Automated performance evaluation of skew-tolerant clocking schemes

D. GUERRERO, M. J. BELLIDO, J. JUAN, A. MILLÁN, P. RUIZ-DE-CLAVIJO, E. OSTUA and J. VIEJO←

Departamento de Tecnología Electrónica de la Universidad de Sevilla, Escuela Técnica Superior de Ingeniería Informática, Avd. de Reina Mercedes, S/N, 41012 Sevilla, Spain
Departamento de Diseño Digital y Mixto del Instituto de Microelectrónica de Sevilla-CNM-CSIC, 41012 Sevilla, Spain

In this paper the authors evaluate the timing and power 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). In order to evaluate the timing performance, the authors introduce algorithms to obtain the clock waveforms required by a synchronous sequential circuit. Separated algorithms were developed for every clocking scheme. From these waveforms it is possible to get parameters such as the non-overlapping time and the clock period. They have been implemented in a tool and have been used to compare the timing performance of the clocking schemes applied to a simple circuit. To analyse the power consumption the authors have electrically simulated a simple circuit for several operation frequencies. The most remarkable conclusion is that it is possible to save about 50% of the power consumption of the clock distribution network by using PALACS.

Keywords: Clock skew tolerance; High speed CMOS design; Low power

1. Introduction

The evolution in the very large scale integration (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 (Tan and Unger 1986, Bakoglu 1990, Bernstein 1998). 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. It should be

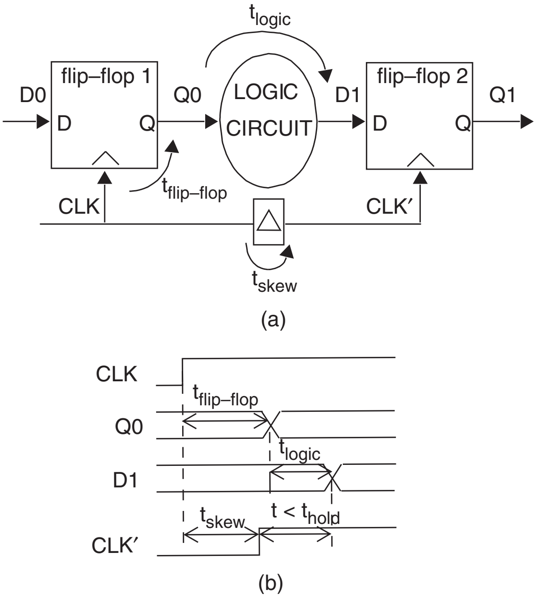


Figure 1. Fast path race problem in a single-phase system with flip-flops: (a) circuit; (b) chronogram.

noted that this problem cannot be solved by enlarging the clock cycle (Horowitz 1992).

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-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 become 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 $t_{\text{separation}}$ in figure 3).

In this work the authors present another two skew tolerant clocking schemes called generically parallel alternating latches clocking scheme (PALACS)

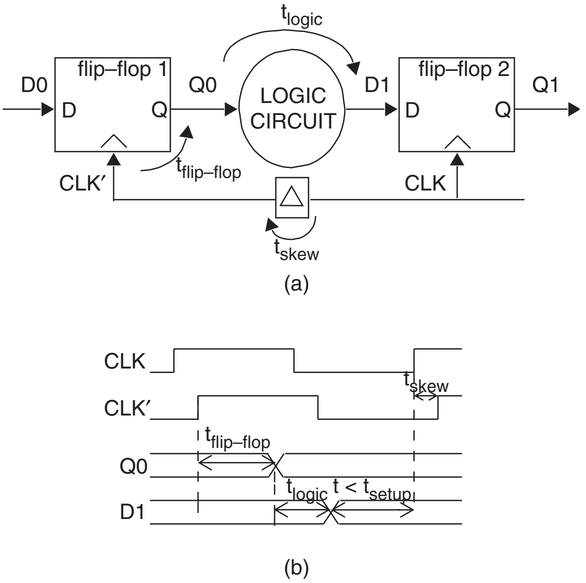


Figure 2. Long path requirement violation in a single-phase system with flip-flops: (a) circuit; (b) chronogram.

(Guerrero *et al.* 2002, 2004). These schemes are based on the one-phase double-edge triggered clocking scheme (Afghahi and Yuan 1991, Oklobdzija 2002). The main objective of this work is to evaluate the performance in terms of speed and power consumption of these three clocking schemes (MSCS, two-phase PALACS and four-phase PALACS).

Targeting this objective, this paper is organized as follows. In the next section we will see the PALACS clocking schemes. In §3 algorithms to obtain the required waveforms for each clocking scheme will be introduced. In §4.1 and 4.2 the correctness of these algorithms will be checked and they will be used to compare the operation speed of each scheme. In §4.3 the power consumption of PALACS and MSCS will be compared and we will see that PALACS provide dramatic power saving. Finally, we will summarize 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 (Afghahi and Yuan 1991, Horowitz 1992). 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

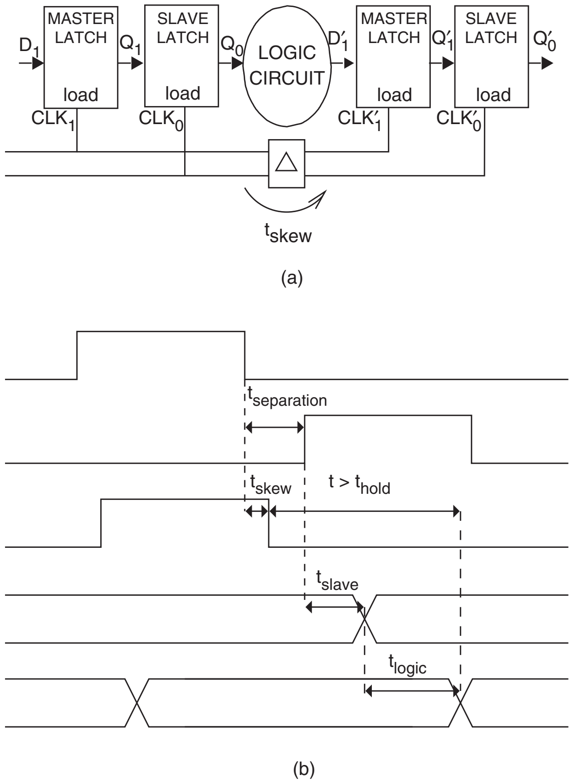


Figure 3. Master–slave clocking scheme: (a) circuit; (b) chronogram.

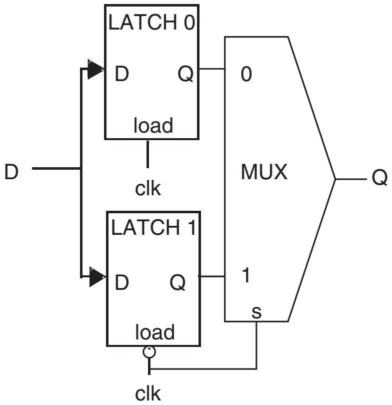


Figure 4. Double-edge triggered flip-flop.

the clock signals is zero. The general MSCS provides tolerance to an arbitrary skew by enlarging the non-overlapping region.

Analogously, the authors have generalized the one-phase double-edge triggered flip-flop clocking scheme to get skew tolerance by using two separated clock signals (Guerrero *et al.* 2002).

2.1 *Two-phase PALACS*

A generalization 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.

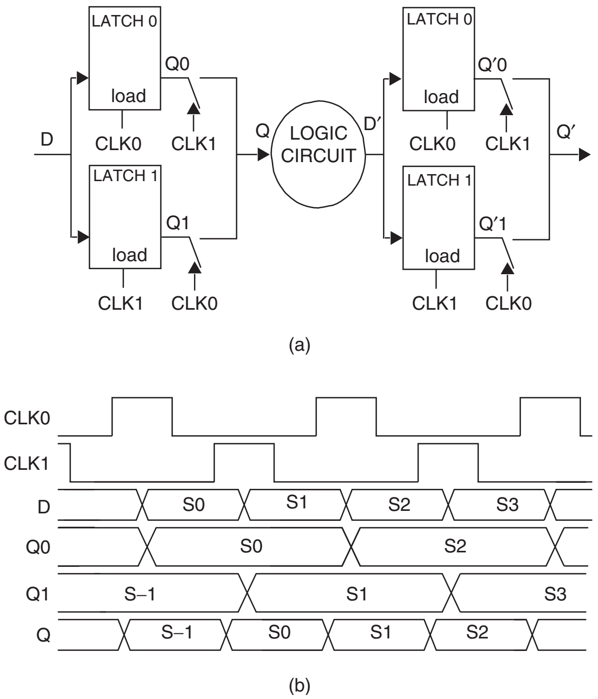
The scheme uses parallel read and write node (PRAWN) structures. A PRAWN structure consists of two latches connected in parallel with the same input and a switch at the output of each latch whose outputs are connected. The loads of both latches are controlled by separate phases, and the switches are also controlled by opposite phases. This scheme, unlike the master–slave scheme, allows reading and writing of the register block, simultaneously, during the active level of each clock phase. In effect, while clock signal CLK_0 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 CLK_0, since its switch is controlled by CLK_0. When CLK_0 becomes inactive, latch 0 stops being transparent. Then both phases remain inactive long enough to avoid clock skew related problems. During this interval both switches are in a high impedance (HI) state, but the previous data value remains loaded at the switches output due to parasitic capacitances. When CLK_1 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 (Afghahi and Yuan 1991, Oklobdzija 2002).

If the node at the output of the switches were almost immediately discharged when a clock signal deactivates due to leakage currents (Nedovic and Oklobdzija 2000), hold time violations may occur. To prevent this, the designer must make sure that

$$K_{\mathrm{discharge}} + LC_{\mathrm{min}} > t_{\mathrm{hold}} + t_{\mathrm{skew}} \tag{1}$$

where $K_{\mathrm{discharge}}$ is the minimum time the node will hold the value before discharging, LC_{min} is the minimum delay of the logic circuit, t_{hold} is the hold time of the latches and t_{skew} is the skew of the clock when it deactivates. The fact that $K_{\mathrm{discharge}}$ is greater than zero and that t_{hold} is usually zero or even negative makes condition (1) easy to meet.

The most important advantage of PALACS versus MSCS is that the clock frequency is reduced by 50% for the same data rate. This has considerable benefits, mainly in the reduction of 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 PRAWN structure is smaller than the propagation delay of the master–slave since in MSCS the input signal has to



(a)

(b)

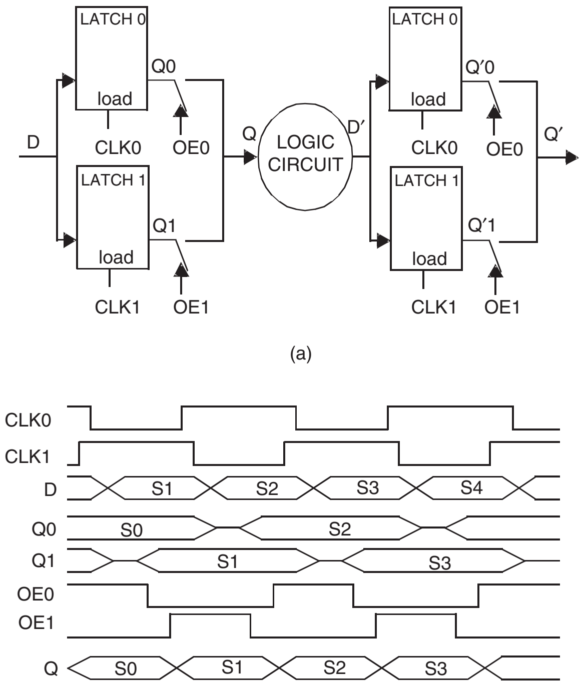
Figure 5. Two-phase PALACS: (a) circuit; (b) chronogram.

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 (Harris 2001). 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 receives 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 is used (Harris 2001). 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.

This scheme also uses PRAWN structures. In the two-phase PALACS, the load control signal of the latches are controlled by opposite phases (CLK_0 and CLK_1 in figure 6), but the switches are controlled by another two separated phases



(a)

(b)

Figure 6. Four-phase PALACS: (a) circuit; (b) chronogram.

(OE_0 and OE_1 figure 6). Therefore, when CLK_0 deactivates, the data item at Q_0 can reach Q by activating OE_0 even if CLK_1 is not active. Furthermore, if the minimum delay of the logic circuit is long enough, 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. 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 see in §4.2, this makes it possible to improve the timing performance of the four-phase PALACS with 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 to be generated according to general timing parameters including logic delay, setup and hold times and maximum tolerable 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 have been implemented in a tool. Given a general

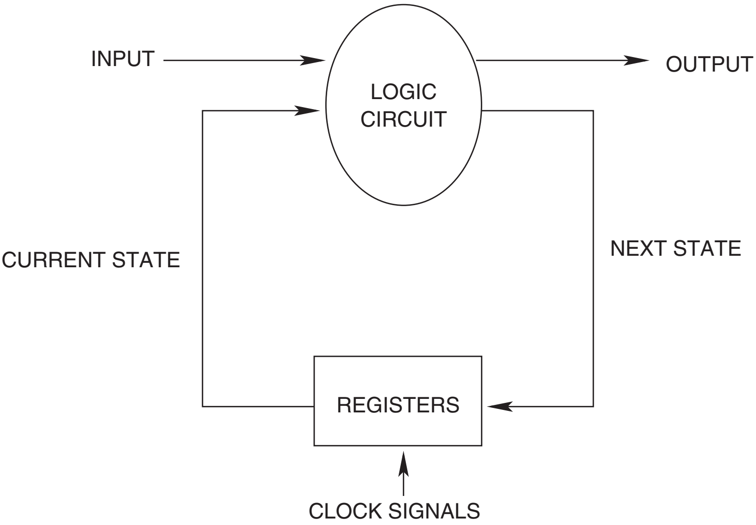


Figure 7. A general synchronous sequential circuit.

synchronous sequential circuit like that shown in figure 7 and a set of timing parameters, the algorithms generate a chronogram where the conditions to ensure the correct operation of the circuit are met.

At this point, it is necessary to give a formal definition of skew to use in the algorithms. Let tc be a clock transition and let C be the set of control inputs connected to the corresponding clock.

- We will define the nominal arrival time of the transition tc as $\min\{a(tc, i)/ i \in C\}$, where $a(t, i)$ denotes the instant when transition tc reaches input i.
- We will define the skew of transition tc as the difference between $\max\{a(tc, i)/ i \in C\}$ and the nominal arrival time of tc.

It should be noted that, by defining the skew in this way, only positive clock skew values make sense.

Starting in a stable initial state, the signals begin to change, affected by the delay of the components. The algorithms set with every signal transition, minimizing the clock period but 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 sufficiently long time 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 described as follows (see figure 8).

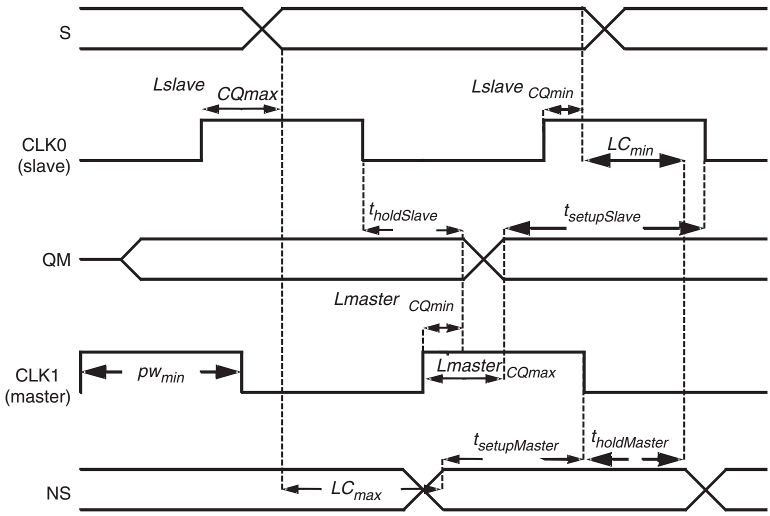


Figure 8. Chronogram generated by the tool for the Master–Slave clocking scheme.

Input parameters

- $t_{\text{skew}0r}$: Upper bound on the skew for the raising transitions of the clock of the master latches
- $t_{\text{skew}0f}$: Upper bound on the skew for the falling transitions of clock of the master latches
- $t_{\text{skew}1ra}$: Upper bound on the skew for the raising transitions of clock of the slave latches
- $t_{\text{skew}1f}$: 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}: Minimum 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
- $L\text{master}_{DQ\text{max}}$: Upper bound on the delay of a master latch when its load control signal is already active and its input changes to a valid value
- $L\text{master}_{CQ\text{max}}$: Upper bound on the delay of a master latch when its input is already stable and valid and its load control signal activates
- $L\text{master}_{CQ\text{min}}$: Minimum delay of a master latch when it load control signal activates
- $L\text{slave}_{DQ\text{max}}$: Upper bound on the delay of a slave latch when its load control signal is already active and its input changes to a valid value
- $L\text{slave}_{CQ\text{max}}$: Upper bound on the delay of a slave latch when its input is already stable and valid and its load control signal activates
- $L\text{slave}_{CQ\text{min}}$: Minimum delay of a slave latch when it load control signal activates
- $t_{\text{setupmaster}}$: Setup time of a master latch
- $t_{\text{setupslave}}$: Setup time of a slave latch
- $t_{\text{holdmaster}}$: Hold time of a master latch

- $t_{\text{holdslave}}$: Hold time of a slave latch
- $pw_{\text{minmaster}}$: 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

Variables

- $S[i]$: 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
- $CLK_{0r}[i]$: Nominal arrival time of the rising edge of the master clock in the computation cycle i
- $CLK_{0f}[i]$: Nominal arrival time of the falling edge of the master clock in the computation cycle i
- $CLK_{1r}[i]$: Nominal arrival time of the rising edge of the slave clock in the computation cycle i
- $CLK_{0f}[i]$: Nominal arrival time of the falling edge of the slave clock in the computation cycle i

Output parameters

- W_0: Active pulse width of the master clock signal
- W_1: Active pulse width of the slave clock signal
- $_{\text{displacemen}}t$: 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 as follows.

```
    /*set the initial state of the chronogram*/

$CLK_{1r}[0] \leftarrow 0$
$CLK_{1f}[0] \quad pw_{\text{minMaster}} + t_{\text{skew}1r}$
$QM[0] \leftarrow L\text{master}_{CQ\text{max}} + t_{\text{skew}1r}$
$CLK_{0r}[0] \quad CLK_{1f}[0] + t_{\text{skew}1f} + t_{\text{holdMaster}} - L_{\text{slave}CQ\text{min}} - LC_{\text{min}}$
$S[0] \leftarrow \max\{CLK_{0r}[0] + t_{\text{skew}0r} + L_{\text{slave}CQ\text{max}},\ QM[0] + L_{\text{slave}DQ\text{max}}\}$
$CLK_{0f}[0] \quad \max\{QM[0] + t_{\text{setupSlave}},\ CLK_{0r}[0] + t_{\text{skew}0r} + pw_{\text{minslave}}\}$
$NS[0] \leftarrow S[0] + LC_{\text{max}}$

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

$i \leftarrow 0$
DO
  $I \leftarrow i + 1$
  $CLK_{1r}[i] \leftarrow CLK_{0f}[i-1] + t_{\text{skew}0f} + t_{\text{holdSlave}} - L_{\text{master}CQ\text{min}}$
  $QM[i] \leftarrow \max\{CLK_{1r}[i] + t_{\text{skew}1r} + L_{\text{master}CQ\text{max}},\ NS[i-1] + L_{\text{master}DQ\text{max}}\}$
  $CLK_{1f}[i] \quad \max\{NS[i-1] + t_{\text{setupMaster}},\ CLK_{1r}[i] + t_{\text{skew}1r} + pw_{\text{minMaster}}\}$
```

```
    $CLK_{0r}[i] \leftarrow CLK_{1f}[i] + t_{\text{skew}1f} + t_{\text{holdMaster}} - L_{\text{slave}CQ\text{min}} - LC_{\min}$
    $CLK_{0f}[i] \quad \max\{QM[i] + t_{\text{setupSlave}},\ CLK_{0r}[i] + t_{\text{skew}0r} + pw_{\text{minSlave}}\}$
    $S[i] \leftarrow \max\{CLK_{0r}[i] + t_{\text{skew}0r} + L_{\text{slave}CQ\text{max}},\ QM[i] + L_{\text{slave}DQ\text{max}}\}$
    $NS[i] \leftarrow S[i] + LC_{\max}$
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 as follows (see figure 9).

Input parameters

- $t_{\text{skew}r}$: Upper bound on the skew for a rising transition of a clock signal
- $t_{\text{skew}f}$: Upper bound on the skew for a falling transition of a clock signal
- $LC_{\max}$: Upper bound on the delay of the circuit
- $LC_{\min}$: Minimum delay of the circuit
- $K_{c\max}$: Upper bound on the delay of a switch when its input is already valid and it activates
- $K_{c\min}$: Minimum delay of a switch when it activates
- $K_{i\max}$: Upper bound on the delay of a switch when it is already on and its input changes

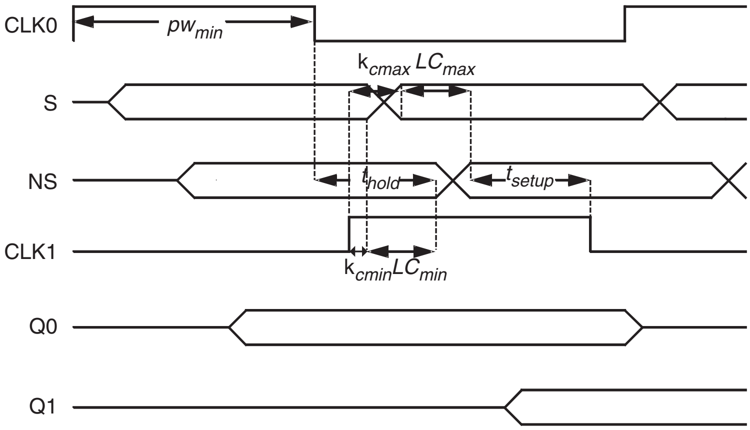


Figure 9. Chronogram generated by the tool for the two-phase PALACS.

- $L_{DQ\max}$: Upper bound on the delay of a latch when its load control signal is already active and its input changes
- $L_{CQ\max}$: Upper bound on the delay of a latch when its input is already 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

Variables

- $S[i]$: 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
- $CLK_r[i]$: Nominal instant of the computation cycle i where the load control signal of the latches labelled with (i mod 2) activates
- $CLK_f[i]$: Nominal 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 $\max\{t_i + K_{i\max}, tc + K_{c\max}\}$ (Tan and Unger 1986). Supposing that the control signals are active in high, the algorithm is the following.

```
    /*set the initial state of the chronogram*/

$CLK_r[0] \leftarrow 0$
$S[0] \leftarrow t_{\text{skew}r} + K_{c\max}$
$NS[0] \leftarrow S[0] + LC_{\max}$
$CLK_f[0] \quad \max\{NS[0] + t_{\text{setup}}, pw_{\min} + t_{\text{skew}r}\}$
$Q[0] \leftarrow \max\{CLK_r[0] + t_{\text{skew}r} + L_{CQ\max}, NS[0] + L_{DQ\max}\}$

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

$i \leftarrow 0$
DO
  $i \leftarrow i + 1$
  $CLK_r[i] \leftarrow CLK_f[i-1] + t_{\text{skew}f} + \max\{0, t_{\text{hold}} - K_{c\min} - LC_{\min}\}$
  $S[i] \leftarrow \max\{CLK_r[i] + t_{\text{skew}r} + K_{c\max}, Q[i-1] + K_{i\max}\}$
  $NS[i] \leftarrow S[i] + LC_{\max}$
  $CLK_f[i] \quad \max\{NS[i] + t_{\text{setup}}, CLK_r[i] + t_{\text{skew}r} + pw_{\min}\}$
  $Q[i] \leftarrow \max\{CLK_r[i] + t_{\text{skew}r} + L_{CQ\max}, NS[i] + L_{DQ\max}\}$
```

```
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 6 labelled with 0 have held the initial state for a time long enough so that state is already valid 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 as follows (see figure 10).

Input parameters

- $t_{\mathrm{skew}rCLK}$: Upper bound on the skew for a rising transition of a load clock signal
- $t_{\mathrm{skew}fCLK}$: Upper bound on the skew for a falling transition of a load clock signal
- $t_{\mathrm{skew}r\mathrm{O}E}$: Upper bound on the skew for a rising transition of a output enable clock signal
- $t_{\mathrm{skew}f\mathrm{O}E}$: 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}: Minimum delay of the circuit
- $K_{c\mathrm{max}}$: Upper bound on the delay of a switch when its input is already valid and it activates

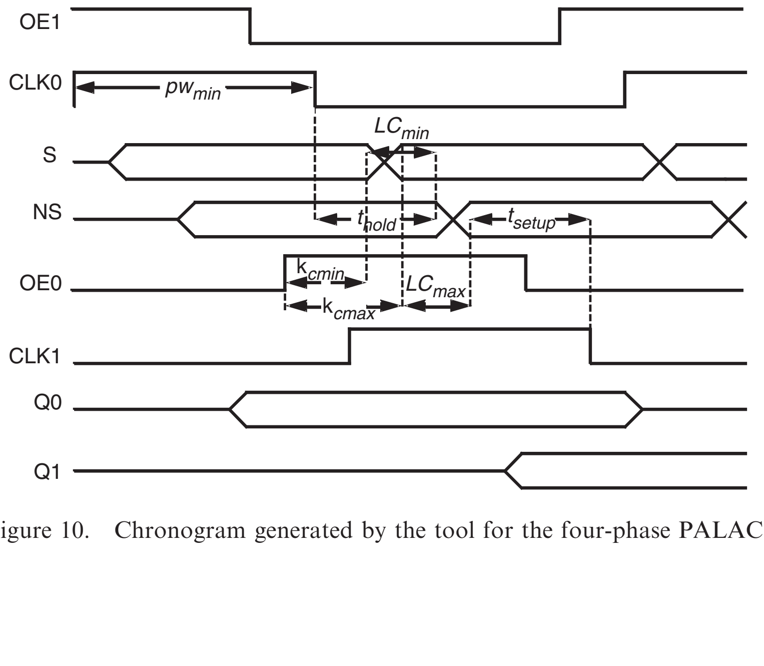


Figure 10. Chronogram generated by the tool for the four-phase PALACS.

- $K_{c\min}$: Minimum delay of a switch when it activates
- $K_{i\max}$: Upper bound on the delay of a switch when its input is already valid and it activates
- $K_{i\min}$: Minimum delay of a switch when its input changes
- $L_{DQ\max}$: Upper bound on the delay of a latch when its load control signal is active and its input changes
- $L_{CQ\max}$: Upper bound on the delay of a latch when its input is already valid and its load control signal activates
- $L_{CQ\min}$: Minimum 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

Variables

- $S[i]$: 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 $(i+1 \bmod 2)$ have reached their new value
- $CLK_r[i]$: Nominal instant of the computation cycle i where the load control signal of the latches labelled with $(i+1 \bmod 2)$ activates
- $CLK_f[i]$: Nominal instant of the computation cycle i where the load control signal of the latches labelled with $(i+1 \bmod 2)$ deactivates
- $\mathrm{O}E_r[i]$: Nominal instant of the computation cycle i where the output enable signal of the latches labelled with $(i \bmod 2)$ activates
- $\mathrm{O}E_f[i]$: Nominal instant of the computation cycle i where the output enable signal of the latches labelled with $(i \bmod 2)$ deactivates

Output parameters

- W_{CLK}: Active pulse width of the load clock signals
- $W_{\mathrm{O}E}$: Active pulse width of the output enable clock signals
- T: Clock signals period
- 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

Again, 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 $\max\{t_i + K_{i\max}, tc + K_{c\max}\}$.

Supposing that the control signals are active in high, the algorithm is the following:

```
/*set the initial state of the chronogram*/

$\mathrm{O}E_r[0] \leftarrow 0$
$CLK_r[0] \leftarrow 0$
$S[0] \leftarrow t_{\text{skew}r\mathrm{O}E} + K_{c\max}$
$NS[0] \leftarrow S[0] + LC_{\max}$
```

```
$CLK_f[0] \quad \max\{NS[0] + t_{\text{setup}},\ pw_{\min} + t_{\text{skew}rCLK}\}$
$Q[0] \leftarrow \max\{CLK_r[0] + t_{\text{skew}rCLK} + L_{CQ\max},\ NS[0] + L_{DQ\max}\}$

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

$i \leftarrow 0$
DO
    $I \leftarrow i + 1$
    $\mathrm{O}E_r[i] \leftarrow CLK_f[i-1] + t_{\text{skew}fCLK} + t_{\text{hold}} - K_{c\min} - LC_{\min}$
    $\mathrm{O}E_f[i-1] \leftarrow \mathrm{O}E_r[i] - t_{\text{skew}f\mathrm{O}E}$
    $CLK_r[i] \leftarrow CLK_f[i-1] + t_{\text{skew}fCLK} + t_{\text{hold}} - L_{CQ\min} - K_{i\min} - LC_{\min}$
    $S[i] \leftarrow \max\{\mathrm{O}E_r[i] + t_{\text{skew}r\mathrm{O}E} + K_{c\max},\ Q[i-1] + K_{i\max}\}$
    $NS[i] \leftarrow S[i] + LC_{\max}$
    $CLK_f[i] \quad \max\{NS[i] + t_{\text{setup}},\ CLK_r[i] + t_{\text{skew}rCLK} + pw_{\min}\}$
    $Q[i] \leftarrow \max\{CLK_r[i] + t_{\text{skew}rCLK} + L_{CQ\max},\ NS[i] + L_{DQ\max}\}$
    UNTIL $\mathrm{O}E_r[i] - \mathrm{O}E_r[i-1] = \mathrm{O}E_f[i] - \mathrm{O}E_f[i-1] = CLK_r[i] - CLK_r[i-1] = CLK_f[i] -$
      $CLK_f[i-1] = \ S[i] - [S[i-1]] = NS[i] - NS[i-1] = Q[i] - Q[i-1]$

    /*set some output parameters*/

$T \leftarrow 2(CLK_r[i] - CLK_r[i-1])$
$\mathrm{O}E_f[i] \leftarrow \mathrm{O}E_f[i-1] + T/2$
$W_{CLK} \leftarrow CLK_f[i] - CLK_r[i]$
$W_{\mathrm{O}E} \leftarrow \mathrm{O}E_f[i] - \mathrm{O}E_r[i]$
Displacement $\leftarrow CLK_r[i] - \mathrm{O}E_r[i]$
```

4. Results

In order to check the algorithms, the binary four-bit counter depicted in figure 11 has been implemented using standard cells of a 0.35 μm CMOS process. The parameters of the MOS transistor model used in our simulation and simulation conditions are shown in table 1.

In order to get realistic clock waveforms, inverter chains like that shown in figure 12 were used to implement each clock net. The latches used in every clocking scheme are transparent at the low level. Because of its simplicity, full electrical simulation of the test circuit is feasible. These characteristics make the proposed example especially 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. Finally, the power consumption of the circuit employing these schemes is measured by electrical simulation for several operation frequencies.

4.1 *Algorithm validation*

To check the implementation of the algorithms, the authors have carried out electrical simulation witch SPECTRE within Cadence's design framework II

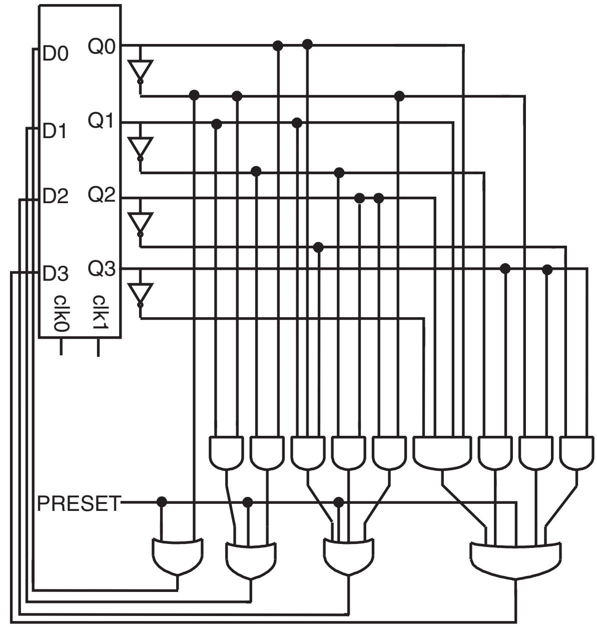


Figure 11. A four-bit counter.

Table 1. Simulation parameters.

Minimum channel length	0.3 microns
Minimum gate width	0.6 microns
MOSFET model	MOS BSIM3V3
Clock slopes (before going through the clock net)	3 picoseconds
Supply voltage	3.3 V
Temperature	25°C

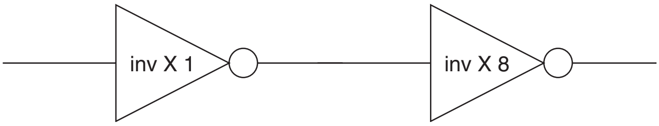


Figure 12. Clock distribution net.

(Cadence 1999). For each clocking scheme, we will proceed as follows.

- First, the circuit will be analysed to get the timing parameters required by the algorithms. The critical path will be obtained by topological analysis.
- The tool will be used to get the clock waveforms in a skew free environment and we will check that the circuit works by electrical simulation.

Table 2. Timing parameters of the four bit counter.

	Master–Slave	PALACS
Maximum delay of the logic circuit	776 ps	887 ps
Minimum delay of the logic circuit	239 ps	372 ps
Maximum DQ delay of the latches	500 ps (master) and 594 ps (slave)	525 ps
Maximum CQ delay of the latches	689 ps (master) and 769 ps (slave)	679 ps
Minimum CQ delay of the latches	415 ps (master) and 444 ps (slave)	602 ps
Setup time of the latches	190 ps	270 ps
Hold time of the latches	0 ps	0 ps
Minimum latching pulse width	370 ps	510 ps
Maximum delay of the switch its input is valid and it is activated		98 ps
Minimum delay of the switch when it is activated		8 ps
Maximum delay of the switch when it the input changes		98 ps
Minimum delay of the switch when it the input changes		8 ps

- Then, using the same clock waveforms, clock skew will be introduced till produce malfunction.
- The introduced clock skew will be measured and the tool will be used to recalculate clock waveforms tolerant to that clock skew.
- Finally, the circuit will be simulated 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, the authors 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 the parameters required by the algorithm were obtained. They are shown in table 2.

The authors got the clock waveforms for two-phase PALACS using these timing parameters and assuming there is no clock skew. As can be seen in figure 13, the circuit works properly. Note that, unlike the clock signals of the example chronograms, the clock signals in the simulations are active at the low level.

Without changing the waveform of the clock signals, the authors introduced skew in the clock signals controlling the latches of the two most significant bits by making them go through an inverter chain. As shown in figure 14, when four inverters are introduced in the clock path the circuit does not work correctly anymore.

The introduced skew was measured and the clock waveforms were recalculated to make the circuit tolerant to that skew. The electric simulation of figure 15 shows that the circuit works correctly.

The authors repeated the entire process for the four-phase PALACS. The results are shown in figures 16–18.

The glitches remarked in figure 18 are not relevant since they do not happen near the end of any active pulse. So, the circuit works correctly.

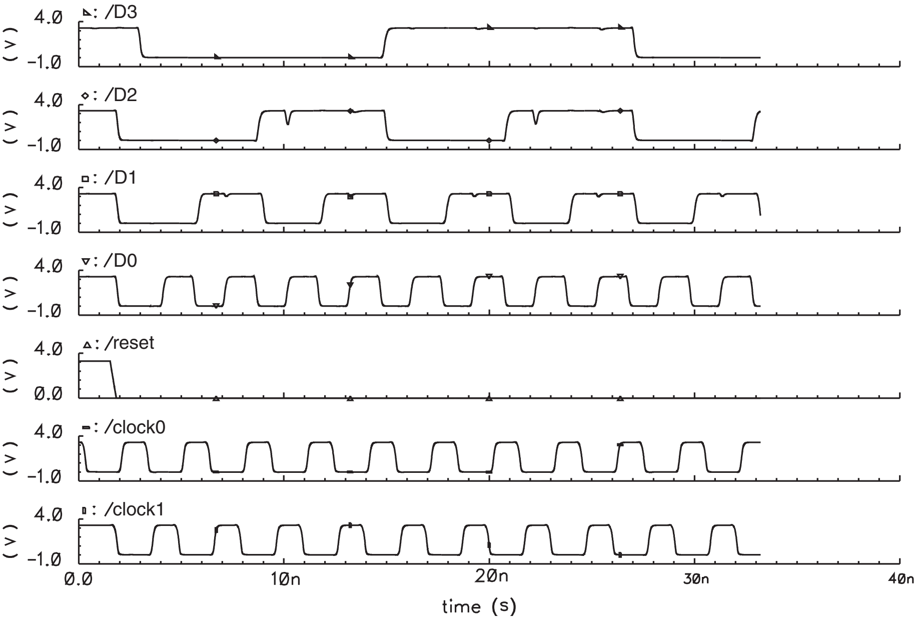


Figure 13. Electric simulation of the four bit counter using the two-phase PALACS in a skew free environment.

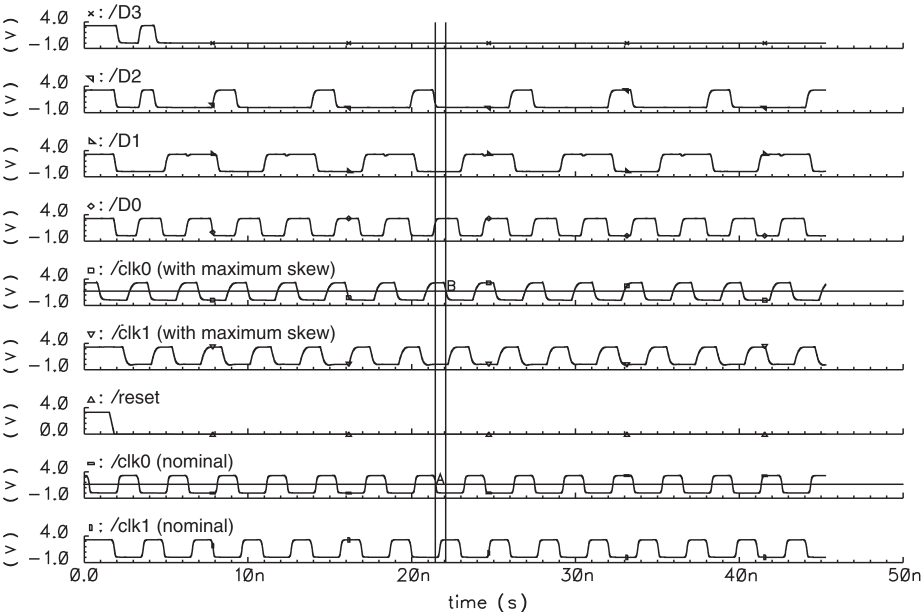


Figure 14. Electric simulation of the four bit counter using the two-phase PALACS under a clock skew equal to the delay of four inverters.

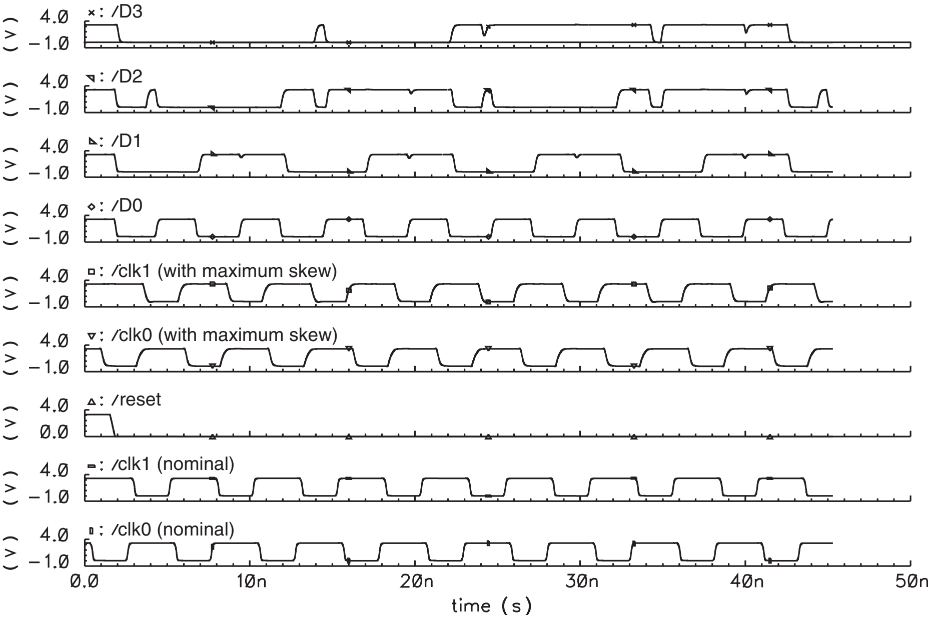


Figure 15. Electric simulation of the four bit counter using the two-phase PALACS tolerant to the introduced skew.

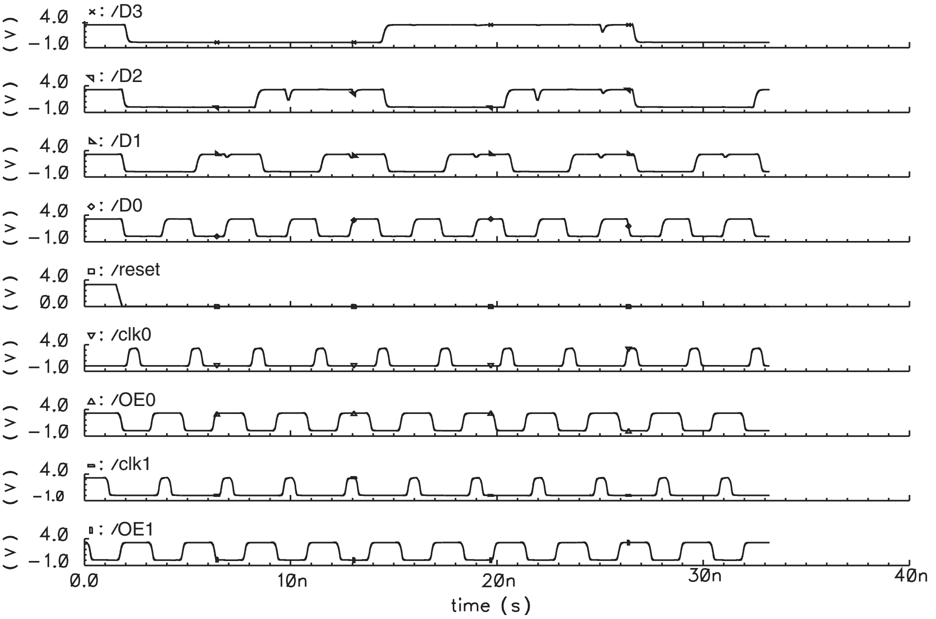


Figure 16. Electric simulation of the four bit counter using the four-phase PALACS in a skew free environment.

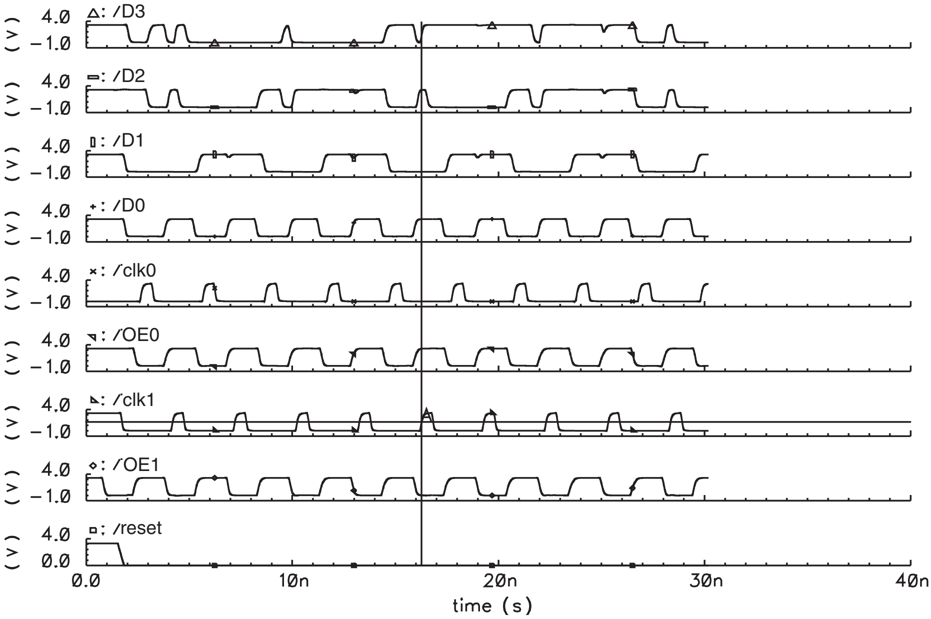


Figure 17. Electric simulation of the four bit counter using the four-phase PALACS under a clock skew causing malfunction.

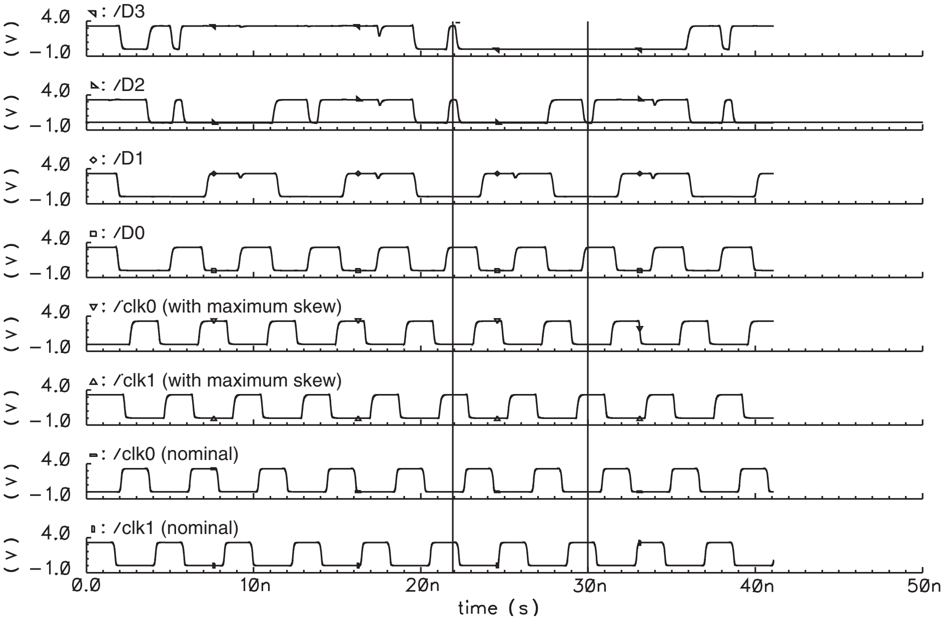


Figure 18. Electric simulation of the four bit counter using the four-phase PALACS tolerant to the introduced skew.

The tool was also checked 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, the maximum computation frequency (minimum computation period) for the three multiphase clocking schemes (master–slave, two-phase PALACS and four-phase PALACS) will be compared. As seen in the previous section, the minimum period depends on the clock skew. So, when $t_{\text{skew}} = 0$ 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, the authors 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 19.

As can be seen in figure 19, 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, the sum of the delays of a master latch, a slave latch and the logic circuit.

All the clocking schemes present a piecewise linear dependence of the minimum cycle time with the maximum allowed skew. PALACS curves show

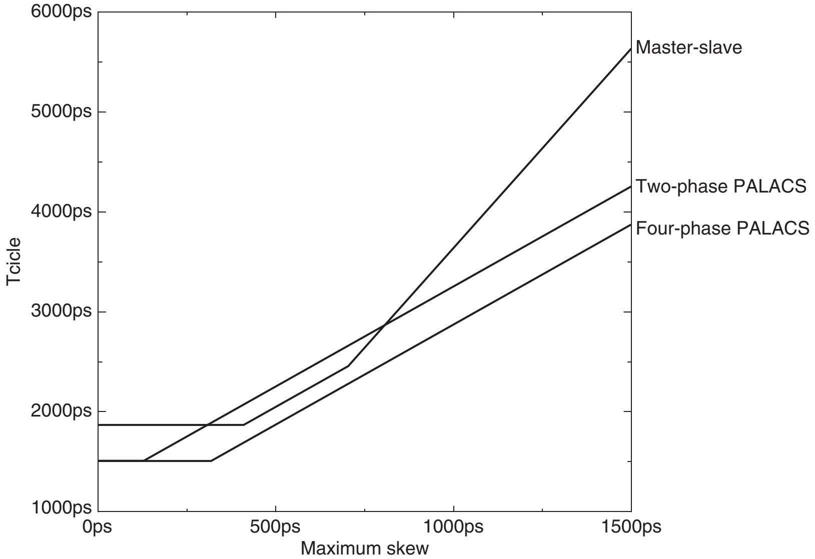


Figure 19. Computation cycle time versus clock skew for each clocking scheme.

two regions: one of slope 0 and a second of slope 2. The transition from the first region to the second region in the two-phase PALACS occurs when $CLK_r[i] + t_{\text{skew}r} + K_{c\max}$ rises above $Q[i-1] + K_{i\max}$, while this transition in the four-phase PALACS happens when $\mathrm{O}E_r[i] + t_{\text{skew}r\mathrm{O}E} + K_{c\max}$ rises above $Q[i-1] + K_{i\max}$.

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 $CLK_{0r}[i] + t_{\text{skew}0r} + pw_{\text{minslave}}$ rises above $QM[i] + t_{\text{setupslave}}$; and the transition from the second region to the third region takes place when $CLK_{1r}[i] + t_{\text{skew}1r} + L_{\text{master}CQ\max}$ rises above $NS[i-1] + L_{\text{master}DQ\max}$.

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.

4.3 *Power consumption*

VLSI digital systems have evolved to big and more complex systems being clocked at high enough frequency. This evolution has reached a point where the overhead of the clock in the form of power consumption has become unacceptable. This is confirmed by what is observed in high-performance microprocessors (Tiwari *et al.* 1998). So, reducing the power due to clock signal distribution is a mandatory issue in digital design.

In this section we will see results regarding the power consumption of the three clocking schemes under analysis. Again, they have been obtained by electrically simulating the circuit of figure 11.

The PALACS schemes should save power with respect to the MSCS for of two reasons. In each computation cycle the number of latches whose state changes with PALACS is half the number of latches whose state changes using the MSCS, but what is more important is the fact that the two-phase PALACS has the same number of clock signal as the MSCS, working at half the frequency for the same computation speed. So the power consumption of the clock distribution network should be smaller. On the other hand, the consumption of the logic circuit should be similar for the three clocking schemes.

To check this power saving, the authors have measured the power consumption of the four bit counter of figure 11 for several computation frequencies using the PALACS and the master–slave schemes. In the simplified example, a buffer plays the role of the clock distribution network. In order to measure the consumption of each component separately, the authors used separated power sources for the clock distribution network, the logic circuit and the latches. The results are shown in table 3.

As expected, there is a remarkable power saving in the two-phase PALACS with respect to the master–slave clocking scheme. The power consumption of the latches is

Table 3. Power consumption of the four-bit counter.

Frequency	25 MHz	50 MHz	100 MHz	200 MHz	500 MHz
Master–slave clock net	85.14 μW	170.25 μW	340.56 μW	681.05 μW	1700.49 μW
Master–slave latches	115.27 μW	230.37 μW	460.35 μW	920.07 μW	2296.47 μW
Master–slave logic circuit	37.36 μW	74.61 μW	149.03 μW	297.63 μW	741.18 μW
PALACS-2 clock net	50.06 μW	100.12 μW	200.24 μW	400.62 μW	1000.56 μW
PALACS-2 latches	100.91 μW	201.60 μW	402.60 μW	804.54 μW	2025.21 μW
PALACS-2 logic circuit	39.40 μW	78.71 μW	157.25 μW	314.06 μW	786.06 μW
Total saving PALACS-2/M-S	25%	25%	25%	25%	24%
PALACS-4 clock net	88.44 μW	174.80 μW	349.47 μW	699.27 μW	1748.01 μW
PALACS-4 latches	101.81 μW	203.38 μW	406.23 μW	811.47 μW	2021.91 μW
PALACS-4 logic circuit	39.47 μW	78.84 μW	157.48 μW	314.52 μW	784.08 μW
Total saving PALACS-4/M-S	4%	4%	4%	4%	4%

reduced by 12% and the power consumption of the clock net is reduced by 42%. However, there is no power saving in the clock net for the four-phase PALACS, since it uses not two, but four separated clock signals.

5. Conclusions

The authors 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. The authors have compared the performance of these schemes with the two-phase master–slave clocking scheme in terms of speed and power consumption. PALACS outperforms master–slave in both speed and power. In the opinion of the authors, the most remarkable improvement is the power saving. The simpler two-phase PALACS, while comparable in complexity to the MSCS, is about 20% faster and greatly improves the consumption of the clock distribution network. 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.

Acknowledgements

This work was supported in part by the MCYT META project TEC 2004-00840 of the Spanish Government.

References

M. Afghahi and J. Yuan, “Double edge-triggered D-flip-flops for high-speed CMOS circuits,” *IEEE Journal of Solid-State Circuits*, 26, pp. 1168–1170, 1991.

H.B. Bakoglu (Ed.). *Circuits, Interconnections and Packaging for VLSI*, Add-Wesley Publishing Company, Menlo-Park, CA, USA, 1990. ISBN 0-201-06008-6.

K. Bernstein, *High Speed CMOS Design Styles*, Kluwer Academic Publishers, 1998. ISBN 0-7923-8220-X.

Affirma Spectre Circuit Simulator User Guide, Cadence Design Systems, Inc., San Jose, CA, USA, 2000.

D. Guerrero, M.J. Bellido, J.J. Chico, A. Millán and P. Ruiz, ''Two phase alternating latches clocking scheme for CMOS sequential circuits'', in *XVII Conference on Design of Circuits and Integrated Systems*, Santander, pp. 159–162, November 2002.

D. Guerrero, M.J. Bellido, J.J. Chico, A. Millán, E. Ostua and P. Ruiz, ''Four phase alternating latches clocking scheme for CMOS sequential circuits,'' in *XIX Conference on Design of Circuits and Integrated Systems*, Bordeaux, pp. 78–83, November 2004.

D. Harris, *Skew-Tolerant Circuit Design*, Morgan Kaufmann Publishers, 2001. San Francisco, CA, USA, ISBN 1-55860-636-X, pp. 14–20.

M. Horowitz, ''Clocking strategies in high performance processors,'' in *Symposium on VLSI Circuits Digest of technical Pagers*, pp. 50–53, 1992.

N. Nedovic and V.G. Oklobdzija, ''Dynamic Flip-Flop with Improved Power'', in *Proceedings of the 26th European Solid-State Circuits Conference*, Stockholm, pp. 376–379, September 2000.

V.G. Oklobdzija, ''Clocking and clocked storage elements in multi-GHz environment,'' in *12th International Workshop PATMOS*, Seville, pp. 128–145, 2002.

Ch. Tan and S.H. Unger, ''Clocking schemes for high-speed digital systems,'' *IEEE Transactions on Computers*, C-35, pp. 880–895, 1986.

V. Tiwari *et al*. ''Reducing power in high-performance microprocessors'', in *Proceedings of 35th Design Automation Conference*,'' San Francisco, pp. 732–737, 1998.