

Reconfigurable Clock Distribution Circuitry

Atanu Chattopadhyay, Zeljko Zilic

Department of Electrical and Computer Engineering, McGill University
Montreal, Quebec, Canada

Abstract— We present the circuitry required for implementing a multi-clock reconfigurable, reprogrammable clock distribution network for integrated circuits using a reference-based scheme for skew compensation. In the scheme, a device is subdivided into multiple regions and a bi-directional clock distribution line is daisy-chained through the device, connecting each region in the domain. Switching structures that can be used to re-route the clock chain are added where needed. The proposed design simplifies layout for irregularly shaped clock domains and provides flexibility to designers by enabling post-fabrication changes to the clock distribution network. Reconfigurable clock distribution networks can be used in some ASICs, SoCs and FPGAs. The reference-based approach used is applicable to both single and multiple clock distributions.

I. INTRODUCTION

Clock distribution for modern integrated circuits represents a challenge due to increasingly complex systems, decreased power supply voltages, larger die sizes and higher clock rates [1],[2]. There are many conflicting requirements for an adequate clock distribution network (CDN). These considerations include clock signal characteristics such as fast transition times and a balanced duty cycle, and clock skew [3]. Typical clock skew budgets have used a 10% of clock period metric to define adequate skew boundaries [4],[5].

Clock tree balancing alone is increasingly insufficient since clock buffer mismatches due to in-die process variations limit the ability to minimize skew [6]-[8]. Traditional H-trees are not well-suited to distributing clocks to asymmetric, irregularly shaped clock domains and add further complication to the floorplanning and layout of designs. The schemes that apply skew reduction techniques on existing H-tree distributions [2],[9],[10] suffer from the same drawbacks of their forbearers: high power consumption and inefficient use of interconnect. Other distributions perform skew compensation at the source, independently for each leaf [2],[6],[11]. This creates a need for long and varying length reference lines returning from each leaf to the source, introducing errors to the skew compensation technique due to the process-dependant delay of each feedback line [12]. The distribution in [13] can perform multi-point skew compensation using multiple independent analog skew compensation buffers. They use a 3-wire method with a raw clock, and forward and reverse reference lines that are tied at the far end of the clock distribution. This technique calibrates the clock network for a reference line that is distinct from the clock signal and is hence susceptible to skew from loading

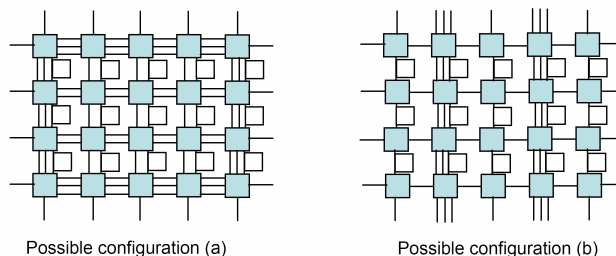


Fig. 1: 2 potential 15-tap Clock Distributions

effects and mismatches. Further, the required omission of clock buffers limits the distance and the number of taps that the raw clock can spawn.

Our method [14] improves upon the design first proposed in [13] by reducing the complexity of the hardware and combining the three reference lines into one bi-directional line. The presence of clock buffers allows us to redirect clocks dynamically at certain pre-defined switchpoints, making the distribution reconfigurable. The circuitry can be used to create fully programmable distributions such as the two possible 15-tap configurations shown in Fig. 1. The unshaded boxes represent clock regions and the shaded boxes represent switchpoints used to reroute a clock domain's distribution line. The proposed CDN is scalable, compatible with irregularly shaped distribution areas, and combines low power operation with tight skew bounds. The reference-based clocking scheme is easy to lay out, automatically compensating for variances in the global clock spine by using a single reference and distribution line. We present here the circuits used and the simulation results for two example circuits using our reference clocking scheme.

II. REFERENCE-BASED CLOCK DISTRIBUTION

To implement a reconfigurable clock distribution using our method, the circuit needs to be divided into roughly equal subregions, each using a small H-tree to distribute the clock from the tap to the leaves. The smaller the area of these subregions, the more taps are required, but the less variability there is within each subregion. All the taps of each clock domain can then be connected together as a "thread" using a single wire, possibly through appropriate switchpoints, to create the desired shape of the clock domain. The clock threads can cross other clock domains easily and are simple to lay out since the taps do not need to be located in close proximity or be spaced regularly. While [13] has shown that

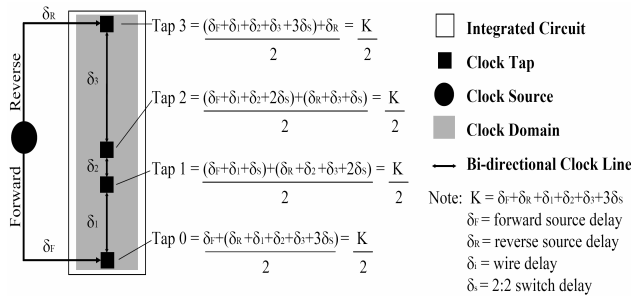


Fig. 2: Reference-based clocking for a single clock domain.

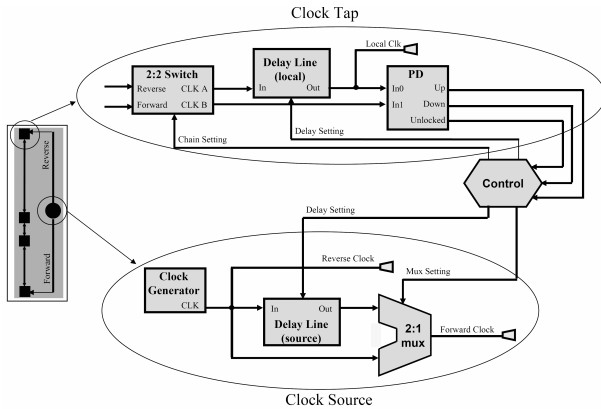


Fig. 3: Components required for our reference-based distribution.

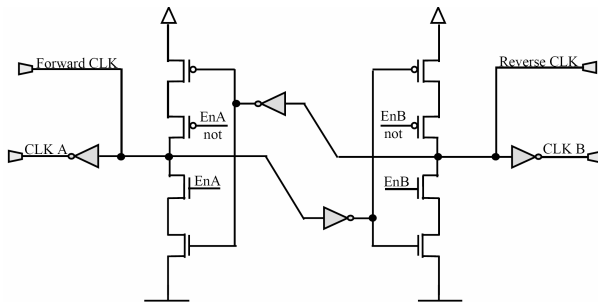


Fig. 4: A 2 in, 2 out switch used to control access to each tap

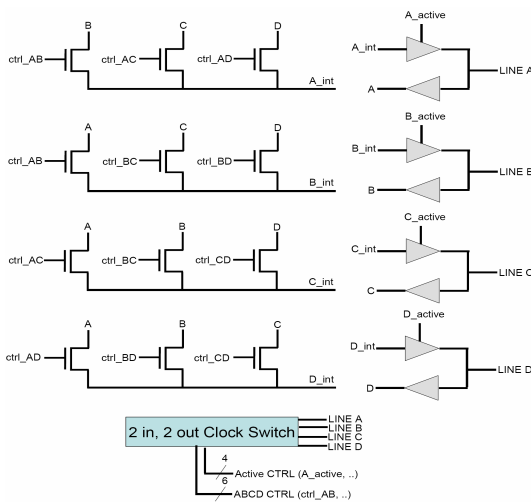


Fig. 5: A 2 in, 2 out clock routing switch

it is possible to synchronize a local clock to a point directly in between forward and reverse traveling reference clocks, we

take that concept one step further. Using bi-directional circuit elements with equal propagation delays for either direction for every segment, it is possible to reuse a single wire to propagate both forward and reverse reference clocks. This wire reuse also allows us to add buffering in the clock path so that the maximum length of the clock distribution is not limited by the drive strength of a single clock driver. The driver strength only limits the maximum distance between buffers. The underlying concept of our CDN is shown in Fig. 2. Each tap contains the necessary hardware to delay the local clock and to route the reference clocks between subregions. Since the clock distribution line has a constant delay ($K + \delta_s$) over its entire length (where δ_s is the delay through a switch), if the delay of the forward clock at a tap is δ_r , the delay of the reverse clock will be $\delta_r = K - \delta_r$. By positioning every local clock directly in between the forward and reverse reference clocks – the average – the resulting rising edges are all at:

$$\frac{\delta_r + \delta_r}{2} = \frac{(K - \delta_r) + \delta_r}{2} = \frac{K}{2} \quad (1).$$

The bi-directional clock line can be routed around obstacles easily without compromising skew tolerance since the clock taps are daisy-chained.

III. CIRCUIT COMPONENTS

The components required to implement the complete system in Fig. 3 includes routing switches, a delay line and a phase detector with their design requirements, discussed next.

A. Routing Switches

The CDN must synchronize each tap sequentially through the 2 in, 2 out switches located along the bi-directional clock line. The switches must propagate forward and reverse clocks to $CLK A$ and $CLK B$, respectively, of the tap being synchronized, Fig. 3. To bypass a tap, the switch in Fig. 4 connects the forward and reverse ports together.

To reconfigure clock domains, other routing switches are required. These have a more stringent design requirement as each port must be able to route its signal to any other port, while matching delays in both directions between associated ports. A 2 input, 2 output (2-clock) routing switch is shown in Fig. 5. It uses pass transistors to control access to an intermediate signal, which gains access to the clock port through a Z-buffer. The Z-buffers establish the direction of the clock signals through the network and are controlled by 4 active control bits of which only 2 can simultaneously be asserted. The 6 $ABCD_ctrl$ bits control the connectivity through the network. This design is scalable to create larger routing constructs. For instance, a 4-clock switch containing 8 ports has been designed in a similar fashion.

B. Variable Delay Line

The variable delay line should be designed to have good linearity between potential delay settings. Here, the maximum delay increment between adjacent settings establishes the worst-case clock skew of the CDN. The complete delay line has a fixed delay component (D) and a variable delay component (δ). The minimum frequency that can be distributed is a function of the total achievable variable delay:

$$F_{\min} = 1/T_{\max} = 1/(2 \cdot \delta) \quad (2).$$

The factor of 2 comes from the ability of the delay line to generate both an inverted and a true version of the input, doubling the effective delay of the delay line for periodic signals. The delay line implementation needs to be as small as possible and should have a minimum of control lines since it needs to be replicated for each tap in the distribution.

The circuit for one suitable delay line is shown in Fig. 6. It achieves equal high-to-high and low-to-low delays, resulting in matching duty cycles for input and output clocks. The variable delay is created through independent coarse and fine delay components. The coarse delay line in Fig. 7 is scalable with additional stages resulting in a lower minimum frequency. A fine delay is needed to fill the gap between coarse settings. The fine delay line uses four instances of the variable delay inverter in Fig. 8. Since process and temperature variations could result in the pull-up and pull-down circuitry behaving differently, the fine delay lines are grouped in pairs, so that each input transition propagates through identically set PMOS and NMOS segments of the variable inverter pair.

C. Phase Detector

There are many phase detectors presented in literature [14],[15]. We propose an original design to solve the unique challenges of our system. Like most detectors, the design in Fig. 9 uses two cross-coupled latch structures to alter the independent *UP* and *DOWN* signals. It is a sample-and-hold type, as it retains the output value for roughly half a clock cycle, independent of the overlap between input clocks. The ordering of the NAND gate inputs in Fig. 9 is critical since the gate is more sensitive to the upper input toggling high since it is connected to the NMOS transistor closer to the output node. The exception is the '=' NAND gate, designed with equal propagation delays. Due to this gate, the phase detector neither asserts an *UP* nor a *DOWN* when clocks *A* and *B* are close. The key difference with other phase detectors is this "nearly locked" state, which is allowed because of the granular nature of the delay line. Our phase detector only needs to be as precise as half the maximum increment between fine delay settings, emphasizing finite resolution time over absolute precision. While this phase detector achieves good resolution near zero skew conditions, it suffers from a lack of sensitivity to nearly non-overlapping inputs. When this condition is detected, an *unlocked* signal can be asserted, triggering either an *UP* or a *DOWN* response, since they are equivalent here.

D. Controller

The clock distribution network requires three distinct phases to work properly: synchronization, calibration and operation. During the synchronization phase, each clock tap is sequentially calibrated, from region 0 to region $n-1$. The forward clock needs to be delayed to align with the reverse clock using equal settings on each of the two (local and source) delay lines. Once all the taps have been sequentially synchronized, the source delay element is removed from the circuit path resulting in the appropriate "average" clock appearing at all the taps. One drawback of averaging the rising

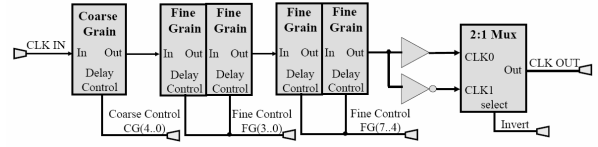


Fig. 6: Variable delay cell capable of periodic delays.

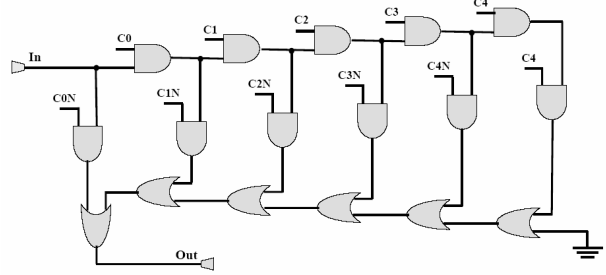


Fig. 7: Coarse delay unit with 6 delay settings.

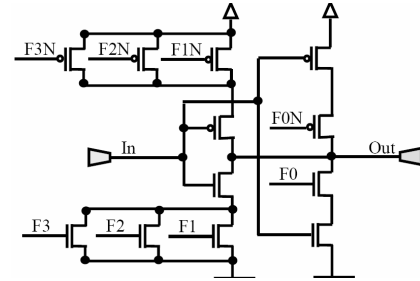


Fig. 8: Fine delay unit.

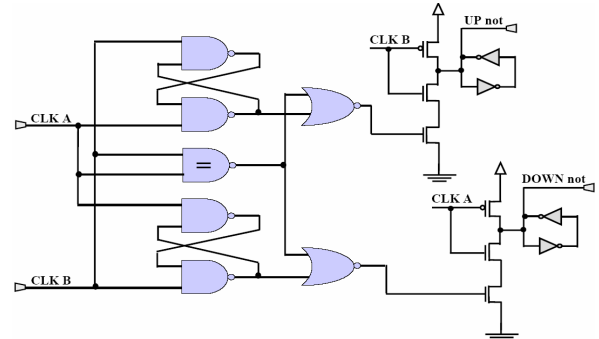


Fig. 9: Phase detector.

clock edges is that each local clock may need inversion, depending on the relative phase of the reference clocks. This dictates an additional step following synchronization to align the polarity of the local clocks by inverting the appropriate clock taps. The controller required to obtain this behavior can be implemented in hardware or with a software-based scheme, possibly using a microprocessor. This dual option for the control is useful to accommodate different applications and architectures. Following calibration, the phase detection and calibration circuitry are disabled to save power.

IV. SIMULATION RESULTS

All the circuits discussed here have been designed for TSMC's 0.18 μm P-well process using the Cadence Virtuoso design environment with the SpectreS and Analog Artist simulators. The simulated circuit behaves as follows. Each of

the five additional coarse grain elements provides a 93.5 ps delay increment over the base setting, resulting in 6 possible coarse delays. Each fine delay line has a range of 0 to 93.5 ps. Retaining sixty-three fine settings results in a 1.51 ps average and a 4.75 ps maximum delay increment. Thus considering both source and local delay lines, the maximum delay that can be achieved (δ) is 1122 ps resulting in a minimum frequency of 446 MHz, Eqn. 2. The resolution of the phase detector is ± 1.5 ps. Since one of the variable delay elements is removed during operation, the net error of the phase detector is halved. The clock distribution circuitry is capable of distributing frequencies up to 1.90 GHz, corresponding to periods of 525 ps and higher. The limiting factor is the 2 in, 2 out switch that drives the bi-directional clock line whose large capacitive load prevents higher performance. The remaining circuits can run up to 2.12 GHz (470 ps period). Should higher performance be required, these circuits can be redesigned, as needed. Simulations show that for a typical 8-tap single clock configuration, the circuit achieves an under 10 ps skew bound. The energy consumption at the maximum frequency is 33.2 mW for synchronization and 18.0 mW at run time. At 891 MHz, the circuit consumes 18.6 mW for synchronization and 9.97 mW at run time.

Fig. 10 shows the synchronized clocks of a 3-clock domain reconfigurable distribution. The 15-tap reference circuit is shown in Fig. 1b and is made up of 6-taps in domain A (1.11 GHz), 4 taps in domain B (1.33 GHz) and 5 taps in domain C (1.66 GHz). The total clock skew ranges from 3.9 ps to 5.5 ps with an overall power consumption of 62.82 mW, or 4.188 mW per tap. Comparable solutions offer similar or worse levels of skew reduction: sub-10 ps for [16], 70 ps for [5], 28 ps for [17] and 15 ps for [18]; [9] is capable of reducing skew to within 10% of the clock period, versus under 4% here; [1] achieves 3 ps skew resolution by modifying an H-tree and requiring a duplicate co-located return path for all leaves. PLL-based distributions typically consume hundreds of mW [17]. [10] shows 0.21 mW power consumption per de-skew tap for a 56 ps skew bound.

V. CONCLUSION

We have designed a novel skew-tolerant multi-point clock distribution that is suitable for irregularly-shaped clock regions and can create a reconfigurable clock distribution for multiple clock domains. This method is useful for both traditional single and multi-clock designs, and reconfigurable and programmable devices like FPGAs. Using a daisy-chain approach reduces clock load and power. The use of a single forward and reverse clock reference line that inhibits skew caused by in-die process variation in the global distribution is unique. The system can also be used to provide beneficial skew between each of the local taps [19] and to correct for variances in the clock distribution, post-fabrication.

REFERENCES

[1] R. L. Aguiar, and D. M. Santos, "Wide-area clock distribution using controlled delay lines," *Proc. ICECS 1998*, vol. 2, 63-66.
 [2] A. Kapoor, N. Jayakumar, and S. P. Khatri, "A novel clock distribution and dynamic de-skewing methodology," *Proc. ICCAD 2004*, 626-631.

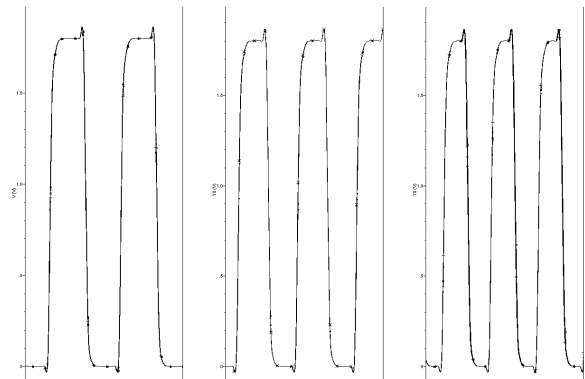


Fig. 10: Synchronized, calibrated 3-clock domain reference circuit (left to right: Domain A, Domain B and Domain C).

[3] C.-Y. Yang, and S.-I. Liu, "A one-wire skew-compensating clock distribution based on bidirectional techniques," *IEEE Journal of Solid-State Circuits*, 36, 2 (Feb. 2001), 266-272.
 [4] M. Omana, D. Rossi, and C. Metra, "Fast and low-cost clock deskew buffer," *Proc. DFT 2004*, 202-210.
 [5] V. Varghese, T. Chen, and P. Young, "Stability analysis of active clock deskewing using a control theoretic approach," *Proc. ACC 2005*, vol. 3, 1758-1763.
 [6] S. Tam, S. Rusu, U. Nagarji Desai, R. Kim, Ji Zhang, and I. Young, "Clock generation and distribution for the first IA-64 microprocessor," *IEEE Journal of Solid-State Circuits*, 35, 11 (Nov. 2000), 1545-1552.
 [7] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, and C. Guardiani, "Analysis of the impact of process variations on clock skew," *IEEE Trans. on Semiconductor Manufacturing*, 13, 4 (Nov. 2000), 401-407.
 [8] E. G. Friedman, "Clock distribution networks in synchronous digital integrated circuits," *Proc. of the IEEE*, 89, 5 (2001), 665-692.
 [9] M. Omana, D. Rossi, and C. Metra, "Low cost scheme for on-line clock skew compensation," *Proc. VTS 2005*, 90-95.
 [10] T. Yamashita, T. Fujimoto, and K. Ishibashi, "A dynamic clock skew compensation circuit technique for low power clock distribution," *Proc. ICICDT 2005*, 7-10.
 [11] H. Lee, H. Q. Nguyen, and D. W. Potter, "Design self-synchronized clock distribution networks in an SoC ASIC using DLL with remote clock feedback," *Proc ASICSOC 2000*, 248-252.
 [12] M. Saint-Laurent, M. Swaminathan, and J. D. Meindl, "On the micro-architectural impact of clock distribution using multiple PLLs," *Proc. ICCD 2001*, 214-220.
 [13] W. D. Grover, J. Brown, T. Friesen, and S. Marsh, "All-digital multipoint adaptive delay compensation circuit for low skew clock distribution," *Electronics Letters*, 31, 23 (9 Nov. 1995), 1996-1998.
 [14] A. Chattopadhyay and Z. Zilic "Reference-based clock distribution architectures," *Proc. MWSCAS 2006*.
 [15] C.-S. Hwang, W.-C. Chung, C.-Y. Wang, H.-W. Tsao, and S.-I. Liu, "A 2 V clock synchronizer using digital delay-locked loop," *Proc. AP-ASIC 2000*, 91-94.
 [16] S. Soliman, F. Yuan, and K. Raahemifar, "An overview of design techniques for CMOS phase detectors," *Proc. ISCAS 2002*, vol. 5, V-457-460.
 [17] D. E. Brueske, and S. H. K. Embabi, "A dynamic clock synchronization technique for large systems," *IEEE Trans. on Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, 17, 3 (Aug. 1994), 350-361.
 [18] B. Doyle, P. Mahoney, E. Fetzer, and S. Naffziger, "Clock distribution on a dual-core, multi-threaded Itanium family microprocessor," *Proc. ICICDT 2005*, 1-6.
 [19] Hong-Yean Hsieh, Wentai Liu, M. Clements, and P. Franzon, "Self-calibrating clock distribution with scheduled skews," *Proc. ISCAS 1998*, 470-473.