Algorithms to get the maximum operation frequency for skew-tolerant clocking schemes

D. Guerrero, M. Bellido, J. Juan, A. Millán, P. Ruiz, E. Ostúa and J. Viejo*

Group of Digital Design, Microelectronic Institute of Seville – CNM &
Dep. of Electronic Technology, University of Seville

ABSTRACT

Nowadays it is not possible to neglect the delay of interconnection lines. The die size is rising very fast, and the delay of the interconnection lines grows quadrically with it. Also, the fact that the gate delay keeps getting smaller increases the importance of the delay of the interconnection lines. The delay of the clock lines is specially important: If the clock skew is underestimated and the clocking scheme is not properly designed, then the system may not work under any clock frequency.

In this paper we evaluate the timing performance of three skew-tolerant clocking schemes. These schemes are the well known Master-Slave clocking scheme (MS) and two schemes developed by the authors: Parallel Alternating Latches Clocking Scheme (PALACS) and four-phase Parallel Alternating Latches Clocking Scheme (four-phase PALACS). To carry out these analysis, the authors introduce new algorithms to obtain the clock waveforms required by a synchronous sequential circuit. Separated algorithms were developed for every clocking scheme. The algorithms take a set of timing parameters as input and generate a chronogram of the circuit trying to minimise the clock period but ensuring the timing restrictions of the circuit are met for a given clock skew. Using these algorithms is it possible to draw a representation of the computation frequency as a function of the clock skew for every clock scheme. Once we have estimated the timing parameters and the skew, these representations can help us to choose the best clocking scheme for our design.

Keywords: Clock skew tolerance, high speed CMOS design, CAD circuit design

1. INTRODUCTION

The evolution in the VLSI digital circuits design makes it mandatory to pay special attention to the clocking scheme used to implement the system and to the clock generation and distribution over the full system. While the gate size and, as a consequence, the gate delay is getting smaller, the die size is rising. Since the delay in interconnection lines increases quadratically with the line length, it becomes longer than gate delay. Because of that the skew increases significantly. Due to the clock skew, the simplest clocking scheme based on edge-triggered flip-flop should not be used for high-speed designs[1,2,3,4]. This is illustrated in Figure 1. As we can see, if the clock skew is very long and the logic circuit is fast enough, the active edge of the clock can reach flip-flop 2 too late, i.e. near the instant when its input is going to change. Note that this problem can not be solved by enlarging the clock cycle[4]. To solve such a problem, it has been suggested that the clock signal should reach first the registers at the end of the data path. Clock skew could cause malfunction anyway, as we can see in Figure 2. If the clock skew is very long, flip-flop 2 could be triggered too early. This could be solved by enlarging the clock cycle, but rerouting the clock path is not a solution if feedback exists in the data path.

In order to prevent the clock skew from causing malfunction, a two-phase clocking scheme may be used. Two-phase clocking systems use two distinct clocks generated from the main clock at the last buffering stage. An example of two-

* {guerre,bellido,jjchico,amillan,pruiz,ostua,julian}@dte.us.es

phase clocking scheme is the two-phase Master-Slave clocking scheme (MSCS), which uses Master-Slave structures to implement the register block. A Master-Slave register working and its chronogram is shown in Figure 3, where it is assumed that the registers are transparent at the high level of the load signal. During the active level of the master clock signal, the values generated by the combinational logic are loaded in the master registers. During the following active level of the slave clock signal, those values are loaded in the slave registers and became the current state. If the input signals do not change between the falling edge of the master clock signal and the raising edge of the slave clock signal, the Master-Slave structure operates from an external point of view like a type D flip-flop register triggered by the raising edge of the slave clock signal. The harmful effects of the clock skew can be prevented by separating enough the active levels of the Master and slave clock signals (i.e. by enlarging $t_{separation}$ in Figure 3).

In this work we present another two skew tolerant clocking schemes called generically Parallel Alternating Latches Clocking Scheme (PALACS)[5,6]. These schemes are based on the one-phase double-edge triggered clocking scheme[3,7]. The main objective of this work is to evaluate the performance in terms of speed of these three clocking schemes (MSCS, two-phase PALACS and four-phase PALACS).

Targeting this objective, this paper is organised as follows: In the next section we will see the PALACS clocking schemes. In section 3 we introduce the algorithms to obtain the required waveforms for each clocking scheme. In section 4 we will check the correctness of these algorithms and will use them to compare the operation speed of each scheme. Finally we will summarise the conclusions.

2. PARALLEL ALTERNATING LATCHES CLOCKING SCHEME

A remarkable alternative to the one-phase single-edge triggered flip-flop clocking scheme is the one-phase double-edge triggered flip-flop clocking scheme[3,7]. This scheme uses the flip-flop shown in Figure 4, which is triggered by both, falling and rising transitions. The power consumption of the clock distribution network in this scheme is smaller than using single-edge triggered flip-flops since there is an only clock transition per computation cycle.

We could say that one-phase single-edge triggered flip-flop clocking scheme is a particular case of the MSCS where the slave clock signal is obtained by inverting the master clock signal, i.e. a particular case where the non-overlapping time between the clock signals is zero. The general MSCS provides tolerance to an arbitrary skew by enlarging the non-overlapping region.

Analogously, we have generalised the one-phase double-edge triggered flip-flop clocking scheme to get skew tolerance by using separated clock signals[5]. In this section we describe two generalizations of the double-edge approach.

2.1 Two-phase PALACS

The first generalisation of the one-phase double-edge triggered flip-flop clocking scheme is the two-phase Parallel Alternating Latches Clocking Scheme (two-phase PALACS) depicted in Figure 5. As we can see, each memory cell consists of two latches connected in parallel with the same input and a switch at the output of each latch whose outputs are connected. The loads of both latches are controlled by separate phases, and the switches are also controlled by opposite phases. This scheme, unlike the Master-Slave scheme, allows reading and writing the register block simultaneously during the active level of each clock phase. In effect, while clock signal CLK0 is active, latch 0 loads the current input while latch 1 holds the previous input. The latch 1 data is read in the active phase of CLK0, since its switch is controlled by CLK0. When CLK0 becomes inactive, latch 0 stops being transparent. Then both phases remain inactive a time interval long enough to avoid clock-skew related problems. During this interval both switches are in high impedance (H.I.) state, but the previous data value remains loaded at the switches output due to parasitic capacitances. When CLK1 activates, the read-write mechanism works again, but both latches alternate their function, i.e. latch 1 loads a new value while latch 0 is read. We could say that this clocking scheme is the two-phase counterpart of the one-phase double-edge triggered flip-flop clocking scheme[3,7].

The most important advantage of PALACS versus MSCS is that the clock frequency is reduced by 50% for the same data rate. This reduces the power consumed by the clock distribution network. In effect, with the PALACS, the number

of clock transitions is two per computation cycle whereas in MSCS it is four. This means that their power dissipation can be reduced up to 50%. Another interesting advantage is that, for some implementations, the propagation delay of the PALACS memory cell is smaller than the propagation delay of the Master-Slave since in MSCS the input signal has to propagate through two latches whereas in PALACS it has to propagate through one latch and a switch (whose delay is usually smaller than the delay of a latch). This produces an improvement in the operation speed of the system.

2.2 Four-phase PALACS

A drawback of the two phase PALACS is that the raising edges of the load control signals are hard edges[8]. This means that, regardless of the instant when a data item reaches a latch output, it will not keep propagating through the circuit until the load control signal of the opposite latch receive the next raising edge. In pipelined designs, hard edges imply that cycle time must be as long as the delay of the slowest segment, so improvements in the delay of other segments are helpless. On the other hand, in hard edge-free systems some segments can have a delay longer than the cycle time if time borrowing[8] is used. Time borrowing techniques compensate the time exceeded in the slow segments for the time saved in the fast segments. The possibility of employing time borrowing gives more freedom to the designer, so it is desirable to remove hard edges. This is the purpose of the four-phase PALACS shown in Figure 6.

In this scheme, the load control signals and the output enable control signals are not the same. So, a data item at the output of a latch can begin to propagate through the circuit even if that item has not been latched yet provided that the contamination delay of the logic circuit is long enough. This was not possible in the two-phase PALACS since the active levels of the load control signal of a latch and the control signal of its associated switch should not be overlapped. As we will in section 4, this makes it possible to improve the timing performance of the four-phase PALACS respect to the two-phase PALACS even without using time borrowing techniques.

3. ALGORITHMS TO GENERATE THE REQUIRED CLOCK WAVEFORMS

The clock signals involved in any clocking scheme need be generated according to general timing parameters including logic delay, setup and hold times and maximum possible clock skew. In order to compare the speed of the clocking schemes presented so far, the process of generating the required clock waveform for a given upper bound of the clock skew and a given circuit has been automated. Several algorithms for that task that have been implemented in a tool. Given a general synchronous sequential circuit and a set of timing parameters, the algorithms generate a chronogram where the conditions to ensure the correct operation of the circuit are met.

Starting in a stable initial state, the signals begin to change affected by the delay of the components. The algorithms set when every signal transition happens minimising the clock period while ensuring the circuit works properly. This is done iteratively till the chronogram becomes periodic. From this chronogram, parameters like non-overlapping time and clock frequency are obtained. This makes it possible to analyse the operation speed as a function of clock-skew.

3.1 Algorithm for the Master-Slave clocking scheme

To generate the chronogram we will suppose that at the beginning the slave latches have held the initial state for a time long enough so the next state signal is already stable and valid at the input of master latches. We will also assume that the first active pulse happens at the master clock. The meaning of the variables and parameters used is the following (see Figure 7):

- Timing parameters of the circuit
  - t_{skew0r}: Upper bound on the skew for the raising transitions of the clock of the master latches
  - t_{skew0f}: Upper bound on the skew for the falling transitions of clock of the master latches
  - $t_{skew1ra}$: Upper bound on the skew for the raising transitions of clock of the slave latches
  - t_{skew1f}: Upper bound on the skew for the falling transitions of clock of the slave latches
  - LC_{max}: Upper bound on the delay of the logic circuit

- LC_{min}: Contamination delay of the logic circuit, i.e. a lower bound on the amplitude of the time interval where the output is stable despite the input is no longer valid
- $Lmaster_{DQmax}$: Upper bound on the delay of a master latch when its load control signal is active and its input changes to a valid value
- $Lmaster_{CQmax}$: Upper bound on the delay of a master latch when its input is stable and valid and its load control signal activates
- $Lmaster_{CQmin}$: Contamination delay of a master latch when it load control signal activates
- $Lslave_{DQmax}$: Upper bound on the delay of a slave latch when its load control signal is active and its input changes to a valid value
- $Lslave_{CQmax}$: Upper bound on the delay of a slave latch when its input is stable and valid and its load control signal activates
- $Lslave_{CQmin}$: Contamination delay of a slave latch when it load control signal activates
- $t_{setupmaster}$: Setup time of a master latch
- $t_{setupslave}$: Setup time of a slave latch
- $t_{holdmaster}$: Hold time of a master latch
- $t_{holdslave}$: Hold time of a slave latch
- $pw_{minmaster}$: Minimum active pulse width at the load control signal of a master latch to ensure that the data will be latched
- $pw_{minslave}$: Minimum active pulse width at the load control signal of a slave latch to ensure that the data will be latched

❖ Geometrical variables of the algorithm

- S[i]: Upper bound on the instant of the computation cycle i where the state signals have reached their new value
- NS[i]: Upper bound on the instant of the computation cycle i where the next state signals have reached their new value
- QM[i]: Upper bound on the instant of the computation cycle i where the output of the master latches have reached their new value
- CLK_{0r}[i]: Upper bound on the instant of the computation cycle i where the load control signal of the master latches activates
- CLK_{0f}[i]: Upper bound on the instant of the computation cycle i where the load control signal of the master latches deactivates
- CLK_{1r}[i]: Upper bound on the instant of the computation cycle i where the load control signal of the slave latches activates
- CLK_{1f}[i]: Upper bound on the instant of the computation cycle i where the load control signal of the slave latches deactivates

❖ Output parameters

- W_0: Active pulse width of the master clock signal
- W_1: Active pulse width of the slave clock signal
- displacement: Time elapsed from the activation of the master clock signal to the activation of the slave clock signal
- T: Clock signals period

Supposing that the load control signals are active in high, the algorithm for the Master-Slave scheme is the following:

/*set the initial state of the chronogram*/

$CLK_{1r}[0] \leftarrow 0$
$CLK_{1f}[0] \leftarrow pw_{minMaster} + t_{skew1r}$

$QM[0] \leftarrow Lmaster_{CQmax}+t_{skew1r}$
$CLK_{0r}[0] \leftarrow CLK_{1f}[0]+t_{skew1f}+t_{holdMaster}-L_{slaveCQmin}-LC_{min}$
$S[0] \leftarrow \max\{CLK_{0r}[0]+t_{skew0r}+L_{slaveCQmax}, QM[0]+L_{DQmax}\}$
$CLK_{0f}[0] \leftarrow \max\{QM[0]+t_{setupSlave}, CLK_{0r}[0]+t_{skew0r}+pw_{minslave}\}$
$NS[0] \leftarrow S[0]+LC_{max}$

/*draw the chronogram iteratively till it becomes periodic*/

$i \leftarrow 0$
DO
$i \leftarrow i+1$
$CLK_{1r}[i] \leftarrow CLK_{0f}[i-1]+t_{skew0f}+t_{holdSlave}-L_{masterCQmin}$
$QM[i] \leftarrow \max\{CLK_{1r}[i]+t_{skew1r}+L_{masterCQmax}, NS[i-1]+L_{masterDQmax}\}$
$CLK_{1f}[i] \leftarrow \max\{NS[i-1]+t_{setupMaster}, CLK1r[i]+t_{skew1r}+pw_{minMaster}\}$
$CLK_{0r}[i] \leftarrow CLK_{1f}[i]+t_{skew1f}+t_{holdMaster}-L_{slaveCQmin}-LC_{min}$
$CLK_{0f}[i] \leftarrow \max\{QM[i]+t_{setupSlave}, CLK_{0r}[i]+t_{skew0r}+pw_{minSlave}\}$
$S[i] \leftarrow \max\{CLK_{0r}[i]+t_{skew0r}+L_{slaveCQmax}, QM[i]+L_{slaveDQmax}\}$
$NS[i] \leftarrow S[i]+LC_{max}$
UNTIL $CLK_{1r}[i]-CLK_{1r}[i-1]=CLK_{1f}[i]-CLK_{1f}[i-1]=QM[i]-QM[i-1]=CLK_{0r}[i]-CLK_{0r}[i-1]=CLK_{0f}[i]-CLK_{0f}\}[i-1]=S[i]-S[i-1]=NS[i]-NS[i-1]$

/*set some output parameters*/

$W_0 \leftarrow CLK_{0f}[i]-CLK_{0r}[i-1]$
$W_1 \leftarrow CLK_{1f}[i]-CLK_{1r}[i-1]$
$displacement \leftarrow CLK_{1r}[i]-CLK_{0r}[i]$
$T \leftarrow CLK_{1r}[i]-CLK_{1r}[i-1]$

3.2 Algorithm for the two-phase PALACS

To generate the chronogram in the two phase PALACS, we will suppose that at the beginning the latches labelled with 1 have held the initial state for a time long enough so that the state is already at their output. We will also assume that the first active pulse happens at the clock 0. The meaning of the variables and parameters used is the following (see Figure 8):

- Timing parameters of the algorithm
  - t_{skewr}: Upper bound on the skew for a rising transition of a clock signal
  - t_{skewf}: Upper bound on the skew for a falling transition of a clock signal
  - LC_{max}: Upper bound on the delay of the circuit
  - LC_{min}: Contamination delay of the circuit
  - K_{cmax}: Upper bound on the delay of a switch when its input is valid and it activates
  - K_{cmin}: Contamination delay of a switch when it activates
  - K_{imax}: Upper bound on the delay of a switch when its input is valid and it activates
  - L_{DQmax}: Upper bound on the delay of a latch when its load control signal is active and its input changes
  - L_{CQmax}: Upper bound on the delay of a latch when its input is valid and its load control signal activates
  - t_{setup}: Setup time of the latches
  - t_{hold}: Hold time of the latches
  - pw_{min}: Minimum active pulse width at the load control signal of a latch to ensure that the data will be latched

- Geometrical variables of the algorithm

- S[i]: Upper bound on the instant of the computation cycle i where the state signals have reached their new value
- NS[i]: Upper bound on the instant of the computation cycle i where the next state signals have reached their new value
- Q[i]: Upper bound on the instant of the computation cycle i where the output of the latches labelled with (i mod 2) have reached their new value
- $CLK_r[i]$: Upper bound on the instant of the computation cycle i where the load control signal of the latches labelled with (i mod 2) activates
- $CLK_f[i]$: Upper bound on the instant of the computation cycle i where the load control signal of the latches labelled with (i mod 2) deactivates

❖ Output parameters

- W: Active pulse width of the clock signals
- T: Clock signals period

We will assume that if the input of a latch gets valid at instant t_i while its load control signal activates at instant t_c then the new value of the input appears at the output at an instant no later than $\max\{t_i+K_{imax}, t_c+K_{cmax}\}$[2]. Supposing that the control signals are active in high, the algorithm is the following:

/*set the initial state of the chronogram*/

$CLK_r[0] \leftarrow 0$
$S[0] \leftarrow t_{skewr}+K_{cmax}$
$NS[0] \leftarrow S[0]+LC_{max}$
$CLK_f[0] \leftarrow \max\{NS[0]+t_{setup}, pw_{min}+t_{skewr}\}$
$Q[0] \leftarrow \max\{CLK_r[0]+t_{skewr}+L_{CQmax}, NS[0]+L_{DQmax}\}$

/*draw the chronogram iteratively till it becomes periodic*/

$i \leftarrow 0$
DO
$i \leftarrow i+1$
$CLK_r[i] \leftarrow CLK_f[i-1]+t_{skewf}+\max\{0, t_{hold}-K_{cmin}-LC_{min}\}$
$S[i] \leftarrow \max\{CLK_r[i]+t_{skewr}+K_{cmax}, Q[i-1]+K_{imax}\}$
$NS[i] \leftarrow S[i]+LC_{max}$
$CLK_f[i] \leftarrow \max\{NS[i]+t_{setup}, CLK_r[i]+t_{skewr}+pw_{min}\}$
$Q[i] \leftarrow \max\{CLK_r[i]+t_{skewr}+L_{CQmax}, NS[i]+L_{DQmax}\}$
UNTIL $CLK_r[i]-CLK_r[i-1]=S[i]-S[i-1]=NS[i]-NS[i-1]=CLK_f[i]-CLK_f[i-1]=Q[i]-Q[i-1]$

/*set some output parameters*/

$W \leftarrow CLK_f[i]-CLK_r[i]$
$T \leftarrow 2(CLK_r[i]-CLK_r[i-1])$

3.3 Algorithm for the four-phase PALACS

To generate the chronogram in the four phase PALACS, we will suppose that at the beginning the latches of Figure labelled with 0 have held the initial state for a time long enough so that state is already at their output. We will also assume that the first active pulse happens at the clock OE0. The meaning of the variables and parameters used is the following (see Figure 9):

- Timing parameters of the circuit

  - $t_{skewrCLK}$: Upper bound on the skew for a rising transition of a load clock signal
  - $t_{skewfCLK}$: Upper bound on the skew for a falling transition of a load clock signal
  - $t_{skewrOE}$: Upper bound on the skew for a rising transition of a output enable clock signal
  - $t_{skewfOE}$: Upper bound on the skew for a falling transition of a output enable clock signal
  - LC_{max}: Upper bound on the delay of the circuit
  - LC_{min}: Contamination delay of the circuit
  - K_{cmax}: Upper bound on the delay of a switch when its input is valid and it activates
  - K_{cmin}: Contamination delay of a switch when it activates
  - K_{imax}: Upper bound on the delay of a switch when its input is valid and it activates
  - K_{imin}: Contamination delay of a switch when its input changes
  - L_{DQmax}: Upper bound on the delay of a latch when its load control signal is active and its input changes
  - L_{CQmax}: Upper bound on the delay of a latch when its input is valid and its load control signal activates
  - L_{CQmin}: Contamination delay of a latch when its load control signal activates
  - t_{setup}: Setup time of the latches
  - t_{hold}: Hold time of the latches
  - pw_{min}: Minimum active pulse width at the load control signal of a latch to ensure that the data will be latched

- Geometrical variables of the algorithm

  - S[i]: Upper bound on the instant of the computation cycle i where the state signals have reached their new value
  - NS[i]: Upper bound on the instant of the computation cycle i where the next state signals have reached their new value
  - Q[i]: Upper bound on the instant of the computation cycle i where the output of the latches labelled with (i+1 mod 2) have reached their new value
  - CLK_r[i]: Upper bound on the instant of the computation cycle i where the load control signal of the latches labelled with (i+1 mod 2) activates
  - CLK_f[i]: Upper bound on the instant of the computation cycle i where the load control signal of the latches labelled with (i+1 mod 2) deactivates
  - OE_r[i]: Upper bound on the instant of the computation cycle i where the output enable signal of the latches labelled with (i mod 2) activates
  - OE_f[i]: Upper bound on the instant of the computation cycle i where the output enable signal of the latches labelled with (i mod 2) deactivates

- Output parameters

  - W_{CLK}: Active pulse width of the load clock signals (output parameter)
  - W_{OE}: Active pulse width of the output enable clock signals (output parameter)
  - T: Clock signals period (output parameter)
  - displacement: Time elapsed from the activation of the output enable clock of a latch to the activation of the load clock signal of the same latch (output parameter)

Again, we will assume that if the input of a latch gets valid at instant t_i ti while its load control signal activates at instant t_c then the new value of the input appears at the output at an instant no later than $\max\{t_i+K_{imax}, t_c+K_{cmax}\}$.
Supposing that the control signals are active in high, the algorithm is the following:

/*set the initial state of the chronogram*/

$OE_r[0] \leftarrow 0$

$CLK_r[0] \leftarrow 0$
$S[0] \leftarrow t_{skewrOE}+K_{cmax}$
$NS[0] \leftarrow S[0]+LC_{max}$
$CLK_f[0] \leftarrow \max\{NS[0]+t_{setup}, pw_{min}+t_{skewrCLK}\}$
$Q[0] \leftarrow \max\{CLK_r[0]+t_{skewrCLK}+L_{CQmax}, NS[0]+L_{DQmax}\}$

/*draw the chronogram iteratively till it becomes periodic*/

$i \leftarrow 0$
DO
$i \leftarrow i+1$
$OE_r[i] \leftarrow CLK_f[i-1]+t_{skewfCLK}+t_{hold}-K_{cmin}-LC_{min}$
$OE_f[i-1] \leftarrow OE_r-t_{skewfOE}$
$CLK_r[i] \leftarrow CLK_f[i-1]+t_{skewfCLK}+t_{hold}-L_{CQmin}-K_{imin}-LC_{min}$
$S[i] \leftarrow \max\{OE_r[i]+t_{skewrOE}+K_{cmax}, Q[i-1]+K_{imax}\}$
$NS[i] \leftarrow S[i]+LC_{max}$
$CLK_{fLD}[i] \leftarrow \max\{NS[i]+t_{setup}, CLK_r[i]+t_{skewrCLK}+pw_{min}\}$
$Q[i] \leftarrow \max\{CLK_r[i]+t_{skewrCLK}+L_{CQmax}, NS[i]+L_{DQmax}\}$
UNTIL $OE_r[i]-OE_r[i-1]=OE_f[i]-OE_f[i-1]=CLK_r[i]-CLK_r[i-1]=CLK_f[i]-CLK_f[i-1]=S[i]-S[i-1]=NS[i]-NS[i-1]=Q[i]-Q[i-1]$

/*set some output parameters*/

$T \leftarrow 2(CLK_r[i]-CLK_r[i-1])$
$OE_f[i] \leftarrow OE_f[i-1]+T/2$
$W_{CLK} \leftarrow CLK_f[i]-CLK_r[i]$
$W_{OE} \leftarrow OE_f[i]-OE_r[i]$
$displacement \leftarrow CLK_r[i]-OE_r[i]$

4. RESULTS

In order to check the algorithms, a binary four-bit counter has been implemented using standard cells of a 0.35 μm CMOS process. The latches used in every clocking scheme are transparent at low level. Because of its simplicity, full electrical simulation of the test circuit is feasible. These characteristics makes the proposed example specially appropriate, to test clocking schemes and to validate the proposed algorithms.

In the following sections, the correct operation of the algorithms is first checked by simulation the operation of the circuit under the clock signals calculated by the tool. The algorithms are then used to compare the operation speed of the three analysed clocking schemes.

4.1 Algorithm validation

To check the implementation of the algorithms we have carried out electrical simulation witch SPECTRE within Cadence's Design Framework II[9]. For each clocking scheme, we will proceed as follows:

1) First we will make a timing analysis of the circuit to get the timing parameters required by the algorithms. The critical path will be obtained by topological analysis.

2) We will use the tool to get the clock waveforms in a skew free environment and we will check that the circuit works by electrical simulation.

3) Then, using the same clock waveforms, we will introduce clock skew till produce malfunction.

4) We will measure the introduced clock skew and we will recalculate clock waveforms tolerant to that clock skew.

5) Finally, we will simulate the circuit with the new clock waveforms to check that it is tolerant to the introduced skew.

The first step, timing analysis of the circuit, is common for both PALACS schemes. For these schemes we used latches of the cell library that had the switch integrated working as an output enable signal. The analysis has been carried out using the Design Framework II environment to get the SDF delay file. From this file we got the parameters required by the algorithm.

We got the clock waveforms for two-phase PALACS using the timing parameters and assuming there is no clock skew. As we can see in Figure 10, the circuit works properly. Note that the clock signals are active at low level.

Without changing the waveform of the clock signals, we introduced skew in the clock signals controlling the latches of the two most significant bits by making then go through an inverter chain. As we can see in Figure 11, when we introduce four inverters in the clock path the circuit does not work correctly anymore.

We measured the introduced skew and recalculated the clock waveforms to make the circuit tolerant to that skew. The electric simulation of Figure 12 shows that the circuit works correctly.

We have repeated all this process for the four-phase PALACS. The results are shown in Figure 13.

Again we skewed the clock signals of the two most significant latches using an inverter chain without changing the waveforms of the nominal clocks. When we introduced four inverters in the clock path, electric simulation showed that the circuit did not work correctly anymore. This can be seen in Figure 14.

Again we measured the introduced skew and calculated waveforms for the clock signals that would tolerate that skew. The simulation of Figure 15 showed that circuit worked correctly again.

The glitches remarked in Figure 15 are not relevant since they do not happen near the end of any active pulse. So, the circuit works correctly.

We have also checked the tool for the Master-Slave scheme in the same way. The results are not shown since it is a well known scheme that has been used for a long time.

4.2 Analysis of operation speed

Here we will compare the maximum computation frequency (minimum computation period) for the three multiphase clocking schemes (Master-Slave, two-phase PALACS and four-phase PALACS). As we have seen in the previous section, the minimum period depends on the clock skew. So, when $t_{skew}=0$ the four bit counter can reach a computation frequency of 534 MHz with the Master-Slave scheme, while with the PALACS schemes can reach a computation frequency of 662 MHz. This means a speed-up of 24% compared to the Master-Slave scheme.

In order to see how clock skew affects computation speed, we have obtained the computation cycle time that can be reached with each scheme for skew values from 0 to T_0, where T_0 is the minimum computation cycle time for the PALACS schemes. This has been done by iteratively running the algorithms assuming that the maximum skew for all the clock signals is the same and that the skew values for rising and falling transitions are equal. The result is shown in Figure 16.

As can be seen in Figure 16, the minimum computation cycle for the PALACS schemes is 1510 ps, what is the sum of the maximum delay of a latch with its switch and the delay of the logic circuit. On the other hand, the minimum computation cycle time reachable with the Master-Slave scheme is 1870 ps, what is the sum of the delays of a master latch, a slave latch and the logic circuit.

All the clocking schemes present a piece-wise linear dependence of the minimum cycle time with the maximum allowed skew. PALACS curves show two regions: one of slope 0 and a second region of slope 2. The transition from the first region to the second region in the two-phase PALACS occurs when $CLK_r[i]+t_{skewr}+K_{cmax}$ rises above $Q[i-1]+K_{imax}$, while this transition in the four-phase PALACS happens when $OE_r[i]+t_{skewrOE}+K_{cmax}$ rises above $Q[i-1]+K_{imax}$.

The MSCS shows three regions of operation with slopes 0, 2 and 4 respectively. The transition from the first region to the second region happens when $CLK_{0r}[i]+t_{skew0r}+pw_{minslave}$ rises above $QM[i]+t_{setupslave}$; and the transition from the second region to the third region takes place when $CLK_{1r}[i]+t_{skew1r}+L_{masterCQmax}$ rises above $NS[i-1]+L_{masterDQmax}$.

As we can see, although the four phase PALACS is always the fastest, the Master-Slave scheme is faster than the two-phase PALACS for a range of values of the clock-skew. Nevertheless, both PALACS schemes behave much better than the MSCS as the clock-skew increases.

In summary, PALACS performs better than MSCS in most cases. In particular, two-phase PALACS is faster than MSCS for low and high skew without including extra complexity in the design of latches or clock distribution network. The four-phase PALACS shows even better timing properties at the expense of extra clock signals.

5. CONCLUSIONS

We have presented two skew tolerant clocking schemes for digital VLSI systems called PALACS. These schemes are inspired on the one-phase double-edge triggered clocking scheme. We have compared the performance of these schemes with the two-phase Master-Slave clocking scheme in terms of speed. Both PALACS looks to outperforms Master-Slave. The simpler two-phase PALACS, while comparable in complexity to the MSCS, is about 20% faster. The four-phase PALACS provides even better timing performance at the expense of a more complex clock distribution network. This makes PALACS a very interesting alternative when designing large digital systems operating at high frequencies.

ACKNOWLEDGEMENT

This work has been supported by the MEC of the Spanish Government, under project TEC2004-00840/MIC.

REFERENCES

1. H. B. BAKOGLU, *Circuits, Interconnections and Packaging for VLSI*, Ed. Add-Wesley Publishing Company. (1990). ISBN 0-201-06008-6
2. S. H. UNGER and CH. TAN, "Clocking Schemes for High-Speed Digital Systems", IEEE transactions on computers. (1986). **Vol. C-35**. Nº10. pp. 880-895
3. M. AFGHAHI and J. YUAN, "Double Edge-triggered D-flip-flops for High-speed CMOS circuits", IEEE Journal of Solid-State Circuits. (1991). **Vol.26** Nº8. pp. 1168-1170
4. M. HOROWITZ, "Clocking Strategies in High Performance Processors", *Symposium on VLSI Circuits Digest of technical Pagers*. (1992). pp. 50-53
5. D. GUERRERO, M. J. BELLIDO,J. J. CHICO, P. RUIZ, A. MILLÁN, "Two phase alternating latches clocking scheme for CMOS sequential circuits", *XVII Conference on Design of Circuits and Integrated Systems*, November 2002, Santander, pp. 159-162
6. D. GUERRERO, M. J. BELLIDO,J. J. CHICO, A. MILLÁN, P. RUIZ, E. OSTUA, "Four phase alternating latches clocking scheme for CMOS sequential circuits", *XIX Conference on Design of Circuits and Integrated Systems*, November 2004, Bordeaux
7. V. G. OKLOBDZIJA, "Clocking and Clocked Storage Elements in Multi-GHz Environment", *12th International Workshop PATMOS* (2002). pp. 128-145
8. D. HARRIS, *Skew-Tolerant Circuit Design*, Morgan Kaufmann Publishers. (2001). ISBN 1-55860-636-X, pp. 14-20

9. *Affirma Spectre Circuit Simulator User Guide*, Cadence Design Systems. (1999)
10. K. BERNSTEIN, *High Speed CMOS Design Styles*, Kluwer Academic Publishers. (1998). ISBN 0-7923-8220-X

FIGURES

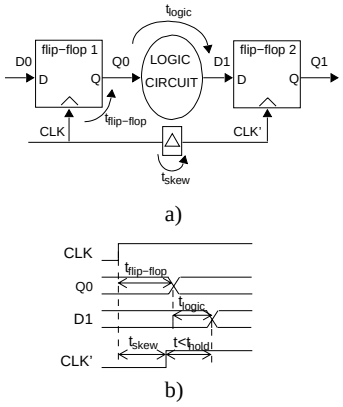

Figure 1: Fast path race problem in a single-phase system with flip-flops. a) Circuit b) Chronogram

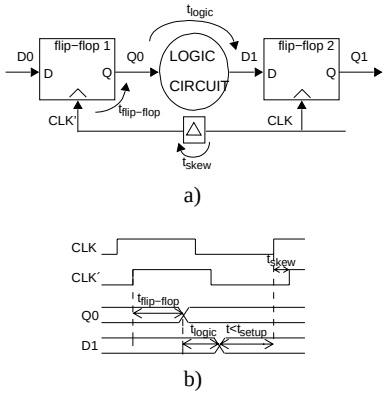

Figure 2: Long path requirement violation in a single-phase system with flip-flops. a) Circuit b) Chronogram

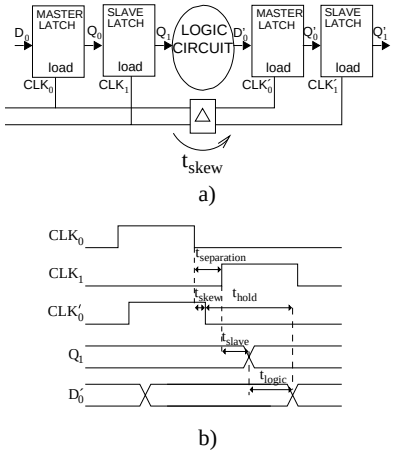

Figure 3: Master-Slave clocking scheme.

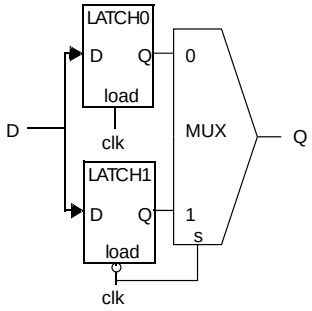

Figure 4: Double-edge triggered flip-flop.

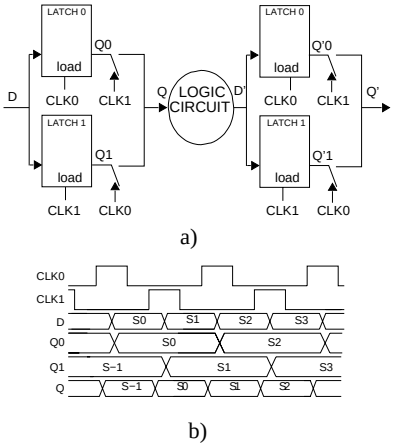

Figure 5: Two-phase PALACS. a) Circuit b) Chronogram

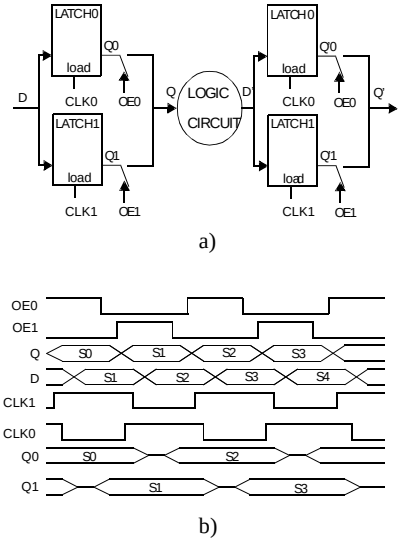

Figure 6: Four-phase PALACS. a) Circuit b) Chronogram

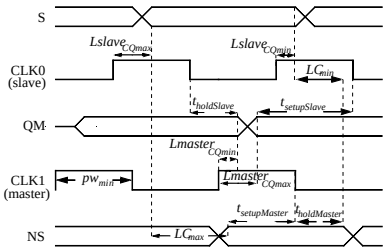

Figure 7: Chronogram generated by the tool for the Master-Slave clocking scheme.

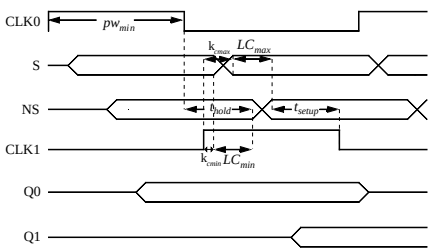

Figure 8: Chronogram generated by the tool for the two-phase PALACS.

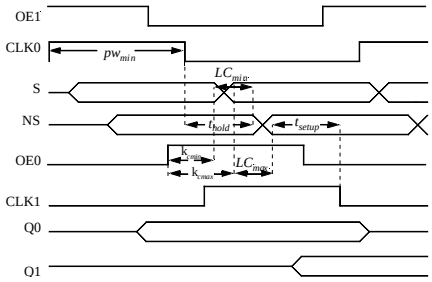

Figure 9: Chronogram generated by the tool for the four-phase PALACS.

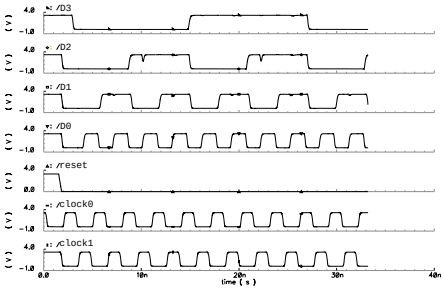

Figure 10: Electric simulation of the four bit counter using the two-phase PALACS in a skew free environment.

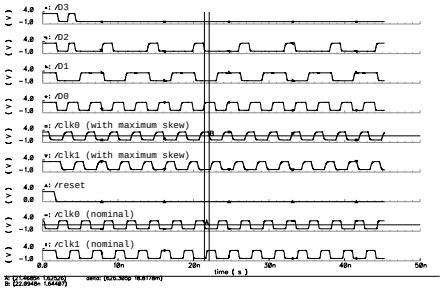

Figure 11: Electric simulation of the counter using the two-phase PALACS under a clock skew equal to the delay of four inverters.

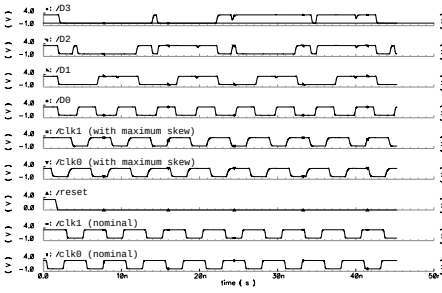

Figure 12: Electric simulation of the four bit counter using the two-phase PALACS tolerant to the introduced skew.

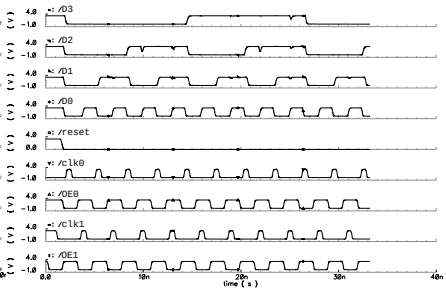

Figure 13: Electric simulation of the four bit counter using the four-phase PALACS in a skew free environment.

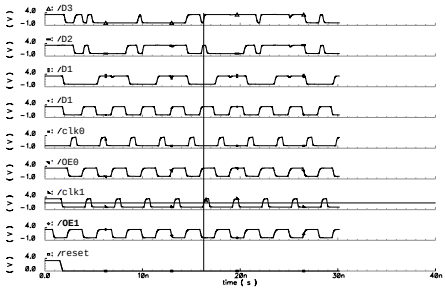

Figure 14: Electric simulation of the four bit counter using the four-phase PALACS under a clock skew causing malfunction.

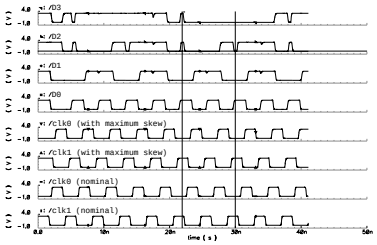

Figure 15: Electric simulation of the four bit counter using the four-phase PALACS tolerant to the introduced skew.

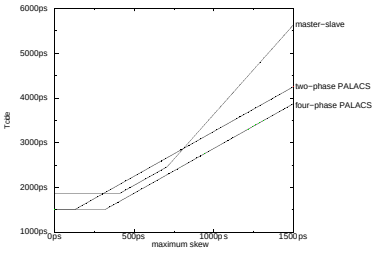

Figure16: Computation cycle time versus clock skew for each clocking scheme.