

Titre: All digital skew tolerant synchronous interfacing methods for high-
Title: Performance point-to-point communication in DSM SoCs

Auteurs: Syed Rafay Hasan, Normand Bélanger, & Yvon Savaria
Authors:

Date: 2008

Type: Rapport / Report

Référence: Hasan, S. R., Bélanger, N., & Savaria, Y. (2008). All digital skew tolerant
Citation: synchronous interfacing methods for high-Performance point-to-point
communication in DSM SoCs. (Rapport technique n° EPM-RT-2008-10).
<https://publications.polymtl.ca/2632/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/2632/>
PolyPublie URL:

Version: Version officielle de l'éditeur / Published version

Conditions d'utilisation: Tous droits réservés
Terms of Use:

 **Document publié chez l'éditeur officiel**
Document issued by the official publisher

Institution: École Polytechnique de Montréal

Numéro de rapport: EPM-RT-2008-10
Report number:

URL officiel:
Official URL:

Mention légale:
Legal notice:

EPM-RT-2008-10

**ALL DIGITAL SKEW TOLERANT SYNCHRONOUS
INTERFACING METHODS FOR HIGH-PERFORMANCE
POINT-TO-POINT COMMUNICATION IN DSM SOCS**

Syed rafay Hasan, Normand Bélanger, Yvon Savaria
Département de Génie électrique
École Polytechnique de Montréal

Décembre 2008

Poly

©2008
Syed Rafay Hasan, Normand Bélanger, Yvon Savaria
Tous droits réservés

Dépôt légal :
Bibliothèque nationale du Québec, 2008
Bibliothèque nationale du Canada, 2008

EPM-RT-2008-10

All Digital Skew Tolerant Synchronous Interfacing Methods for High-Performance Point-to-Point Communication in DSM SoCs

par : Syed Rafay Hasan, Normand Bélanger, Yvon Savaria
Département de génie électrique
École Polytechnique de Montréal

Toute reproduction de ce document à des fins d'étude personnelle ou de recherche est autorisée à la condition que la citation ci-dessus y soit mentionnée.

Tout autre usage doit faire l'objet d'une autorisation écrite des auteurs. Les demandes peuvent être adressées directement aux auteurs (consulter le bottin sur le site <http://www.polymtl.ca/>) ou par l'entremise de la Bibliothèque :

École Polytechnique de Montréal
Bibliothèque – Service de fourniture de documents
Case postale 6079, Succursale «Centre-Ville»
Montréal (Québec)
Canada H3C 3A7

Téléphone : (514) 340-4846
Télécopie : (514) 340-4026
Courrier électronique : biblio.sfd@courriel.polymtl.ca

Ce rapport technique peut-être repéré par auteur et par titre dans le catalogue de la Bibliothèque :
<http://www.polymtl.ca/biblio/catalogue/>

High-performance clocking of IPs, within a skew budget, is becoming difficult in Deep Sub-Micron technologies. Therefore, the concept of local islands of independent clocks prevails in SoCs, which can communicate using various synchronous and asynchronous interfacing methodologies. However, asynchronous methods are inadequately supported in the context of conventional synchronous design flows, and are also associated with substantial failure rates. By contrast, synchronous interfacing methods often require PLL based synchronization, which requires phase correction that consumes useful bandwidth and mixed signal components. This work proposes a novel and all digital synchronous design method for point-to-point communications, using n interfacing registers and locally delayed clocks with phase adjustments. An overall improvement in skew tolerance of up to $n/2$ to n times, compared to conventional designs, is obtained depending on the context. This is proven analytically. The modules are assumed to have same or integer multiple frequencies. Gate-level simulations are used to validate the analytical results. A proof of concept implementation of the proposed design is demonstrated using a Virtex-II Pro FPGA from Xilinx.

All Digital Skew Tolerant Synchronous Interfacing Methods for High-Performance Point-to-Point Communication in DSM SoCs

Syed Rafay Hasan*, Normand Bélanger, Yvon Savaria

*Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada; Groupe De Recherche EnMicroelectronique École Polytechnique de Montréal Montréal, Canada

*sr_hasan@ece.concordia.ca, normand.belanger@polymtl.ca, Yvon.Savaria@polymtl.ca

Abstract—High-performance clocking of IPs, within a skew budget, is becoming difficult in Deep Sub-Micron technologies. Therefore, the concept of local islands of independent clocks prevails in SoCs, which can communicate using various synchronous and asynchronous interfacing methodologies. However, asynchronous methods are inadequately supported in the context of conventional synchronous design flows, and are also associated with substantial failure rates. By contrast, synchronous interfacing methods often require PLL based synchronization, which requires phase correction that consumes useful bandwidth and mixed signal components. This work proposes a novel and all digital synchronous design method for point-to-point communications, using n interfacing registers and locally delayed clocks with phase adjustments. An overall improvement in skew tolerance of up to $n/2$ to n times, compared to conventional designs, is obtained depending on the context. This is proven analytically. The modules are assumed to have same or integer multiple frequencies. Gate-level simulations are used to validate the analytical results. A proof of concept implementation of the proposed design is demonstrated using a Virtex-II Pro FPGA from Xilinx.

Index Terms — SoC, Multiple Clock Domains, GALS, Synchronizers, Inter-Module Communication

INTRODUCTION

The solutions described in this technical report use predictable timing methods to overcome the problem of inter-module communication. It follows that if a clean synchronous solution, utilizing apriori timing information under a given context, provides an at-par or better performance than more complex or more exotic solutions, then it will be better appreciated by industry. Therefore, the proposed solutions attempt to show that synchronous methods, involving locally delayed clocks with phase adjustments using advance knowledge of worst timing constraints, are viable and effective solutions for interfacing multiple-clock IP modules. These solutions do not require any mixed signal component.

It is assumed, in this work, that there are at least two IP modules, a sending and a receiving IP module, in each system. Sending and receiving modules are referred to as terminating modules in the rest of this paper, unless otherwise stated. Terminating modules are supplied with the clock from the same clock source, but with a certain phase difference between each other. This phase difference is due to skew in the clock distribution network [10][11]. It is further assumed that the maximum skew bound is known. Links in the system are assumed to be subject either to positive or negative skew, and the comprehensive mathematical treatment is performed on all the possible sub-cases of timing constraints.

The proposed design leverages wider bus-width (as wires are usually abundant in on-chip systems) to alleviate the problems in inter-module communication. In [4], a method is devised to know the locations of the terminating modules which experience the most skew in an H-tree CDN (Clock Distribution Network). Utilizing the same concept in this work, interfacing registers are introduced between the two modules which are assumed to be experiencing the worst skew. The terminating modules work at a higher frequency and the interfacing modules run at a slower frequency. By careful adjustment of the interfacing clock (slow) phases, and thorough knowledge of maximum skew bound, higher skew tolerance is achieved.

Based on analytical modeling, simulation results, and practical implementation using a Virtex-II Pro Xilinx FPGA board, this paper shows that under certain design constraints, the utilization of synchronous interfacing methods leads to a peak-to peak skew tolerance of, depending on the context, up to $n/2$ to n

clock cycles, as compared to the best previously reported result of 2 clock cycles [2] tolerance for a state-of-the-art asynchronous FIFO based design. Hardware overhead (as with any interfacing mechanism) and latency are subject to trade offs with this design. A special case of inter-module IP communication is also suggested where IP modules work at frequencies related by integer multiple. Analytical comparison with conventional synchronous designs shows that the suggested solution tolerates up to $n/2$ to n times more skew, where n is the number of interfacing registers in one stage. Simulation results validate these novel analytical findings, and a prototype implementation further confirms the viability of the proposed design.

The rest of this report is organized as follows: Section II states some related work, section III explains the concept of wider bus width; Section IV presents a mathematical analysis the proposed interfacing method, three different cases are discussed; Section V discusses simulation setup and results as well as a comparison with conventional synchronous designs; Section VI explains the prototype implementation using Virtex-II Pro FPGA from Xilinx; Section VII compares the proposed design with state-of-the-art interfacing mechanisms; and Section VIII concludes the paper.

I. RELATED WORK

Fig. 1 shows different possible implementations of inter-module communication. It suggests that SoCs with Multiple Clock Domains (MCD) can be classified into four main classes: point-to-point communication [3][13][14][15]16], point-to-multipoint communication [17], Bus and NOC based [18][19][20][21] design schemes. Circuit designers put a special emphasis on a good point-to-point communication method. Indeed, if a point-to-point communication method is broad enough to encompass all possible variations under different timing constraints, or can at least suggest a good solution under a given set of conditions, then it should lead to a better overall system architecture. Therefore, this work is directed towards proposing new and improved methods for point-to-point communication.

Several methods have already been suggested in the literature to ease the communication among multi-clocked IPs communicating using a point-to-point method. It can be seen in Fig. 1 that some of the state-of-the-art solutions follow a synchronous design style for IP communication, while other designers seek

solutions in asynchronous or partially synchronous and partially asynchronous, such as GALS, design methods. Most of these asynchronous solutions can be broadly divided into two main categories: Pausible clocking [3][13]-[16], and self-timed FIFO-based [2][3][22] designs. Pausible clocking provides a MTBF (Mean Time Between Failures) free communication method between two different clock domains, but it requires separate clock-generating and arbitration/control mechanisms for each clock domain. Furthermore, it has problems associated with clock jitter, throughput, and clock-overflow issues [2][5]. FIFO-based design techniques are free from most of the hardware and clocking issues. But these designs have MTBF problems [2]. In addition, the introduction of asynchronous circuit elements within synchronous designs affects the design uniformity, which, in turn, adversely affects the optimality of the design.

Fig. 1 further shows that synchronous solutions have their own three sub-categories, when the modules are working at the same/similar frequencies. These sub-categories are plesiochronous, mesochronous, and a class of systems where the maximum possible phase offset is known. Plesiochronous schemes are usually found in telecommunication systems, where the two interacting modules typically have independent clocks generated from separate crystal oscillators. Such a scenario is not likely to be present in on-chip or even in on-board systems [6]. Therefore plesiochronous systems are not discussed further. Mesochronous synchronization solutions [23][24] are the most established synchronous design methods for on chip communication among modules that have the same frequency, but different phases. Mesochronous designs require some synchronizing schemes that are implemented by PLL and/or phase detection circuits [23][24]. Furthermore, these solutions require re-synchronization after every ‘suitable’ amount of time [24], which reduces the effective bandwidth of the system. In the third category, in which the maximum phase offset is known, synchronization is conventionally resolved by using the mesochronous synchronizing techniques such as the Self-Test Self-Synchronizing (STSS) method [23] or the Globally Updated Mesochronous (GUM) method [24]. In this work, it is shown that such systems can be treated differently compared to mesochronous designs. One of the solutions proposed in this paper, discussed in section III-A, addresses this category and provides a robust and efficient means of communication. This solution exploits the fact

that in most of the on-chip systems, the worst case timing constraints are known in advance. Hence, it avoids the conventional PLL based synchronization methods, yet achieves very high performance and desirable skew tolerance.

Another branch of synchronous methods, shown in Fig. 1, consists of solutions for the case when the interacting modules are working at different frequencies. The sub-categories are divided on the basis of the frequency ratio between the communicating modules, which may be an integer, fractional or arbitrary rational number. DDS, Direct Digital Synthesis [25] is a popular synchronizing method for communicating among the modules having fractional frequency ratio. But due to its dependency on analog components, DDS has limitations with the maximum attainable frequency. Furthermore, DDS is unable to generate rational number frequencies if the denominator cannot be expressed by 2^n , where n is an integer [26]. Calbaza et al. have proposed a solution, called DDPS (Direct Digital Period Synthesis) [27], for communicating at fractional frequency ratio by using only digital logic. Boyer et al. used a similar concept to produce variable speed processors [28]. Even though DDPS and VPCS provide flexibility in clock multiplication, their control signals require a separate frequency phase detector and low pass filter [29]. The third sub-category, for interacting modules working at different frequencies, is when the frequency ratio is exactly an integer. DDPS and VPCS can generate the required clocks, but producing clocks related by fractional frequency ratios remains a challenge.

A further subdivision of synchronous methods is performed on the knowledge of maximum phase offset, Fig. 1. This is based on the fact that maximum clock phase offset between the clock inputs of the two IPs may change, but the worst case (or maximum) phase offset (skew) can be accurately estimated. Based on this knowledge [Svensson et al.] a FIFO based methodology (without the need of synchronizer) is introduced. In this methodology the clock signal from the sender end is sent to the receiver, called as strobe signal, along with the data. Data is latched as and when data arrives by the strobe signal and the receiver clock that latches the data from a particular clock is delayed enough so that it tolerates maximum skew in the system. Such designs require stringent matching delay property between data and clock [Nguven et al.].

However this design methodology tolerates large skew but the maximum attainable frequency for the terminating modules is a function of skew. Hence the maximum attainable frequency by the processing elements or any other IPs is still dependent on the skew magnitude.

II. CONCEPT OF WIDER BUS WIDTH

It is known that synchronous systems, in certain regions of their Clock Distribution Network (CDN), experience more skew than in others [9] [4]. In [4], a methodology is devised to split the CDN, after identifying the regions most affected by skew. It is also proven that with the introduction of an interfacing mechanism between two split regions, higher system operating frequency can be achieved [4]. Keeping in mind the same splitting mechanism, an all digital synchronous interfacing method is introduced in this section, as shown in Fig. 2. This method provides a point-to-point communication without the need of any asynchronous or mixed signal design. In Fig. 2, two mutually asynchronous IPs are shown. These IPs are running at frequencies F_S (sender frequency) and F_R (receiver frequency). Consequently, the time periods of the sender and receiver modules are denoted by $T_1=1/F_S$ and $T_2 = 1/F_R$, respectively. According to the assumptions stated in section I, F_S and F_R are sourced with the same clock, but may vary in phases. In a conventional design strategy, clock frequency is usually limited due to the skew between the communicating modules. In order to tolerate skew limits, and achieve higher frequency for communicating modules, interfacing modules are introduced. In Fig. 2, the intermediate modules, in the middle, are the interfacing registers.

It is assumed, as generally is the case, that each terminating module can individually (when isolated) run at a faster rate, compared to the fastest rate at which mutually asynchronous terminating modules may communicate with each other.

The proposed solution conceals the skew limitation from the terminating modules through the introduction of interfacing registers, which run at a slower frequency as compared to the fastest clock frequency of any terminating module within the system. Each bank of interfacing registers is represented as

a stage. Therefore, in Fig. 2, interfacing registers 1 to m on the left side belong to stage I, and interfacing registers 1 to n on the right side belong to stage II. This general formulation can represent cases where m and n are related by a fractional ratio. A simpler situation would be the case where that ratio or its inverse is an integer. For example if $m=2n$, the sender would operate at twice the speed of the receiver. However, in the sequel, we will first focus on the situation where $m=n$ and then generalize to integer ratios. Means to support fractional ratios are not investigated further in this paper.

Assume that skew limits the sender and receiver interfacing registers to run at frequencies of f_{S_int} and f_{R_int} , respectively. In the proposed method, each interfacing register can be clocked at a moment appropriate for that register according to the sending and receiving clocks (delayed version of respective clocks are used). Thus no data is lost in the interfacing process. A proposed phase adjustment mechanism of delayed clocks is described in Section III. In order to make the data path transparent to the terminating modules, the designer only has to decide on the number of interfacing registers. This number can be determined using Eq. (1) that expresses the fact that the sustained bandwidth of all stages of the interface must be the same,

$$F_S W_S = W_{S_int} f_{S_int} m = W_{R_int} f_{R_int} n = F_R W_R \quad (1)$$

In this expression, W_{S_int} and W_{R_int} , denote the interfacing bus widths for the sending and receiving ends respectively, while W_S and W_R stand for the bus width of the sending and receiving (terminating) modules, In general, the width W_S and W_{S_int} can be different as well as W_R can be different from W_{R_int} .

The case where all stages of the interface work at the same frequency is analyzed in Section III-A, whereas the case when sending and receiving modules are working at different frequencies, $F_S > F_R$, is considered in Section III-B. The register arrangement for such a case is shown in Fig. 2b. If we have $W_S = W_{S_int}$, F_S is slower than F_R and their ratio is n , as explored in Section III-C, then (1) can be simplified to obtain (2), when it is further assumed that $F_S = f_{S_int}$

$$n F_S W_S = n W_{R_int} f_{R_int} = F_R W_R \quad (2)$$

III. ANALYSIS OF THE PROPOSED INTERFACING METHOD

A. Interfacing two synchronous modules communicating at the same frequency

Assume $F_S = F_R$, $W_S = W_R$ and $W_{S_int} = W_{R_int}$ in (1). Accordingly, $f_{S_int} = f_{R_int}$ and $m = n$. Fig. 3a shows a practical implementation of such a design. Ideally, as will be shown, it allows a throughput that is up to n times higher than a conventional design, under a given skew constraint. Here n is also the number of interfacing registers in each stage.

Referring to Fig. 3a, the interface has two clock inputs, sender clock (CLK1) and receiver clock (CLK2). For analysis purposes, the interface can be divided into three regions. Let these regions be denoted as A, B and C, as shown in Fig. 3a. Region A consists of the sender module (IP TX), and n 1st stage interfacing registers that are clocked by a state machine. These n sending end registers are labeled SR0 to SR($n-1$) are clocked by the Sender State Machine outputs, labeled SSMCLK0 to SSMCLK($n-1$). All registers in region A are clocked, directly or using the sender state machine outputs, derived from the sender clock (CLK1). Region B comprises $2n$ interfacing registers; n from each of the 2 stages. First stage registers on the left side overlap with region A, and second stage registers on the right side overlap with region C. Region C consists of all the modules that are driven by the receiver clock (CLK2) or its derivatives. This includes n second stage interfacing registers that overlap with region B, the receiver module, a state machine, and the multiplexer. The n second stage receiver end interfacing registers are labeled as RR0 to RR($n-1$). Each of these registers is clocked by the shifted outputs of the receiver state machine labeled RSMCLK(0) to RSMCLK($n-1$). Notice the shifts of indices between the first and second stages. This is further explained later.

1) Preliminaries

This section elaborates some of the preliminaries required to better understand the timing constraints of the proposed and conventional designs. For simplicity and without loss of generality, it is assumed that the setup time (t_{SU}), hold time (t_{hold}), and clock to Q (t_{cq}) delay of all the registers in the system, are equal. It is

further assumed that the maximum skew bound is known. Skew or phase discrepancy between the two clock phases are represented by t_{sk} , and the maximum possible drift (jitter) on each side of the nominal edge of the clock is denoted t_j . The skew is assumed to be random but fixed (or changing very slowly) while the drift or jitter component may change rapidly or very rapidly. Therefore, the maximum possible deviation between the two clock domains, due to jitter, is $2t_j$.

It was stated earlier that region A of Fig. 3a is clocked by CLK1, having time period $T_1=1/F_S$, and region C of Fig. 3a is clocked by CLK2, having time period $T_2=1/F_R$. Remember that in this first considered case $F_S = F_R$, so this leads to $T_1 = T_2$. Furthermore, as $f_{S_int} = f_{R_int}$, in the rest of this section, interfacing module frequencies are denoted as f_{int} . As all the modules in region A are clocked by signals derived from the same clock source, therefore, timing constraints in region A of Fig. 3a are deterministic. The same is true of modules in region C. By contrast, region B of Fig. 3a is the most uncertain region of this design, as it combines the two clocks with an unknown phase difference. In such circumstances, there are two kinds of uncertainties to address, namely, the magnitude of the clock phase difference or skew (t_{sk}), and the skew orientation, i.e. whether the skew is positive or negative. In order to tolerate the skew, the proposed design introduces two sets of interfacing registers, sender registers (SR(x)) and receiver registers (RR(x)), as shown in Fig. 3a. These interfacing registers are clocked by different phases of the clock, for example SR(0) is clocked by SSMCLK0 and RR(0) is clocked by RSMCLK(n/2). So each pair of communicating interfacing registers nominally receives clocks separated by a delay of $(n/2)T_1$. This phase adjustment allows almost the same relaxation to both the setup and the hold time constraints. This in turn made the design equally tolerant to both orientations of the skew.

Fig. 3b shows the expected waveform of the proposed design, where the top wave is the sending IP (TX) clock, denoted CLK1. The next three waveforms correspond to Sender State Machine clocks, denoted SSMCLK(x), where x is an integer number from 0 to n-1 and which represents the clock phase. Similarly, the next (5th) waveform represents the receiving IP RX clock, denoted CLK2. The last three waveforms belong to the Receiver State Machine, denoted RSMCLK(x). The numbers below the CLK1 and CLK2

waveforms, in Fig. 3b, indicate the clock sequence number. The time period of the two state machine clocks, SSMCLK(x) and RSMCLK(x) is T , which is nT_1 , with the interfacing frequency $f_{\text{int}} = 1/(nT_1)$. This in turn implies that the terminating modules at both ends are running at frequency nf_{int} , as shown in Fig. 3a. It can further be noticed that any two successive interfacing registers in region A and in region C respectively, RR0 and RR1 for example (or SR0 and SR1), receive clock signals separated by a T_1 time interval (or T_2 as $T_1 = T_2$). Moreover, in Fig. 3b, t_{SK} represents a negative skew between CLK1 and CLK2. Generally, the skew is defined as the time difference between the nominal edges of the sending and receiving clocks, further discussion on the skew orientation is deferred to subsequent sub-sections. The three arrows linking the rising edges represent the expected drift (jitter) in the clock. Arcs X, Y and Z relate to one complete data-path from the sender module to the receiver module. Sender and receiver module are shown as IP TX and IP RX in Fig. 3a, respectively. In order to better understand the proposed solution and its advantages, the following mathematical analysis is performed. As a first step in the analysis, the following inequalities for the different regions shown in Fig. 3a are obtained. In the following analysis it is observed that sender and receiver state machines are identical in their operations. In the implementation reported in section IV the state machine is implemented as a one hot counter. Thus these state machines produce output phases with some delay with respect to their respective clocks (CLK1 and CLK2). Nominally, these delays are equal and are denoted t_{SM} . Region A can operate according to one of the following two sets of timing constraints, the first set assumes one sender clock cycle (CLK1) latency, while the second set assumes zero clock cycle latency from the sender module (IP TX) to the first stage of the interfacing register, SR(x),

$$t_{\text{cq}} + t_{\text{SU}} < T_1 + t_{\text{SM}} \quad (3) \quad \& \quad t_{\text{cq}} > t_{\text{hold}} + t_{\text{SM}} \quad (4)$$

or

$$t_{\text{cq}} + t_{\text{SU}} < t_{\text{SM}} \quad (3a) \quad \& \quad T_1 > t_{\text{SM}} + t_{\text{hold}} - t_{\text{cq}} \quad (4a)$$

where, all the timing parameters in the above inequalities correspond to Fig. 3a. t_{cq} represent clock to Q delay of register IPTX. t_{SU} and t_{hold} are the setup and hold time of SR(x) registers, T_1 is the time period of

CLK1 and t_{SM} is the delay in producing the output phases of the sender state machine.

In region C of Fig. 3a, getting zero cycle latency between RR(x) and IP RX is not realistic. Indeed, the output signals from RR(x) go to IP RX through a large Multiplexer and the RR(x) registers are clocked by control signals arriving a t_{SM} delay after IP RX receives its clock. Thus we only derive equations for one cycle latency in Region C, which leads to inequalities (5) and (6):

$$\max(t_{SM} + t_{MUX_D} + t_{cq} + t_{SU}, t_{SEL} + t_{MUX_S} + t_{SU}) < T_1 \quad (5)$$

$$\max(t_{SM} + t_{cq} + t_{MUX_D}, t_{SEL} + t_{MUX_S}) > t_{hold} \quad (6)$$

where in the above inequalities, t_{MUX_D} is the delay of the multiplexer from the data input to output, and t_{MUX_S} is the multiplexer delay from the selector signal to the output. t_{SEL} is the delay of the selector signal for the multiplexer, which is implemented as a counter. t_{SM} is control logic delay of the receiver state machine, t_{cq} is the clock to Q delay of the RR(x) registers, t_{SU} and t_{hold} are the setup and hold time of IP RX register.

2) Timing constraint in a conventional design, similar to region B

As explained earlier, analysis for region B is the most critical as it involves two different clocks which feature some phase difference. In order to better illustrate the timing constraint in this region, first an analysis assuming a conventional design is performed. Foundations of this analysis can be found in the literature [6][11], but the complete set of constraints devised in this analysis are either implicit or not explained in other references. For example, the introduction of delay insertion, as denoted by Δ in the following analysis, and its upper and lower limits are usually not discussed in the literature. Also, the notations vary in different references. Hence, it is hard to avoid ambiguity while analyzing the complete set of equations of the proposed designs (developed in section III-A-3) with the equations provided in different references. Moreover, the following analysis allows establishing the notation in a complete self contained derivation, and leverages as a basis to obtain timing constraint relation of the proposed schemes.

Hardware implementation of a conventional design is shown in Fig. 4a. CON_CLK1 and CON_CLK2, in Fig. 4a, are derived from the same clock tree network. CON_CLK1 has the time period of T_1 same as

CLK1 in Fig. 3a. Similarly CON_CLK2 has the same time period T_2 as CLK2 in Fig.3a.

Due to the non-idealities in the clock tree network, as stated earlier, there may be some phase discrepancies between the CON_CLK1 and CON_CLK2. There are two possible cases either CON_CLK2 clocks IP RX, in Fig. 4a, arrives before CON_CLK1 clocks IP RX or after. Depending on the direction of data flow, these phase discrepancies are treated differently.

The two terminating modules in Fig. 4a, IP TX and IP RX, may act as a sender or receiver at a time. If IP TX is the sending module and IP RX is the receiver module and CON_CLK2 clocks IP RX, in Fig. 4a, after CON_CLK1 clocks IP RX, then such a phase discrepancy is called positive skew, t_{sk+} . Similarly, if CON_CLK2 clocks IP RX, in Fig. 4a, before CON_CLK1 clocks IP RX, then the phase discrepancy is called negative skew, t_{sk-} . The expected waveforms for t_{sk+} and t_{sk-} are shown in Fig. 4b. and Fig. 4c, respectively. In these figures, it is assumed that the two modules of this conventional design are communicating with a latency of one clock cycle.

To generalize the analysis, the phase discrepancy between the two clock modules in Fig. 4a can always be represented as t_{sk} defined as follows

$$t_{sk} = t_{sk+} - t_{sk-} \quad (7)$$

$$\text{where } t_{sk} = t_{sk+} \text{ when } t_{sk} > 0$$

$$\text{and } t_{sk} = - t_{sk-} \text{ when } t_{sk} < 0$$

In a system subject to phase discrepancy, setup time and hold time constraints of the registers require further careful analysis. These timing constraints are treated one by one in the following analysis. In all of the subsequent analysis, the latency is assumed to be of one clock cycle, unless otherwise stated. The following inequality shows how the setup time affects the time period of CON_CLK1 of IP TX as shown in Fig. 4a. Provided the data is traversing between IP TX and IP RX, irrespective of the sign of the skew, the setup time constraint can be expressed as follows,

$$T_1 + t_{sk} > t_{cq} + 2t_j + t_{SU} \quad (8)$$

where, T_1 is the time period of CON_CLK1 and CON_CLK2, t_{cq} is the clock to Q delay for the IP TX and

t_{SU} is the setup time for IP RX. The term $2t_j$, in the above inequality (8), shows the worst case, i.e. CON_CLK2 rises t_j duration before the nominal edge, and CON_CLK1 rises t_j duration after the nominal edge. The above inequality shows that if $t_{sk} > 0$, i.e. there is a positive skew (t_{sk+}), then it is advantageous to the system as a shorter T_1 can be acceptable, this is mathematically shown in inequality (9). Likewise, when $t_{sk} < 0$ in inequality (8), i.e. there is a negative skew (t_{sk-}), then T_1 becomes longer and in turn, it makes the system slower, this is illustrated by inequality (10).

$$T_1 > t_{cq} + 2t_j + t_{SU} - t_{sk+} \quad (t_{sk} > 0, \text{ waveform shown in Fig. 4b}) \quad (9)$$

$$T_1 > t_{cq} + 2t_j + t_{SU} + t_{sk-} \quad (t_{sk} < 0, \text{ waveform shown in Fig. 4c}) \quad (10)$$

The following analysis considers the other timing-constraint with respect to the hold time. Conventionally, the hold time violation for the design shown in Fig. 4a can be avoided if inequality (11) holds, with the same assumption that data is traversing from IP TX to IP RX,

$$t_{cq} > t_{sk} + 2t_j + t_{hold} \quad (11)$$

where, t_{cq} is the clock to Q delay for IP TX, and t_{hold} is the hold time for IP RX. Again $2t_j$ is associated with the worst case assumption that CON_CLK1 rises t_j duration before the nominal edge and CON_CLK2 rises t_j after the nominal edge. Mathematically, utilizing inequality (11), hold time constraints for the two possible cases of skew can be written as follows,

$$t_{cq} > t_{sk+} + 2t_j + t_{hold} \quad (t_{sk} > 0, \text{ waveform shown in Fig. 4b}) \quad (12)$$

$$t_{cq} > -t_{sk-} + 2t_j + t_{hold} \quad (t_{sk} < 0, \text{ waveform shown in Fig. 4c}) \quad (13)$$

If clock to Q delay of IP TX does not meet the requirements of the hold time constraints in the above inequalities, then additional delay insertion, Δ , is required to cope with the phase discrepancy, t_{sk} . Hence the above inequalities are modified as follows,

$$t_{cq} + \Delta > t_{sk+} + 2t_j + t_{hold} \quad (t_{sk} > 0, \text{ waveform shown in Fig. 4b}) \quad (14)$$

$$t_{cq} + \Delta > -t_{sk-} + 2t_j + t_{hold} \quad (t_{sk} < 0, \text{ waveform shown in Fig. 4c}) \quad (15)$$

The insertion of delay, Δ , is not limited to cope with phase discrepancy alone. In DSM processing technologies, due to long interconnects between interfacing modules, additional buffers are inserted in the

interconnects, which in turn increases the phase discrepancy between the two modules, Δ is used to manage such problems as well.

However, insertion of delays has its limitations. For example for a system with maximum latency of one clock cycle, the delay should be inserted in such a manner that it does not affect the subsequent clock cycles. Such a limitation can be avoided by observing the following constraint:

$$t_{cq} + \Delta < T_1 - 2t_j - t_{hold} + t_{sk}$$

$$\Delta < T_1 + t_{sk} - 2t_j - t_{SU} - t_{cq} \quad (16)$$

where, t_{cq} is the clock to Q delay of IP TX, t_{SU} is the setup time of the IP RX, and $2t_j$ shows the worst case timing jitter.

Relations (7) to (16) are general guidelines for the timing constraints of conventional designs. Depending upon the context of the system, different set of relations is valid. Following, different contexts are discussed, it is assumed that all the systems use different forward and backward channels to pass the data.

System is only subjected to unidirectional communication with positive skew, i.e. $t_{sk} > 0$ or $t_{sk} = t_{sk+}$.
In such systems, inequality (9), holds for setup time constraint and inequalities (12) and (14) holds for hold time constraint. Inequality (14) and (16) leads to the following relationship for the delay insertion,

$$t_{sk+} + 2t_j + t_{hold} - t_{cq} < \Delta < T_1 + t_{sk+} - 2t_j - t_{SU} - t_{cq} \quad (17)$$

System is only subject to unidirectional communication with negative skew, i.e. $t_{sk} < 0$ or $t_{sk} = -t_{sk-}$.

Here, inequality (10) defines the setup time limit and inequality (13) is valid for hold time constraint. As the only possible orientation of skew is negative, therefore, there is no requirement of any delay insertion.

System is subject to bi-oriented (bi-directional) skew, the maximum value of t_{sk+} and t_{sk-} is known.
When the system communicates in either direction then skew orientation has converse relationship for each pair of communication. So under this context, the designer has to choose the worst case design consideration. This leads to the choice of following worst case inequalities. Inequality (10) is valid for the worst case setup time constraint, hence indicates the most restricted limit on time period, T_1 . Similarly, inequality (12) is valid for the worst case hold time constraint when no delay is inserted. Inequality (17)

leads to the worst case relation for the magnitude of delay insertion.

System is subject to bi-oriented (bi-directional) unknown skew orientation, the maximum value of t_{sk+} and t_{sk-} is known. This is a context in which the digital system designer is provided only with the maximum phase discrepancy and no knowledge of skew orientation. In such a case, the choice of the magnitude of the inserted Δ delay is of real importance. Indeed, increasing the magnitude of Δ is advantageous for one skew orientation (t_{sk+}), but concurrently it is disadvantageous for the other skew orientation (t_{sk-}). The value of Δ is obtained using following set of rules. Firstly, the minimum value of Δ is obtained using inequality (14). Secondly, this minimum value of Δ is checked for the maximum bound using inequality (16) for the negative skew. These two steps can be summarized mathematically as follows, where t_{sk} on either side of the relation has the same magnitude but differs in their orientation for a given context of communication,

$$t_{sk+} + 2t_j + t_{hold} - t_{cq} < \Delta < T_1 - t_{sk-} - 2t_j - t_{SU} - t_{cq} \quad (18)$$

Finally, the setup time constraint is checked for this minimum value of Δ using inequality (10-A), which in turn advise the designer about any requirement for incrementing the time period T_1 to accommodate the given t_{sk} . Inequality (10-A) is a modified form of inequality (10) as shown below,

$$T_1 > t_{cq} + \Delta + 2t_j + t_{SU} + t_{sk-} \quad (10-A)$$

3) *Timing Analysis in region B of the Proposed Design*

In the above inequalities (9) and (10), and later from inequalities (16) to (18), it is shown that in conventional designs, the skew tolerance is proportional to the time period of the clock, which is denoted by T_1 in the above relations. In the proposed design, this limit on skew tolerance is relaxed by introducing a suitable number of interfacing registers activated by a version of the clock with an appropriate phase. The resulting skew tolerance becomes proportional to an integer multiple of T_1 . This integer number may range from, depending upon the skew conditions, the total number of interfacing registers n to half of the number of interfacing registers $n/2$ (assuming n is even, this assumption can be relaxed at the expense of more elaborate analysis).

In the rest of this section, a timing analysis of the proposed design is performed that encompasses several possible design variations. Fig. 3b shows the expected waveform of the proposed design. To simplify the following derivations, it is assumed, that the number of interfacing registers in each stage is even. All the interfacing registers, SR(x) or RR(x) in Fig. 3a, are clocked once, in a successive manner by the SSMCLK(x) or RSMCLK(x) signal, within duration of nT_1 . At time nT_1 , the state machine sends out the reference signal again, and the cycle repeats itself.

To exploit the benefit of interfacing registers, clocking order of sending and receiving registers are kept different, as shown in Fig. 3. For example, the SR1 interfacing register at the sender end or Ist stage is clocked by SSMCLK1. SR1 is connected with RR1 at the receiving end or IInd stage of interfacing registers and it is clocked by RSMCLK((n/2)+1), where n is an even number, as stated earlier. Therefore, a phase difference of $n/2T_1$ is created between the SR(x) and RR(x). This in turn makes the digital system more tolerant to the phase discrepancy, while trading off a latency of n/2 times the shorter clock period, T_1 .

Arcs X, Y, and Z in Fig. 3b illustrate the propagation of Data 'A'. Initially, Data 'A' transfer is initiated by sender clock CLK1 on the sender register, IP TX. Arc X shows that Data 'A' is clocked by SSMCLK1 at the sender end interfacing register, SR1. Arc Y shows the data path between the interfacing registers of stage I (sending end), SR1 in this case, and stage II (receiving end), RR1 in this case. This arc also shows that the RR1 interfacing register, is clocked by RSMCLK(n/2+1). Arc Z shows the data path between the RR1 interfacing register and the receiver register, IP RX. The receiver end register, IP RX, safely receives Data 'A' at the n/2+2 clock edge, as shown in Fig. 3b.

In order to understand the effect of setup and hold time constraints on time period and delay insertion requirements in the proposed design, an analysis is performed in line with the conventional design analysis, carried out in section III-A-1. Fig. 3b shows the expected waveform of the proposed design when the system has zero skew, $t_{sk} = 0$. For illustration purposes signals associated with arc X and arc Y are drawn separately in Fig. 4d. Arcs X and Y show that the data going from IP TX to IP RX. If the rising edge of the signal RSMCLK(n/2+1) shifts towards left from the position shown in Fig. 4d, then it indicates a negative

skew between CLK1 and CLK2, and the converse shift indicates positive skew between CLK1 and CLK2. As described earlier, $t_{sk} < 0$ is a negative skew and $t_{sk} > 0$ is a positive skew.

In order to avoid any data loss, phase discrepancy, t_{sk} , has to be within one large clock period, nT_1 . As the RSMCLK(x) is $(n/2)T_1$ apart from SSMCLK(x) at $t_{sk} = 0$, therefore timing constraints of such designs can be split into two cases. The following discussion first neglects set up and hold time which will be considered later. Let us first consider the case where $|t_{sk}| < (n/2)T_1$, with reference to Fig. 4d, in this case the data sent on the ‘First’ rising edge of SSMCLK1 signal has to be latched by ‘First’ rising edge of RSMCLK(n/2+1). For this case, the latency in terms of the slow (nT_1) clock period is zero. Other cases considered are when $(n/2)T_1 < |t_{sk}| < nT_1$. Of course, here timing relations of the two interfacing registers depends upon the skew orientation. For positive skew, the first rising edge of RSMCLK(n/2+1) moves towards right and shifts beyond the boundary point C shown in Fig. 4d. So the data from the ‘First’ rising edge of SSMCLK1 is latched beyond boundary point C by the ‘First’ following rising edge of RSMCLK(n/2+1). For negative skew, the ‘First’ rising edge of RSMCLK(n/2+1) move towards left with respect to the position shown in Fig. 4d and shifts beyond the boundary point A. Therefore, the data sent by the ‘First’ rising edge of SSMCLK1 is latched by following rising edge of RSMCLK(n/2+1), which is one cycle later, with the system exhibiting a one nT_1 cycle latency. Off course, this ‘Second’ rising edge of RSMCLK(n/2+1) is a nT_1 duration away from the ‘First’ nominal rising edge of RSMCLK(n/2+1).

For the first case, when $|t_{sk}| < (n/2)T_1$, setup timing constraints of the proposed design are modeled by the following inequality, (see inequality(8) where T_1 has been replaced by $n/2T_1$),

$$(n/2) T_1 + t_{sk} > t_{cq} + t_{su} + 2t_j \quad (19)$$

where t_{cq} is clock to Q delay of the SR1 register in Fig. 3a, and t_{su} is the set up time of the RR1 register. $2t_j$ is the worst case jitter time duration. This indicates that the clock edge defining the data path that arrives first rises t_j before its nominal time, while the clock edge that arrives second, due to phase discrepancy, rises t_j after its nominal time. Inequality (19) can be written as follows for the two different skew orientations, similar to inequality (9) and (10),

$$(n/2) T_1 > t_{cq} + t_{SU} + 2t_j - t_{sk+} \quad (\text{for } t_{sk} > 0) \quad (20)$$

$$(n/2) T_1 > t_{cq} + t_{SU} + 2t_j + t_{sk-} \quad (\text{for } t_{sk} < 0) \quad (21)$$

For the second case, when $(n/2)T_1 < |t_{sk}| < nT_1$, then the setup time constraint leads to the following generalized inequality, which is independent of skew orientation, with the assumption that latency is one clock cycle in terms of the slow clock period and t_{sk} may have either orientation,

$$nT_1 + t_{sk} > t_{cq} + t_{SU} + 2t_j \quad (22)$$

where, the definition of t_{cq} , t_{SU} and $2t_j$ are the same as in inequality (19). In this case t_{sk} may be formulated as follows,

$$|t_{sk}| = (n/2)T_1 + \delta \quad (23)$$

where, δ shows the portion of t_{sk} which is more than $(n/2)T_1$. Substituting (23) into (22),

$$nT_1 + [\delta + (n/2)T_1] > t_{cq} + t_{SU} + 2t_j \quad (24)$$

This inequality can be written as follows when $t_{sk} > 0$,

$$(3/2)nT_1 > t_{cq} + t_{SU} + 2t_j - \delta \quad (25) \quad (\text{for } t_{sk} > 0)$$

In inequality (25), t_{cq} represents the clock to Q delay of the first rising edge of SSMCLK1, shown in Fig. 4d, while t_{SU} is the setup time of 'First' rising edge of RSMCLK($n/2 + 1$), shown in Fig. 4d. When $|t_{sk}| > n/2T_1$, RSMCLK($n/2+1$) shifts to the right beyond the boundary point C, shown in Fig. 4d, and this may cause an ambiguity condition with respect to latching data with the first or the second rising edge of SSMCLK1. This possible system malfunction is always avoided in worst case designs, because the worst case setup timing constraint condition takes care of this ambiguity condition. This condition is derived by rewriting inequality (24) for $t_{sk} < 0$ as follows,

$$nT_1 > t_{cq} + t_{SU} + 2t_j + \delta \quad (26) \quad (\text{for } t_{sk} < 0)$$

In inequality (26) definitions of t_{cq} and $2t_j$ are same as above. But t_{SU} is the setup time of the second rising edge of RSMCLK($n/2+1$). This edge may be visualized in Fig. 4d when RSMCLK($n/2+1$) is considered to have moved towards left farther than $n/2T_1$. At this instant the next rising edge of RSMCLK($n/2+1$) not shown in Fig. 4d appears before boundary point C.

In the same way as inequalities for setup time constraints are obtained, the hold time constraint is also divided into two cases. When $|t_{sk}| < (n/2)T_1$, the latency in terms of slow clock period is zero clock cycle, hence there is no possibility of hold time violation. For the second case, when $(n/2)T_1 < |t_{sk}| < nT_1$, following inequality holds (similar to inequality (11), with the addition of Δ (which can be equated to 0 if no delay insertion is required) to generalize the inequality, and $n/2T_1$ because $(n/2)T_1 < |t_{sk}| < nT_1$),

$$n/2 T_1 + t_{cq} + \Delta > t_{sk} + t_{hold} + 2t_j \quad (27)$$

In inequality (27), t_{cq} belongs to the clock to Q delay of SR1, while t_{hold} is the hold time of the RR1. For positive skew $t_{sk} > 0$, the inequality (27) can be written as follows,

$$n/2 T_1 + t_{cq} + \Delta > t_{sk+} + t_{hold} + 2t_j \quad (28) \text{ (for } t_{sk} > 0\text{),}$$

substituting (23) into (28) leads to (29) which is same as inequality (14), where t_{sk+} is replaced by δ ,

$$t_{cq} + \Delta > \delta + t_{hold} + 2t_j \quad (29)$$

here, t_{cq} is with respect to the ‘Second’ rising edge of SSMCLK1 in Fig. 4d. t_{hold} is the hold time of the RR1, when the ‘First’ rising edge of RSMCLK($n/2+1$) has moved further right to the ‘Second’ rising edge of SSMCLK1, i.e. beyond the boundary point C in Fig. 4d. The definition of $2t_j$ remains the same. This inequality solves the problem associated with the ambiguity conditions as well.

Likewise, for negative skew, $t_{sk} < 0$, the inequality (27) becomes,

$$n/2 T_1 + t_{cq} + \Delta > -t_{sk-} + t_{hold} + 2t_j \quad (30) \text{ (for } t_{sk} < 0\text{)}$$

substituting (23) into (30), and knowing that the RSMCLK($n/2+1$) signal repeats itself after nT_1 duration, the above inequality can be written as inequality (31), which is similar to inequality (15), where $-t_{sk-}$ is replaced by $-\delta$,

$$t_{cq} + \Delta > -\delta + t_{hold} + 2t_j \quad (31)$$

here, t_{cq} is the clock to Q delay of the Second rising edge of SSMCLK1 in Fig. 4d. While t_{hold} is with respect to the second rising edge of RSMCLK($n/2+1$). This edge may be imagined to appear from the shown hypothetical point C in Fig. 4d and move towards left with the increment of δ . $2t_j$ still has the same definition.

In order to maintain the maximum latency of one clock cycle, there is an upper limit on delay insertion. The following inequality shows the maximum value of this delay insertion,

$$t_{cq} + \Delta < n/2 T_1 + t_{hold} - 2t_j$$

$$\Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq} \quad (32) \text{ (for } |tsk| > n/2 T_1 \text{)}$$

where t_{cq} is the clock to Q delay of SR1 with respect to Second rising edge of SSMCLK1 in Fig. 4d, and similarly t_{hold} is the hold time for the RR1 register. The $2t_j$ term reflects the case where the rising edge of SSMCLK1 is t_j after its nominal time while the clock edge of RSMCLK is t_j before its nominal time.

In DSM processing technologies due to long interconnects between interfacing modules, additional buffers are inserted in the interconnects, which in turn increases the phase discrepancy between the two modules more than the maximum $|t_{sk}|$ specified in the analysis by the relation $(n/2)T_1 < |t_{sk}| < nT_1$. The proposed design may be extended to support such interconnect delays by leveraging wave pipelining concept to achieve high performance, but it is beyond the scope of this paper.

In line with the conventional design analysis performed in section III-A-2, the same contexts are investigated here. So that a performance comparison can be made between the conventional design and the proposed design. These investigated contexts are as follows,

System only subject to unidirectional communication with positive skew, i.e. $t_{sk} > 0$ or $t_{sk} = t_{sk+}$. In such systems, inequality (20) holds for setup time constraint when $t_{sk} < (n/2) T_1$. When $t_{sk} > (n/2)T_1$, then inequality (25) follows the setup time constraint. Similarly, inequalities (28) and (29) set a limit for hold time constraint (hold time constraint is only valid for $tsk > (n/2)T_1$, as explained in section III-A-2). The delay insertion limit is obtained from inequalities (29) and (32),

$$\delta + t_{hold} + 2t_j - t_{cq} < \Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq} \quad (33)$$

System only subject to unidirectional communication with negative skew, i.e. $t_{sk} < 0$ or $t_{sk} = -t_{sk-}$. Here, inequality (21) defines the setup time constraint when $t_{sk} > -(n/2)T_1$ (or $|t_{sk}| < (n/2)T_1$). While for $t_{sk} < -(n/2)T_1$ (or $|t_{sk}| > (n/2)T_1$), inequality (26) follows the setup time constraint. Inequalities (30) and (31) are valid for hold time constraint. Whereas the delay insertion limit is obtained from inequalities (31) and (32),

$$-\delta + t_{\text{hold}} + 2t_j - t_{\text{cq}} < \Delta < n/2 T_1 + t_{\text{hold}} - 2t_j - t_{\text{cq}} \quad (34)$$

System subject to bi-directional communication and the orientation of the skew is known, (the maximum values of phase discrepancy is t_{sk}). This analysis is must be divided into two cases, when $|t_{\text{sk}}| < (n/2)T_1$, there is no possibility of hold time violation and hence only the worst case inequality of setup time constraint, inequality (21), is sufficient for the whole system.

When this bidirectional system is subjected to $(n/2)T_1 < |t_{\text{sk}}| < nT_1$, then both the hold time constraint and the setup time constraint have to be considered. Inequality (26) observes the worst case setup time constraint, whereas inequality (28) and (29) define the worst case hold time constraint. The worst case delay insertion value can be within the bounds shown in inequality (33).

System subject to bi-directional communication and the skew orientation is not known, (the maximum values are $t_{\text{sk}+}$ and $t_{\text{sk}-}$). As the maximum values of phase discrepancy is known, therefore it is known apriori that whether $|t_{\text{sk}}|$ is more than $(n/2)T_1$ or not. If $|t_{\text{sk}}| < (n/2)T_1$, then, as analyzed in the case of bi-directional communication, inequality (21) describes the time period of the system and there is no possibility of hold time violation. When the system is subject to $|t_{\text{sk}}| > (n/2)T_1$, then, following an analysis similar to that performed for the conventional design under the identical condition, a set of rules are devised to find the maximum value of Δ that it can tolerate. Firstly, inequality (29) imposes the minimum value of Δ . Secondly, this minimum value of Δ is checked with inequality (32) to find out whether Δ , for a given set of conditions and T_1 , is within the bounds or not. Finally, the setup time constraint is checked for this minimum value of Δ using inequality (26-A), which in turn advise the designer about any requirement for incrementing the time period T_1 to accommodate certain t_{sk} . Inequality (26-A) is a modified form of inequality (26) provided below,

$$nT_1 > t_{\text{cq}} + \Delta + t_{\text{SU}} + 2t_j + \delta \quad (26-A)$$

The above inequalities can be compared with their corresponding inequalities in the conventional design analysis, to estimate the benefits of the proposed design. This comparison is tabulated in Table I and II for uni-directional and bidirectional communication respectively. From second column of Table I, it is

observed that inequality (21) and (26) of the proposed design and inequality (10) of the conventional design indicate worst case setup time constraint. Comparing these inequalities, it is found that the skew tolerance is increased in the proposed design in proportion to the number of interfacing registers n , up to $n/2$ times for zero cycle latency and up to n times for one cycle latency.

Similarly, from the first column of Table I, it is observed that relation (33) for the proposed design and relation (17) for the conventional design show that the limit of delay insertion, to cope with hold time constraint, is increased in the proposed design. This increment is proportional to $n/2$.

Column1 of Table II leads to exactly the same result for setup and hold time violations as obtained by analyzing Table I. On the other hand column2 of Table II, which shows the results of bidirectional communication for unknown skew orientation, leads to similar results but via different inequalities.

Another aspect on which the proposed design fares better is the requirement of delay insertion. Inequality (12) indicates that in the conventional design, delay insertion is required when,

$$t_{sk+} > t_{cq} - 2t_j - t_{hold} \quad (35)$$

whereas, in the proposed design, delay insertion is required only when $t_{sk} > (n/2)T_1$. It is known that in most digital systems to meet setup time constraint, $t_{cq} \ll T_1$, as can be seen in inequality (10) for example. Therefore, it is concluded that there is a substantial relaxation in timing budget to enable delay insertion with the proposed system. This relaxation is also proportional to $n/2$.

The time period of the proposed design can be estimated by using all the relevant relations, if $|t_{sk}| < (n/2)T_1$,

$$T_1 = \max\left(\frac{(t_{cq} + \Delta + t_{SU} + 2t_j + t_{sk})}{n}, \max(t_{SM} + t_{MUX_D} + t_{cq} + t_{SU}, t_{SEL} + t_{MUX_S} + t_{SU}), (t_{cq} + t_{SU} - t_{SM}), (t_{SM} + t_{hold} - t_{cq})\right) \quad (36)$$

$$\text{and if } nT_1 < |t_{sk}| < (n/2)T_1$$

$$T_1 = \max\left(\frac{(t_{cq} + \Delta + t_{SU} + 2t_j + t_{sk})}{n}, \frac{2}{n}(\Delta + t_{hold} + 2t_j + t_{cq}), \max(t_{SM} + t_{MUX_D} + t_{cq} + t_{SU}, t_{SEL} + t_{MUX_S} + t_{SU}), (t_{cq} + t_{SU} - t_{SM}), (t_{SM} + t_{hold} - t_{cq})\right) \quad (37)$$

where, the first term is the generalized form of inequalities (26), (26-A) and (21), second, third, fourth and

fifth term in (37) are directly taken from inequalities (32), (5), (3) and (4a), respectively. This equation can be compared with the similar expression for conventional design which is obtained from inequality (10-A) and (16) and shown as follows,

$$T_1 = \max (t_{cq} + \Delta + 2t_j + t_{SU} + t_{sk}, \Delta + 2t_j + t_{hold} + t_{cq}) \quad (38)$$

The 1st term of equations (36), (37) and (38) contain t_{sk} and it is likely in DSM designs that this term would dominate. Hence, comparison of these first terms shows that the proposed design may allow using local frequencies up to n times faster while meeting some predefined timing constraints.

B. Faster sender module communicating with slower receiver module (frequencies are integer multiples)

The second case study is performed under the condition in which two modules are working at different frequencies, and the ratio of their frequencies is an integer. Such a case has two sub-classes: communication between Fast-to-Slow modules, denoted as F-to-S systems or designs, and communication between Slow-to-Fast modules, denoted as S-to-F systems or designs, as shown in Fig. 1. S-to-F and F-to-S variants of the proposed design may also act as parallel to serial and serial to parallel data conversion, respectively. These kinds of circuits are useful in the design of serializers and deserializers (SERDES).

In this subsection, communication between Fast-to-Slow modules, F-to-S system, is first studied. The same study can be applied for the reverse case to obtain S-to-F systems. Two variants of this method are proposed in the subsequent sub-sections. The first method proposes complete removal of Region C from Fig. 3a as shown in Fig. 6a. This method assumes that n IP RX registers, shown as RR(x) in Fig. 6a, are clocked at different phases to tolerate the skew. This proposed method is elaborated in sub-section III-B-1 and called as F-to-S system–I. The second method proposes a change in the width of IP RX, along with an intermediate register module shown as RRIM in Fig. 7a. This proposed method is elaborated further in the sub-section III-B-2, and called as F-to-S system–II.

1) *Timing constraints in conventional and proposed F-to-S system-I under Different contexts*

In conventional design schemes, if the sender module is n times faster than the receiver module, a F-to-S system, then the receiving module has to be designed such that it latches n data items at each receiving end (slower) clock cycle edge. Mathematically the frequency relationship between the terminating modules can be represented as: $F_S = nF_R$ and $f_{int} = F_R$, and it implies that $F_S = 1/T_1$, $F_R = 1/T_2$ and $n = T_2/T_1$. The hardware implementation and expected waveform of conventionally implemented F-to-S systems is shown in Fig. 5a and Fig. 5b, respectively. A detailed analysis similar to the one presented in section III-A, was applied on conventional F-to-S systems. The timing constraints were obtained by assuming the data is traversing between IP TX to IP RX(X) via the Demultiplexer in Fig. 5a. The inequalities obtained for this design are summarized in table III. For simplicity, it is assumed in inequalities 39-45 in Table III that the Demultiplexer has the same delay in both selector to data output and data input to data output paths. Table III also shows the corresponding inequality obtained in section III-A to illustrate the similarity of these relations.

The proposed interfacing mechanism and expected waveforms for F-to-S systems-I are respectively shown in Fig. 6a and Fig. 6b. A careful examination shows the similarity between Fig. 3a and Fig. 6a. The region A of Fig. 6a is almost identical to region A in Fig. 3a, with the exception of the introduction of the Demultiplexer and the corresponding Selector units in Fig. 6a. Region B is exactly the same in both cases. The clock phases applied to the interfacing registers are identical to the case described for Fig. 3a in section III-A-1.

The timing constraints on region A of the proposed F-to-S system-I can be expressed by replacing t_{cq} with $t_{cq} + t_{DMUX}$ in inequalities 3, 3a, 4 and 4a. A comparison of conventional and the proposed F-to-S system-I design under different contexts are shown in Table IV and V. These comparisons lead to the same conclusion as obtained in section III-A. Skew tolerance and relaxation in delay insertion requirement can be seen by comparing Conventional Design row of Table V with the two Proposed Design rows in Table V. This comparison shows that an overall improvement in skew tolerance is in proportion to $(n/2)T_1$ for $|t_{sk}|$

$< (n/2)T_1$ and in proportion to nT_1 for $(n/2)T_1 < |t_{sk}| < nT_1$. Again utilizing the analysis provided in section III-A, time period of the proposed design, given in inequalities (36) and (37), can be compared with inequality (41), and which again leads to the conclusion that the operating frequency can be increased by up to n times with the proposed design.

2) *Timing constraints in conventional and proposed F-to-S system–II under Different contexts*

In the proposed F-to-S system–II it is considered that the slow receiver module is able to latch the data only once in the entire slow cycle nT_1 . Fig. 7a and 7b respectively show the proposed interfacing mechanism and expected waveforms for the proposed F-to-S systems–II. A careful inspection of Fig. 7a demonstrates that Region B of this design is identical to Fig. 3a, while Region A is identical to Fig. 6a. Region C of Fig. 7a is also comparable to Region C of Fig. 3a, where the Multiplexer of Fig. 3a is replaced by a wider width register RRIM and IP RX of Fig. 3a is replaced by a wide width IP RX in Fig. 7a.

Fig. 7b explains the expected waveform of this design. It can be seen that the phase adjustment in Region B leads to similar advantages as derived for the proposed F-to-S system–I. Due to the different nature of IP RX, a wide width register is required to latch the entire n data items at once at the receiving end, Region C. This requirement of IP RX put additional timing constraint at Region C. These local timing constraints are alleviated by the introduction of the intermediate register RRIM. This register latches half of the data items from the RR(x) interfacing registers, which in turn allows the designer $n/2$ different clock phases for clocking IP RX. In Fig. 7a and 7b the IP RX is clocked by $RSMCLK(n/2)$, but if Region C internal delay exceeds the duration T_1 then this design allows IP RX to clock as late as $RSMCLK(n-1)$ allowing a local Region C delay of $n/2T_1$.

For performance comparison with the conventional design, the preceding analysis shown in Table IV and V are still valid. Region C delay equations can be easily found using a similar analysis, and it is not developed in detail due to its lesser significance.

3) Summary of the Advantages of the Proposed Design over Conventional Designs:

The terminating Modules timing margins are virtually independent of the clock skew – As seen in section IV-A-1, inequalities 4 to 6 and 36 show that T_1 , the terminating-IP time period, is independent of the clock skew. A closer inspection identifies that the relaxation in T_1 is directly proportional to the number of registers in each stage.

No possibility of clock-data delay mismatch – No clock signal needs to traverse the entire length of the data path. Therefore, there is no possibility of clock data delay mismatch, which is a major performance bottleneck **Erreur ! Source du renvoi introuvable.**

Skew Tolerance is increased – Inequalities (21) and (26) of the proposed design and inequality (10), shown in Table I, of the conventional design indicate worst case setup time constraints. Comparing these inequalities, it is found that the skew tolerance is increased in the proposed design in proportion to the number of interfacing registers n , up to $n/2$ times for zero-cycle latency and up to n times for one-cycle latency.

Delay Insertion limit is increased – It is observed that relation (33) for the proposed design and relation (17) for the conventional design show that the limit of delay insertion, to cope with the hold time constraint, is increased in the proposed design. This increase is proportional to $n/2$.

Relaxation in Delay Insertion – Inequality (17), in Table I, indicates that, in the conventional design, the worst case delay insertion is required when

$$t_{sk+} > t_{cq} - 2t_j - t_{hold} \quad (35)$$

whereas, in the proposed design, delay insertion is required only when $t_{sk} > (n/2)T_1$. It is known that, in most digital systems, to meet setup time constraint, t_{cq} should be less than clock period or $t_{cq} \ll T_1$. Therefore, it can be concluded, after comparing delay insertion requirement for the proposed design with (35), that there is a substantial relaxation in timing budget to enable delay insertion with the proposed system. This relaxation is also proportional to $n/2$.

In summary, it can be observed that, when utilizing the interfacing registers, the terminating module

frequency becomes independent of the clock skew and this skew is completely absorbed in the interfacing registers. There was no delay matching requirement, as is the case in **Erreur ! Source du renvoi introuvable.****Erreur ! Source du renvoi introuvable.****Erreur ! Source du renvoi introuvable.**

IV. SIMULATION SETUP AND RESULTS

In order to verify the validity of the proposed solution, detailed design and simulations were performed for the case explained in the section III-A, where two terminating modules are communicating at the same frequency. Gate level synthesis, simulations and static timing analysis are performed for 0.18 micron TSMC CMOS process technology, using Synopsys' Design Compiler (DC) [7], and Prime Time for static timing analysis [8]. This section describes the assumptions and simulation setup, along with the results and their comparison with the conventional synchronous designs.

Two different sub-cases are simulated. The first sub-case simulates the hardware implementation of the structure shown in Fig. 4a, which is a conventional design and does not contain any interfacing registers. The other sub-case simulates an example of the proposed hardware implementation shown in Fig. 3a. This simulated design example has four interfacing registers at each terminating end.

It can be seen in Fig. 8 that for a given skew value under unidirectional communication, when the system is subject to positive skew only, the proposed design with four interfacing registers at each stage can communicate at a higher frequency, compared to the conventional design. The proposed solution with four interfacing registers at each terminating end is called the Quadruple Bus Width (QBW) solution.

Initially, to keep the speed of terminating modules within practical limits, it is observed from gate level simulations that a very simple circuit, the TFF (Toggle Flip Flop), can operate at a maximum frequency of approximately 500 MHz with the 0.18 micron TSMC CMOS process technology. Therefore, the maximum considered frequency for this technology is 500 MHz. In Fig. 8 it is shown that the conventional design can run up to 125 MHz, for a 7.5 nsec skew, whereas the proposed design allows the terminating modules to work at a frequency as high as 500 MHz for the same skew. This improvement in positive skew tolerance is

in accordance with the analytical study performed in section III-A. Here the skew is bounded by, $n/2T_1 < |t_{sk}| < nT_1$; and the simulation results is within the limit provided by analytical inequality (25).

The second simulation result, as shown in Fig. 9, addresses the timing constraints associated with the unidirectional systems that are subject to a negative skew only. The simulation results show that with a 3.3 nsec negative skew, a conventional design can work at a maximum frequency of 250 MHz, while the proposed design for the same skew can work at double the frequency, 500 MHz. This means QBW works two ($n/2 = 4/2$) times faster compared to conventional design for the same skew budget. This is in agreement with the analytical results obtained for the proposed design when $|t_{sk}| < (n/2)T_1$, in inequality 21.

Fig. 10 shows a third case where the bidirectional communication is simulated under the assumption that skew orientation is not known. Data points for conventional design are shown with the squares. Data points for QBW design are represented by diamonds. The simulation results in Fig. 10 for the zero clock cycle latency case show for a skew of 3.3 nsec, the terminating modules of the proposed design communicate at 500 MHz, as compared to 142 MHz for conventional design under the same context. It can be noticed that 3.3 nsec is less than $(n/2)T_1$; hence the obtained frequency values are in compliance with inequality (10-A) for the conventional design and with inequality (21) for the proposed design. Similarly, the proposed design tolerates about 13.3 nsec of skew at the frequency of 142 MHz. This result is still within the limits of $(n/2)T_1$ time duration, in compliance with the theory, and also shows a tolerance of approximately 4 times more skew than its conventional counterpart running at the same frequency.

Designs explained in section III-B were also simulated, under similar design constraints. Due to the identical characteristics in region 'B' of the systems for all the three cases, the results of the simulation are almost exactly the same. These simulations further confirmed the analytical designs, but due to their identical nature, simulation results for the designs of section III-B are not shown separately.

V. PROTOTYPE IMPLEMENTATION AND BACK-ANNOTATED SIMULATION RESULTS

Prototype implementations of the proposed designs were implemented using a Virtex-II Pro FPGA board from Xilinx. A gate-level HDL description was written for the proposed design shown in Fig. 3a. In this prototype implementation, a 3-bit binary counter is the (data generating) sender module, along with two stages of four interfacing registers and the receiver unit comprising a 4-to-1 Multiplexer and a receiver register. Fig. 11 shows the back-annotated simulation results of the design where the terminating modules are working at 250 MHz and the skew is negative and of 12 nsec amplitude, which is more than $(n/2)T_1$. Both the design parameters are limited because of the FPGA technology used. Virtex-II Pro FPGA XC2VP30-7FF896 can run at a maximum clock frequency of 250 MHz under the slow model and at 320 MHz under the fast model, when an internal clock of 100 MHz is used to synthesize frequencies through Digital Clock Managers (DCM) **Erreur ! Source du renvoi introuvable.** Around 12 nsec is the maximum delay limit that can be introduced by this FPGA technology **Erreur ! Source du renvoi introuvable.** Figure 12 shows a similar waveform for this solution with positive skew of about 12 nsec under the same operating frequency of the terminating modules.

Figures 13 and 14 show the back-annotated simulated waveforms of the proposed F-to-S design described in Fig. 7a under the constraint of positive and negative skew, respectively. It can be seen in the waveform that the sender module is working at 250 MHz and that the skew, in either orientation, is 12 nsec. It is shown that, with proper phase adjustment and interfacing registers, the data is safely latched at the receiver. Due to the limitation of the available oscilloscope, all the data signals can not be shown concurrently. The design chosen for demonstration purposes is shown in Fig. 7a where the output is slowed down to one fourth of the sender frequency. Fig. 15 shows a representative waveform of the prototype implementation. This figure demonstrates the case when the sender module is working at 350 MHz, which corresponds to a time period of 2.67 nsec. Note that, although the frequency limit according the manual is 320 MHz **Erreur ! Source du renvoi introuvable.**, in practice it is seen that a relatively higher frequency, 350 MHz., is achieved. This may be attributed to the fact that the system designed in this work does not exactly

match the loading criterion and the process parameters for which the value is reported in the design manual. As it is of lesser significance for our design goal, we did not investigate further on it.

The phase shift or skew applied to this design is of magnitude 12 nsec in either orientation. This skew is chosen to observe the effect of skew of a magnitude $|t_{sk}| > (n/2)T_l$ on the proposed skew tolerant design. Two separate tests are performed for positive and negative skew, applying the same phase discrepancy magnitude.

The three bit binary counter counts up to 7 and then resets to 0. Each set of three bits is called a count value in this paper. Therefore, the 3 bit counter has a total of 8 possible count values, from 000 to 111. The time to complete the counting from 000 to 111 and back to 000 is called a count cycle in this text. The top waveform of Fig. 15 represents the MSB (most significant bit) of the count values from one of the four sending-end interfacing registers. The second waveform from the top shows the MSB of the corresponding IPRX register at the receiving end. As the three bit counter is working four times faster than the corresponding receiver, and since, as explained above, each count cycle is performed in eight fast clock cycles, therefore, each receiving-end register holds two different count values in one count cycle. Each count value in the receiving end register lasts for four sending-end (faster) clock cycles. Therefore, the fastest transition that can be seen at the receiver end is one fourth of the faster clock cycle.

It has been observed through the FPGA implementation, that the receiver register receives all the count values. Fig. 15 to 18 show the output of the first stage interfacing registers and compare them with the values latched in corresponding IPRX register. This experiment not only demonstrates the correct functionality at the maximum frequency that such an FPGA may work at, but also shows the phase difference. It is not possible to distinguish, in these waveforms, whether the skew is positive or negative (only the phase difference is shown) but the experiment is performed for both types of skew. The decreasing phase difference of IPRX with each of the interfacing register validates the functionality. The frequency of the analog waveform is measured to be 43.6 MHz, which is very close to $(350/8)$ MHz. The delay between the two MSB waveforms ranges from 10 to 20 nsec, depending on the four different phases

used to clock the interfacing registers. Hence, overall this implementation shows that the proposed design scheme is able to retrieve all the data elements sent by the four-times faster sender module, along with a phase shift of more than $(n/2)T_1$.

VI. CONCLUSION AND FUTURE WORK

This work shows that an all digital and completely synchronous solution may be adopted for IPs communicating in multiple clock domains and subject to severe phase discrepancies. Rather than a paradigm shift to asynchronous or other non-synchronous solutions, the synchronous solutions can be readily applied to the system and can be accommodated in the design flow with a far lesser effort. Different cases of multiple clock domains are addressed, including terminating modules communicating at same and integer-multiple frequencies. The timing constraints, for all the possible cases of the proposed solutions, have been established. It is observed that when the number of interfacing registers is increased, skew tolerance increases linearly. Gate level simulations were performed for different clock frequencies using TSMC 0.18 technology library. These results are in complete compliance with the analytical results. Comparison with conventional design shows, depending on the context, up to $n/2$ to n clock cycles of tolerance in skew budget. The best previously reported state-of-the-art FIFO based inter-module communication method [2] allows up to two clock cycles of skew tolerance, hence the proposed design is better in terms of skew tolerance. A prototype implementation of the proposed F-to-S system-I is performed using a FPGA, Virtex-II Pro from Xilinx, to help validating the concept and conforming the analytical results quantitatively. It is observed that data sent from a 375 MHz IP TX is safely received by the IPs running at one fourth of that frequency, even though the experimental phase discrepancy was set up to 10 nsec which is approximately nT_1 . This is in compliance with the analytical results obtained in section III for unidirectional communication.

Qualitatively, the proposed method uses purely digital synchronous methodology, which is advantageous and desired in the industry as compared to the PLL based system that involves mixed signal components or

asynchronous interfaces.

References

- [1] Semiconductor Industry Association, “International technology roadmap for semiconductors”, 2007.
<http://www.itrs.net/Links/2007ITRS/ExecSum2007.pdf>
- [2] A. Chakraborty and M. R. Greenstreet, “Efficient Self-Timed Interfaces for Crossing Clock Domains”, in Ninth IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'03), pp 78-88, 2003
- [3] M. De Clercq and R. Negulescu, “1.1-GDI/s Transmission between Pausible Clock Domains”. In Proc. IEEE International Symposium on Circuits and Systems (ISCAS), Vol 2, pp II-768- II-771, 2002
- [4] S. R. Hasan, Y. S. and Mohamed Nekili, “Split H-tree Design Method for High-Performance GALS Systems”, in 4th IEEE NorthEast Workshop on Circuit and Systems (NEWCAS 2006), pp 161-164, 2006.
- [5] J. Mekie, S. Chakraborty, and D. K. Sharma., “Evaluation of pausable clocking for interfacing high-speed IP cores in GALS systems”, in Proc. of VLSI'04, pp. 559 -564, 2004
- [6] J.M. Rabaey, A. Chandrakasan, B. Nikolic, “Digital Integrated Circuits; A Design Perspective: Second Edition”, Prentice Hall Electronics and VLSI Series, 2003.
- [7] Synopsys' Design Compiler, User guide, Version V-2004.06, June 2004
- [8] Synopsys' PrimeTime Tutorial, Version V-2004.06, June 2004
- [9] P. Zarkesh-Ha, “Global Interconnect Modeling for a Gigascale Sytem-on-a-Chip”, Ph.D. thesis, Georgia Institute of Technology, February 2001, [online]
http://www.ece.gatech.edu/research/labs/gsigroup/publications/Dissertation_Payman.pdf
- [10] H. Bakoglu “Circuits, Interconnections and Packaging for VLSI”, Reading M.A: Addison-Wesley, 1990.
- [11] W. J. Dally and John W. Poulton, “Digital Systems Engineering”, Cambridge University Press, 1998
- [12]Xilinx's Virtex-II Pro, Platform FPGA Handbook, UG012 (v2.0), October 14, 2002.

- [13]J. Muttersbach, T. Villiger, W. Fichtner, “practical Design of Globally-Asynchronous Locally-Synchronous systems”, in Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, pp. 52-59, April 2000.
- [14]S. Moore, G. Taylor, R. Mullins, P. Robinsion, “Point to point GALS interconnect” in 8th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'02), pp 69-75, 2002.
- [15]R. Dobkin, R. Ginosar, C. P. Sotiriou, “Data synchronization issues in GALS SoCs”, in 10th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'04), pp 170-179, 2004.
- [16]R. Mullins and S. Moore, "Demystifying Data Driven and Pausible Clocking Scheme", in 13th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'07), pp 175-185, March 2007.
- [17]J. Bainbridge and S. Furber, “Chain: a delay-insensitive chip area interconnect”, Micro IEEE Volume 22, Issue 5, pp. 16 -23, Sept. Oct. 2002
- [18]ARM, Technical Specification: AMBA Specification, Doc No: ARMIHI-0011A, Issued: May 2001.
- [19]IBM, Technical Specification: 32-bit Processor Local Bus – Architecture Specification, Doc NO: SA-14-2531-01, Issued May 2001.
- [20]HyperTransport Technology Consortium, Technical Specification: Hypertransport I/O Link Specification, Doc NO: HTC2001021-0009-0022, Issued: Aug. 2003.
- [21]W. J. Dally, and B. Towles, “Route Packets not Wires: On-Chip Interconnection Networks,” in Proceeding of Design Automation Conference, 2001, DAC'01.
- [22]T. Chelcea and S. M. Nowick, “Robust-Interfaces for Mixed-timing Systems ”, in IEEE Transaction on Very Large Scale Integration (VLSI), Volume 12, Issue 8, pp. 857-873, August, 2004.
- [23]F. Mu, and C. Svensson, “Self-tested self-synchronization circuit for mesochronous clocking”, in IEEE Transaction on Circuits and Systems II: Analog and Digital Signal Processing, Volume 48, Issue 2, Feb. 2001 pp. 129 – 140.

- [24]I. Soderquist, “Globally updated mesochronous design style”, in IEEE Journal of Solid State Circuits, Volume 38, Issue 7, July 2003 pp.1242 – 1249.
- [25]M. Kihara, “Digital clocks for synchronization and communications:”. Boston : Artech House, chapter 11, c2003.
- [26]D. E. Calbaza, and Y. Savaria, “Direct digital frequency synthesis of low-jitter clocks”, in IEEE Journal of Solid-State Circuits, Volume 36, Issue 3, March 2001, pp. 570 – 572.
- [27]D. E. Calbaza, and Y. Savaria, “A direct digital period synthesis circuit”, in IEEE Journal of Solid-State Circuits, Volume 37, Issue 8, Aug. 2002, pp. 1039 – 1045.
- [28]F. R. Boyer, H. G. Epassa, Y. Savaria, “Embedded power-aware cycle by cycle variable speed processor”, in IEE Proceedings on Computers and Digital Techniques, Volume 153, Issue 4, 3 July 2006, pp. 283 – 290.
- [29]D. E. Calbaza, I. Cordos, N. Seth-Smith, Y. Savaria, “An ADPLL circuit using a DDPS for genlock applications”, in Proceedings of the 2004 International Symposium on Circuits and Systems, 2004, ISCAS’ 04, Volume 4, 23-26 May 2004, pp. IV- 569-72.
- [30]A. Edman, C. Svensson, “Timing closure through a globally synchronous, timing partitioned design methodology”, in Proceedings of the 41st Design Automation Conference, 2004, pp 71 – 74.
- [31]A. Edman, C. Svensson, “Synchronous latency-insensitive design for multiple clock domain”, Proceedings of IEEE international SOC Conference, 25-28 Sept. 2005, pp. 83 – 86.
- [32]P. Caputa, C. Svensson, “An on-chip delay- and skew-insensitive multicycle communication scheme”, in IEEE International Conference of Technical Papers on Solid-State Circuits, Feb. 6-9, 2006, pp. 1765 – 1774.
- [33]P. Teehan, M. Greenstreet, G. Lemieux, “A Survey and Taxonomy of GALS Design Styles” Design & Test of Computers, IEEE Volume 24, Issue 5, Sept.-Oct. 2007 pp. 418 – 428.

[34]H. T. P. Nguyen, “Conception d’un module de synchronisation pour l’intégration à l’échelle de la tranche de routeurs de communication”, Masters’ Thesis, Department of Electrical Engineering, University of Montréal, QC, Canada, December 2005.

Table I. Timing constraints for unidirectional communication between the terminating modules of conventional and proposed designs, running at same frequency

	$t_{sk} > 0$	$t_{sk} < 0$
Conventional Design	$T_1 > t_{cq} + 2t_j + t_{SU} - t_{sk+}$ (9) $t_{cq} > t_{sk+} + 2t_j + t_{hold}$ (12) $t_{sk+} + 2t_j + t_{hold} - t_{cq} < \Delta < T_1 + t_{sk+} - 2t_j - t_{SU} - t_{cq}$ (17)	$T_1 > t_{cq} + 2t_j + t_{SU} + t_{sk-}$ (10) $t_{cq} > -t_{sk-} + 2t_j + t_{hold}$ (13) No requirement of any Δ insertion in this case
Proposed Design ($ t_{sk} < (n/2)T_1$)	$(n/2) T_1 > t_{cq} + t_{SU} + 2t_j - t_{sk+}$ (20) No hold time violations	$(n/2) T_1 > t_{cq} + t_{SU} + 2t_j + t_{sk-}$ (21) No hold time violations
Proposed Design ($(n/2)T_1 < t_{sk} < nT_1$)	$(3/2)nT_1 > t_{cq} + t_{SU} + 2t_j - \delta$ (25) $t_{cq} + \Delta > \delta + t_{hold} + 2t_j$ (29) $\delta + t_{hold} + 2t_j - t_{cq} < \Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq}$ (33)	$nT_1 > t_{cq} + t_{SU} + 2t_j + \delta$ (26) $t_{cq} + \Delta > -\delta + t_{hold} + 2t_j$ (31) $-\delta + t_{hold} + 2t_j - t_{cq} < \Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq}$ (34)

Table II. Timing constraints for bidirectional communication between the terminating modules of conventional and proposed designs, running at same frequency (for known and unknown skew orientations)

	Bi-directional Communication with known skew orientation	Bi-directional Communication with unknown skew orientation
Conventional Design	$T_1 > t_{cq} + 2t_j + t_{SU} + t_{sk-}$ (10) $t_{sk+} + 2t_j + t_{hold} - t_{cq} < \Delta < T_1 + t_{sk+} - 2t_j - t_{SU} - t_{cq}$ (17)	$T_1 > t_{cq} + \Delta + 2t_j + t_{SU} + t_{sk-}$ (10-A) $t_{sk+} + 2t_j + t_{hold} - t_{cq} < \Delta < T_1 - t_{sk-} - 2t_j - t_{SU} - t_{cq}$ (18)
Proposed Design ($ t_{sk} < (n/2)T_1$)	$(n/2) T_1 > t_{cq} + t_{SU} + 2t_j + t_{sk-}$ (21) no hold time violation	$(n/2) T_1 > t_{cq} + t_{SU} + 2t_j + t_{sk-}$ (21) no hold time violation
Proposed Design ($(n/2)T_1 < t_{sk} < nT_1$)	$nT_1 > t_{cq} + t_{SU} + 2t_j + \delta$ (26) $\delta + t_{hold} + 2t_j - t_{cq} < \Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq}$ (33)	$nT_1 > t_{cq} + \Delta + t_{SU} + 2t_j + \delta$ (26-A) $\delta + t_{hold} + 2t_j - t_{cq} < \Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq}$ (33)

Table III. A Summary of timing constraints for conventional F-to-S system-I, along with the corresponding inequalities of conventional design of Fig. 4a, i.e. when the terminating modules have same frequency

	F-to-S conventional System shown in Fig. 5a	Conventional Design shown in Fig. 4a
Setup time constraint	$T_1 + t_{sk} > t_{cq} + t_{DMUX} + t_{SU} + 2t_j$ (39) $T_1 > t_{cq} + t_{DMUX} + 2t_j + t_{SU} - t_{sk+}$ ($t_{sk} > 0$) (40) $T_1 > t_{cq} + t_{DMUX} + 2t_j + t_{SU} + t_{sk-}$ ($t_{sk} < 0$) (41)	$T_1 + t_{sk} > t_{cq} + 2t_j + t_{SU}$ (8) $T_1 > t_{cq} + 2t_j + t_{SU} - t_{sk+}$ (9) $T_1 > t_{cq} + 2t_j + t_{SU} + t_{sk-}$ (10)
Hold time constraint	$t_{cq} + \Delta + t_{DMUX} > t_{hold} + 2t_j + t_{sk}$ (42) $t_{cq} + \Delta + t_{DMUX} > t_{sk+} + 2t_j + t_{hold}$ ($t_{sk} > 0$) (43) $t_{cq} + \Delta + t_{DMUX} > -t_{sk-} + 2t_j + t_{hold}$ ($t_{sk} < 0$) (44)	$t_{cq} > t_{sk} + 2t_j + t_{hold}$ (11) $t_{cq} + \Delta > t_{sk+} + 2t_j + t_{hold}$ (14) $t_{cq} + \Delta > -t_{sk-} + 2t_j + t_{hold}$ (15)
Delay insertion limit	$\Delta < T_1 + t_{sk} - 2t_j - t_{hold} - t_{cq} - t_{DMUX}$ (45)	$\Delta < T_1 + t_{sk} - 2t_j - t_{SU} - t_{cq}$ (16)

Table IV. Summary of timing constraints for conventional and proposed F-to-S system-I under unidirectional communication

	$t_{sk} > 0$	$t_{sk} < 0$
Conventional Design	$T_1 > t_{cq} + t_{DMUX} + 2t_j + t_{SU} - t_{sk+}$ (40) $t_{cq} + \Delta + t_{DMUX} > t_{sk+} + 2t_j + t_{hold}$ (43) $t_{sk+} + 2t_j + t_{hold} - t_{cq} - t_{DMUX} < \Delta < T_1 + t_{sk+} - 2t_j - t_{hold} - t_{cq} - t_{DMUX}$ (46)	$T_1 > t_{cq} + t_{DMUX} + 2t_j + t_{SU} + t_{sk-}$ (41) $t_{cq} + t_{DMUX} > -t_{sk-} + 2t_j + t_{hold}$ (44-A) No requirement of any Δ insertion in this case
Proposed Design ($t_{sk} < (n/2)T_1$)	$(n/2)T_1 > t_{cq} + t_{SU} + 2t_j - t_{sk+}$ (20) No hold time violations	$(n/2)T_1 > t_{cq} + t_{SU} + 2t_j + t_{sk-}$ (21) No hold time violations
Proposed Design (($n/2$)$T_1 < t_{sk} < nT_1$)	$(3/2)nT_1 > t_{cq} + t_{SU} + 2t_j - \delta$ (25) $t_{cq} + \Delta > \delta + t_{hold} + 2t_j$ (29) $\delta + t_{hold} + 2t_j - t_{cq} < \Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq}$ (33)	$nT_1 > t_{cq} + t_{SU} + 2t_j + \delta$ (26) $t_{cq} + \Delta > -\delta + t_{hold} + 2t_j$ (31) $-\delta + t_{hold} + 2t_j - t_{cq} < \Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq}$ (34)

Table V. Summary of timing constraints for conventional and proposed F-to – S system-I, bidirectional communication (with known and unknown skew orientations)

	Bi-directional Communication with known skew orientation	Bi-directional Communication with unknown skew orientation
Conventional Design	$T_1 > t_{cq} + t_{DMUX} + 2t_j + t_{SU} + t_{sk-} \text{ (40)}$ $t_{sk+} + 2t_j + t_{hold} - t_{cq} - t_{DMUX} < \Delta < T_1 + t_{sk+} - 2t_j - t_{hold} - t_{cq} - t_{DMUX} \text{ (46)}$	$T_1 > t_{cq} + t_{DMUX} + \Delta + 2t_j + t_{SU} + t_{sk-} \text{ (41-A)}$ $t_{sk+} + 2t_j + t_{hold} - t_{cq} - t_{DMUX} < \Delta < T_1 - t_{sk-} - 2t_j - t_{hold} - t_{cq} - t_{DMUX} \text{ (47)}$
Proposed Design ($t_{sk} < (n/2)T_1$)	$(n/2) T_1 > t_{cq} + t_{SU} + 2t_j + t_{sk-} \text{ (21)}$ <p style="text-align: center;">no hold time violation</p>	$(n/2) T_1 > t_{cq} + t_{SU} + 2t_j + t_{sk-} \text{ (21)}$ <p style="text-align: center;">no hold time violation</p>
Proposed Design ($(n/2)T_1 < t_{sk} < nT_1$)	$nT_1 > t_{cq} + t_{SU} + 2t_j + \delta \text{ (26)}$ $t_{cq} + \Delta > \delta + t_{hold} + 2t_j \text{ (29)}$ $\delta + t_{hold} + 2t_j - t_{cq} < \Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq} \text{ (33)}$	$nT_1 > t_{cq} + \Delta + t_{SU} + 2t_j + \delta \text{ (26-A)}$ $\delta + t_{hold} + 2t_j - t_{cq} < \Delta < n/2 T_1 + t_{hold} - 2t_j - t_{cq} \text{ (33)}$

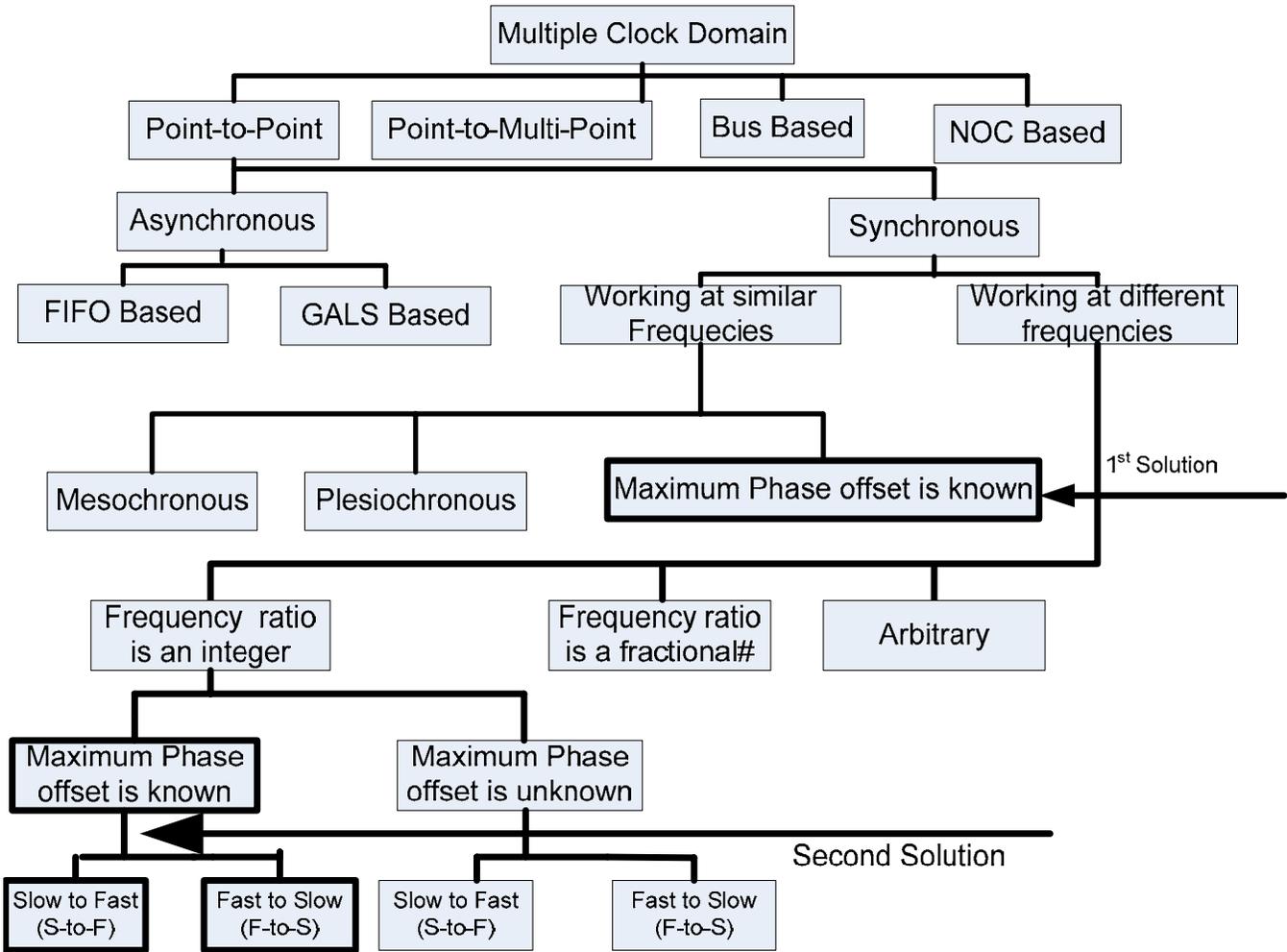


Figure 1. Possible inter-module communication methods

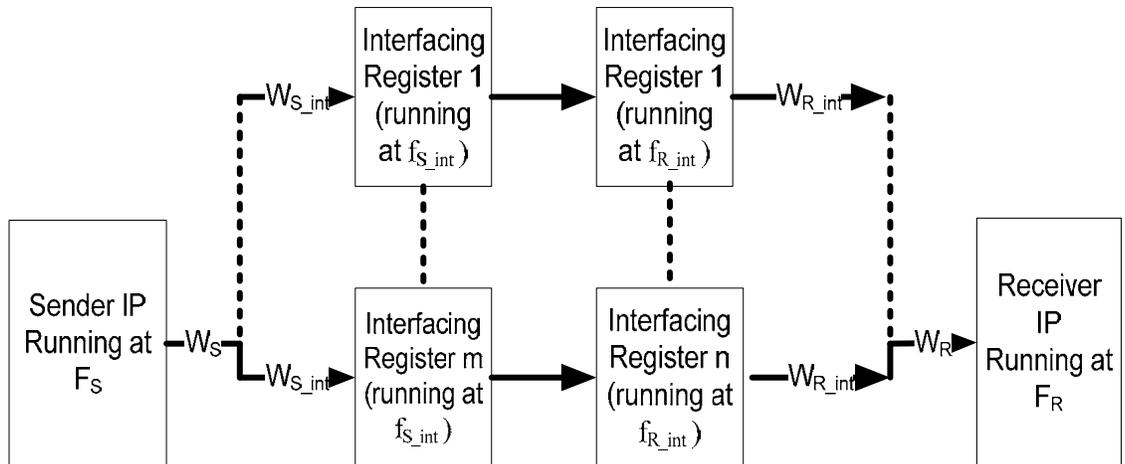


Figure 2a. Elaboration of two IP modules with interfacing registers

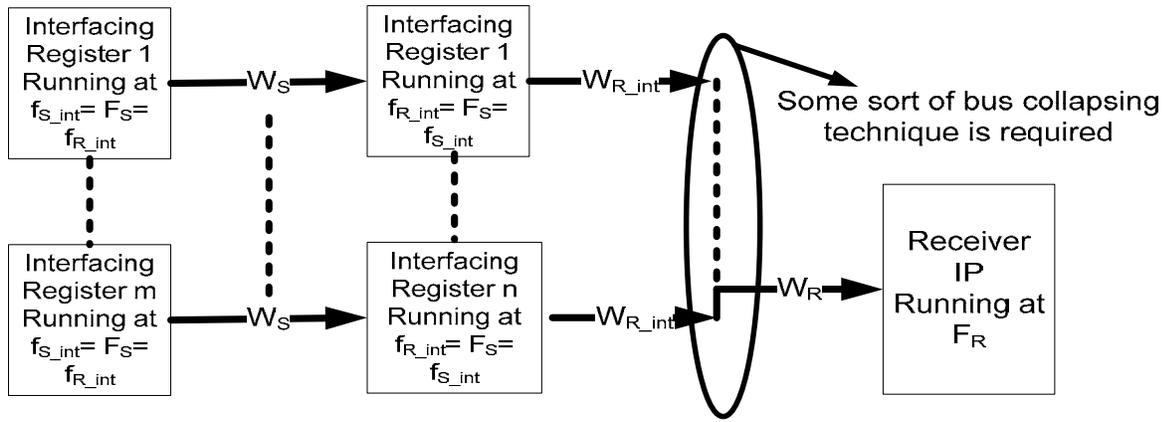


Figure 2b. Elaboration of Interfacing register scheme for slow sender and fast receivers

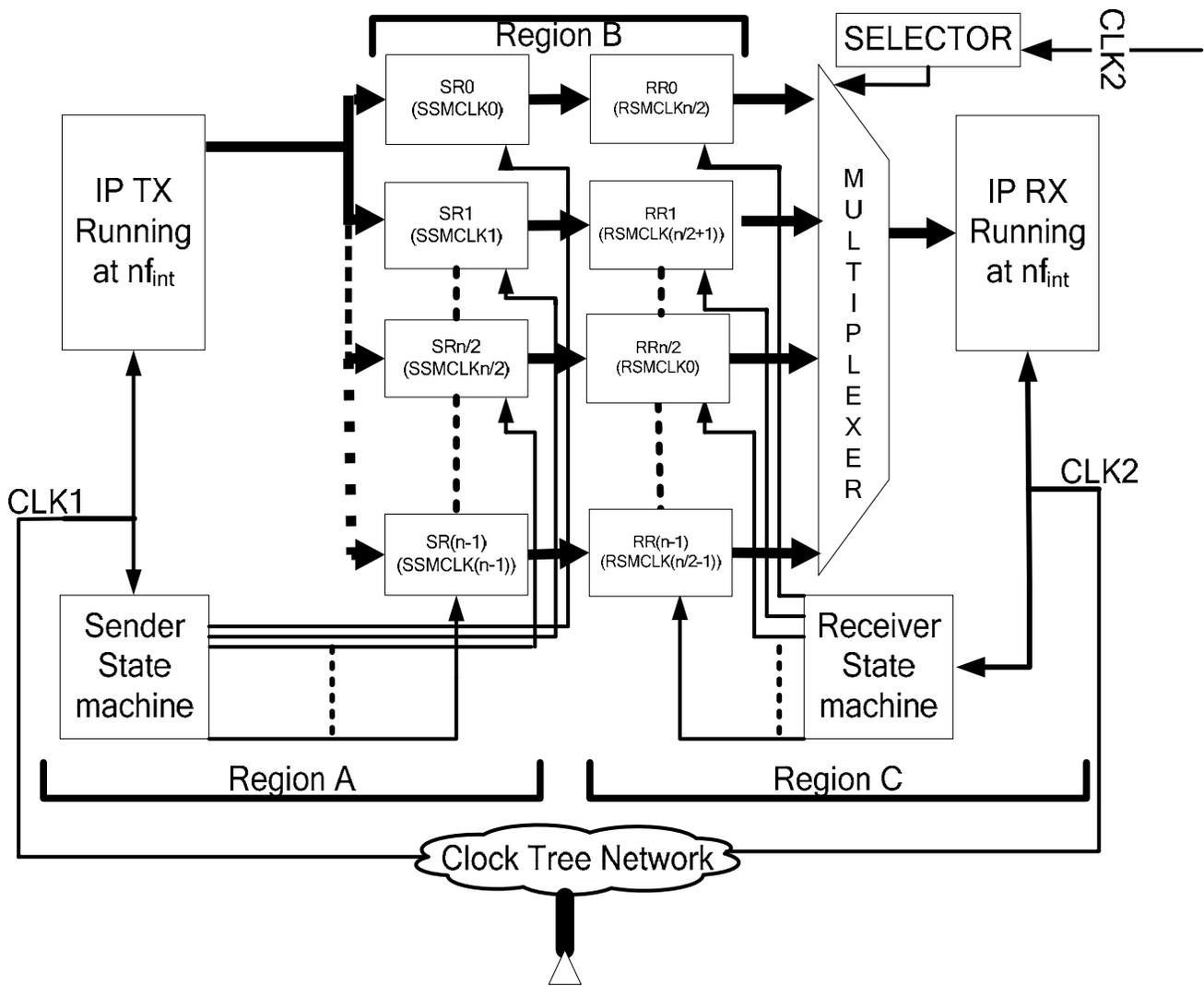


Figure 3a. Hardware implementation of interfacing registers for the n modules with same frequency and bus-width (n assumed even here for simplicity)

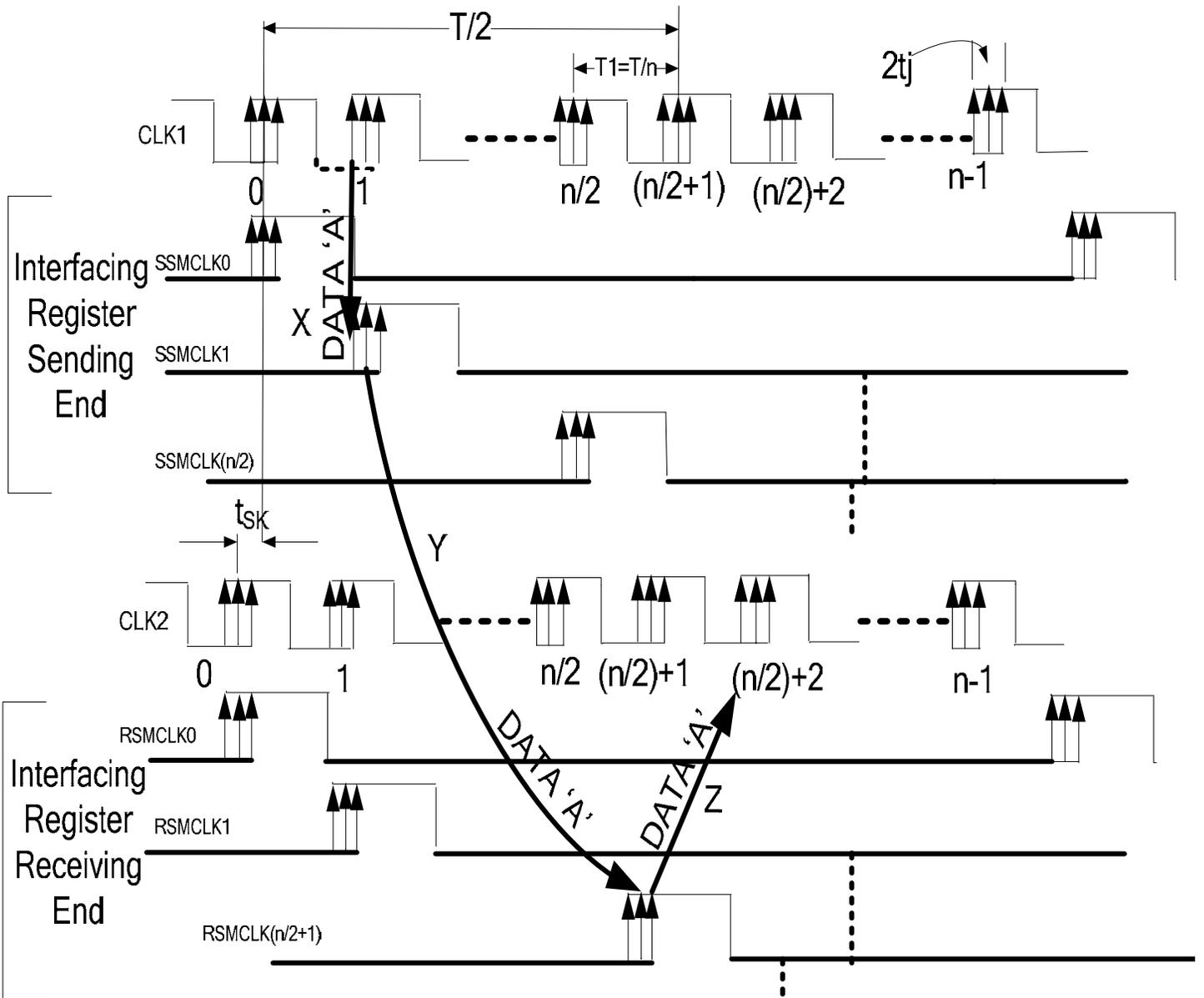


Figure 3b. Waveform representation of interfacing registers for the n modules with same frequency and bus-width (Arrows X, Y and Z, are showing one complete data path)

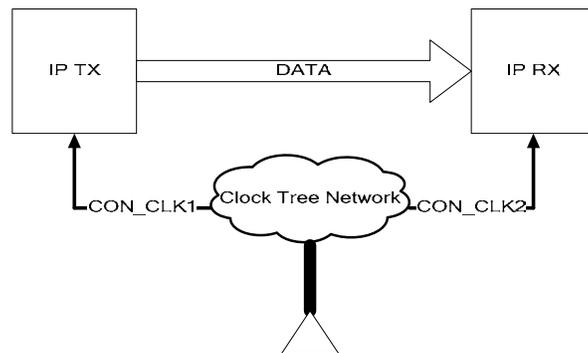


Figure 4a. Conventional two register communication. Both registers are working with the same frequency but different phase relationship, due to skew caused by non-idealities of the clock tree network.

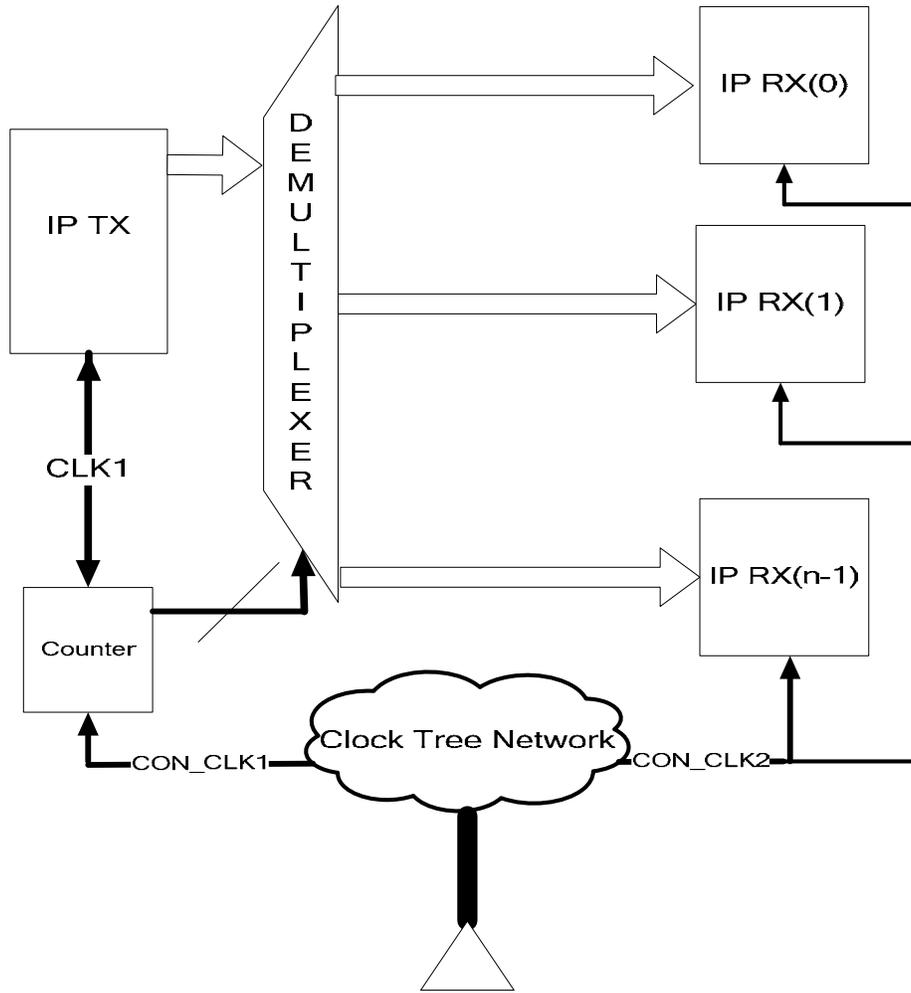
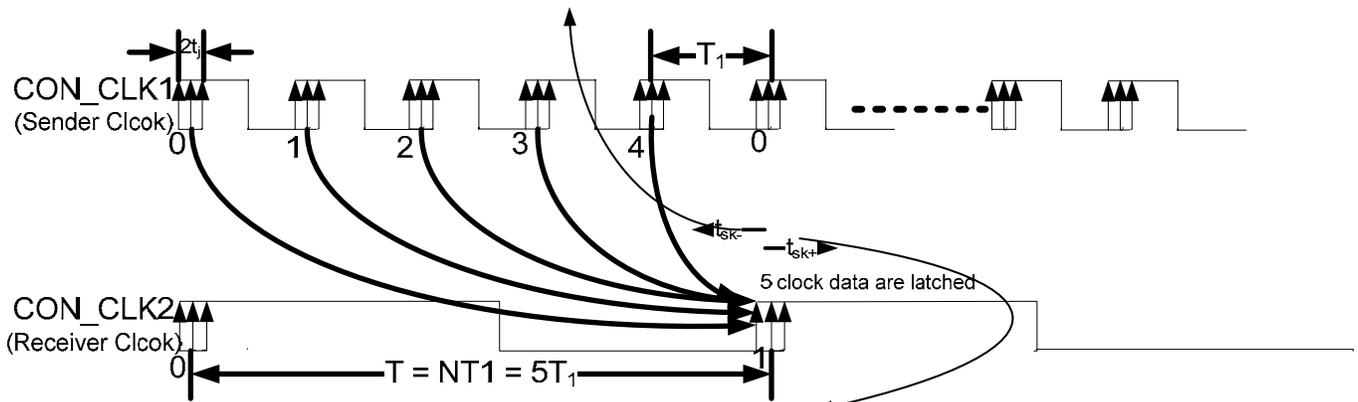


Figure 5a. Hardware implementation of F-to-S Conventional Design.

To avoid Negative Skew $T_1 > t_{cq} + \Delta + t_{DMUX} + t_{SU} + 2 t_j + |t_{sk-ve}|$



To avoid Positive Skew $t_{cq} + \Delta + t_{DMUX} > t_{hold} + 2 t_j + |t_{sk+ve}|$

Figure 5b. Waveform Representation of conventional design, with faster sender module and slower receiver module (F-to-S systems)

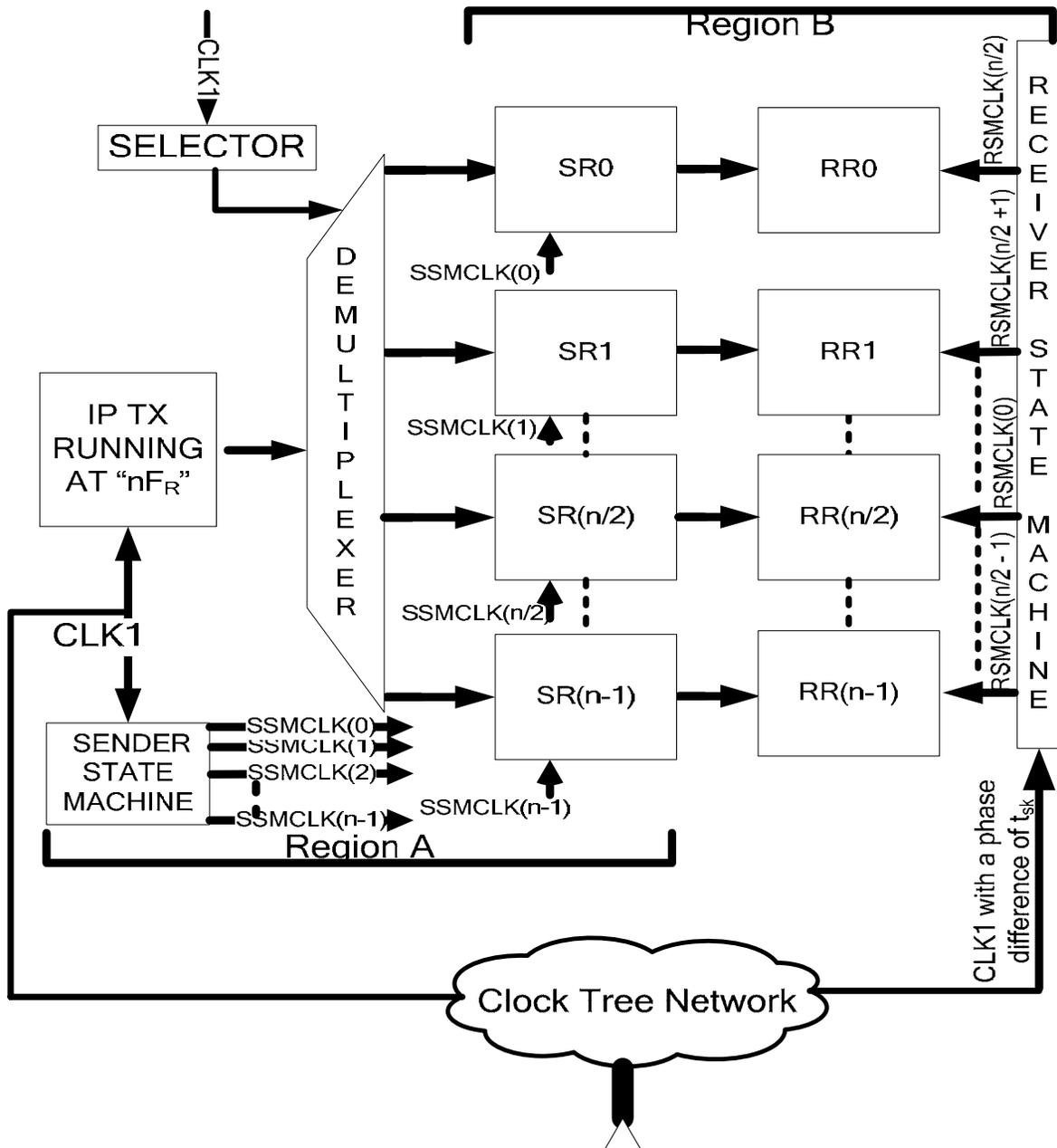


Figure 6a. Hardware implementation of Interfacing registers for the n modules with integer multiple frequencies. Faster sending module and slower receiving module (F-to-S system-I)

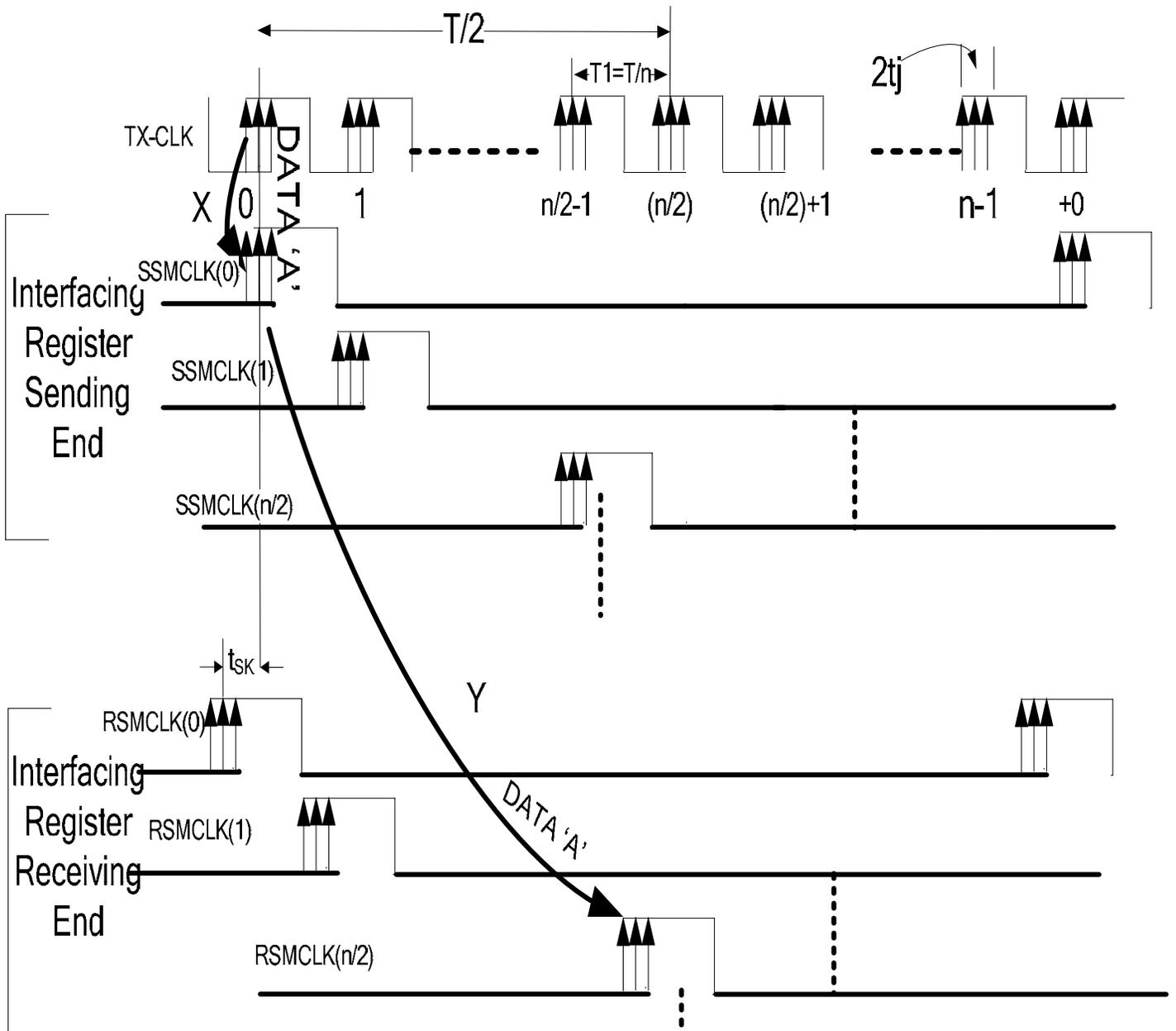


Figure 6b. Waveform representation of Interfacing registers for the n modules with integer multiple frequencies. Faster sending module and slower receiving module (F-to-S system-I)

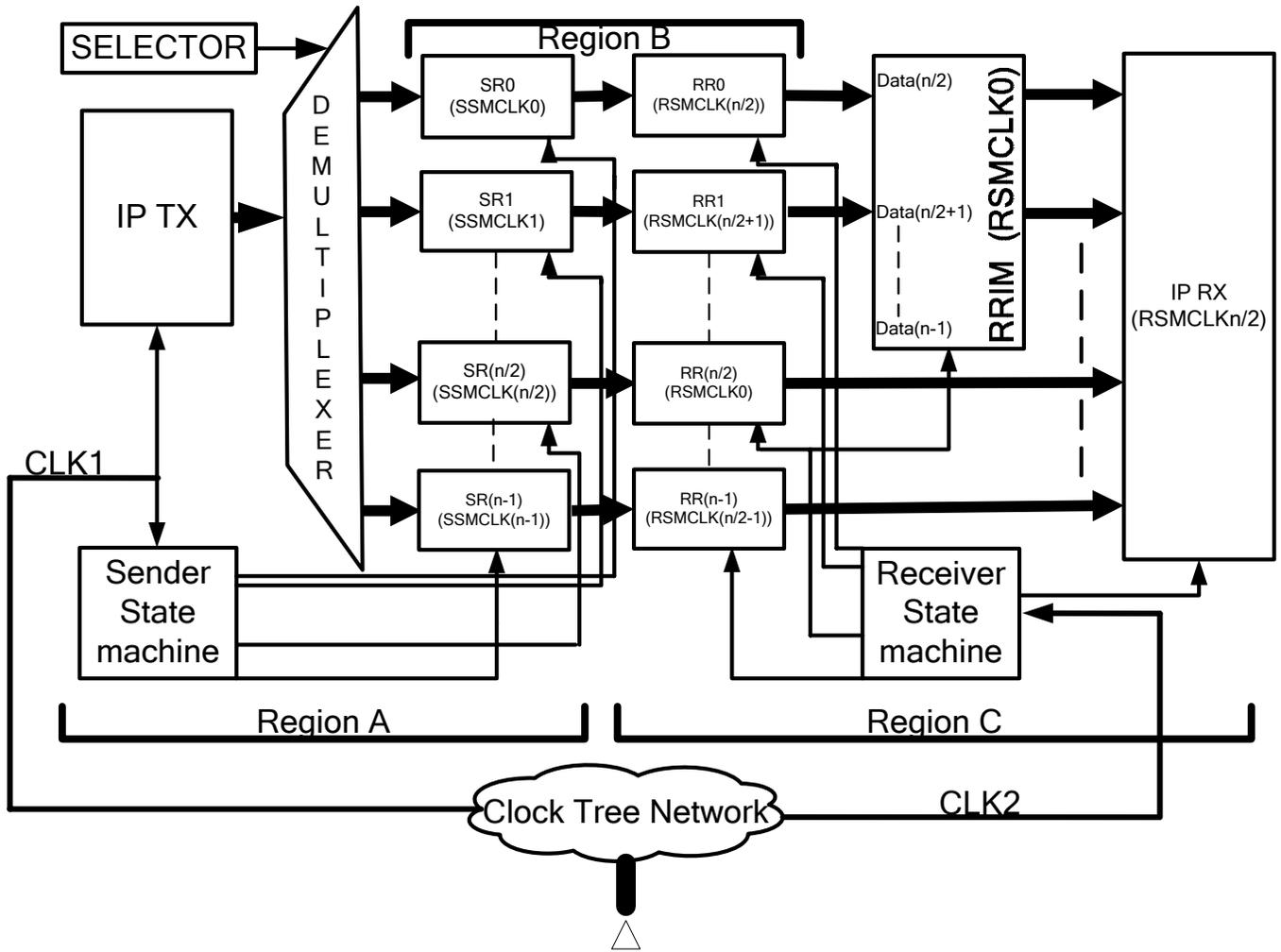


Figure 7a. . Hardware implementation of Interfacing registers for the n modules with integer multiple frequencies. Faster sending module and slower receiving module (F-to-S system-II)

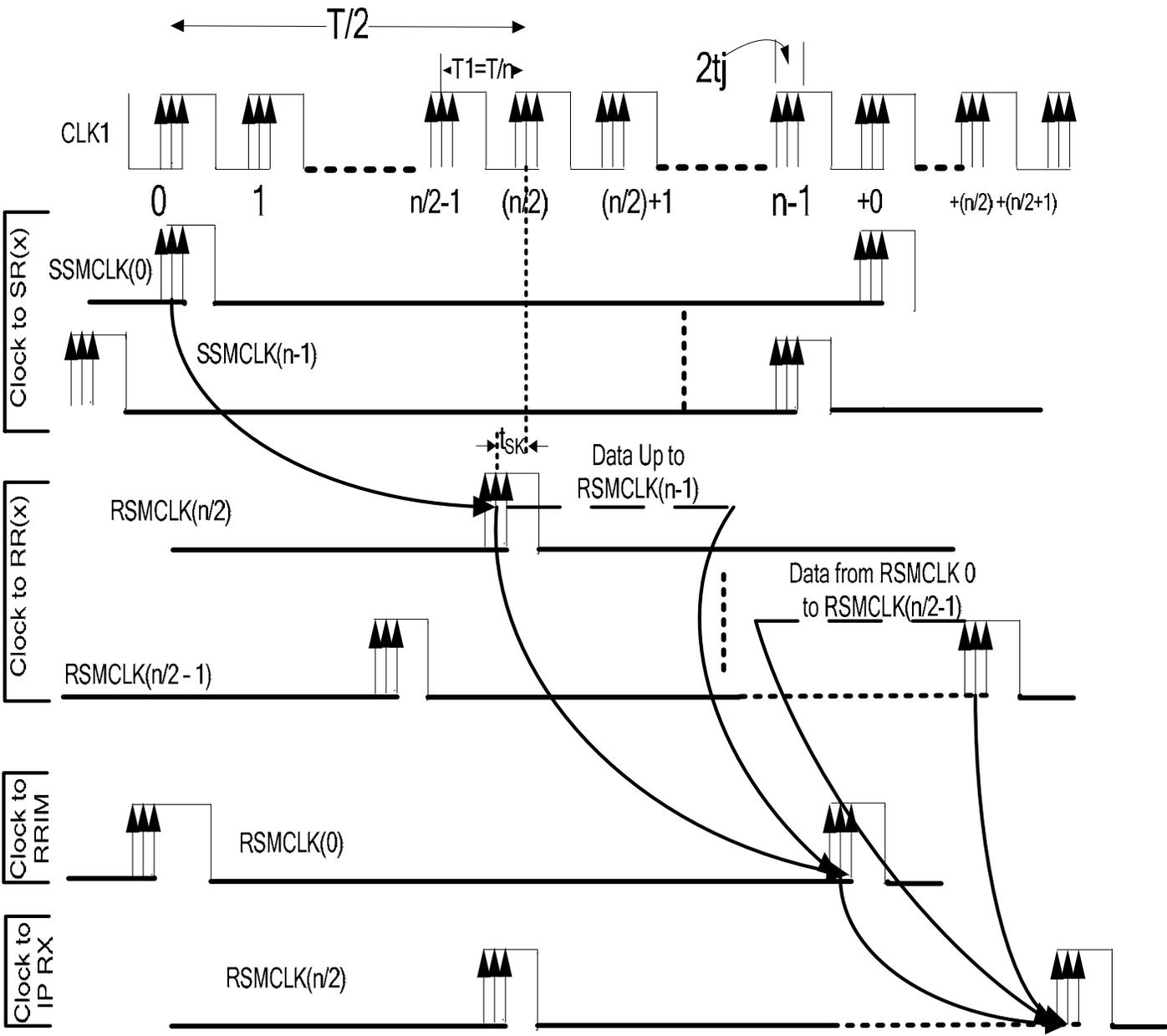


Figure 7b. Waveform representation of Interfacing registers for the n modules with integer multiple frequencies. Faster sending module and slower receiving module (F-to-S system-II)

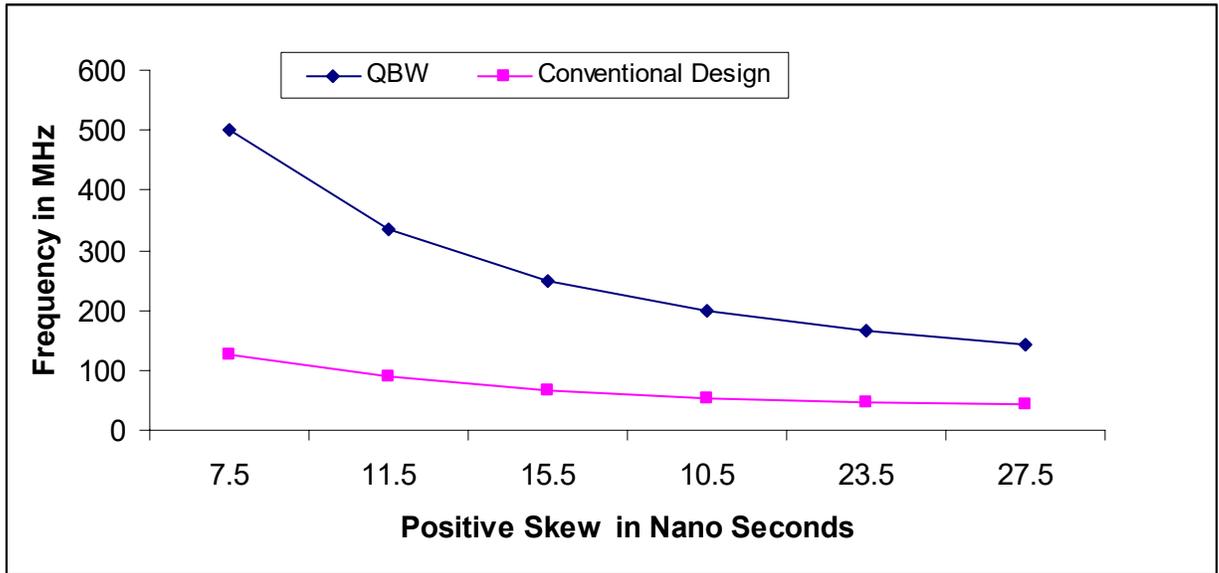


Figure 8. Simulation results showing effect on frequency with the increase in positive skew for unidirectional communication (QBW and Conventional Designs).

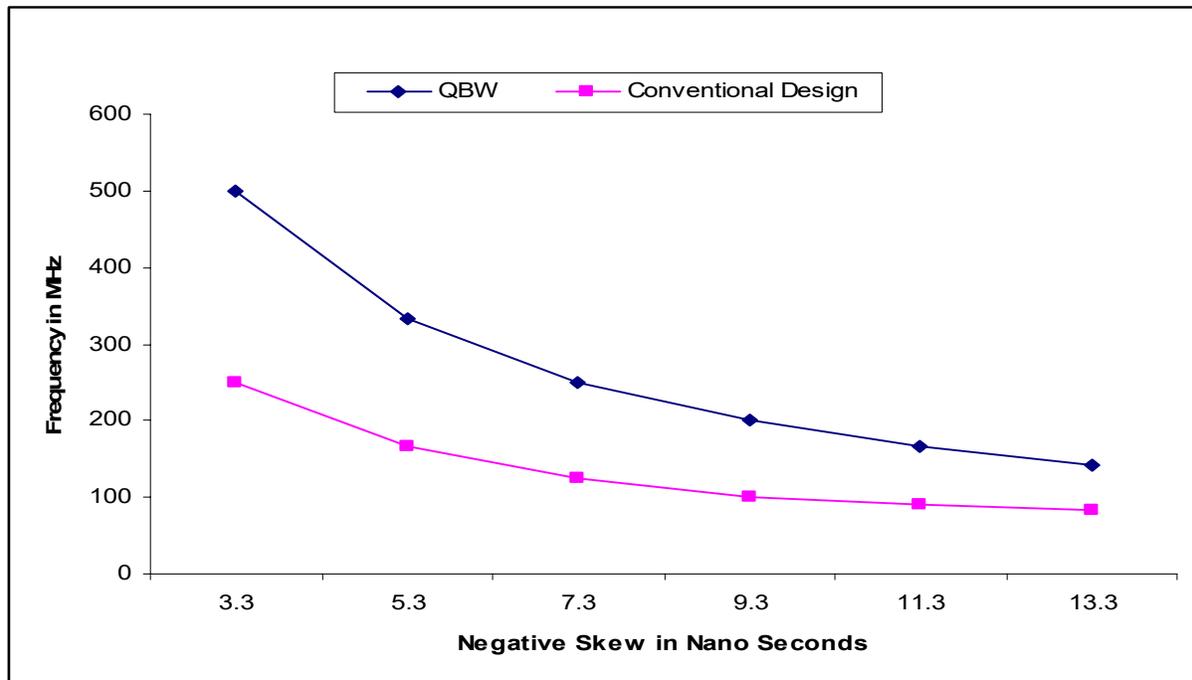


Figure 9. Simulation results showing effect on frequency with the increase in positive skew for unidirectional communication. (QBW and Conventional Designs)

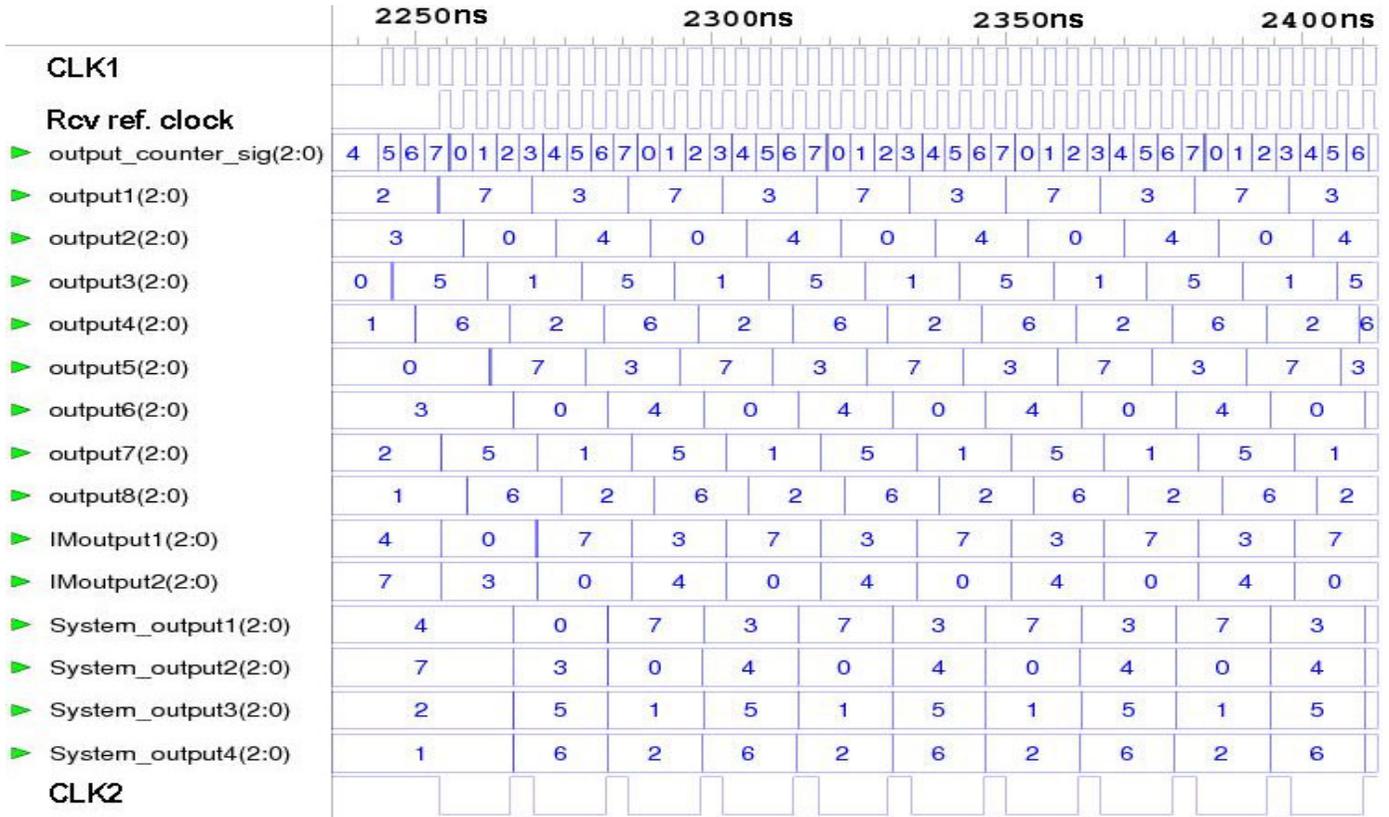


Figure 14. Back Annotated Simulation Results using Xilinx Virtex II-Pro, for the proposed design shown in Fig. 7a, with terminating module working at 250 MHz. and a positive skew of 10 nsec, $> (n/2)T_1$

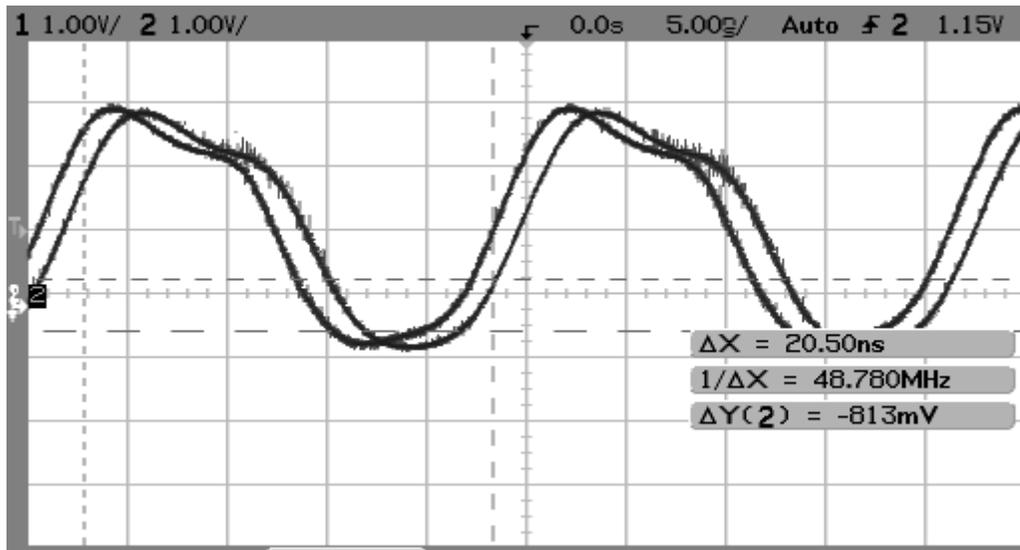


Figure 15. Waveform of a prototype implementation of the proposed F-to-S system-I using 3 bit counter as an example. This experiment was performed using Xilinx Virtex-II Pro FPGA (XC2VP30-7FF896)

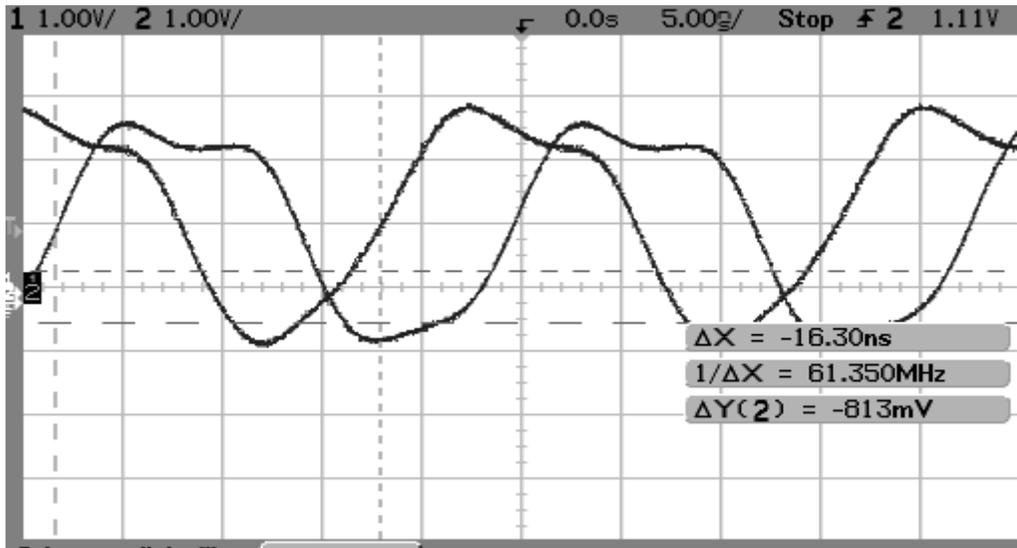


Figure 16. Waveform of a prototype implementation of the proposed F-to-S system-I using 3 bit counter as an example. This experiment was performed using Xilinx Virtex-II Pro FPGA (XC2VP30-7FF896)

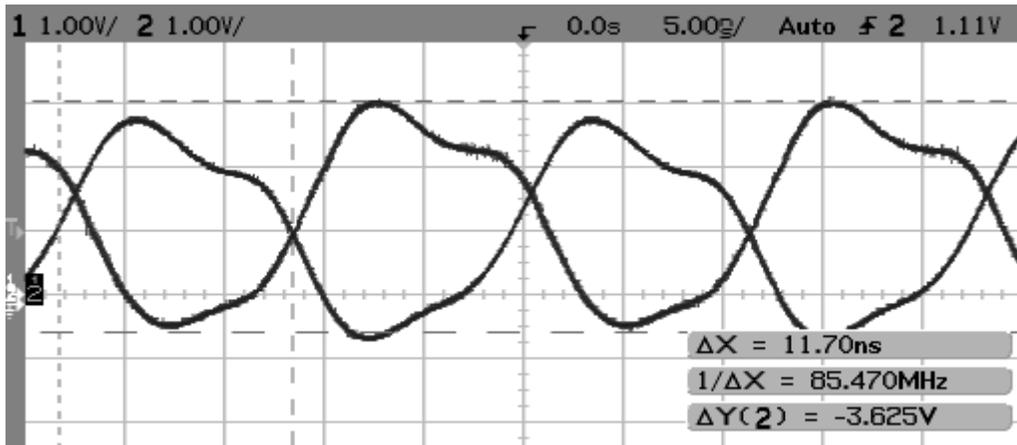


Figure 17. Waveform of a prototype implementation of the proposed F-to-S system-I using 3 bit counter as an example. This experiment was performed using Xilinx Virtex-II Pro FPGA (XC2VP30-7FF896)

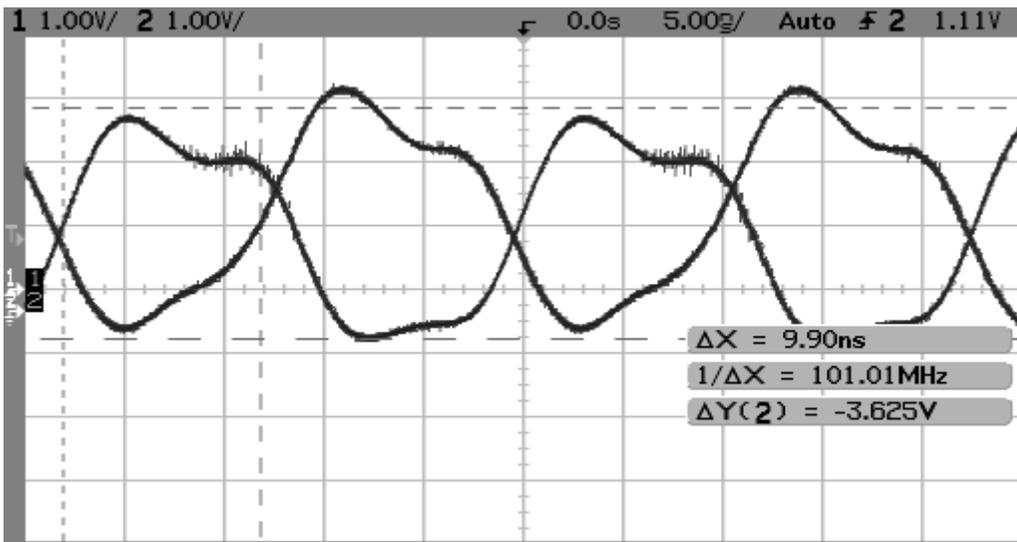


Figure 18. Waveform of a prototype implementation of the proposed F-to-S system-I using 3 bit counter as an example. This experiment was performed using Xilinx Virtex-II Pro FPGA (XC2VP30-7FF896)

L'École Polytechnique se spécialise dans la formation d'ingénieurs et la recherche en ingénierie depuis 1873



École Polytechnique de Montréal

**École affiliée à l'Université
de Montréal**

Campus de l'Université de Montréal
C.P. 6079, succ. Centre-ville
Montréal (Québec)
Canada H3C 3A7

www.polymtl.ca

