

# A Clock Tuning Circuit for System-on-Chip

Yaron Elboim, Avinoam Kolodny and Ran Ginosar  
VLSI Systems Research Center, Electrical Engineering Department  
Technion--Israel Institute of Technology, Haifa 32000, Israel  
[ran@ee.technion.ac.il]

## Abstract

*Clock distribution in System-on-Chip (SoC) designs has become a problem for integrating IP cores into a single synchronous SoC, because of different clock delays in the IP cores. We propose an on-chip clock tuning circuit. Programmable delays are inserted in the clock distribution network, facilitating clock alignment and synchronization. Design iterations are eliminated, saving design effort and cost. The method also compensates for unbalanced clock trees. The circuit was implemented in a commercial chip, and demonstrated good functionality and high productivity.*

## 1. Introduction

In SoC design, a buffered clock distribution network is typically used to drive the large clock load. Chip design involves a clock alignment step, which equalizes the delay from the clock source to each and every clock target [1][2]. Accurate clock alignment is important, because unwanted differences or uncertainties in clock network delays may degrade performance or cause functional errors. Clock distribution and alignment has become an increasingly challenging problem in VLSI design, consuming an increasing portion of resources such as wiring area, power and design time [3].

Ideally, IP cores ("IPs") should be treated as "black-boxes" to support "plug-and-play" [4], such that IPs can be inserted or removed without affecting other blocks. However, the clock distribution network does not support this concept because each change influences the complete network [1]. Redesign and verification of the global clock distribution network may be required after each change. Such iterations are undesirable and should be minimized.

In a competitive commercial environment, IC design is typically optimised for shortest time to market. Physical design may be performed in parallel with logic design, although in theory the former should follow the completion of latter. In such cases, global physical features of the IC, such as the global clock distribution network, may have to be redesigned multiple times, where each change in the logic incurs painful and expensive redo of the global nets.

*Clock tuning* can be used to eliminate repetitive redesign of the clock network. A tuning circuit can be

used statically or dynamically to perform clock alignment according to the uncertainty of the system [1]. Multiple PLLs may be employed to align the clock dynamically [5], but are expensive and difficult to design.

We propose an efficient method for clock alignment in SoC, using a programmable circuit for static delay tuning. The main goals of static delay tuning are to enable quick and easy integration of IP cores into SoC and to ease the design of the SoC clock distribution network. In Section 2 we demonstrate the problem of SoC integration due to different clock delays, and compare the common solutions of signal and clock delay insertion. Clock tuning is presented in Section 3, and its additional application to balancing the clock distribution network is discussed in Section 4. Sections 5 and 6 describe the clock tuning circuit and the experimental results.

## 2. Internal IP Core Clock Delays

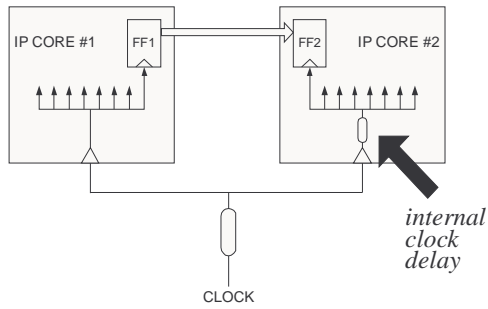
Consider the SoC of Figure 1. A global clock is distributed such that it arrives at exactly the same phase to all IP cores. However, since IP Core #2 has an internal clock delay larger than that of IP Core #1, the flip-flops of the two IP Cores are not synchronized. An output registered in FF1 would be missed by FF2 because, by the time FF2 receives the clock edge, the output of FF1 (and correspondingly the input of FF2) has already changed. This is a classic min-delay problem, caused in this case by non-uniform internal clock delays of various IP Cores. A non-negligible internal clock delay is typical in deep sub-micron processes [6][7].

*Data Delay Insertion* provides one solution to this problem, as in Figure 2. Data lines are delayed to match the clock phase difference. This is not a desirable approach: Many delay elements may be required for wide data buses, incurring heavy area and power penalties, and circular dependencies may prohibit a solution altogether.

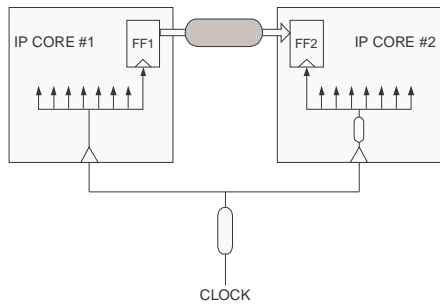
*Clock Delay Insertion* is a better solution (Figure 3). Delay is inserted in front of the clock input port of IP Core #1, adjusted to assure that FF1 and FF2 are synchronized.

In typical SoC designs the clock delays are added manually between the clock distribution network and the

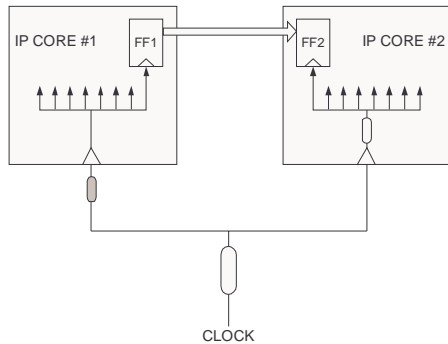
clock port of each IP core. This paper proposes a programmable method for inserting clock delays.



**Figure 1: SoC Clock Synchronization Problem. FF1 and FF2 are not synchronized due to non-uniform internal IP core clock delays**



**Figure 2: Data Delay Insertion**



**Figure 3: Clock Delay Insertion**

### 3. Clock Delay Insertion Methodology

#### 1.1. Delay Insertion Algorithm

A typical clock distribution network is shown schematically in Figure 4a. The network consists of a balanced clock tree where the delay from the root to each leaf is the same. Thus, the clock inputs of all IP cores receive the same clock phase. As shown in the previous section, this approach leads to data delay insertion and is hence not desirable. Alternatively, the clock delay insertion method enables a different total clock delay to each IP core, as demonstrated in Figure 4b. These delays compensate for the different internal clock delays of the various IP cores. The complete SoC is thus clock aligned

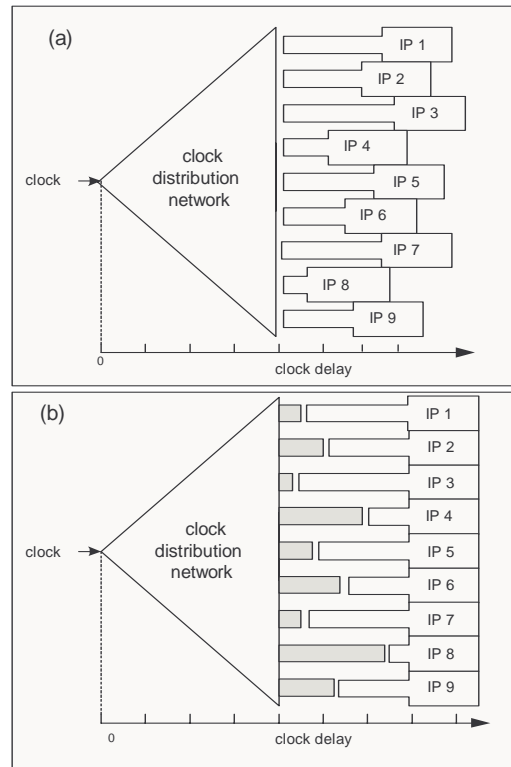
with zero skew among all state elements in all IP cores. The clock insertion method is based on the following algorithm:

$$D := \max\{d_i\}$$

$$\text{for each IP core } i = 0 \dots N$$

$$\text{Add clock delay } \Delta_i = D - d_i;$$

Where  $d_i$  is the internal clock delay of IP core  $i$ . Optionally,  $D' = D + \Psi$  may be employed instead of  $D$ , with some  $\Psi > 0$ . The added delay  $\Psi$  leaves margin for future changes, in case the largest internal clock delay exceeds  $D$ .



**Figure 4: (a) An aligned clock distribution network driving IP cores having different internal clock delays: The SoC is not clock aligned. (b) Clock delay insertion compensates for the different internal clock delays, leading to a clock aligned SoC.**

#### 1.2. Global Clock Re-design

The design of clock distribution networks is not straightforward. Ideally, when designing a global clock distribution network, changes in one IP core should not affect other parts of the system. In practice, however, changing an IP core might change its layout, wire capacitances, resistances, etc. Such changes may affect the entire clock distribution network and may require its redesign. Each such redesign involves adding the various delay elements according to the algorithm and performing timing verification of the result, iterating if needed.

The process is repeated whenever any of the system parts is changed. Therefore, proposed changes to the system are not easily accepted. The programmable clock

tuning proposed here eliminates the need for repetitive global clock re-design.

### 1.3. Clock Tuning

We propose a novel and efficient implementation of clock delay insertion. Programmable clock delay lines [1] are inserted at the clock input port of each IP core (Figure 5). Delay values are computed at the very last stage of the design, once the rest of the SoC design is finalized. The delay units are programmed by hard-wiring their control bits.

The most important advantage of the programmable clock delay is the elimination of repeated clock network redesign every time any IP core is changed. Another benefit of this method is the ability to employ an unbalanced global clock distribution network, as explained in the next section.

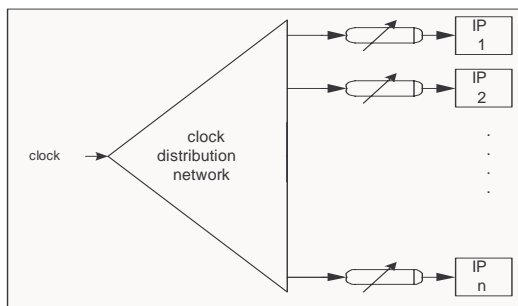


Figure 5: SoC with clock tuning circuits.

## 4. Unbalanced Global Clock Distribution Network

The global clock distribution network of Figure 4b is balanced. This balance is typically achieved at a high cost in terms of design time and effort, as well as chip area and power. In many cases, the clock distribution network must be pre-designed before the other parts of the SoC (because it demands placement and routing resources, which might not be available at a later phase of the design, and due to time-to-market considerations). These complex demands are major obstacles to modular design and are also heavy consumers of time and effort.

Unbalanced clock distribution networks may save a lot of time and effort in modular design. The clock skew of the unbalanced network is compensated for by the same inserted clock delays that also compensate for different internal clock delays inside the IP cores, as in Figure 6. Notice again that the clock tuning process is carried out only once at the end of the design process.

## 5. Clock Delay Tuning Circuit

A tapped delay line has been employed (Figure 7). Two circuits are concatenated, where the first one contains three buffers and can be programmed for 0, 1, 2 or 3 buffer delays, and the second block comprises three

stages of four delay buffers each, providing 0, 4, 8 or 12 buffer delays. The two blocks can thus be programmed for 0—15 buffer delays. Note that even with zero delay buffers the total delay is non-zero due to the taps.

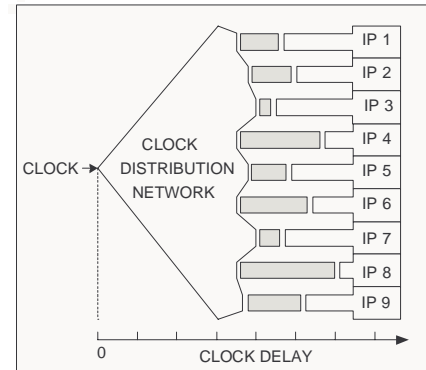


Figure 6: An unbalanced clock distribution network. Inserted clock delays (grey rectangles) compensate for the unbalanced clock network as well as the different internal clock delays, resulting in a clock aligned SoC.

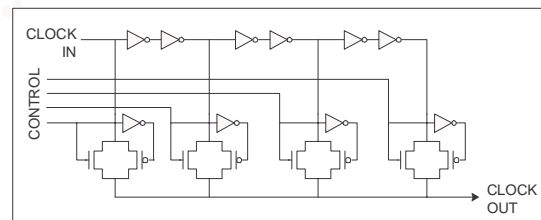


Figure 7: Tapped delay line circuit.

## 6. Experimental Results

The SoC that incorporates the programmable clock delay circuits is a multi-standard demodulator and decoder for terrestrial and cable DTV and analog TV reception (Figure 8 and Table 1). Table 2 describes the final programming of the clock delay units in the ten IP cores of the SoC. The programmable clock delay units were placed in each one of the modules marked in Figure 8.

As explained above, an important advantage of the programmable clock delay circuits is the ability to use an unbalanced global clock distribution network. The programmable clock delay units compensate for the unbalanced distribution network and enable easy clock balancing at the IP level. Figure 9 schematically shows the layout of the unbalanced clock tree of the SoC.

Productivity of the proposed clock tuning method was proven very high. Weeks of iterative clock distribution network design were reduced to several days in which the complete network was designed, tuned and tested. The implementation demanded several design flow changes with standard CAD tools (such as synthesis, scan generation and static timing analysis).

Device Count	12M
Die Size	6.3x7.1mm <sup>2</sup>
Frequency	200MHz
Supply Voltage	1.8/3.3V
Power Dissipation	<1W
Metal Layers	6
Minimum Feature Size	0.18μm
Package	128 pin QFP

**Table 1: SoC Parameters**

Block	Internal Clock Delay (ns)	Added Delay (ns)	Total Clock Delay (ns)
Audio	0.95	0.95	1.9
DSP	1.1	0.75	1.85
GIF	0.7	1.25	1.95
FEC	1.05	0.85	1.9
FT	0.6	1.35	1.95
OFDM	0.85	1.1	1.95
EX	0.4	1.55	1.95
IN	0.8	1.1	1.9
SEQ	0.7	1.2	1.9

**Table 2: Delay programming**

## 7. Conclusion

Two methods for timing integration of IP cores in SoC were discussed. Data delay insertion was shown inefficient in terms of power and area. Clock delay insertion requires frequent redesign of the clock distribution network every time any of the IP cores is changed. This clock network re-design carries a high price in design time and engineering resources.

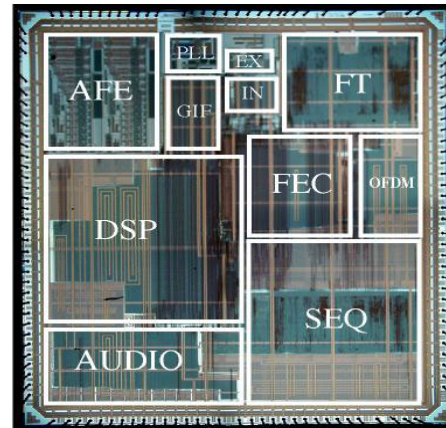
Clock tuning, employing programmable clock delay units, alleviates the need for global clock re-design. In addition, it can compensate for unbalanced clock distribution networks. Design changes are possible at any stage of the design, and they do not incur any burden of clock network re-design. Clock tuning is carried out only once, after all logic and physical design is complete.

In summary, a clock distribution strategy for integrating IP cores in SoCs has been proposed, analyzed and demonstrated in a commercial chip. It improves ease of IP cores reuse by enabling simple clock tuning. Using clock-tuning circuits with programming options enables easy integration of many IP cores into a complete SoC, and eliminates design iterations in the engineering flow. Thus, design effort is reduced, design modularity is improved, and “last minute changes” are enabled.

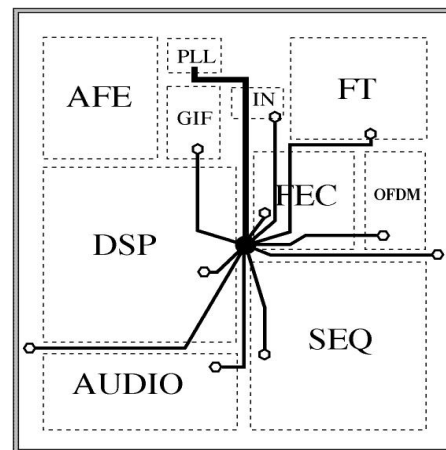
## References

- [1] J.M. Rabaey, Digital Integrated Circuits—a Design Perspective, Prentice Hall Electronics and VLSI series 1996.
- [2] E. G. Friedman, “Clock Distribution Networks in VLSI Circuits and Systems,” New York: IEEE Press, 1995.
- [3] P.J Restle et al., “A Clock Distribution Network for Microprocessors,” IEEE Journal of Solid State Circuits, Vol.36, No.5, May 2001, pp.792-797.

- [4] C.K. Lennard and E. Granata, “The Meta-methods: Managing Design Risk during IP Selection and Integration”, IP 99 Europe pp.285-299.
- [5] H.Mizuno, K.Ishibashi, “A Noise-immune GHz-Clock Distribution Scheme using Synchronous Oscillators,” IEEE International Solid-State Circuits Conference 1988, pp. 404-405.
- [6] D. Sylvester and K. Keutzer, “Rethinking Deep-Submicron Circuit Design,” IEEE Computer magazine, November 1999, pp.25-33.
- [7] R. Ho, K. Mai, and M. Horowitz, “The Future of Wires,” Proceedings of the IEEE, April 2001, pages 490-504.



**Figure 8: Chip micrograph**



**Figure 9: Chip layout and its (unbalanced) clock distribution network. Programmable clock tuning circuits (the small circles) compensate for both the unbalanced network and the internal IP core clock delays.**