[3] A. Castro, M. Nicolaidis, P. Lestrat, and B. Courtois, "Built-in self test for multi-port RAMs," presented at the ICCAD, Santa Clara, CA, Nov. 1991.

[4] A. J. van de Goor, *Testing Semiconductor Memories, Theory and Practice*. Chichester, U.K.: Wiley, 1991.

[5] A. J. van de Goor, "Using march tests to test SRAMs," *IEEE Des. Test Comput.*, vol. 10, no. 1, pp. 8–14, 1993.

[6] A. J. van de Goor and C. A. Verruijt, "An overview of deterministic functional RAM chip testing," *ACM Comput. Surveys*, vol. 22, no. 1, pp. 5–33, Mar. 1990.

[7] M. Marinescu, "Simple and efficient algorithms for functional RAM testing," in *Proc. IEEE Int. Test Conf.*, 1982, pp. 236–239.

[8] V. N. Yarmolik, S. Hellebrand, and H.-J. Wunderlich, "Symmetric transparent BIST for RAMs," presented at the DATE, Munich, Germany, Mar. 1999.

[9] V. N. Yarmolik, I. V. Bykov, S. Hellebrand, and H.-J. Wunderlich, "Transparent word-oriented memory BIST based on symmetric march algorithms," in *Proc. Eur. Dependable Comput. Conf.*, 1999, pp. 339–350.

[10] I. Voyiatzis, "Test vector embedding into accumulator-generated sequences: A linear-time solution," *IEEE Trans. Comput.*, vol. 54, no. 4, pp. 476–484, Apr. 2005.

[11] A. Stroele, "BIST patter generators using addition and subtraction operations," *J. Electron. Test.: Theory Appl.*, vol. 11, pp. 69–80, 1997.

[12] R. Dorsch and H.-J. Wunderlich, "Accumulator-based deterministic BIST," in *Proc. Int. Test Conf.*, 1998, pp. 412–421.

[13] I. Voyiatzis, A. Paschalis, D. Gizopoulos, N. Kranitis, and C. Halatsis, "A concurrent built-in self-test architecture based on a self-testing RAM," *IEEE Trans. Reliab.*, vol. 54, no. 1, pp. 69–78, Mar. 2005.

[14] W. L. Wang, K. J. Lee, and J. F. Wang, "An on-chip march pattern generator for testing embedded memory cores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 5, pp. 730–735, Oct. 2001.

[15] C.-T. Huang, J.-R. Huang, C.-F. Wu, C.-W. Wu, and T.-Y. Chang, "A programmable BIST core for embedded DRAM," *IEEE Des. Test Comput.*, vol. 16, no. 1, pp. 59–70, Jan./Mar. 1999.

[16] J. -F. Li, R.-S. Tzeng, and C.-W. Wu, "Diagnostic data compression techniques for embedded memories with built-in self-test," *J. Electron. Test.: Theory Appl.*, vol. 18, pp. 515–527, 2002.

[17] S. Hamdioui and J. E. Q. D. Reyes, "New data-background sequences and their industrial evaluation for word-oriented random-access memories," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 6, pp. 892–904, Jun. 2005.

[18] S. Hamdioui, Z. Al-Ars, and A. J. van de Goor, "Opens and delay faults in CMOS RAM address decoders," *IEEE Trans. Comput.*, vol. 55, no. 12, pp. 1630–1639, Dec. 2006.

[19] W.-L. Wang, K.-J. Lee, and J.-F. Wang, "An on-chip march pattern generator for testing embedded memory cores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 5, pp. 730–735, Oct. 2001.

[20] D.-C. Huang and W.-B. Jone, "A parallel built-in self-diagnostic method for embedded memory arrays," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 4, pp. 449–465, Apr. 2002.

[21] B. H. Fang and N. Nicolici, "Power-constrained embedded memory BIST architecture," in *Proc. 18th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst. (DFT)*, 2003, pp. 451–458.

Injection-Locked Clocking: A Low-Power Clock Distribution Scheme for High-Performance Microprocessors

Lin Zhang, Aaron Carpenter, Berkehan Ciftcioglu, Alok Garg, Michael Huang, and Hui Wu

Abstract—We propose injection-locked clocking (ILC) to combat deteriorating clock skew and jitter, and reduce power consumption in high-performance microprocessors. In the new clocking scheme, injection-locked oscillators are used as local clock receivers. Compared to conventional clocking with buffered trees or grids, ILC can achieve better power efficiency, lower jitter, and much simpler skew compensation thanks to its built-in deskewing capability. Unlike other alternatives, ILC is fully compatible with conventional clock distribution networks. In this paper, a quantitative study based on circuit and microarchitectural-level simulations is performed. Alpha21264 is used as the baseline processor, and is scaled to 0.13 μm and 3 GHz. Simulations show 20- and 23-ps jitter reduction, 10.1% and 17% power savings in two ILC configurations. A test chip distributing 5-GHz clock is implemented in a standard 0.18-μm CMOS technology and achieved excellent jitter performance and a deskew range up to 80 ps.

I. INTRODUCTION

Clock distribution is a crucial aspect of modern multi-gigahertz microprocessor design. Conventional distribution schemes are more or less monolithic in that a single clock source is generated by an on-chip phase-locked loop (PLL) and then fed through hierarchies of clock buffers and interconnects to eventually drive the entire chip (see Fig. 1). This raises a number of challenges. First, the nonuniform load of the clock network and deteriorating process, voltage, and temperature (PVT) variations give rises to spatial timing uncertainties known as clock *skews*. To minimize the global clock skew, the global clock-distribution network has to be balanced by meticulous design of the interconnects and buffers [5]. This practice puts a very demanding constraint on the physical design of the chip. Another practice is to use a grid instead of a tree for clock distribution, as shown in the upper-left local clock region in Fig. 1. A grid has a lower resistance than a tree between two end nodes, and hence can reduce skew. However, a grid usually has much larger parasitic capacitance (due to larger metal area) than an equivalent tree, and therefore takes more power to drive. Passive and active deskew methods have also been employed to compensate skew *after* chip fabrication. Unfortunately, these approaches often increase the circuit complexity, chip area, and power consumption.

Second, given the substantial load of the clock, sending a high quality clock signal to every corner of the chip requires driving the clock distribution network "hard," usually in full swing of the power supply voltage. Not only does this mean high power expenditure, but it also requires a chain of clock buffers, which are subject to power supply noise and hence add delay uncertainty-jitter. Unlike skew, (short-term) jitter is very difficult to compensate due to its random nature and thus poses an even larger threat to microprocessor performance and power consumption. To reduce jitter, the interconnect
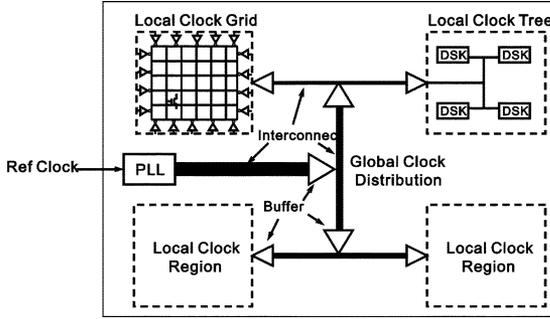
Fig. 1. Conventional clock distribution, showing an H-tree topology with interconnects and clock buffers [5].
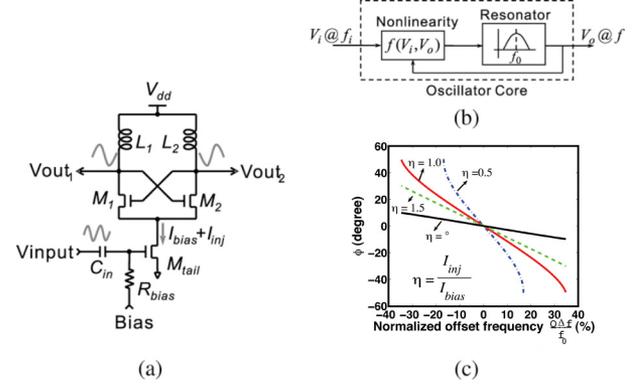

Fig. 2. ILO. (a) A divide-by-2 ILO based on a common differential LC oscillator. The input signal is injected into the oscillator core through the tail transistor M_{tail}. (b) A generic model of an ILO, similar to a PLL. (c) Phase tuning characteristics for an ILO in Fig. 2(b). $\eta \equiv I_{\mathrm{inj}}/I_{\mathrm{bias}}$ is the injection ratio, f_0 is the free-running frequency, $\Delta f \equiv f - f_0$ is the frequency shift, and Q is the LC resonator quality factor.
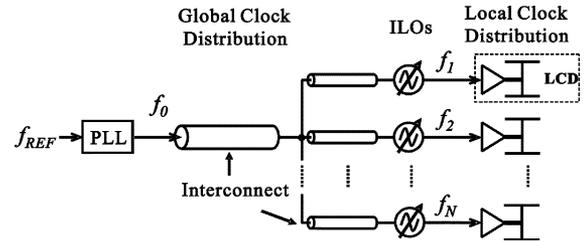

Fig. 3. ILC.

wires in the global clock distribution network need to be well shielded from other noise sources. This is usually done by sandwiching them between power/ground wires and layers. Unfortunately, shielding inevitably increases the parasitic capacitance of the clocking network and thus requiring more and larger clock buffers. Extra buffers not only increase power dissipation but also introduce their own jitter. Jitter and skew combined represent about 18% of cycle time currently [11]. For a fixed cycle time budget, any improvement in jitter and skew generates timing slack that can allow the logic circuit to operate more energy-efficiently.

As commercial microprocessors are rapidly becoming multi-core systems, monolithic clock distribution will be even less applicable. In the era of billion-transistor microprocessors, a single chip is really a complex system with communicating components and should be treated as such. In communication systems, synchronizing clocks is also a rudimentary and crucial task. In this paper, we apply the concept of *injection locking* and the latest innovation in circuit implementation to clock distribution in microprocessor.

The rest of this paper is organized as follows. Section II discusses proposed clocking scheme based on injection locking. Section III presents a case study. A test chip is demonstrated in Section IV. Finally, Section V concludes.

II. INJECTION-LOCKED CLOCKING (ILC)

A. Injection-Locked Oscillators (ILOs)

Injection locking [1], [10] is a special type of forced oscillation in nonlinear dynamic systems. Suppose a signal of frequency f_i is injected into an oscillator, which has a self-oscillation (free-running) frequency f_0. When f_i is quite different from f_0, "beats" of the two frequencies are observed. As f_i approaches f_0, the beat frequency ($|f_i - f_0|$) decreases. When f_i enters some neighborhood very close to f_0, the beats suddenly disappear, and the oscillator starts to oscillate at f_i. The frequency range in which injection locking happens is called the *locking range* Injection locking also occurs when f_i is close to the harmonic or subharmonic of f_0 ($n f_0$ or $(1/n) f_0$), which can be used for frequency division or multiplication.

An ILO can be considered a first-order PLL [see Fig. 2(a)], in which nonlinearity of the oscillator core functions as a phase detector. For example, in a typical divide-by-2 ILO [see Fig. 2(b)] [13], the oscillator core (consisting of M_1, M_2, and M_{tail}) also serves as a single-balanced mixer for phase detection. Because of the simple structure, ILOs consume much less power than a full-fledged PLL, and can operate at frequencies as high as tens of gigahertz [17]. The fact that the built-in "phase detectors" are mixer-based explains why ILOs can operate at the harmonic and subharmonic frequencies of the input signal.

Once locked to the input signal, the output of ILOs will maintain a determined phase relative to the input signal. The phase difference from the input signal to the output is determined by the injection signal amplitude, the frequency shift from its free-running frequency, and the frequency characteristics of the oscillator resonator. As shown in Fig. 2(c),

the phase shift φ is a monotonic function of the frequency shift $\Delta \omega$, and the function is quite linear within the locking range except when close to the edges. By tuning the ILO free-running frequency, we can change the phase of the output signal [21]. This can be utilized to compensate the timing skew (deskew) between different clock domains.

B. Clocking Using ILOs

Recently, we proposed a new clocking scheme as shown in Fig. 3. Similar to conventional clocking, the global clock is generated by an on-chip PLL and distributed by a global distribution network. The difference is that we use ILOs to regenerate local clocks, which are synchronized to the global clock through injection locking. Another major change is that most global clock buffers in conventional clocking are removed because the sensitivity of ILOs are much greater than digital buffers (see a detailed discussion in the following). Essentially, we use ILOs as local clock receivers, similar to the idea of clock recovery in communication systems. Note that this is different from resonant clocking [4], where all the oscillators are coupled together [20]. ILOs can also be constructed as frequency multipliers [8] or dividers [13], [18], and hence this scheme enables local clock domains to have higher ($n f_0$) or lower clock speed (f_0/m) than the global clock (f_0). Such a global-local clocking scheme with multiple-speed local clocks offers significant improvements over conventional single-speed clocking scheme in terms of power consumption, skew, and jitter.

C. Power Savings

ILC can lead to significant power savings in high-performance microprocessors. First, it is straightforward to use a low-speed global clock combined with high-speed local clocks in ILC. This reduces the power consumption in the global clock distribution network. In comparison, conventional global-local clocking would require multiple power-hungry PLLs for frequency multiplication. An ILO consumes much less power than a PLL because of their circuit simplicity [17].
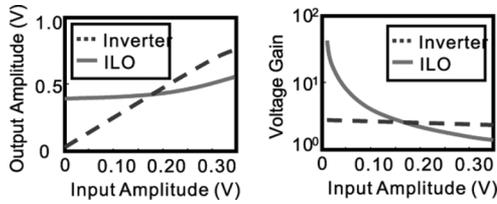
Fig. 4. (a) Voltage transfer and (b) gain comparison of an inverter and an ILO. As an oscillator, the output signal amplitude of an ILO stays constant at small input amplitude, which translates into high voltage gain.

The benefit will become more evident as future processors incorporate more and more cores.

Second, ILOs have higher sensitivity than clock buffers (inverters). An ILO effectively has very large voltage gain (much larger than an inverter) when the input signal amplitude is small (see Fig. 4). This is because that synchronization in an ILO is usually achieved in tens to hundreds clock cycles, and hence in each clock cycle only a small amount of injection locking force is needed. While an inverter needs to change its state twice in every clock cycle. Therefore, the signal amplitude of the global clock can be much smaller in ILC, which leads to smaller number of (or none) clock buffers and less power loss on the interconnects of the global clock distribution network.

More importantly, because ILC significantly lowers skew and jitter in the global clock (see the following), the timing margin originally allocated can be recovered for higher circuit speed, or trade for lower power supply voltage (V_{dd}). The latter saves power from not only clock distribution, but all the logic gates on the chip. In Section III, we demonstrate the power savings from all aspects using a quantitative case study.

D. Skew Reduction and Deskew Capability

Because the number of buffers is reduced in the new clocking scheme, skew due to clock buffers mismatch is reduced compared to conventional clocking. Further, ILC provides a built-in mechanism for deskew. From Section II-A, the phase difference between the input and output signals of an ILO can be tuned by adjusting its free-running frequency. This phase tuning capability enables ILOs to serve as built-in "deskew buffers," and conventional deskew architectures can be applied directly. For example, similar to active deskewing in conventional clocking, phase detectors can be placed between local clock domains to check skew, and then tune corresponding ILOs. Removing dedicated deskew buffers not only saves power, but also reduces their vulnerability to power supply noise. Note that ILC deskewing is different from the distributed PLL approach [6], [12], where phase detectors have to be added between all adjacent clock domains for frequency synchronization, and then possibly for deskew. In ILC, frequency synchronization is achieved by injection locking, and the phase detection is used for deskew only. In other words, ILC with deskew tuning is a dual-loop feedback system, and therefore provides both good tuning speed and small phase error (residue skew). Because of the excellent built-in deskew capability of ILOs, it can be expected that an injection-locked clock tree has much more freedom in its physical design (layout).

E. Jitter Reduction and Suppression

ILC can significantly reduce jitter in global clock distribution networks. First, reduced number of global clock buffers also means less pickup of power supply and substrate noise, and hence less jitter generation and accumulation. Second, because of the design freedom in layout, clock interconnect can be placed where there is minimal noise coupling from adjacent circuits and interconnects. In addition, similar to a PLL, an ILO can suppress both its internal noise through high-pass filtering and input noise through low-pass filtering, and hence can pos-

sibly lower jitter at its output. Using a differential structure, an ILO can be less insensitive to the common-mode power supply and substrate noise than an inverter by design. Therefore, ILC is likely to achieve better jitter performance than conventional clocking.

F. Potential Applications

With the numerous technical advantages, ILC can be used to improve high-performance microprocessors and the design process in many ways. First, ILC reduces jitter and skew compared to conventional clocking. This reduces cycle time, and therefore, allows a faster clock speed. As technology scaling improves transistor performance but does not reduce jitter and skew (which actually increase), the improvement in clock speed will be more pronounced over time. Although further increasing whole-chip clock speed finds limited practical appeal in today's setting, it may still be effective in certain specialized engine inside a general-purpose architecture.

Second, using ILC, clock distribution for a multi-core system is a natural extension from a single-core system. A conventional clocking scheme would require adding chip-level PLLs. PLLs are vulnerable to noise and hence usually placed at the very edge of a chip. In future multi-core systems, it represents a significant challenge to place PLLs and route high-speed clock signal to the destination cores. In contrast, in ILC, a single medium-speed global clock signal can be distributed throughout the chip and locally each core can multiply the frequency according to its need.

Third, even in a single-core architecture, different macroblocks can run at different frequencies. This is referred to as the multiple clock domain (MCD) approach [15]. Using ILC, we can locally multiply (or divide) the frequency of the single global clock. One significant advantage of using ILC to enable multiple clock domains is that the local clocks have a well-defined relationship as they are all synchronized to the global clock. As a result, cross-domain communication can still be handled by synchronous logic without relying on asynchronous circuits.

III. CASE STUDY

In this paper, we quantitatively demonstrate some benefits of ILC in a most straightforward setting, a single-core processor running at a single clock frequency. We focus on the energy benefits in this case study and compare processors that only differ in the *global* clock distribution. Due to the limited availability of detailed characterization of clocking network in the literature, our choice of the clocking network closely resembles that of the baseline processor. Note that this is far from the optimal ILC design for the given processor, but demonstrates significant benefits of ILC nonetheless.

Our baseline processor is Alpha 21264, which has the most details in public domain on its clock distribution network [2], [3]. In this processor, an on-chip PLL drives an X-tree, which in turn drives a two-level clocking grid containing a *global clock* grid and several *major clock* grids. The major clock grids cover about 50% of the chip area and drive local clock chains in those portions. The remaining part of the chip is directly clocked by the global clock grid. The densities of the two levels of grids are different. This configuration is illustrated in Fig. 5(a). The three planes X, G, and M represent the three layers of clock distribution networks: the X-tree, the global clock grid, and the major clock grids, respectively.

In the first ILC configuration [see Fig. 5(b)], we only replace the very top level of the clock network (X). We remove all buffers in the X-tree trunk and replace the final level of buffers (a total of four) with ILOs. The rest of the hierarchy remains unchanged. Note that in contrast to the Alpha implementation, we send low-swing signals on the X-tree, which reduces the energy consumption of the top level clock network. Furthermore, as discussed before, clock jitter and skew will also reduce. We convert this timing advantage into energy reduction by slightly reducing the supply voltage.
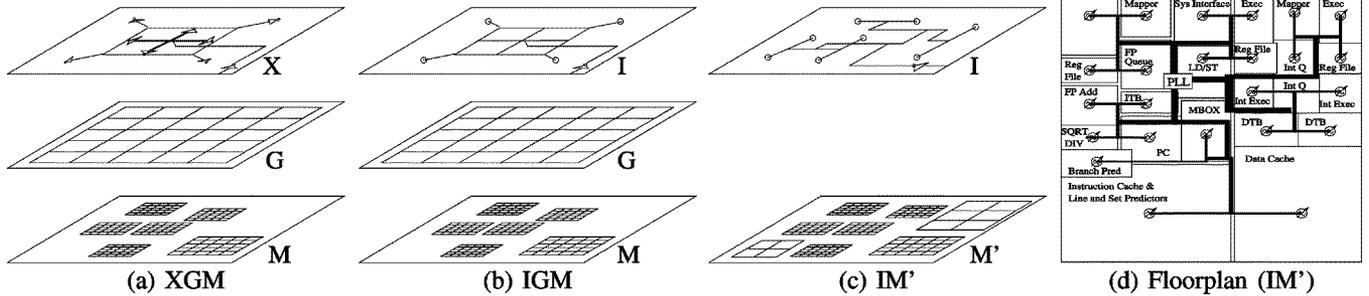
Fig. 5. Illustration of the three different configurations (a)–(c) of global clock distribution, and a possible floorplan (d) for the ILC-based global clock distribution in Alpha 21264. Each configuration is designated according to its clocking network: XGM, IGM, and IM'.
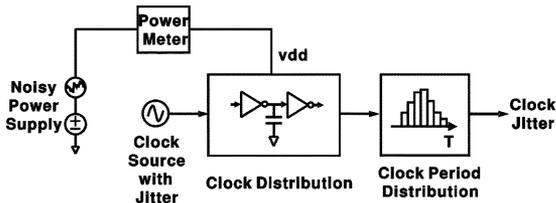

Fig. 6. Circuit-level jitter simulation setup.

While such a simple approach of using ILC as a *drop-in* replacement already reduces energy consumption, it is not fully exploiting the benefits of ILC. As discussed before, numerous ILOs can be distributed around the chip to clock logic macro-blocks. Thanks to the built-in deskew capability, we can avoid using power-hungry clock grids altogether. However, to faithfully model and compare different approaches, we need parameters (e.g., capacitance load of individual logic macroblocks) for circuit-level simulation which we could not find in the literature. As a compromise, in the second ILC configuration, we still use grids, but use only a single level of grids, which consist of all the major clock grids and the portion of the global grid that directly feeds logic circuit [see Fig. 5(c)]. With this configuration, the load of the clock network can be derived based on results reported in [2] and [3] and technology files. Finally, thanks to the deskew capability of ILOs, there is no need to use a balanced global clock tree. In Fig. 5(d), we show an example clock tree design. In this example, each macroblock in the floorplan is driven by an ILO which is at the leaf of the global clock tree.

To evaluate the benefits of injection-locked clocking, we perform both circuit- and architecture-level simulations on the baseline processors with each clock distribution configuration in Fig. 5. In order to reflect the state of the art, we scale the global clock speed from 600 MHz to 3 GHz, and correspondingly the process technology from 0.35 to 0.13 μm. The validity of scaling is verified using Pentium 4 Northwood 3.0 GHz processor as the reference.

A. Circuit and Architectural Simulation Setup

At the circuit level, we use a commercial circuit simulator, Advanced Design Systems (ADS), to evaluate power consumption and jitter performance of the clock distribution network with different configurations. The simulations are based on extracted models of the clock distribution networks, including the buffer size, interconnect capacitance, and local clock load capacitance. Then the distribution network model is applied in the circuit simulation with ILOs and clock buffers constructed using SPICE models of transistors.

Since jitter is largely introduced by power supply and substrate noise through clock buffers, a noise voltage source with a Gaussian distribution is inserted to the power supply node, as shown in Fig. 6. Transient simulation is used to calculate the voltage and current waveforms along

the clock distribution. Output clock waveform is analyzed statistically to get the distribution of the clock period. Jitter at the output is then calculated based on this distribution. Jitter is first measured in the baseline conventional clocking configuration, and the noise source amplitude is determined by matching measured jitter with reported value in [9] 35 ps. The same noise voltage source is then used in the subsequent jitter simulation for the ILC configurations, and the results are compared to the baseline configuration. We believe this approach is actually pessimistic considering the target jitter number (35 ps) is among the lowest in conventional clocking reported [11]. The source jitter from on-chip PLL is represented using a built-in ADS model of clock with jitter, and the clock jitter is chosen to be 5 ps, which is consistent with jitter of on-chip PLLs published.

For architectural simulations, we use a heavily modified version of SimpleScalar toolset extended with Wattch for the dynamic energy component, and HotSpot and BSIM3 models for temperature-dependent leakage modeling in 0.13 μm technology with a V_{dd} of 1.5 V. For brevity, the detailed parameters of the simulation are left in the technical report [20].

B. Analysis of Jitter and Skew

In the circuit simulation, the PLL source jitter is set to 5 ps, and the value of the added power supply noise source is chosen so that the output clock jitter for the baseline processor [see Fig. 5(a)] is 35 ps [9]. Apparently, there is 30-ps jitter added along the clock distribution, which comes from the power supply noise coupled through the buffers. For the clock speed of 3 GHz, the overall jitter in the baseline processor therefore corresponds to 10.5% of the clock cycle. In the case of ILC with IGM configuration [see Fig. 5(b)], under the same power supply noise and source jitter, the output clock jitter is lowered to 15 ps—a 57% reduction. This translates into recovering 6% of a clock cycle at 3 GHz, a significant performance improvement. As described in Section II-E, the jitter reduction can be attributed to the reduced number of clock buffers and good noise rejection of ILOs. When ILOs are used to directly drive the local clock grids without the global grid as in IM' configuration [see Fig. 5(c)], thanks to the further reduction in the buffer stages, jitter is lowered to 12 ps, or 66% lower than the baseline. These results clearly demonstrate that ILC can achieve better jitter performance than conventional clocking.

In the current study, it is assumed that built-in deskew capability of ILOs can reduce the skew to below 15 ps, resulting in 10 ps savings in timing margin compared to the baseline processor (without any deskew). This estimate is consistent with the results using existing deskew schemes [11], and hence quite reasonable. In fact, we believe ILC should lead to even lower skew as discussed in Section II-D, which can be supported by a test chip measurement shown in the following.

C. Chip-Wide Power Impact of ILC

The results of using different clocking structures are summarized in Fig. 7. In this comparison, all configurations achieve the same cycle
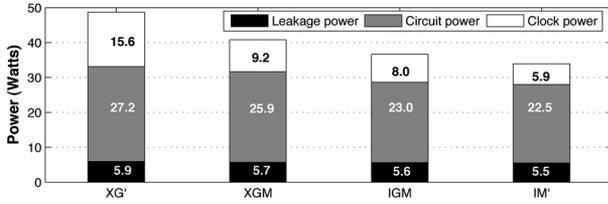
Fig. 7. Breakdown of processor power consumption with different clock distribution methods.

time. The density of the grids and the driving capabilities are determined using circuit simulation. We choose the design point where energy is minimized.

1) Baseline Processor: Simulations show that the power consumption of the baseline processor ranges from 30.4 to 50.4 W with an average of 40.7 W. The power can be divided into three categories: global clock distribution power, leakage, and the dynamic power of the rest of the circuit. The breakdown of the power is shown in Fig. 7. The global clock is unconditional and consumes 9.2 W or (23%).

2) ILC Configurations: Now, we analyze the power savings of ILC. For IGM [see Fig. 5(b)], power savings come from two factors. First, the power consumed in the top-level X-tree is reduced from 1.72 to 1.56 W because of the reduction of the total levels of buffers used and the lowered voltage swing on the X-tree. Second, as explained before, jitter and skew all improved when using ILC: a 20 ps reduction in jitter and 10 ps in skew are achieved. These savings increase the available cycle time for logic from 273 to 303 ps. This, in turn, allows a reduction in $V_{\rm dd}$ without affecting the clock speed. We use the following voltage-delay equation from [14] to calculate the new $V_{\rm dd}$, which is 1.415 V

$$ t = \frac{C}{k' \left(\frac{W}{L}\right)(V_{\rm dd} - V_t)} \left[\frac{2V_t}{V_{\rm dd} - V_t} + \ln\left(\frac{3V_{\rm dd} - 4V_t}{V_{\rm dd}}\right) \right] $$

the power reduction for the tested applications ranges from 3 to 5.2 W with an average of 4.1 W or 10.1%. The reduction is mainly due to the lowering of supply voltage.

The second ILC configuration, IM' [see Fig. 5(c)], further reduces clock distribution power by reducing the size of the grid. For IM', the global clock power is reduced to 5.9 W (from 9.2 W in XGM) and the combined jitter and skew reduction is 33 ps, which allows us to scale $V_{\rm dd}$ to 1.41 V. The overall effect is an average of 6.8 W (17%) total power reduction. Compared to IGM, IM' further reduces power by 2.7 W or 7%.

For reference, we also show the result of replacing the two levels of grids by a single grid in the conventional configuration. Note that this grid is different from the M' grid as it needs higher density and larger buffers to achieve the same overall cycle time target. We designate this grid G', and the configuration XG'. We use the same methodology to compute its jitter performance, clocking load, and power consumption. From the results, it is clear that ILC significantly improves power consumption. It is also clear that using a single-level grid per se is not the source of energy savings for IM': using a single grid in the conventional design leads to a significant 7.9 W of extra power consumption.

Overall, we see that ILC can be introduced to a processor in various levels of ease. With minimum design intrusion, when only the very top level of the clock tree is modified to use injection locking, energy reduction is already significant (10%), thanks to the lowered jitter and skew. When we further optimize the clocking grid, the power savings become more pronounced (17%). All these are achieved without affecting performance or the design methodology of the processor.

IV. TEST CHIP

A test chip was fabricated in a standard 0.18-μm digital CMOS technology [see Fig. 8(a)] to demonstrate ILC advantages and verify its
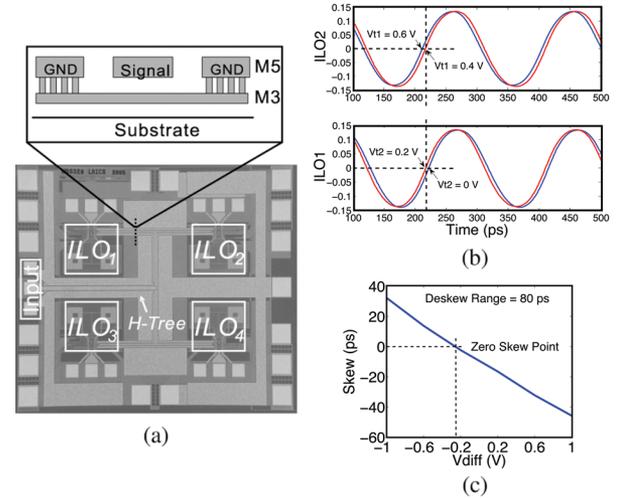

Fig. 8. (a) Die photo of the test chip with the transmission line structure of the interconnect showing in the inset. The spiral inductors in each ILO are also constructed using 1-μm-thick metal5, occupying 80 μm by 80 μm. Each inductor is 2.2 nH with a quality factor of 4 at 5 GHz. To demonstrate the deskew characterisitc of ILC, (b) waveforms and (c) phase difference (skew) of local clock signals generated by ILO1 and ILO2 are measured when they are tuned differentially. $\boldsymbol{V}_{\rm diff} = \boldsymbol{V}_{t2} - \boldsymbol{V}_{t1}$ is the differential tuning voltage. Note that the imbalance between ILO1 and ILO2 is caused by mismatch in the clock distribution tree and measurement system.

performance [19]. ILC on the test chip consists of four identical ILOs interconnected by an H-tree. Divide-by-2 ILOs as shown in Fig. 2(a) were used because the design was readily available. ILOs with fundamental or subharmonic frequency input will be introduced in the future. It is noteworthy that resonator-less oscillators can potentially be used as ILOs with some penalty in jitter performance if on-chip spiral inductors have very low Q. Each ILO drives an open-drain buffer, which then drives 50 Ω impedance of microwave instruments. A 10-GHz global clock is distributed to the ILOs to generate 5-GHz local clocks. Locking range of each ILO is designed and verified in measurement to be as large as 17% (when the input amplitude is 0.7 V). Long-term RMS jitter of generated local clocks is from 1.7 to 1.9 ps (0.0085UI to 0.0095UI) at different injection power, much less than that of the input signal (1.8 to 2.4 ps or 0.018UI to 0.024UI). This clearly demonstrates that an ILO can serve as a PLL and clean up the clock signal. The deskew capability of ILC is demonstrated by tuning the control voltage V_t of two ILOs [see Fig. 8(b)]. As shown in Fig. 8(c), skew up to 80 ps can be compensated. The deskew resolution of ILC depends on the skew detection and deskew control circuit, and 7 ps or less residue skew is achieable [16]. Thus, the assumption of 15 ps skew in an ILC system is valid. The ILOs consumes 7.3 mW with a 1.4-V power supply when injection signal is 6 dBm. In a real microprocessor, local clocks generated by ILOs will be buffered to drive the mostly capacitive local clock load. We plan to apply the resonant clocking technique to both the global and local clock distribution in ILC to further reduce the power consumption.

V. CONCLUSION

Thanks to the high sensitivity, good noise rejection, and built-in deskewing capability of injection-locked oscillators, the proposed injection-locked clocking can significantly improve skew and jitter performance of a multi-gigahertz clock distribution network. Reduced number of clock buffers, and recovered timing margin from skew and jitter lead to substantial power savings for the whole processor. Initial results from circuit and architectural simulations confirmed our analysis. More detailed modeling and characterization is under way, particularly in skew simulation. A chip prototype has also been recently demonstrated [19]. We expect the benefits of this new clocking

scheme will be even greater when it is applied to high-performance multi-core microprocessors and other high performance SoC systems.

REFERENCES

[1] R. Adler, "A study of locking phenomena in oscillators," in *Proc. IRE*, Jun. 1946, vol. 34, pp. 351–357.

[2] D. Bailey and B. Benschneider, "Clocking design and analysis for a 600-MHz alpha microprocessor," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1627–1633, Nov. 1998.

[3] W. Bowhill, S. L. Bell, B. J. Schneider, A. J. Black, S. M. Britton, R. W. Castelino, D. R. Donchin, J. H. Edmonson, H. R. Fair, P. E. Gronowski, A. K. Jain, P. L. Kroesen, M. E. Lamere, B. J. Loughlin, S. Mehata, T. A. Shedd, S. C. Thierauf, R. O. Mueller, R. P. Preston, and M. J. Smith, "Circuit implementation of a 300-MHz 64-bit second-generation CMOS alpha CPU," *Digit. Technol. J.*, vol. 7, no. 1, pp. 100–118, 1995.

[4] S. Chan, K. Shepard, and P. Restle, "1.1 to 1.6 GHz distributed differential oscillator global clock network," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2005, pp. 518–519.

[5] E. Friedman, "Clock distribution networks in synchronous digital integrated circuits," *Proc. IEEE*, vol. 89, no. 5, pp. 665–692, May 2001.

[6] V. Gutnik and A. Chandrakasan, "Active GHz clock network using distributed PLLs," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1553–560, Nov. 2000.

[7] A. Hajimiri, S. Limotyrakis, and T. Lee, "Jitter and phase noise of ring oscillators," *IEEE J. Solid-State Circuits*, vol. 34, no. 6, pp. 896–909, Jun. 1999.

[8] K. Kamogawa, T. Tokumitsu, and M. Aikawa, "Injection-locked oscillator chain: A possible solution to millimeter-wave MMIC synthesizers," *IEEE Trans. Microw. Theory Technol.*, vol. 45, no. 9, pp. 1578–1584, Sep. 1997.

[9] N. Kurd, J. Barkatullah, R. Dizon, T. Fletcher, and P. Madland, "A multigigahertz clocking scheme for the Pentium 4 microprocessor," *IEEE J. Solid-State Circuits*, vol. 36, no. 11, pp. 1647–1653, Nov. 2001.

[10] K. Kurokawa, "Injection locking of microwave solid-state oscillators," *Proc. IEEE*, vol. 61, no. 10, pp. 1386–1410, Oct. 1973.

[11] A. V. Mule, E. N. Glytsis, T. K. Gaylord, and J. D. Meindl, "Electrical and optical clock distribution networks for gigascale microprocessors," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 5, pp. 582–594, Oct. 2002.

[12] G. Pratt and J. Nguyen, "Distributed synchronous clocking," *IEEE Trans. Parallel Distrib. Syst.*, vol. 6, no. 3, pp. 314–328, Mar. 1995.

[13] H. Rategh and T. Lee, "Superharmonic injection-locked frequency dividers," *IEEE J. Solid-State Circuits*, vol. 34, no. 6, pp. 813–821, Jun. 1999.

[14] A. Sedra and K. Smith, *Microelectronic Circuits*. Oxford, U.K.: Oxford Univ., 2004.

[15] G. Semeraro *et al.*, "Dynamic frequency and voltage control for a multiple clock domain microarchitecture," in *Proc. Int. Symp. Microarch.*, Nov. 2002, pp. 356–367.

[16] S. Tam, R. Limaye, and U. Desai, "Clock generation and distribution for the 130-nm itanium 2 processor with 6-MB on-die L3 cache," *IEEE J. Solid-State Circuits*, vol. 39, no. 4, pp. 636–642, Apr. 2004.

[17] H. Wu and A. Hajimiri, "A 19 GHz, 0.5 mW 0.35 μm CMOS frequency divider with shunt-peaking locking-range enhancement," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2001, pp. 412–413.

[18] H. Wu and L. Zhang, "A 16-to-18 GHz 0.18 μm Epi-CMOS divide-by-3 injection-locked frequency divider," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2006, pp. 602–3.

[19] L. Zhang, B. Ciftcioglu, M. Huang, and H. Wu, "Injection-locked clocking: A new GHz clock distribution scheme," in *IEEE Custom Integrated Circuits Conf. Dig. Tech. Papers*, 2006, pp. 785–788.

[20] L. Zhang, A. Carpenter, B. Ciftcioglu, A. Garg, M. Huang, and H. Wu, "Injection-locked clocking: A low-power clock distribution scheme for high-performance microprocessors," 2007 [Online]. Available: www.ece.rochester.edu/research/laics/publication.html

[21] L. Zhang and H. Wu, "A double-balanced injection-locked frequency divider for tunable dual-phase signal generation," in *IEEE Radio-Frequency Int. Circuits (RFIC) Symp. Dig. Papers*, 2006, pp. 137–140.