Globally Asynchronous, Locally Synchronous Clocking and Shared Interconnect for Large-Scale SFQ Systems

Gleb Krylov, *Student Member, IEEE*, and Eby G. Friedman, *Fellow, IEEE*

***Abstract*—A globally asynchronous, locally synchronous clocking scheme for large-scale single flux quantum (SFQ) systems is proposed. In this scheme, the width of each data bus is extended to carry the corresponding clock signal. This signal activates the distribution of the clock signals within the receiving block. Based on this approach for intra-chip interconnect within SFQ systems, a configurable shared bus is also proposed. The data are attached to a tag, and a resulting data packet is sent to the shared bus. This packet is received by each block, but only processed if the tag matches the block identifier. By avoiding expensive comparators and multiplexers, the overhead of the global bus connection is reduced. The proposed approaches exploit the pulse-based nature and ambiguity of clock and data in SFQ technology – the data packet propagating through the interconnect carries a local clock signal.**

***Index Terms*—Single flux quantum, superconducting integrated circuits, superconductor digital electronics.**

I. Introduction

RECENT advances and ongoing research efforts in the area of superconductive electronics, including the development of EDA tools and methodologies, will greatly enhance the development of large scale single flux quantum (SFQ) systems [1], [2]. With improved integration and complexity of these superconductive systems, the number of functional units will significantly increase.

In modern CMOS systems, communication among the functional blocks within a system-on-chip is achieved using configurable interconnects [3]. These interconnects provide sophisticated interfaces for scheduling and ordering the requests for the different devices and memory. This approach supports modular design and reuse of IP cores with minimal modifications, while significantly reducing overall development time. An additional benefit of this approach is the flexibility and scalability of the resulting system - functional blocks can be added or removed without extensive modifications to the existing system.

Manuscript received October 30, 2018; accepted April 4, 2019. Date of publication April 9, 2019; date of current version May 22, 2019. This work was supported by the Department of Defense (DoD) Agency – Intelligence Advanced Research Projects Activity (IARPA) through the U.S. Army Research Office under Contract W911NF-17-9-0001. *(Corresponding author: Gleb Krylov.)*

The authors are with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: gleb.krylov@rochester.edu; friedman@ece.rochester.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASC.2019.2909985

Another important issue in large scale SFQ systems is timing. Most SFQ logic gates as well as all flip flops require a clock signal to operate. Furthermore, SFQ circuits are clocked at tens [4], [5] and even hundreds [6] of gigahertz, resulting in clock periods on the order of several to tens of picoseconds. Timing tolerances in these circuits are therefore extremely narrow, on the order of a few picoseconds. This issue will become more severe in future SFQ VLSI circuits as a primary application is high speed computation. One commonly used technique to mitigate timing variations across functional blocks and to simplify global timing in CMOS is globally asynchronous, locally synchronous (GALS) clocking [7]. In this approach, each functional block is locally clocked; a global clock distribution network is not required. Consequently, the area, power, and timing issues of global clock networks are avoided.

In this paper, two novel approaches exploiting a pulse-based signal representation of both data and clock in SFQ circuits are proposed. The GALS clocking scheme provides a means for data packets to carry a local clock, avoiding the overhead of a global clock network. This topic is discussed in section II. The proposed bus is an extension of the GALS scheme, and provides another means to exploit the ambiguity of clock and data in SFQ circuits to enhance asynchronous and self-timed communication while requiring reasonable overhead. This topic is described in section III. In section IV, some simulation results are provided, followed by some concluding remarks in section V.

II. GALS Clocking Scheme for SFQ Circuits

In this section, specific features of SFQ are discussed, and a means to exploit these features are proposed. In subsection II-A, the ambiguity of clock and data in SFQ is introduced, and early work exploiting this property is discussed. In subsection II-B, SFQ specific advantages for clock distribution are highlighted. In subsection II-C, a GALS clock activation approach utilizing this property is presented.

A. Ambiguity of Clock and Data

A distinctive quality of SFQ technology is the pulse-based nature of both data and clock. Information is encoded by the presence (logical "one") or absence (logical "zero") of an input SFQ pulse within a specific clock period. The logic gates process the inputs at the time of arrival of the clock pulse. The clock pulse triggers the junctions to produce an output based on the inputs.

The clock pulses arriving at the clock input are indistinguishable from the data pulses arriving at the data inputs. The clock and data inputs within an SFQ logic gate can therefore be arbitrarily exchanged to achieve a desired behavior. Moreover, the clock signal can be locally regenerated from the data at each gate.

This distinctive feature has been explored in data driven self-timed (DDST) RSFQ circuits [8]. In DDST circuits, the data are carried by complementary signals using two parallel lines for each bit. When data arrive at the next logic gate, the clock is generated by a logical OR function. At the output of the logic gate, the complementary signal is provided by a D flip flop with complementary outputs. While DDST circuits require significant overhead for routing as well as additional circuitry for generating complementary signals by each logic gate, this approach enhances controllability of the clock timing, improves robustness to process variations, and reduces the overhead of the global clock network.

Other important methods of a pulse-based data representation for clocking large circuits are the concurrent and counterflow clocking schemes [9], [10]. In the concurrent scheme, the clock pulse travels together with the data through the logic pipeline. In the counterflow scheme, the clock pulse travels in the opposite direction relative to data. In the clock-follow-data scheme, which is a type of concurrent scheme, the clock signal follows the data, marking the end of a clock period.

B. Clock Generation and Distribution

Efficient on-chip clock generation is necessary to operate multi-gigahertz systems within a cryogenic environment, as each high speed connection to/from room temperature is costly. SFQ technology, as opposed to CMOS, provides multiple efficient ways for on-chip clock generation with low overhead. Every Josephson junction with an applied DC voltage generates a train of SFQ pulse through the Josephson effect. Other methods for clock generation, such as arrays of junctions and long Josephson junctions, have also been proposed [11], [12]. For analog applications, the high Q factor of an on-chip oscillator is critical to reduce clock jitter. For digital applications, a simple ring oscillator is typically sufficient to generate a repetitive clock waveform.

Pulse-based signal representation leads to another beneficial property of SFQ circuits for clock generation and distribution. A switching Josephson junction regenerates the incoming SFQ pulse, restoring the shape of the signal. For a clock network within a functional block, the entire block can be clocked by a single clock pulse through a combination of ring oscillators and splitter gates. This property lessens the need for a global clock distribution network.

C. Clock Activation Scheme

A combination of DDST and clock-follow-data techniques – a GALS clock activation scheme – is proposed here. In this scheme, each multi-bit signal connection between functional blocks is extended by one additional signal, carrying the clock. Within each block, the clocking scheme is individually chosen to satisfy the local design criteria, and can be one of several synchronous schemes [13], [14] (binary tree, concurrent, counterflow) or an asynchronous scheme, where the incoming clock signal serves as a handshake signal. The proposed approach exploits the pulse-based nature and ambiguity of clock and data in SFQ technology - the data propagating through the interconnect carry a local clock signal. This approach, depicted in Figure 1, can be applied to any infrequently used circuit or block within a larger circuit that can benefit from the temporary absence of a clock signal.

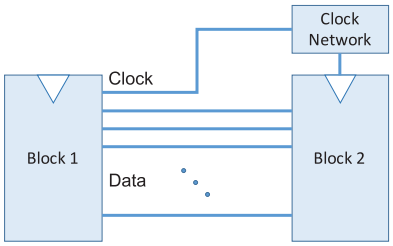

Fig. 1. Proposed GALS scheme.

The overhead of this approach is one additional signal line for every data bus. The area overhead of routing an additional line within the data bus is 3.1% for a 32-bit bus and 1.6% for a 64-bit bus.

The clock distribution network within the receiving circuit introduces a delay for each gate between the reception of the activation signal and generation of the appropriate clock signals. In addition, the driving circuit provides a clock activation signal for the receiving circuit that signals the end of the current operation. The overhead of the generating and activating circuits depends upon the particular circuit. The receiving circuit remains unchanged.

The benefits of the proposed GALS topology include a reduction in dynamic power dissipation and lower overall clocking complexity. In addition, this approach enables safe clock domain crossings [15] for SFQ systems with multiple clock domains.

III. Shared Interconnect

Heretofore, the low complexity of SFQ circuits did not justify the significant overhead of advanced bus architectures and interconnect networks widely used in modern CMOS systems-on-chip. These interconnects are therefore frequently designed manually on an *ad hoc* basis. With higher complexity superconductive systems, the development of a structured interconnect network for SFQ circuits will become necessary. SFQ technology provides higher speed and lower power interconnects as compared to CMOS technology. In the following subsections, these advantages are described, and relevant applications for the proposed bus are discussed. In subsection III-A, the primary types of interconnect for SFQ technology are briefly reviewed. In subsection III-B, the input discrimination characteristics of SFQ circuits, particularly beneficial for bus structures, are discussed. Based on these properties, in subsection III-C, a novel shared bus topology for SFQ circuits is proposed.

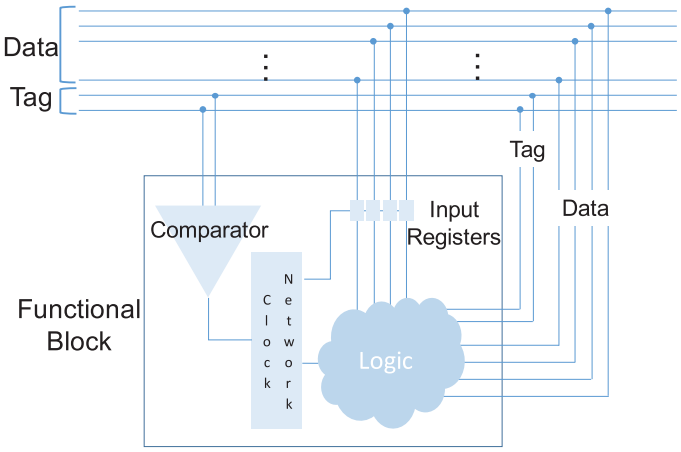

Fig. 2. Proposed shared bus topology.

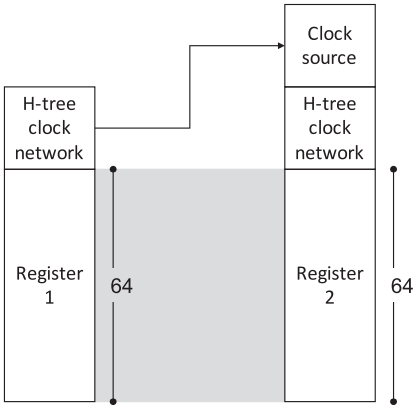

Fig. 3. 64-bit GALS clock activation scheme for two registers with an H-tree clock distribution network.

A. Types of SFQ Interconnect

Two types of signal interconnect used in SFQ integrated circuits are passive transmission lines (PTL) and Josephson transmission lines (JTL) [16], [17]. A JTL is an active structure composed of biased and grounded JJs connected by inductive lines. SFQ pulses are regenerated at each stage of a JTL, expending energy and adding delay. A PTL is a passive stripline connecting a driver and a receiver. SFQ pulses in a PTL propagate ballistically over significant distances at the speed of light within the medium with negligible loss. These distances can exceed several millimeters before a repeater is needed [18]. Multiple pulses can also simultaneously propagate along a PTL (in a wave pipelined fashion), further increasing throughput.

B. Input Discrimination

Another important consequence of pulse-based data representation of SFQ circuits is improved discrimination of unwanted inputs. Each input of an SFQ circuit supports multiple fan-in – multiple input lines from different sources which can be electrically connected to a circuit input, assuming no pulses arrive simultaneously. If the input registers support the rejection of multiple data pulses within a clock period through an escape junction or if blocking gates are used [19], the erroneous data are constrained to just one clock period. This property is useful for bus structures, similar to three state buses in conventional CMOS circuits [20].

C. Bus Topology

In this subsection, a shared bus topology is proposed, exploiting the aforementioned SFQ circuit properties and the GALS clock activation scheme described in section II. In this proposed bus topology, as depicted in Figure 2, multi-bit data originating from each device are attached to a tag identifying the recipient block, forming a data packet, *e.g.*, the memory access request is attached to a tag identifying the memory controller. The resulting packet is passed to the shared bus, consisting of passive transmission lines, Josephson transmission lines, and functional block interfaces containing a hard wired block identifier.

At each block interface, the tag in the incoming data packet is compared to the block identifier. As the identifier is predetermined, the comparator does not require significant overhead. For a matching condition, a control pulse is generated. This pulse is a write enable signal for the input register of the block, a handshake signal for an asynchronous block, or a clock signal for an entire block. In the latter case, the data inputs are converted into clock signals, as described in subsection II-B – a feature not possible in CMOS.

The data reach all of the functional blocks connected to the bus, and are stored within the input registers. Processing, however, is only initiated if the tag matches the identifier. In this way, fewer expensive decoders at each block interface are needed. To reach the previous block connected to this interconnect, the bus is configured as a circular topology with the end connected to the beginning. A similar approach has been previously proposed in a CMOS-based optical ring bus [21].

IV. Simulation Results and Discussion

In this section, simulation results describing the GALS clock activation scheme are presented. In subsection IV-A and IV-B, globally asynchronous communication between two synchronous shift registers is demonstrated and certain timing characteristics are illustrated. In subsection IV-C, several approaches for clock activation methodology are discussed. In subsection IV-D and IV-E, respectively, the applicability of the proposed approach to multi-chip modules and energy efficient SFQ is discussed.

A. Simulation Results for Two H-Tree Networks

Two 64-bit wide shift registers with a depth of eight bits are generated via a Python script. These SFQ shift registers utilize an H-tree clocking topology as a binary splitter tree with zero skew between the bits and stages, implemented as an array of D flip flops. The clock signal from the last stage of the first register is bundled with a 64-bit data connection and connected to the second register. The clock signal, after some delay, is distributed within the second register by a similar H-tree. The

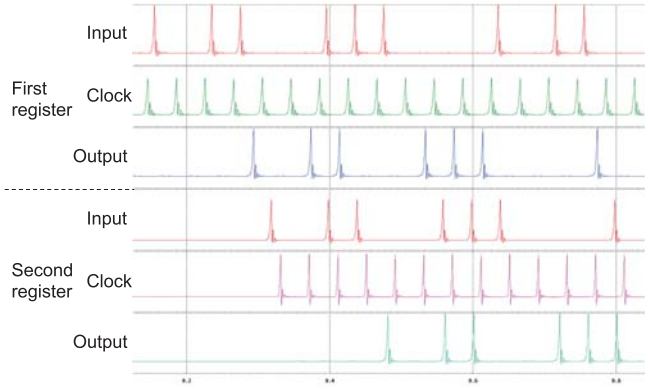

Fig. 4. Waveforms of a one bit slice of the circuit shown in Fig. 3.

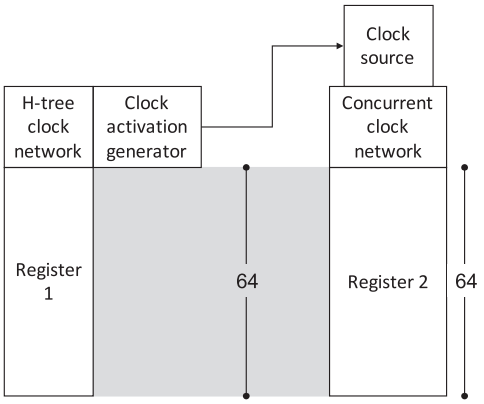

Fig. 5. 64-bit GALS clock activation scheme for an H-tree shift register connected to a shift register with a concurrent clock network.

circuit is schematically shown in Figure 3, and WRSpice [21] waveforms are depicted in Figure 4.

In this example, no additional area or energy savings from the proposed clocking scheme is achieved. The overall timing complexity is reduced since the clock signal of the receiving register is derived from the clock signal of the transmitting register. This dependent clock signal does not necessarily require zero skew relative to the primary clock source. By relaxing the requirements on the clocking system, the timing constraints on the EDA routing tools are also relaxed.

B. Simulation Results for H-Tree and Concurrent Networks

A 64-bit shift register with a depth of eight bits and a concurrent clock distribution network are connected to one of the registers, as described in subsection IV-A. The resulting topology is depicted in Figure 5. The clock activation signal is generated by the AND-OR function between the clock signal of the transmitting block and the output data. This signal, connected through a delay line, is a concurrent clock signal within the receiving block. A simulation of this topology is depicted in Figure 6.

This circuit demonstrates the generation of a clock activation signal and a connection between two different clock networks. In the case of a concurrent clock network within the receiving block, no additional clock source is needed as the clock signal travels together with the data packet.

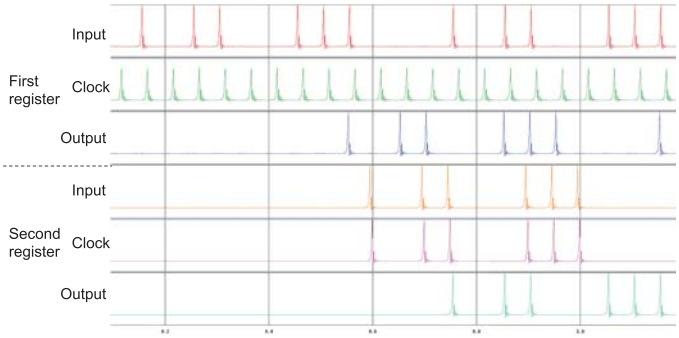

Fig. 6. Waveforms of a one bit slice of the circuit shown in Fig. 5.

C. Discussion

Two possible approaches exist for clock activation and distribution in the proposed scheme. The first approach is to distribute the activation signal through the clock distribution network. The resulting number of clock pulses precisely matches the number of incoming data pulses, and the clock is gated when data are not present. The benefit of this scheme is the absence of any activity when no data are being processed. The disadvantage, however, is careful control of the clock-data timing relationship is necessary [22]. The time to propagate the activation signal between blocks and through the clock network needs to be less than the time for the data to propagate between the blocks plus one clock period. If this condition is not satisfied, fewer clock signals than data packets will be available. To satisfy this condition, additional delay may need to be added to the data path.

The second approach is to use the activation signal to turn on the source of the local clock. In this approach, the timing requirements are relaxed – the activation signal is only required to arrive before the data packet. The disadvantage is that additional unnecessary clock pulses will be generated.

D. Multi-Chip Modules

Prospective SFQ systems will extensively utilize multi-chip modules (MCM) to improve integration and reduce the number of connections to a room temperature environment. Several functional blocks can be placed on different integrated circuits (IC) within the same MCM. High speed communication between the ICs is possible through MCM bumps, where a data transmission rate of 93 GHz has been demonstrated [23]. These connections are suitable for the proposed GALS technique, supporting prospective MCM systems.

E. Compatibility of Energy Efficient SFQ With Proposed Approaches

Energy efficient SFQ (ERSFQ) circuits utilize JTLs connected to the clock distribution network as a source of the highest average voltage within a circuit [24]. As both the GALS clock activation scheme described in section II and the shared bus system described in section III gate the primary clock signal, current regulation within the energy efficient bias network is affected. To maintain compatibility with ERSFQ, the feeding JTLs of those

circuits that utilize these proposed schemes are connected to an external average voltage reference.

Bias distribution for ERSFQ requires a power-up time – the time required to distribute the bias currents. This time can be lowered by using smaller bias inductors or a larger feeding JTL. Larger feeding JTLs with a lower bias inductance, which results in a faster power-up process, can be used in those circuits that can tolerate a power-up delay.

V. Conclusion

A globally asynchronous, locally synchronous clocking scheme based on clock activation signals and an intra-chip interconnect network for large scale SFQ systems are proposed in this paper. The GALS clock activation scheme utilizes an enable signal generated by a functional block as a clock or a handshaking mechanism for another block. The proposed GALS clock activation scheme reduces the complexity of the clock network and provides timing flexibility and power savings with a routing area overhead of 3.1% for a 32-bit bus. The proposed approaches combine existing CMOS asynchronous network-on-chip architectures, such as an intra-chip shared circular bus, with the unique features of SFQ circuits, thereby enabling high complexity SFQ VLSI and MCM-based systems.

References

[1] K. Gaj, Q. P. Herr, V. Adler, A. Krasniewski, E. G. Friedman, and M. J. Feldman, "Tools for the computer-aided design of multigigahertz superconducting digital circuits," *IEEE Trans. Appl. Supercond.*, vol. 9, no. 1, pp. 18–38, Mar. 1999.

[2] C. J. Fourie, "Digital superconducting electronics design tools - status and roadmap," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 5, Aug. 2018, Art. no. 1300412.

[3] B. Mathewson, "The evolution of SOC interconnect and how NOC fits within it," in *Proc. ACM/IEEE Design Autom. Conf.*, Jun. 2010, pp. 312–313.

[4] T. V. Filippov *et al.*, "20 GHz operation of an asynchronous wave-pipelined RSFQ arithmetic-logic unit," *Phys. Procedia*, vol. 36, pp. 59–65, May 2012.

[5] A. Fujimaki, M. Tanaka, T. Yamada, Y. Yamanashi, P. Heejoung, and N. Yoshikawa, "Bit-serial single flux quantum microprocessor CORE," *IEICE Trans. Electron.*, vol. 91, no. 3, pp. 342–349, Mar. 2008.

[6] W. Chen, A. V. Rylyakov, V. Patel, J. E. Lukens, and K. K. Likharev, "Rapid single flux quantum T-flip flop operating up to 770 GHz," *IEEE Trans. Appl. Supercond.*, vol. 9, no. 2, pp. 3212–3215, Jun. 1999.

[7] E. G. Friedman, *High Performance Clock Distribution Networks*. Boston, MA, USA: Springer, 1997.

[8] Z. J. Deng, N. Yoshikawa, S. R. Whiteley, and T. V. Duzer, "Data-driven self-timed RSFQ digital integrated circuit and system," *IEEE Trans. Appl. Supercond.*, vol. 7, no. 2, pp. 3634–3637, Jun. 1997.

[9] K. Gaj, E. G. Friedman, and M. J. Feldman, "Timing of multi-gigahertz rapid single flux quantum digital circuits," *J. VLSI Signal Process. Syst. Signal Image Video Technol.*, vol. 16, no. 2, pp. 247–276, Jun. 1997.

[10] R. N. Tadros and P. A. Beerel, "A robust and self-adaptive clocking technique for SFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 7, Oct. 2018, Art. no. 1301211.

[11] I. V. Vernik and D. Gupta, "Two-phase 50 GHz on-chip long Josephson junction clock source," *IEEE Trans. Appl. Supercond.*, vol. 13, no. 2, pp. 587–590, Jun. 2003.

[12] Y. Zhang and D. Gupta, "Low-jitter on-chip clock for RSFQ circuit applications," *Supercond. Sci. Technol.*, vol. 12, no. 11, p. 769, Dec. 1999.

[13] E. G. Friedman, "Clock distribution design in VLSI Circuits – an overview," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1993, pp. 1475–1478.

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

[15] R. Ginosar, "Fourteen ways to fool your synchronizer," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, May 2003, pp. 89–96.

[16] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock-frequency digital systems," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, Mar. 1991.

[17] T. Jabbari, G. Krylov, S. Whiteley, E. Mlinar, J. Kawa, and E. G. Friedman, "Interconnect routing for large scale RSFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1102805.

[18] R. L. Kautz, "Picosecond pulses on superconducting striplines," *J. Appl. Phys.*, vol. 49, no. 1, pp. 308–314, Jan. 1978.

[19] G. Krylov and E. G. Friedman, "Design for testability of SFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 8, Dec. 2017, Art. no. 1302307.

[20] P. Horowitz and W. Hill, *The Art of Electronics*. New York, NY, USA: Cambridge Univ. Press, 1989.

[21] S. Pasricha and N. Dutt, "ORB: An on-chip optical ring bus communication architecture for multi-processor systems-on-chip," in *Proc. ACM/IEEE Asia South Pacific Design Autom. Conf.*, Jan. 2008, pp. 789–794.