School of Electrical, Electronic & Computer Engineering

Robust Synchronization using the Wagging Technique

Mohammed Alshaikh, David Kinniment, and Alex Yakovlev

Technical Report Series

NCL-EECE-MSD-TR-2010-165

December 2010

Contact:
m.s.a.alshaikh@ncl.ac.uk
Alex.Yakovlev@ncl.ac.uk

NCL-EECE-MSD-TR-2010-165

School of Electrical, Electronic & Computer Engineering,
Merz Court,
University of Newcastle upon Tyne,
Newcastle upon Tyne, NE1 7RU, UK
http://async.org.uk/

Robust Synchronization using the Wagging Technique

Mohammed Alshaikh, David Kinniment, and Alex Yakovlev:
School of EECE
University of Newcastle,
Newcastle upon Tyne, NE1 7RU UK
{m.s.a.alshaikh,david.kinniment,alex.yakovlev}@ncl.ac.uk

Abstract—**As integrated circuits technology sizes shrink, variability in process parameters, such as the threshold voltage, are expected to increase and become worse under low supply voltage (V_{DD}). Circuit parameters, such as the propagation delay in logic gates and the resolution time from metastability in flip-flops, will vary more. As a consequence, the synchronizer failure rate would be unpredictable. In this paper, we present the concepts of the conventional cascaded flip-flops synchronizer and wagging synchronizers, particularly how the wagging synchronizer can tolerate such variability. After that, based on simulation results, we show the effects of process and Vdd variability on both synchronizer circuits in terms resolution time constant, MTBF and latency. Then, we propose a control circuit to drive the clock phases of the wagging synchronizer. The control circuit tolerated 6σ process variations up to 3GHz clock frequency and 1.0V V_{DD}.**

Keywords; Wagging, synchronizer, MTBF, latency, resolution time constant, clock control circuit.

I. INTRODUCTION

Increasing unpredictability and vulnerability to process, and voltage variations in sub-nano CMOS process technologies suggest that current optimal designs in cell libraries must be reviewed and refined. Parametric variability is expected to worsen with every new technology node and significantly increase variability in circuit performance, in terms of power consumption and delay [1].

Many VLSI systems and architectures, such as Network-on-Chip (NoC) [2], are designed with multi-synchronous elements, which need to be made more resilient to such variations. Cells which particularly affect the performance of systems on silicon include synchronizers, which affect the latency between independently clocked processors systems, and register bits which require a time to set and hold data. In addition, timing speculation designed to improve performance as in Razor, [3], which involves the late arrival of data to a pipeline register and therefore may require close control of recovery from metastability as well as high throughput in the register bit cell. Currently, a multiprocessor system on chip may have many hundreds of these synchronizers whose performance is critical to the systems performance and reliability. If the reliability is too low with a single clock cycle time for metastability resolution two or more cycles times are used by pipelining the synchronizer in order to maintain the data throughput [4].

The key contribution of this paper is in exploring how the structural technique called *wagging* can improve the design of synchronizers. Wagging is typically known to be a way of alternating the data flow between two or more parallel paths, effectively providing a form of time-division (de-)multiplexing. For example, let's consider a ripple FIFO which consists of a number of buffers, in series, such that the first input is loaded into the first buffer. The second input can then be loaded into the first buffer, only after the first input has been moved into the second, and so on until the output appears at the final buffer. A wagging FIFO [5] on the other hand, consists of a number of buffers in parallel, with an input de-multiplexer to control the sequential loading of the buffers, and an output multiplexer to control the sequential outputting of the buffers. The de-multiplexer loads the first input into the first buffer, followed by the second input into the second buffer, and so on. Here the loading of the second input does not depend on the first input being moved out of the first buffer, as in the case of the ripple FIFO, thus increasing the FIFO's throughput. The cost of wagging in a flip-flop is the extra multiplexing and de-multiplexing circuitry needed to create these parallel paths, but the direct benefit from wagging is the removal of some of the delay associated with inactive paths from the critical path of the data flow. In the case of a synchronizer, it allows recovery of the synchronizer latch to be separated from the outputting of the result so that the available recovery time is greater.

In this contribution, we have:

- studied current synchronizer circuits tolerance to supply voltage and process variations.
- introduced the wagging synchronizer technique and how it could be modified to improve its robustness and reliability.
- evaluated synchronizers implemented as cascaded flip-flops and as wagging structure using different latch circuits.

- proposed a control circuit to clock the wagging synchronizer.

The structure of the paper is as follows. Section II introduces the basic definitions of a synchronizer circuit and the measurement test bench. Section III discusses the current synchronizer designs. Section 0 introduces the wagging technique and its positive effects on the design of synchronizers and proposes how improve its robustness considers the. Section V presents a set of comparisons between the designs discussed in the previous two sections with respect to failure rate and latency. Section VI proposes a clock control circuit for the wagging synchronizer.

II. Synchronizer Parameters Definitions and Measurements Methods

In the following timing parameters for flip-flops are described, and in particular for synchronizers. In general, they include setup/hold time, clock-to-Q propagation delay and D-to-Q propagation delay, where D is the input signal to latch or flip-flop and Q is the output. In addition, metastability time constant, resolution time and metastability window give a measure of reliability for a synchronizer. We also define latency and power consumption.

A. Propagation delay

For a flip-flop, the clock-to-Q propagation delay ($t_{CLK\text{-}Q}$) is the delay time difference measured between the clock triggering edge and the output Q edge, when D-to-clock time ($t_{D\text{-}CLK}$) is wide enough and does not violate the setup and hold times. The value of this delay is a function of the $t_{D\text{-}CLK}$, V_{DD}, temperature, process parameters and the output load [6]. The D-to-Q delay time (t_{DQ}) for a flip-flop is just the sum of the clock-to-Q delay time and the D-to-clock time.

The output time $t_{CLK\text{-}Q}$ can be measured against input time $t_{D\text{-}CLK}$, as shown in Fig. 1. It can be plotted by closing the arrival time of the D-edge signal to the triggering-edge of the clock, while measuring input time difference $t_{D\text{-}CLK}$ and the clock-to-Q delay from 50% to 50% of the edges. This plot gives a clear view of the normal and failure operation regions of the flip-flop.

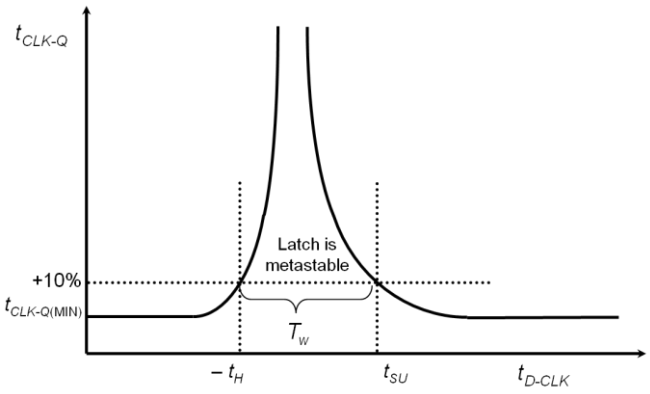

Figure 1. Flip-flop timing parameters.

B. Setup/Hold time

Setup time (t_{SU}) is defined as the minimum time between a D transition and the triggering edge of the clock pulse that will produce a valid output Q. Hold time (t_H) is defined as the minimum time for which the D signal must be kept constant after the clock triggering edge to maintain a stable output Q.

Using Fig. 1, the setup and hold times can be determined at the points where clock-to-Q delay is increased by 10% [7]. Another method would determine the setup and hold times at the optimum D-to-Q delay [6].

C. Metastability window T_w

There are a number of definitions of T_w in the literature [8][9][10][11][14][16]. In the context of using flip-flops as synchronizers, we could define T_w as the region where metastability may occur when the setup and hold times are violated, see Fig. 1. We could say T_w and the setup plus hold time window are related, and they are good approximation of the actual T_w region.

D. Metastability resolution time constant τ

The metastability resolution time constant τ is an important factor in the synchronizer reliability. It was modeled and analyzed by [11] [18] using the small signal model of a cross-coupled inverters, which showed that τ is equivalent to the transconductance g_m and inverse of the node capacitance C of the cross-coupled inverters. When a crossed-coupled inverter latch enters metastability, it takes some time to resolve its metastability, which is directly dependent on the value of τ. The larger value of τ, the slower metastability resolution, and the smaller value of τ value, the faster metastability resolution.

The time constant τ can be determined from the exponential region shown in Fig. 1, for D-to-clock time values within the metastability window, however, this only gives an estimation of the true value of τ, because true metastability occurs within 60fs time difference between the edge of the data signal and the clock edge and should be time stepped at 10fs or less [9]. The slope of the exponential region is a semi-log slope, and can be written as

$$\tau = \frac{t_{C-Q1} - t_{C-Q2}}{\ln\left|\frac{t_{D-CLK2}}{t_{D-CLK1}}\right|} \quad (1),$$

Alternatively, we could use a direct method [9] [18] to find the true value of τ by shortening both latch nodes by switch (see Fig. 2), forcing the latch to be in deep metastability. Then, opening this switch at t_0 and let the latch node voltages diverge away, one to V_{DD}, while the other one to ground. The value of τ is the slope of the nodes difference $V_{A\text{-}B}$ from the start of resolution, using the following equation:

$$\tau = \frac{t_1 - t_2}{\ln\left|\frac{V_2}{V_1}\right|} \quad (2).$$

E. Resolution time

The resolution time or settling time [11][13] is dependent on the remainder of the clock period after deducting the Clock to Q delay of a flip-flop and the setup time of the second, which can be interpreted as "lost time". We could write as:

$$t_R = ClockPeriod - LostTime$$
$$LostTime = t_{SU} + t_{CLK-Q} \quad (3).$$

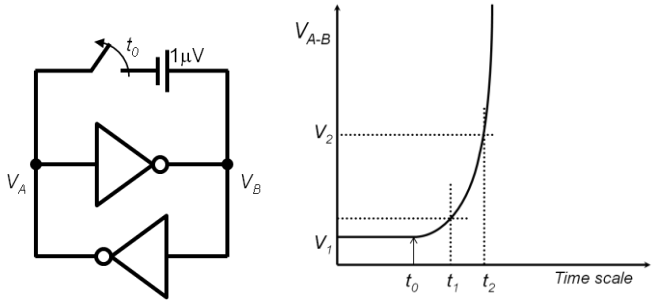

Figure 2. Direct τ measurment.

F. Failure rate (MTBF)

A flip-flop synchronizer failure occurs when input data generated from one clock violates the setup and hold times of the synchronizer. When this occurs the most important parameter is the recovery time of the flip-flop τ, and how much time needs to be allowed in order to reduce the failure rate of the synchronizer to an acceptable level. The failure rate is measured by the Mean Time Between Failures (MTBF), which is related to the synchronizer τ and T_w by the following formula [8]:

$$MTBF = \frac{e^{t_R/\tau}}{T_w f_d f_c} \quad (4),$$

where f_c is the receiver clock frequency and f_d is the data frequency.

G. Latency

Latency can be defined as the time taken by a data signal to go through synchronizer input and arrive at its output. For a synchronizer, latency time is combined of D to Q time in flip-fops in addition to time required for resolution. For example, a two flip-flop synchronizer has latency of two D to Q delays plus available clock time for resolution, or as in (2) below.

$$Latency = 2 \cdot t_{DQ} + t_R \quad (5).$$

Another measure of the synchronizer's effectiveness is the total latency for a required resolution time, usually 30τ to 40τ resolution time [17].

H. Simulation setup

All circuits were implemented in the UMC 90nm process with most p-type transistors held at twice the width of the n-type transistors; so that a buffer width of 1μ means that the n-type transistors were 1μ wide and the p-type transistors 2μ. Other circuits have stated separate widths for n-type and p-type transistors. All transistors used in this paper were sized to minimum channel length in this process technology, i.e. 80nm. We included a double 2μ inverter in series as a load on the output of all circuits to ensure a fair comparison. All input signals were buffered using a double 1μ inverter. A nominal supply voltage of 1.0V was used all over. The simulation setup is shown in Fig. 3. Input signal and its inversion were derived using a double inverter.

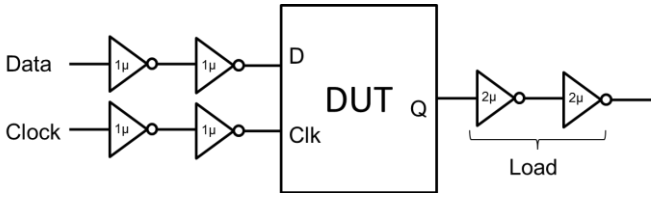

Figure 3. Simulation test bench.

By means of a series of SPECTRE simulations we measured the Clock to Q time from 50% values to 50% of the Clock as well as the τ time constant. The values of τ were simulated using the short circuit method [9]. Voltage supply impact was simulated from 1.0V to 0.3V with a 0.1V step. Process variability simulations were carried out using Monte Carlo statistical analysis under process variations of 3σ. In terms of worst case scenarios, the values of mean (m) plus three standard deviations (3std) were used, and in terms of variability, the values of three times std divided by m were used.

III. Current Synchronizer Designs

A. Jamb latch synchronizer

The jamb latch synchronizer circuit is considered a simple latch with short resolution time, shown in Fig. 4. It is similar to the one in [9][11], but without the reset part, where the inversion of data signal is used instead. A small output buffer was used in order to enhance its resolution time constant. The Jamb latch has total transistors area of 13.5μm×80nm

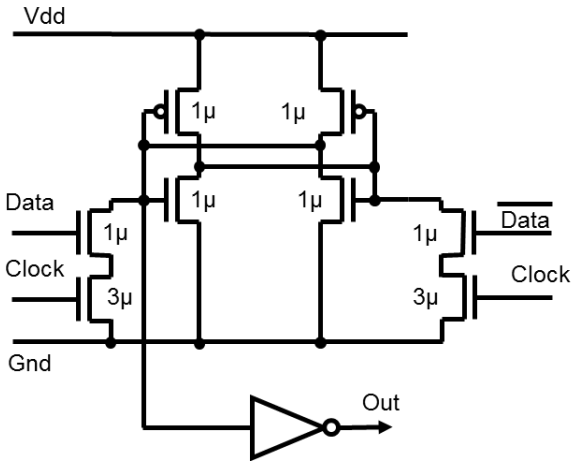

Figure 4. Jamb latch synchronizer.

The circuit of Fig. 4 achieves a τ value of 8.9ps, faster than the flip-flops discussed in [14]. The barrier to further improvement is that the input driving transistors cannot be reduced in size because they would then be unable to pull down the latch nodes. Another problem for all synchronizer circuits in future processes is the lower V_{DD} associated with lower power circuits and processes. Low V_{DD} means low

transistor current at metastable levels giving low g_m and high τ. If V_{DD} falls to below 0.7V the value of τ starts to increase steeply and the Jamb latch performance is severely degraded, and its variability also increases so that the worst case value of τ is more than three times its nominal value. The τ and delay degradation can be seen in TABLE I.

TABLE I. JAMB LATCH SIMULATION RESULTS.

27°C	Jamb Latch					
	τ			$t_{Clk\text{-}Q}$		
V_{DD} (V)	typical	m+3std	3std/m	typical	m+3std	3std/m
1.0	8.90ps	13.76ps	179%	66.8ps	77.2ps	15%
0.9	10.66ps	16.38ps	186%	77.2ps	90.6ps	17%
0.8	14.19ps	22.53ps	186%	93.1ps	112.2ps	20%
0.7	22.35ps	37.79ps	191%	120.2ps	151.9ps	25%
0.6	49.20ps	79.13ps	190%	175.2ps	245.3ps	37%
0.5	132.68ps	265.84ps	236%	326.4ps	656.7ps	87%
0.4	439.75ps	900.37ps	241%	1.07ns	4.80ns	274%
0.3	1.95ns	4.12ns	252%	9.87ns	27.80ns	337%

B. Robust Latch

Fig. 5 shows a robust latch synchronizer circuit, [12], in which the size of the p-type latch transistors has been reduced to 0.25µ width. This allows the Data and Clock transistors to be smaller than in Fig. 4. The feedback gain uses small p-type transistors would normally increase the recovery time constant τ. In this circuit the presence of metastability is detected, and two extra p-type transistors are switched in to increase the current and hence improve g_m. This produces and a τ value of 10.2ps at nominal V_{DD} (1V), see TABLE I and TABLE II. It has a significantly better performance at low voltages than the Jamb latch, e.g. at 0.5V τ of around 55ps for the robust latch compared with about 133ps for the Jamb latch. On the other hand, the Clock to Q delay is in the Jamb latch is faster by 30ps than the Robust latch at nominal 1.0V , whereas at 0.5V worst case the Robust latch delay is limited by the feedback gain with faster with nearly 600ps compared to 656ps for the Jamb latch. The Robust latch consumes a total transistors area of 13.6µm×80nm, which is not far from that of the Jamb latch.

C. Cascaded Flip-flops Synchronizer

A conventional synchronizer is typically composed of two flip-flops connected in series, FF1 and FF2, where each flip-flop has a master and slave latch. Latch circuits of Fig. 4 or Fig. 5 could be used as the master and slave latches of each flip-flop. This is shown at the top of Fig. 6. This configuration is used to reduce the probability of metastable events occurring in the input flip-flop FF1 from progressing into the system. In this configuration there is one clock cycle between capturing the state of the input, resolving metastability, and holding the result in the output flip-flop FF2. If the time available to resolve metastability is not enough, based on (3), a synchronizer failure may occur quite frequently. The amount of time actually available to resolve metastability is less than one clock cycle, due to the clock to Q delay time taken by the master latch, the time taken to pass through the slave latch, and the setup time for the following slave flip-flop, FF2. This time effectively adds up to two D to Q times, which can be a significant part of the clock cycle. If the reliability of the two flip-flop synchronizer is insufficient within a single clock cycle, a third flip-flop is often added as in the bottom of Fig. 6. In this scheme any remaining metastability is passed on from FF2 and FF3 and resolved in the next cycle while another sample in the input is taken by FF1. This maintains the throughput of the synchronizer at the cost of two cycles of latency but has the disadvantage of adding another D to Q time.

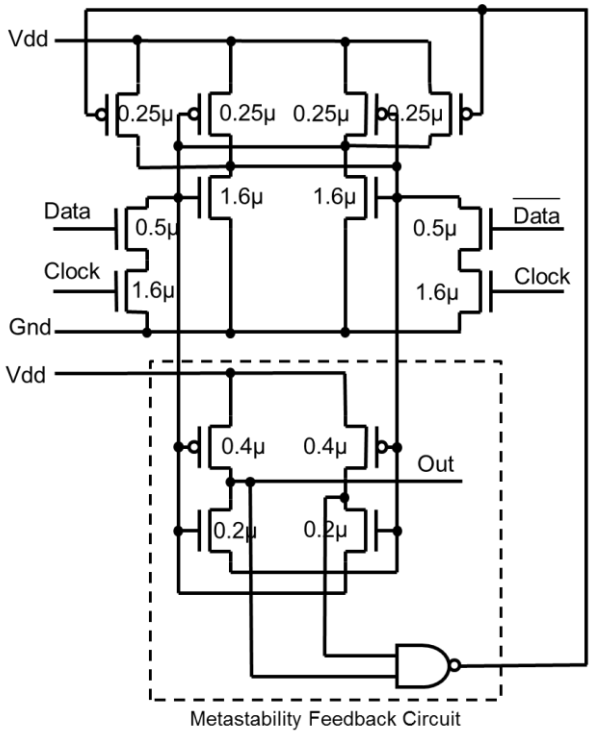

Figure 5. Robust Latch Synchronizer with low buffer to latch ratio.

TABLE II. ROBUST LATCH SIMULATION RESULTS.

27°C	Robust Latch					
	τ			$t_{Clk\text{-}Q}$		
V_{DD} (V)	typical	m+3std	3std/m	typical	m+3std	3std/m
1.0	10.18ps	13.67ps	193%	91.5ps	102.8ps	12%
0.9	11.78ps	15.56ps	186%	105.4ps	120.3ps	14%
0.8	14.45ps	22.83ps	223%	126.6ps	147.8ps	16%
0.7	19.50ps	27.57ps	203%	162.1ps	195.9ps	20%
0.6	30.83ps	35.67ps	179%	230.5ps	296.8ps	27%
0.5	54.93ps	57.30ps	167%	399.9ps	596.8ps	44%
0.4	129.70ps	200.70ps	255%	1.07ns	3.06ns	143%
0.3	611.00ps	2.556ns	426%	6.19ns	20.81ns	191%

In the two flip-flop synchronizer, the available resolving time t_R is limited by the clock cycle T_{CLK} and lost time in the input to output path. This lost time is equivalent to the clock to Q time in FF1 and the setup time in FF2 as shown in Fig. 7. For a number N flip-flop cascaded synchronizer, the available time and latency:

$$t_R = (N-1)\cdot T_{CLK} - N\cdot t_{DQ} \qquad (6),$$

$$Latency = N \cdot t_{DQ} + t_R \quad (7).$$

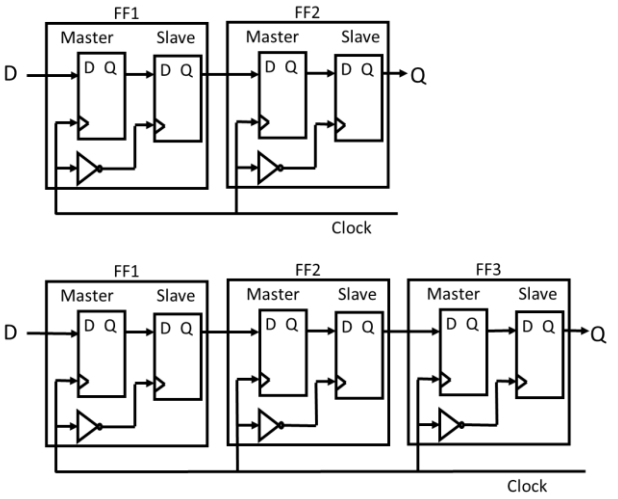

Figure 6. Cascaded flip-flops synchronizer.

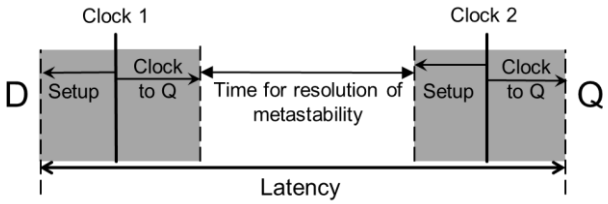

Figure 7. Timing for two flip-flops synchronizer.

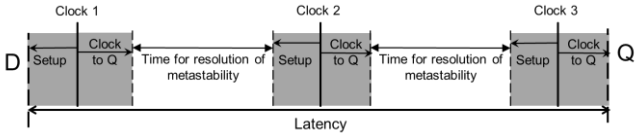

Figure 8. Timing for three flip-flops synchronizer.

IV. Wagging Synchronizer Design

A. Wagging Synchronizer

An alternative structure is proposed in [19] based on the wagging principle. This is shown in Fig. 9 and called the 'wagging' synchronizer. This structure is a three-way wagging synchronizer, which uses three similar paths controlled by three clock phases. Each path has a switched buffer/latch and a switched output buffer, where all buffers drive the output node Q. The input buffer/latch and the output buffer are controlled by two clock phases from the three phases (Clk1, Clk2 and Clk3), as shown in Fig. 10, where each clock phase pulse is equivalent to one clock cycle of the receiver clock frequency and each clock phase is non-overlapping with the others. Each path pair has a different clock signal combination. In Fig. 9, Clk1 drives input buffer I1, latch 1 and buffer B2, whereas Clk2 drives I2, latch 2 and B3, while Clk3 drives I3, latch 3 and B1. All latches are identical and have the same value of τ and setup and delay times. We have used 1µ inverters and 1µ switched inverters to construct the wagging synchronizer in Fig. 9. This gives a total transistors area of 63µm×80nm.

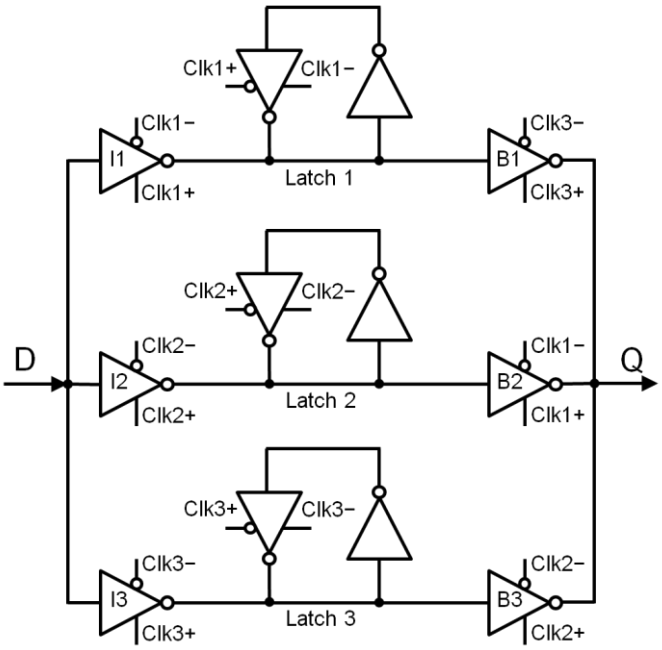

Figure 9. Three way Wagging Synchronizer.

The aim of the wagging synchronizer is to increase the time allowed for metastability to resolve, hence improve the synchronizer reliability. As shown in Fig. 10, when Clk1 is high, latch 1 is set to a new value of input D, while B2 drives the value stored in latch 2 to the output node Q, whereas latch 3 is allowed to recover from any metastability for one clock cycle. Similarly, during Clk2, latch 1 recovers while latch 2 is set and latch 3 drives Q. In Clk3 phase, latch 2 recovers while latch 3 is set and latch 1 drives Q. The only reduction in the clock cycle time allocated to recover from metastability is the clock to Q time of the latch, and this slightly reduced time is always available in one path, while the D input is stored in another and Q is read in a third.

TABLE III shows that the metastability time constant and the delay values for the switched latch in Fig. 9 degrades with supply voltage reduction.

Fig. 11 indicates the available resolving time t_R for the wagging synchronizer is limited by the clock phase width T_{CLK} and lost time in the input to output path. Following setup, all of the time between the fall of Clk1 and the rise of Clk3 is available for the resolution of metastability. One property of the wagging synchronizer is that it can be expanded to *N* way wagging synchronizer (where $N \geq 3$), which expands the available resolution time without penalty of path delays. The resolution time and latency of *N*-way wagging synchronizer can be expressed as below in (8) and (9).

$$t_R = (N-2) \cdot T_{CLK} - t_{DQ} \quad (8),$$

$$Latency = t_{DQ} + t_R \quad (9).$$

The wagging structure can be applied using the Jamb latch circuit instead of the input switched buffer/latch, as shown in Fig. 12. This arrangement provides synchronizer with better performance in terms of latency and failure rate, because it will has the faster resolution time constant of the Jamb latch and the longer resolution time of the wagging structure. In this case the total transistor area equals 45µm×80nm. Later in section V, this design will be evaluated against other designs.

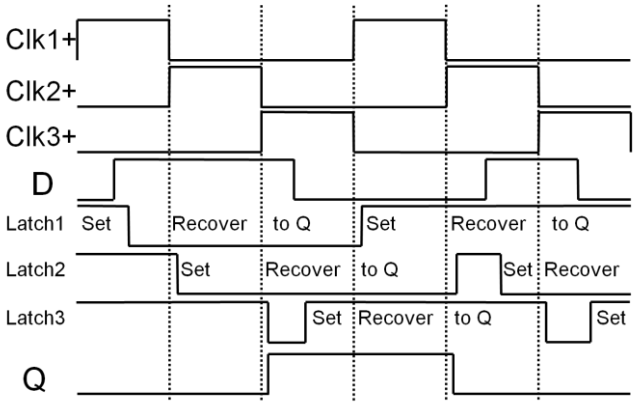

Figure 10. Three way Wagging Synchronizer operation.

TABLE III. WAGGING SYNCHRONIZER SIMULATION RESULTS.

27°C	Wagging Synchronizer Single Path					
	Input Latch τ			*Output Buffer* $t_{Clk\text{-}Q}$		
V_{DD} (V)	typical	m+3std	3std/m	typical	m+3std	3std/m
1.0	10.3ps	14.70ps	73%	31.5ps	35.4ps	13%
0.9	13.4ps	17.67ps	72%	36.0ps	40.9ps	15%
0.8	18.8ps	23.96ps	72%	42.6ps	49.4ps	17%
0.7	32.7ps	40.87ps	81%	53.5ps	63.8ps	21%
0.6	73.1ps	93.11ps	96%	73.8ps	92.3ps	26%
0.5	206.6ps	282.75ps	111%	121.4ps	165.9ps	37%
0.4	722.1ps	991.79ps	120%	283.9ps	462.3ps	61%
0.3	1.125ns	4.195ns	131%	1.259ns	2.605ns	98%

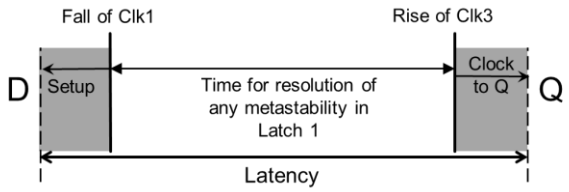

Figure 11. Timing for a three-way wagging synchronizer.

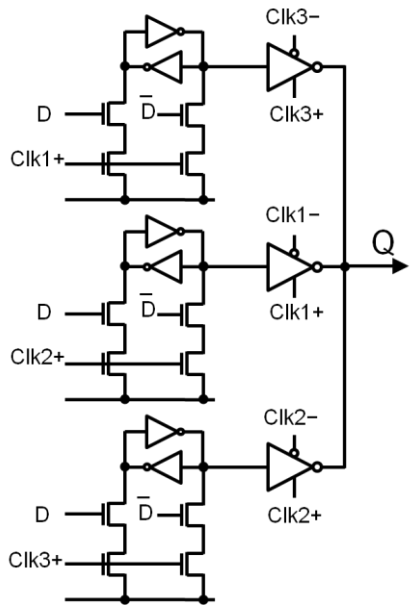

Figure 12. Fast τ Wagging synchronizer circuit.

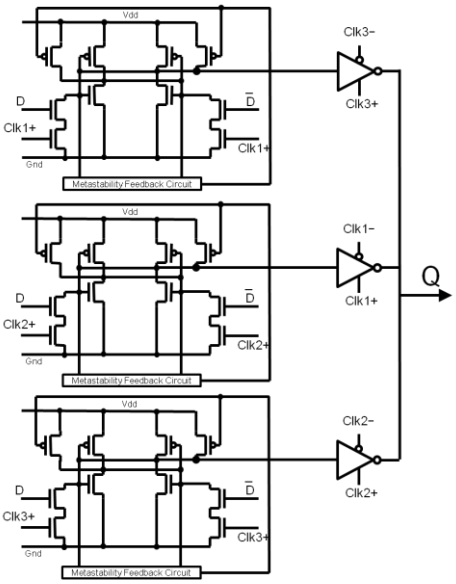

Figure 13. Robust Wagging synchronizer circuit.

B. Robust Wagging Synchronizer

In order to improve the reliability of the wagging synchronizer under low V_{DD}, we could replace all input buffers/latches of Fig. 9 with the robust latches, presented previously in section III. This arrangement is illustrated in Fig. 15. The output of the latch is taken straight from either node of the cross-coupled inverters, which will either drive out Q or inverted Q. The connection showed in Fig. 13 drives the output buffer with the inverted store value to drive out Q which follows D. This design consumes transistor area of 49.8µm×80nm, which smaller than that of Fig. 9.

V. SYNCHRONIZER DESIGN COMPARISIONS

In this section, we show a comparison between five synchronizer structures; two cascading synchronizers that have either Jamb latches or Robust latches; and three 3-way wagging synchronizers where each one uses either Fig. 9, the Jamb latch or the Robust latch. The comparisons are based on failure rate and latency at nominal and worst case conditions and at 1.0V and 0.3V supply voltage. The comparison results are shown in TABLE IV to TABLE VII. In each table, there are two sets of computations; the first shows the computed failure rate (MTBF) using (4) based on the available resolution time (t_R), and the second shows latency based on a required 40τ resolution time. The available resolution time and latency were computed using (6) and (7) for the two flip-flop synchronizer, and (8) and (9) for the wagging synchronizer. The values of T_w in 90nm process at 1.0V is 10ps and at 0.3V is 50ns [16] [13].

Based on TABLE IV, operating at 1.0V supply voltage clock frequency of 1GHz with no process variations, the wagging synchronizer shows significant improvement in the value of MTBF compared to the 2 flip-flop synchronizer, with around 5000x for the jamb latch and 20000 times for the robust latch. On the other hand, latency shows an expected reduction of about 80ps for the jamb latch and 100ps for the Robust latch shows the comparison at low supply voltage (0.3V) and 5MHz clock frequency without any process variations. The failure rate and latency show great improvement for the Jamb latch and Robust latch using the wagging structure.

TABLE IV. COMPARISON AT NOMINAL CONDITIONS.

27°C 1.0V	**Synchronizer Design (f_C =f_D = 1GHz and T_W = 10ps)**				
	2 Flip-flop		**3-Way Wagging**		
Latch	**Jamb**	**Robust**	**Switched**	**Jamb**	**Robust**
τ	8.9ps	10.18ps	10.3ps	8.9ps	10.18ps
t_{DQ}	76.8ps	102ps	41.5ps	76.8ps	102ps
t_R	846.4ps	796ps	958.5ps	923.2ps	898ps
MTBF	6.5×10^{26} years	3.3×10^{19} years	7.8×10^{25} years	3.7×10^{30} years	7.1×10^{23} years
$t_R = 40\tau$	356.0ps	407.2ps	412ps	356.0ps	407.2ps
Latency	509.6ps	611.2ps	453.5ps	432.8ps	509.2ps

TABLE V. COMPARISON AT LOW VDD.

27°C 0.3V	**Synchronizer Design (f_C =f_D = 5MHz and T_W = 50ns)**				
	2 Flip-flop		**3-Way Wagging**		
Latch	**Jamb**	**Robust**	**Switched**	**Jamb**	**Robust**
τ	1.95ns	611ps	1.125ns	1.95ns	611ps
t_{DQ}	59.9ns	56.2ns	51.3ns	59.9ns	56.2ns
t_R	80.2ns	87.6ns	148.7ns	140.1ns	143.8ns
MTBF	2.1×10^{4} years	4.8×10^{48} years	7×10^{43} years	4.9×10^{17} years	4.3×10^{88} years
$t_R = 40\tau$	78.0ns	24.44ns	45.0ns	78.0ns	24.44ns
Latency	198ns	137ns	96.3ns	138ns	80.64ns

TABLE VI. COMPARISON AT WORST CASE AND VDD =1V.

27°C 1.0V	**Synchronizer Design (f_C =f_D = 1GHz and T_W = 10ps) Worst case (m+3std)**				
	2 Flip-flop		**3-Way Wagging**		
Latch	**Jamb**	**Robust**	**Switched**	**Jamb**	**Robust**
τ	13.76ps	13.67ps	14.7ps	13.76p	13.67ps
t_{DQ}	87.2ps	113ps	45.4ps	87.2ps	113ps
t_R	825.6ps	774ps	954.6ps	912.8ps	887ps
MTBF	3.6×10^{11} years	1.3×10^{10} years	5×10^{13} years	2×10^{14} years	4.9×10^{13} years
$t_R = 40\tau$	550.4ps	546.8ps	588ps	550.4ps	546.8ps
Latency	725ps	773ps	633.4ps	638ps	660ps

TABLE VII. COMPARISON AT WORST CASE AND LOW VDD.

27°C 0.3V	**Synchronizer Design (f_C =f_D = 5MHz and T_W = 50ns) Worst case (m+3std)**				
	2 Flip-flop		**3-Way Wagging**		
Latch	**Jamb**	**Robust**	**Switched**	**Jamb**	**Robust**
τ	4.12ns	2.556ns	4.195ns	4.12ns	2.556ns
t_{DQ}	77.8ns	70.8ns	52.6ns	77.8ns	70.8ns
t_R	44.4ns	58.4ns	147.4ns	122.2ns	129.2ns
MTBF	38ms	1.8 hrs	46 years	72 days	222.4×10^{6} years
$t_R = 40\tau$	164.8ns	102.24ns	167.8ns	164.8ns	102.24ns
Latency	320.4ns	243.8ns	220.4ns	242.6ns	173ns

TABLE VI and TABLE VII shows synchronizer performance during worst case condition due to process variations. At 1.0V V_{DD}, the wagging synchronizer MTBF values are reduced from at no process variations, but there still greatly better than the 2 flip-flop synchronizer. Latency showed reduction of nearly 110ps in favor for the wagging synchronizer. At 0.3V V_{DD}, the wagging synchronizer with Robust latches outstands the other structures by over 200 million years of MTBF and 173ns latency. This improvement is due to the feedback circuits of the Robust latch, which increases the transconductance and helps to keep the value of τ small at reduced supply voltage, and the increased resolution time in wagging structure.

The wagging synchronizer can easily be extended from a single cycle resolution time to two cycles by adding a further latch to the three of Fig. 9 and Fig. 13. This then allows one latch to be loaded while two are resolving and the fourth is outputting, thereby improving the reliability of the synchronizer. The effect of this extension on latency is different for the two types of synchronizer considered here. According to (7), a three flip-flop Jamb latch based synchronizer with a 40τ total resolution time, which is split into two periods one between FF1 and FF2 and other between FF2 and FF3, incurs an additional D to Q time, leading to 586ps latency. In contrast, a four Jamb latch wagging synchronizer only requires an additional 4ps for the extra output buffer fan in, or 437ps latency. Therefore, the relative improvement for the wagging synchronizer is 25%.

VI. CONTROL CIRCUIT FOR THE WAGGING SYNCHRONIZER

One requirement of the wagging structure is that clock phases must be ordered and non-overlapping. In order to maintain the relationship between the N clock phases for N-

way wagging synchronizer we propose the solution using a Signal Transition Graph (STG) of the required functionality in Fig. 14. In this STG, the signal Clk is the input clock signal, which indicates the receiver frequency, whereas signals Clk1, Clk2 and Clk3 and the output clock phases required to drive the 3-way wagging synchronizer. Internal signals S1, S2 and S3 are based on a timing assumption [15] that the negative pulse of the input clock signal long enough to make two signal transitions before the rising edge of the second clock cycle, i.e. the transitions {Clk–/1 → S1+ → S3– → Clk+/2} must maintain their sequence.

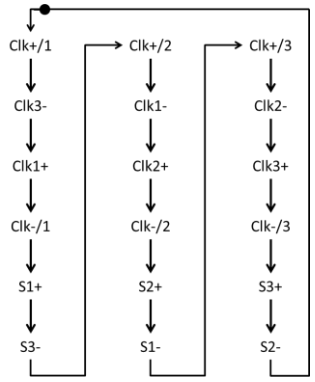

Figure 14. STG for the clocking control circuit (N=3).

The STG in Fig. 14 was synthesized and the sequential circuit in Fig. 15 is proposed to control the clocking of a 3-way wagging synchronizer. This circuit implementation uses symmetric optimized OAI gates and inverters which has symmetric delays between signals transitions. In other words, the time required from Clk+/1 to Clk1+ is equivalent to the time from Clk+/2 to Clk2+ as well as the time from Clk+/3 to Clk3+. This is also true for the case between transitions from Clk–/1 to S3–, Clk–/2 to S1– and Clk–/3 to S2–. The timing diagram of the control circuit signals with data signal D and output Q in the wagging synchronizer are shown in Fig. 16. The output clocks of this circuit are buffered to drive the wagging synchronizer.

The circuit has a minimum functional frequency due to the timing assumption in the STG. This timing restriction between Clk–/1 and Clk+/2 has to be at least 130ps at nominal operation. This gives a minimum clock period of 260ps (f_{CLK}≈ 3.8GHz) at 1.0V supply voltage and no process variations. The circuit produces a delay of 85ps to produce a clock signal; from the rising edge of the input clock the rising edge of the next clock phase. A 53ps proportion of the 85ps delay is make sure that the previous clock phase signal has fallen to 0 before the rise of the next clock phase signal. This is to maintain the non-overlapping output clock signals. The pulse width of the output clocks is less than the clock cycle by 53ps, which is the delay between the adjacent clock phases. The cycle of the clock phase is three times that of the original input clock, i.e. 780ps at the minimum clock period in our case.

The proposed clock control circuit was tested under six-sigma process variations at 1.0V supply voltage using Monte Carlo simulations to show the acceptable minimum frequency of operation. The simulation results are shown in TABLE VIII. The control circuit showed 100% tolerance of failure at input clock period of 325ps (≈3GHz) with insignificant 6-sigma variations. We conclude this is the maximum input clock frequency to produce high yield clocking signals driving a 3-way wagging synchronizer.

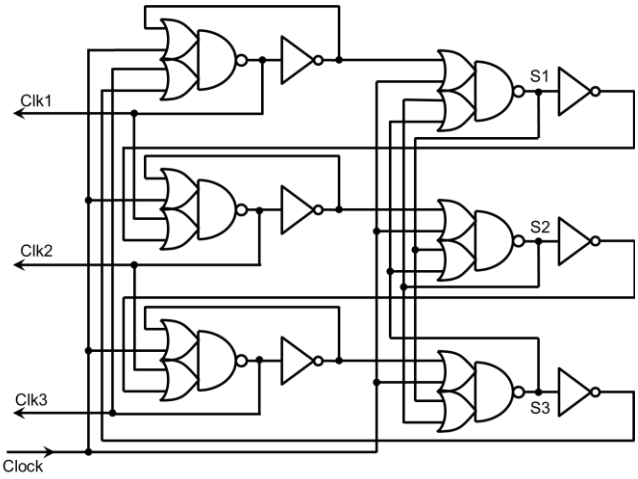

Figure 15. The proposed circuit.

The design of the control circuit can be expanded for a number N clock phases. This can be done by adding extra sequence in the STG diagram, shown in Fig. 14, for each additional signal of the clock phases. For example, if we intend to design a control circuit for a 4-way wagging synchronizer, we could replace the sequence {S2– →Clk+/1 → Clk3– } in the STG by the following sequence: {S2– →Clk+/4 → Clk3– → Clk4+ → Clk–/4 → S4+ → S3– →Clk+ → Clk4–}. Then, we could synthesize a new circuit in a similar fashion to the circuit presented in Fig. 15. The cycle of the clock phases in this case is four times that of the input clock signal.

TABLE VIII. CLOCK CONTROL CIRCUIT MONTE CARLO RESULTS.

27°C 1.0V	**Output Clock Phases (Clk1, Clk2, Clk3)**				
Input Clk period	**Pulse width**		**Phase Period (3×Clk)**		**Overall Yield**
	m-6std	**6std/m**	**m-6std**	**6std/m**	
260ps	391ps	90%	831.7ps	6.33%	74.8%
300ps	293ps	22%	904.2ps	0.46%	99.5%
305ps	294ps	19%	916.6ps	0.17%	99.7%
325ps	310ps	15%	975.0ps	0.002%	100.0%
350ps	331ps	12%	1.050ns	0.002%	100.0%
400ps	378ps	9%	1.200ns	0.0019%	100.0%

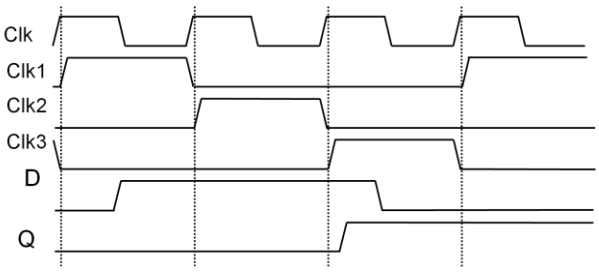

Figure 16. Timing diagram of the clocking signals with the wagging synchronizer.

VII. CONCLUSION

The wagging synchronizer does not suffer from long latencies or the additional complication of master and the slave latches. The idea of wagging is applied to the synchronizer structure, only a single latch is necessary to capture the state of the input. This significantly shortens the path from unsynchronized input to synchronized output when compared with the conventional two flip-flop synchronizer. The proportion of time available for resolution of metastability is also increased, and the total latency reduced by 15% compared with a two flip-flop synchronizer and 25% for a three flip-flop synchronizer. This allows that a reliable wagging synchronizer can be built with significantly lower latency than more conventional designs. The robustness of a wagging synchronizer can be improved by using a Robust latch instead of typical ones. This improves the robustness at low supply voltages and under process variations when compared to other structures. A clock control circuit was proposed, which provides the clock signals of the wagging structure, It showed reliable operation of sequencing the clocks under process variations at clock frequency upto 3GHz.

ACKNOWLEDGMENT

The authors acknowledge the support by a scholarship from Umm Al-Qura University at Saudi Arabia, and UK EPSRC research grants EP/C007298/1, and EP/G066361/1 for the work described here.

REFERENCES

[1] "International Technology Roadmap for Semiconductors," http://public.itrs.net 2008 Update.

[2] J. Sparso, "Asynchronous design of networks-on-chip," *Norchip, 2007*, pp.1-4, Nov. 2007.

[3] D. Ernst, et al., "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," in *Proc. IEEE/ACM Intl. Symp. Microarchitecture (MICRO-36)*, pp. 7-18, Dec. 2003.

[4] J. Jex and C. Dike, "A fast resolving BiNMOS synchronizer for parallel processor interconnect," *Solid-State Circuits, IEEE Journal of* , vol.30, no.2, pp.133-139, Feb 1995.

[5] C.H. Kees van Berkel, M. Rem, R. Saeijs, "VLSI programming", in: *IEEE International Conference on Computer Design: VLSI in Computers and Processors*, 1998, pp. 152–156.

[6] V. Stojanovic and V. G. Oklobdzija, "Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems," *IEEE Journal of Solid State Circuits*, Vol. 34, pp536-548, 1999.

[7] B. Rebaud, M. Belleville, C. Bernard, M. Robert, P. Maurine and N. Azemard, "A Comparative Study of Variability Impact on Static Flip-Flop Timing Characteristics," in *Proc. ICICDT 2008*, pp. 167 – 170.

[8] D. J. Kinniment and D. B. G. Edwards, "Circuit Technology in a large computer System," *Proc. of the conference on Computers-Systems and Technology*, London, October 1972, pp. 441-449.

[9] C. Dike and E. Burton, "Miller and Noise Effects in a Synchronizing Flip-Flop," *IEEE Journal of Solid State Circuits*, Vol. 34 No. 6, pp. 849-855, June 1999.

[10] I. W. Jones, S. Yang and M. Greenstreet, "Synchronizer Behavior and Analysis," *ASYNC 2009 15th IEEE Symposium on Asynchronous Circuits and Systems*, May 2009 pp117-226.

[11] D. J. Kinniment, A. Bystrov and A. V. Yakovlev, "Synchronization circuit performance," *Solid-State Circuits, IEEE Journal of*, vol. 37, pp. 202-209, 2002.

[12] J.Zhou, D.J.Kinniment, G. Russell and A. Yakovlev, "A Robust Synchronizer Circuit," *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'06),* pp442-443, March 2006.

[13] J. Zhou, M. Ashouei, D. Kinniment, J. Huisken and G. Russell,, "Extending Synchronization from Super-Threshold to Sub-threshold Region," *Asynchronous Circuits and Systems (ASYNC), 2010 IEEE Symposium on*, pp.85-93, May 2010.

[14] D. Li, P. Chuang and M. Sachdev, "Comparative analysis and study of metastability on high-performance flip-flops," *Quality Electronic Design (ISQED), 2010 11th International Symposium on*, pp.853-860, March 2010.

[15] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno and A. Yakovlev. Logic Synthesis of Asynchronous Controllers and Interfaces, Springer Series in Advanced Microelectronics, vol. 8, Springer, 2002.

[16] D. Kinniment, K. Heron and G. Russell, "Measuring deep metastability," Proceedings of the 12th IEEE International Symposium on Asynchronous Circuits and Systems, March 2006.

[17] D. Kinniment, Synchronization and arbitration in digital systems, Wiley & Sons, 2007.

[18] S. Beer, R. Ginosar, M. Priel, R. Dobkin and A. Kolodny, "The Devolution of Synchronizers," *Asynchronous Circuits and Systems (ASYNC), 2010 IEEE Symposium on* , pp.94-103, May 2010.

[19] M. Alshaikh, D. Kinniment and A. Yakovlev, "A Synchronizer Design Based on Wagging," *accepted in International Conference on Microelectronics (ICM) 2010 ,IEEE, Dec. 2010* .