Distributed Active Clock Network

Vadim Gutnik
Massachusetts Institute of Technology
50 Vassar Street, 38-107
Cambridge, MA 02139
Tel. +1 617 258 8075
gutnik@mit.edu

Anantha Chandrakasan
Massachusetts Institute of Technology
50 Vassar Street, 38-107
Cambridge, MA 02139
Tel. +1 617 258 7619
anantha@mtl.mit.edu

Abstract

A distributed clock network can alleviate problems of skew and jitter in GHz-clock-speed microprocessors. A proof-of-concept chip fabricated in a $0.6\mu m$ process achieves 354MHz operation of 4 phase-locked oscillators. Supply noise insensitive, current controlled oscillators; nonlinear, modelock-resistant phase detectors; and the loop filters needed for multiple interconnected PLLs are integrated on chip, with a total area per PLL of less than $0.02mm^2$.

1. Introduction

The majority of modern microprocessors use a balanced tree to distribute the clock [1, 2]. Tree distribution has worked for many generations of processors, but it suffers from a number of drawbacks that make it less attractive at the gigahertz clock speeds required in the near future. We propose a distributed clock network as an alternative to clock trees.

The primary metrics for clock distribution are skew, jitter, and power dissipation. To evaluate the performance in these terms, we consider a representative section of a clock tree in Fig. 1. Skew is caused by mismatch in the delay along buffers A_i with respect to buffers B_j. (If, as shown, the tree drives a common global clock line or grid, RC delays in the clock line still allow different parts of the chip to see different clocks.) Since delay through clock trees has grown to multiple clock cycles, even a small variation in gate delay (caused primarily by process variations, and temperature and power supply gradients across the chip), or wire delay, (caused by variations in dielectric thickness or lithography) can add up to a large skew. As trees grow deeper to minimize systematic skew at high clock speeds, skew caused by random process variations begins to dominate, and that skew can only be compensated actively.

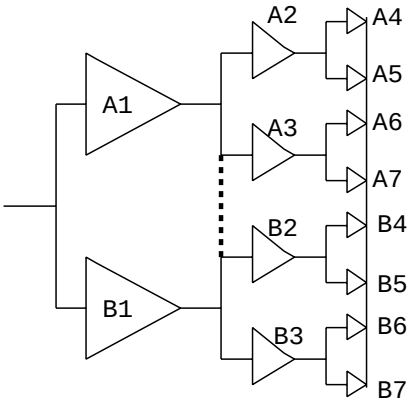

Figure 1. Simplest model of clock tree driving a global clock net.

The majority of jitter is introduced by buffers and interline coupling to the clock wires; a relatively small amount comes from noise in the source oscillator [3, 4]. In many cases outputs of the clock buffers are strapped together at intermediate stages as well, as shown by the dotted line in Fig. 1. To the extent that this helps control skew and jitter, the clock edges become slower and there is an increase in short circuit power dissipation.

2. Distributed Clocking

A distributed clock network aims to address these weaknesses by generating the clock signal with phase locked loops at multiple points (“nodes”) across a chip, and distributing each only to a small section of the chip (“tile”); Fig. 2 shows a representative section of a chip.

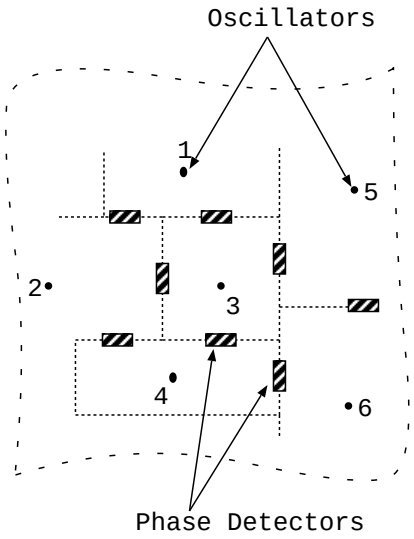

Figure 2. Local Synchronization

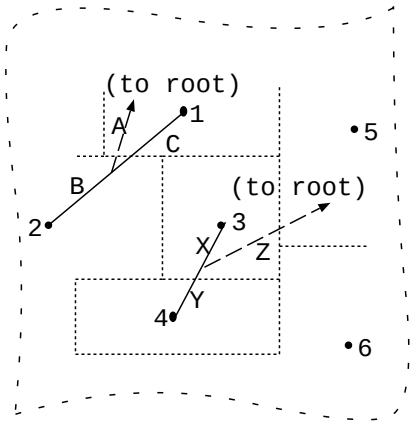

Figure 3. Tree Distribution

Phase comparators at the boundaries of each tile produce an error signal that is summed by an amplifier in center of each tile and used to adjust the frequency of the node oscillator. Compare this to a clock tree, shown in Fig. 3, where adjacent tiles (*e.g.*, tiles 1 and 3) are driven by different branches of the tree which have uncorrelated noise and offsets.

And thus, with locally generated clocks, there are no chip-length clock lines to couple in jitter; skew is introduced only by asymmetries in phase detectors instead of mismatches in physically separated buffers; and perhaps most importantly, the clock is regenerated at each node, so jitter does not accumulate. Unlike earlier work on having multiple clock domains which suggested the use of multiple independent clocks [5], this approach produces a single, fully synchronized clock. Furthermore, the arbitrary network of tiles, each with its own PLL, is far more general than the recently introduced single-DLL active skew management approach [6].

However, because there are many elements, the power and size constraints on each element of a distributed clock generation architecture are even more stringent than the constraints on a single, global PLL. Furthermore, there must be a way to ensure that the multiple nodes get and stay synchronized. The oscillator, phase detector, and loop filter of a working demonstration chip fabricated in a 0.6 μm process are considered in turn below.

2.1. Oscillator

The primary metric in the design of oscillators for clock generation is jitter, and the majority of that is due to power supply noise [3, 4]. For this reason, the oscillator used in this chip is optimized to reject noise from the power supplies. The schematic of the current controlled oscillator, or "CCO" used at the nodes, is shown in Fig. 4. (It is a differential relaxation oscillator. Integrated LC oscillators tend to be very sensitive to substrate noise, and are typically on the order of 100x100 μm^2 in size; the differential ring oscillators simulated did not have as much power supply rejection.)

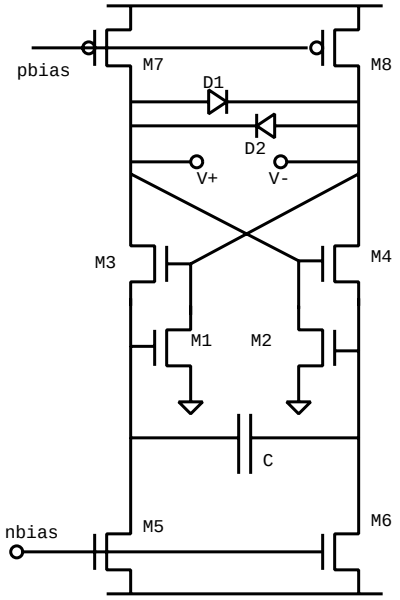

Figure 4. Relaxation Oscillator Schematic

Transistors M_3, M_4, M_5, and M_6, along with capacitor C make up a conventional source-coupled multivibrator, with M_7 and M_8 as active loads and **nbias** controlling oscillation frequency through $gm_{3,4}$. The drawback is that that circuit has a feedthrough of -6db to nodes V+ and V- from V_{DD}, and almost 0dB to the capacitor from ground via C_{bs} of $M_{3,4}$, so there is no way to reject supply noise. In the proposed oscillator, M_1 and M_2 provide shunt-shunt feedback around M_3 and M_4 respectively, lowering the output impedance at V+ and V-. D_1 and D_2 limit the amplitude of oscillation to avoid saturation of M_3 and M_4. Frequency can be adjusted by adding common-mode current into nodes V+ and V-.

2.2. Phase Detector

Synchronization of multiple oscillators is complicated by the fact that phase is periodic, so the average phase of a set of oscillators is multiply defined. For example, consider four oscillators with phase comparators between neighbors, shown in Fig. 5. Although they are not in phase, the net phase error at each is 0, so they are in equilibrium. Worse, the equilibrium is stable if the phase comparators output a signal proportional to the phase error.

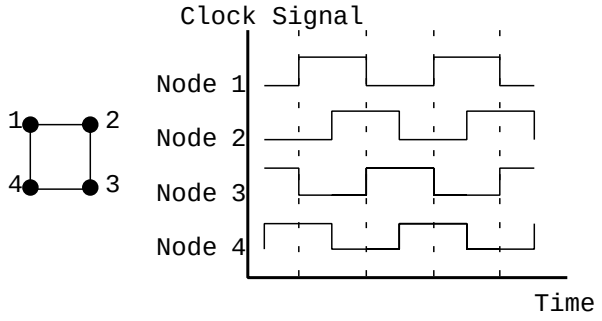

Figure 5. Modelock Condition

This phenomenon, dubbed "modelock," was described by Pratt and Nguyen, who also suggested that modelock can be avoided in regular arrays by using nonlinear phase detectors whose response decreases monotonically beyond a phase difference of $\pi/2$ [7]. The phase detector Pratt and Nguyen suggest (a flip-flop delay and an **XOR** gate) is not well-suited for integrated PLLs, however. First, it has relatively low gain, so mismatch can lead to large input-referred phase offsets. Second, it generates a lot of high-frequency noise that must be attenuated in the loop filter.

The phase detector proposed here, shown in Fig. 6, has the right nonlinearity, higher gain at small $\Delta\phi$ and has much less high-frequency content than an **XOR**. (Only half of the circuit is drawn. The other half is the symmetrical counterpart, with **clock1** and **clock2** switched.) M_1, M_2, and M_3 comprise an arbiter. The voltage at

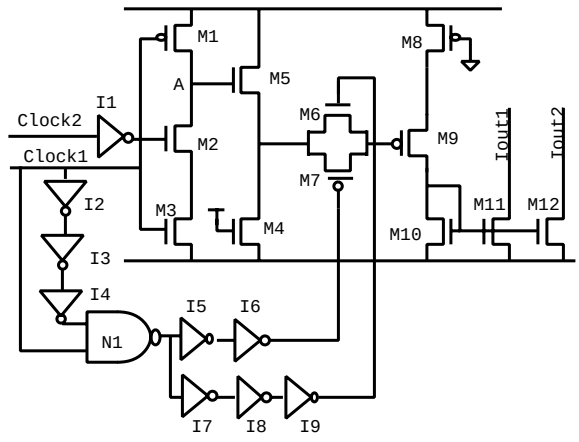

Figure 6. Phase Detector Schematic

node **A** is buffered, sampled, and converted to a current, so that multiple inputs can be summed at each oscillator node. Synchronous sampling of the arbiter output by M_6 and M_7 demodulates it, removing high frequency content. The phase of the sampling instant also affects the transfer function, shown in Fig. 7. The shape is set by $\theta_c = \theta_f - \theta_s$, where θ_f is the phase of the falling edge of **clock2** relative to its rising edge (so for a 50% duty cycle, $\theta_f = \pi$), and θ_s is the phase of the sampling instant. If the output of the arbiter is sampled immediately after the rising edge of **clock1**, and **clock2** has 50% duty cycle, the transfer function is exactly that of an arbiter. By delaying the sampling instant with $I_5 \ldots I_9$, θ_c can be lowered to less than $\pi/2$, which is the constraint to avoid modelock. Adding the output from the unshown half of the circuit gives the other half of the phase response. The full circuit fits in 80μm x 40μm.

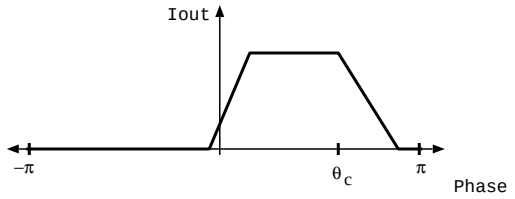

Figure 7. Sampled Phase Detector Transfer Function

2.3. Loop Filter

One loop filter is associated with each CCO. Conventional loop filters use a charge pump with an RC pole-zero pair, and often put the large capacitor and resistor off chip. To avoid inconveniently large resistor and capacitors, we propose a feed-forward compensation method. The loop filter of Fig. 8 consists of two differential amplifiers. (Note that because the frequency control to the oscillator consists of two currents, both amplifiers have twin outputs.) M_3, M_4, M_5, and M_6 make up amplifier A_1, biased by M_9, while M_1, M_2, M_7, M_8, M_{11} and M_{12} make up A_2, biased by M_{10}. The differential output currents from the phase comparators at the edges of each tile are summed at nodes I_{in}+ and I_{in}-, and drive both amplifiers. A_1 is a single stage differential pair, so it has relatively low gain but a bandwidth limited by $g_{m3,4}/C_{gs3,4}$, since nodes I_{out1} and I_{out2} drive a low impedance. A_2 has two stages, much like a prototypical op-amp. The first is biased at very low current to give high gain at DC and allow the use of a relatively small compensation capacitor, and the second provides the needed gain and isolates the high impedance pole from the output. In this amplifier, the DC gain was simulated at 31dB with a 16kHz pole, a compensating zero at 7.6MHz, and a high frequency pole well above the PLL target frequency. The use of feed-forward compensation allowed the use of very small capacitors; the loop filter, including the poly-poly capacitor, and the CCO with its output buffers together take up 88μm x 88μm.

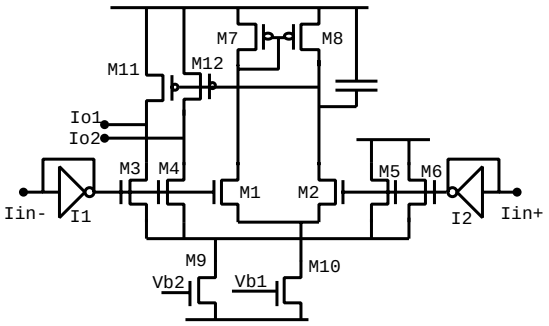

Figure 8. Loop Filter Schematic

3. Results

A chip was fabricated with a 2x2 array of nodes and phase comparators between nearest neighbors. In a larger network there would be on average two phase comparators per node, so the total overhead is approximately $0.014mm^2$ per node. Another phase comparator was placed between one of the nodes and the input reference frequency to lock the network to an external reference. At $V_{DD} = 3V$, the oscillators were observed to achieve phaselock for a range of input frequencies from 238.3 MHz to 354 MHz. This speed is comparable to commercial microprocessors at similar technologies – clock speeds at the $0.5\mu m$ generation ranged up to 300 MHz [8]. Off-chip measurement of jitter (through unavoidably noisy output buffers) was 22 picoseconds rms; the input signal was measured to jitter 12ps rms. The entire chip, including all biasing, draws 15mA. A chip plot is shown in Fig. 9.

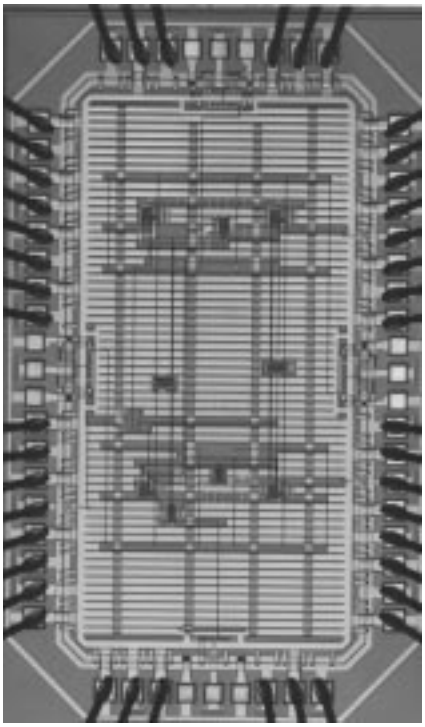

Figure 9. Distributed Clock Chip

4. Conclusions

Design and measurements on this chip confirm that generating and synchronizing multiple clocks on chip is feasible. Neither the power nor the area overhead of multiple oscillators and PLLs is substantial compared to the cost of distributing the clock by conventional means — typical clock distribution networks for microprocessors consume on the order of 10W. Most importantly, a distributed clock network can take advantage of improved devices by shrinking the size of the cells, lowering the overall skew and jitter, so performance will scale with the speed of devices, rather than with the much slower improvement of on-chip interconnect speed.

5. Acknowledgments

This work was funded by the MARCO Interconnect Focus Center and Intel.

6. References

[1] Daniel W. Bailey and Bradley J. Benschneider, "Clocking design and analysis for a 600 MHz Alpha microprocessor," *Journal of Solid State Circuits*, vol. 33, no. 11, pp. 1627–1633, November 1998.

[2] Shantanu Ganguly, Daksh Lehther, and Satyamurthy Pullela, "Clock distribution methodology for the PowerPC microprocessors," *Journal of VLSI Signal Processing*, vol. 16, no. 2/3, pp. 181–189, June/July 1997.

[3] Ian A. Young, Monte F. Mar, and Bharat Bhushan, "A 0.35μm CMOS 3-880MHz PLL N/2 clock multiplier and distribution netowrk with low jitter for microprocessors," in *ISSCC Digest of Technical Papers*, February 1997, pp. 330–331.

[4] Raghunand Bhagwan and Alan Rogers, "A 1GHz dual-loop microprocessor PLL with instant frequency shifting," in *ISSCC Digest of Technical Papers*, February 1997, pp. 336–337.

[5] Francois Ançeau, "A synchronous approach for clocking VLSI systems," *Journal of Solid State Circuits*, vol. SC-17, no. 1, pp. 51–56, February 1982.

[6] G. Geannopoulos and X. Dai, "An adaptive digital deskewing circuit for clock distribution networks," in *ISSCC Digest of Technical Papers*, February 1998, pp. 400–401.

[7] Gill A. Pratt and John Nguyen, "Distributed synchronous clocking," *IEEE Transactions on Parallel and Distributed Systems*, February 1995.

[8] Bradley J. Benschneider *et al.*, "A 300-MHz 64-b quad-issue CMOS RISC microprocessor," *Journal of Solid State Circuits*, vol. 30, no. 11, pp. 1203–1214, November 1992.