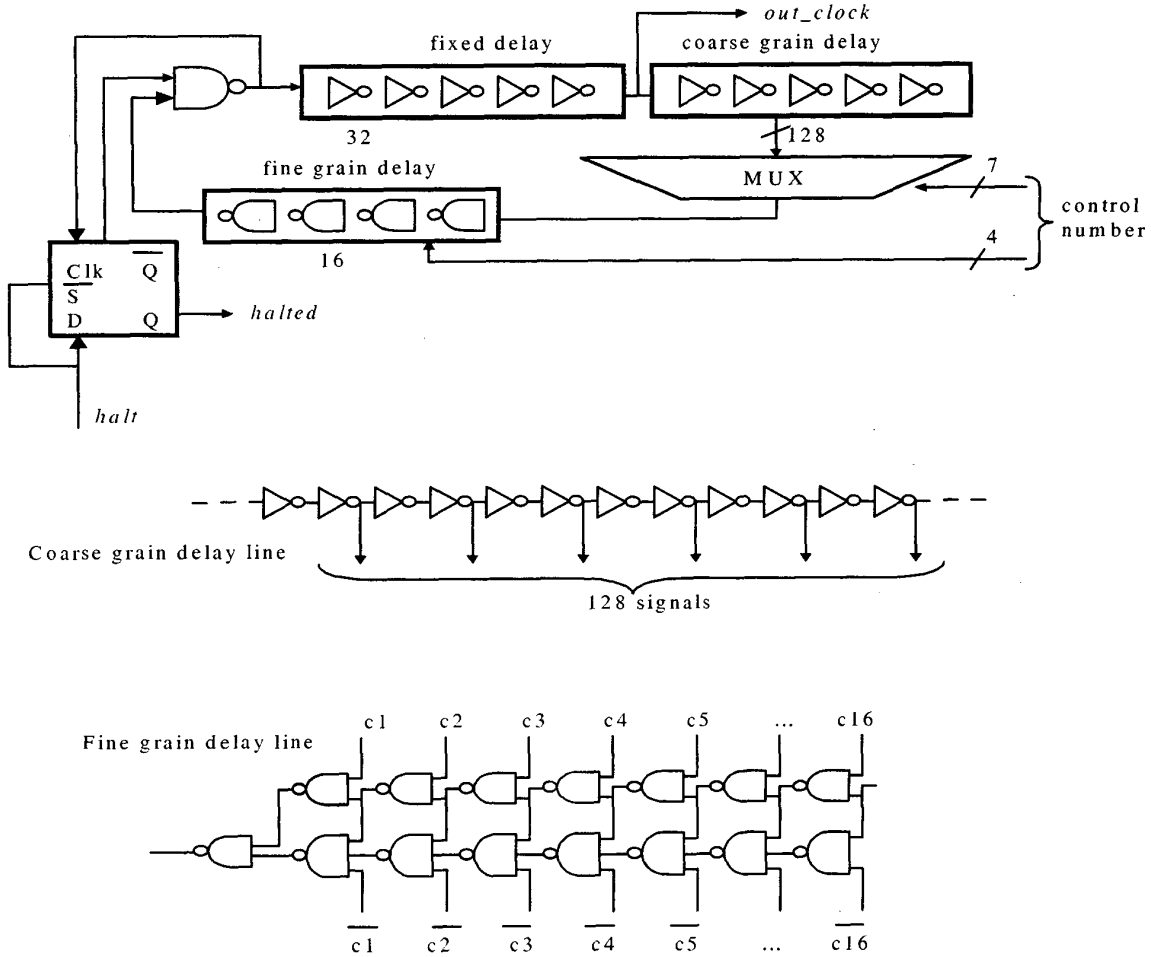


Fig. 4 – Gate level design of the DCO

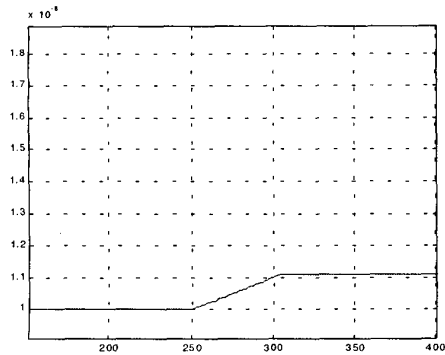


Fig. 6 – Clock period change (seconds vs cycles)

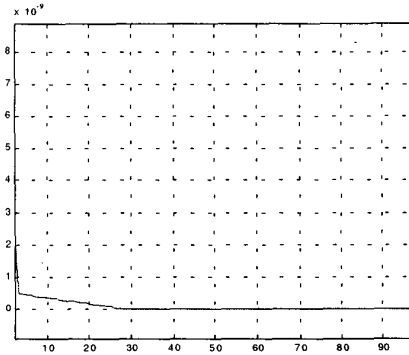


Fig. 5 – Jitter suppression after wakeup (secs vs cycles)