A Unified Single-Phase Clocking Scheme for VLSI Systems

MORTEZA AFGHAHI, MEMBER, IEEE, AND CHRISTER SVENSSON

***Abstract*—Two of the main consequences of advances in VLSI technologies are increased cost of design and wiring. In CMOS synchronous systems, this cost is partly due to tedious synchronization of different clock phases and routing of these clock signals. In this paper a single-phase clocking scheme is described that makes the design very compact and simple. It is shown that this scheme is general, simple, and safe. It provides a structure that can contain all components of a digital VLSI system including static, dynamic, and precharged logic as well as memories and PLA's. Clock and data signals are presented in a clean way that makes VLSI circuits and systems well-suited for design compilation.**

***Index terms*—Clocking, compilation, edge-triggered flip-flops, latches, one-phase clocking, skew, synchronous systems, timing, VLSI digital systems.**

I. Introduction

VLSI technologies are continuously being developed towards an increased level of integration and higher speeds. This has several consequences. Two of the most important consequences are an increased design cost and increased cost of wiring.

The increased cost of design is due to the increased complexity of the design. One way to cope with this is to use more systematic approaches to the design, sometimes called structured design [1]. So far structured design has been mainly applied to the structural part of the design (architecture, floorplan, etc.) and not so much to the timing part. In this paper we will discuss a structural timing design. Our goal is to develop a clocking strategy that is general, simple, and safe. Generality and simplicity make design easier and therefore decrease the design cost. Safety decreases the need for careful timing analysis and will also reduce design cost. These factors also make the CMOS circuits and systems well suited for design compilation.

Part of the high wiring cost of CMOS systems is caused by the many clock phases used; often four different clock phases are to be distributed all over a design [2]. The reason for this is that multiphase clocking has been recognized as both safe and transistor-count effective. In this paper we will show that a recently proposed scheme for single-phase clocking [3], [4] is just as safe and transistor-count effective as the previous four-phase clocking techniques. This technique will be shown to be general enough to handle static, dynamic, precharged, and precharged domino logic as well as array logic like memories or PLA's. Furthermore, it allows considerably higher clock frequencies than the normal multiphase techniques [4].

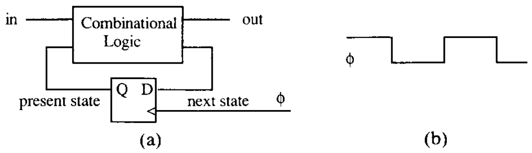


Fig. 1. (a) Single-phase state machine structure and (b) the clock signal.

II. Safe Clocking

In this paper we will limit our discussions on "safety" to timing behavior. We will thus use the term "safe" to denote safety against clock-skew and clock-slope problems, races, hazards, etc. Safe clocking is easily discussed in connection to a finite-state-machine model of logic (see Fig. 1) [1], [5]. The machine in this figure uses a positive edge-triggered D-flip-flop (ETDFF) as storage element. Input signals together with present state signals are inputs to a combinational logic block whose outputs are outputs and next-state signals of the machine. At the positive clock edge the next state becomes the present state and a new state is generated. The circuit is safe if the flip-flop is always nontransparent; that is, a state signal cannot pass the logic block more than once during one clock cycle. The discussion is easily extended to several interconnected finite state machines, a system which can be used as a model for any synchronous digital system. Designing circuits with ETDFF's is rather simple and straightforward and is used in most basic digital design texts [5].

However, if instead of edge-triggered elements data latches are used, it is possible to obtain faster and more transistor-count-effective circuits. But, it is shown that when traditional latches are used, single-phase clocking is neither safe nor fast [6] and a multiphase clock is required. This is due to the lower bound constraint on the minimum short-path delay of the combinational circuits, that could result in a data racethrough problem. For CMOS, when

Manuscript received June 13, 1989; revised September 25, 1989.

The authors are with the LSI Design Centre, Linköping University, S-581 83 Linköping, Sweden.

IEEE Log Number 8932697.

0018-9200/90/0200-0225$01.00 ©1990 IEEE

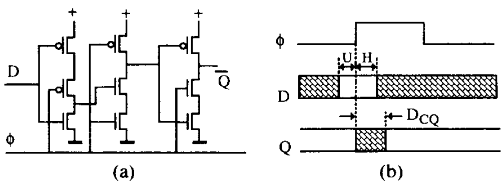


Fig. 2. Single-phase positive ETDFF: (a) circuit diagram and (b) parameters.

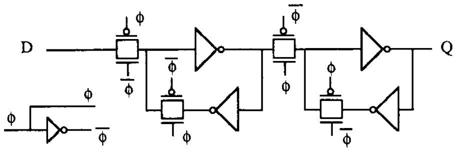


Fig. 3. Static single-phase ETDFF.

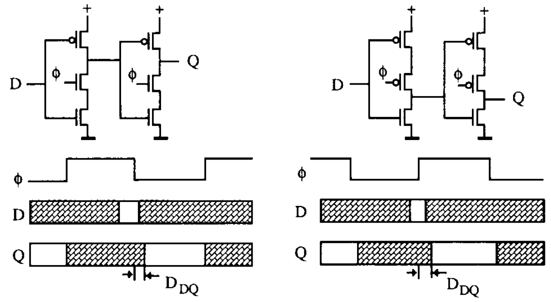


Fig. 4. Circuits and parameters of latches: (a) N-latch and (b) P-latch.

using latches, several schemes exist, in most cases requiring a four-phase clock signal (or pseudo-two-phase, which also means four clock signals) [2]. A simple example in this respect is a circuit according to Fig. 1, where the flip-flop is replaced by two dynamic latches (tristate inverters) clocked by a pseudo-two-phase clock [2]. Some of these schemes are used in connection with other techniques to save transistors, like precharged or precharged domino techniques. In the following, however, it is our intention to show that it is possible to combine the simple and safe single-phase clocking with transistor-count-effective latches. Furthermore, we propose a method that can safely handle up to half a clock period clock skew between two communicating machines without any constraint on the minimum short-path delay of the combinational logic circuit.

In this paper, we divide signals in a system into two classes, namely, clock signal and data signal. The only constraint on the clock signal is that its reference edges must occur only when specified. On the other hand, there are data signals that must obey certain specifications with respect to the clock-signal reference edges. This clear picture of the signals in a system makes the design and verification simpler. To gate the clock signal with a data signal or to delay the primary clock signal destroys this picture. Moreover, gating the clock signal usually results in considerable clock skew. Circuit and bus structures will be considered to selectively transfer data between different parts of the system without gating the clock signal with data signals.

III. Basic Components of Single-Phase Clocking

A. Storage Elements

In a synchronous system the movement of data is controlled by clocked storage elements. Thus, in this section we consider the timing behavior of storage elements of a single-phase clocking scheme.

A dynamic positive ETDFF [4] is shown in Fig. 2(a). The timing behavior of this device is the same as that of other ETDFF's, i.e., the input data must be valid during the interval given by the setup (U) and hold (H) times of the device and the output data are available and stable D_{CQ} after the active clock edge. The value of the hold time H of this kind of flip-flop is close to zero (see Section V). It is also possible to design a similar circuit which is triggered on the negative clock edge instead of the positive edge [4]. One may note that the proposed flip-flop is inverting. This is normally of little importance. The only limitation of this class of circuits is that the clock rise time must be short enough. This is, however, not a strong limitation, as a rise time of the order of 20 times the logical delay can be accepted [4], [12].

This dynamic flip-flop can thus replace the flip-flop in Fig. 1. It uses nine transistors and one clock line, which is more effective than the standard pseudo-two-phase CMOS solution using eight transistors and four clock lines (two tristate inverters as flip-flop).

If a static flip-flop is needed, a traditional CMOS master–slave flip-flop based on transmission gates [2] fits well into this timing scheme. In Fig. 3 this static ETDFF is demonstrated. This circuit is not really a true single-phase clocked circuit but is safely clocked by a single-phase clock if properly designed. By using a local inverter, as in the figure, the clock skew is under local control so that a properly designed cell will be safe in any environment. This cell uses 18 transistors.

The reason for using a two-phase nonoverlapping clock (pseudo-four-phase in CMOS) when latches are used is mainly to create two different points within each clock period to latch the data. In this way, a holding period is created before each latch that prevents the racethrough problem. This requirement may also be achieved by designing two latches that are transparent on different parts of the same clock signal, i.e., high and low part of the signal. Fig. 4(a) shows a circuit for a latch that is transparent when the clock signal is high and latches the data when the clock signal goes low. Fig. 4(b) shows the counterpart of this circuit, i.e., a circuit that is transparent when the clock signal is low and latches the data when the clock signal goes high [4]. These circuits may be called N- and P-latches, as the clock signal is applied to n and p transis-

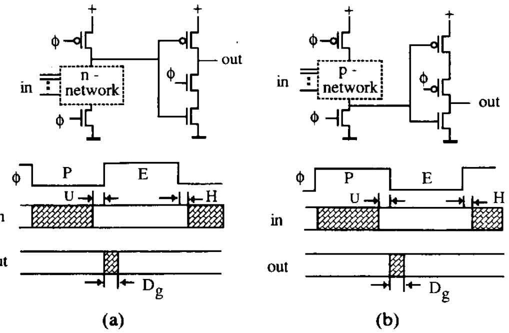


Fig. 5. Circuits and timing diagrams of (a) N-block and (b) P-block.

tors in these devices. In the same figure timing diagrams of these circuits are also depicted. Parameters of these latches are similar to other latches. Again, these devices have a forbidden window and the output signal is stable following the propagation delay of the element after the latching edge of the clock signal. As in the case of the dynamic ETDFF above, the clock edge must be fast enough.

B. Precharged Logic and Domino Logic

Precharged dynamic logic utilizes the MOS device characteristics to their best advantages. In fact, another reason for appealing to a multiphase clocking scheme was to enable several levels of nMOS dynamic logic evaluation to be performed per clock cycle [7]. This is now made possible for CMOS circuits even when a single-phase signal is used [3]. Single-phase precharged dynamic logic is based on N- and P-blocks. These blocks, with their respective timing diagrams, are demonstrated in Fig. 5(a) and (b). An N-block is precharging (P) when the clock signal is low and evaluates (E) its stable inputs when the clock signal is high. On the contrary, a P-block is in the precharge phase when the clock is high and in the evaluation phase when the clock signal is low. Therefore, a P-block using the same clock as an N-block can follow an N-block as it is in the precharging phase when the N-block is evaluating; that is, it can change its outputs. Obviously, the output of the N-block cannot be used to drive another N-block as they are in the evaluation phase at the same time. Again, these blocks can be characterized by a forbidden window during which the input data must stay stable and a delay after which the output data are stable.

As in the case of multiphase precharged logic, several levels of logic can be evaluated during one clock cycle using the domino technique [4]. This is done in the usual way, by replacing the logical stage in Fig. 5 with several cascaded logical stages separated by inverters [4], [15]. The single-phase clock thus imposes no limitations on the flexibility of the precharged/domino technique.

Comparing Fig. 2(a) and Fig. 5(a), one notices that the last stage of the positive ETDFF is identical to the last stage (the latching stage) of an N-block and, therefore, they have the same output timing. Thus a positive ETDFF can always drive a P-block and a negative ETDFF can always drive an N-block.

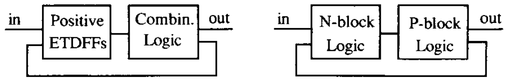


Fig. 6. Block diagram of a single-phase system with ETDFF's.

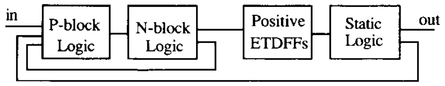


Fig. 7. A mixed technique.

TABLE I
ALLOWABLE COMBINATIONS OF FLIP-FLOPS AND PRECHARGE LOGICS

From :

To : ↓	+edge	- edge	N - Block	P - Block
+edge	ok	ok	ok	ok
- edge	ok	ok	ok	ok
N-Block		ok		ok
P-Block	ok		ok	

+edge= Positive ETDFF - edge= Negative ETDFF

IV. Single-Phase Subsystem Circuits

In this section we will show that all types of circuits used in a CMOS design fit well into the single-phase clocking scheme. To this end, we employ the general concept of the finite state machine used in Section II. It is also important to include array logics, such as memory and PLA circuits, in the same regular clocking scheme.

A. Single-Phase Finite State Machine

When ETDFF's are employed as storage elements, it is relatively simple to fulfill the requirements of the state machine. The structure of the circuit becomes as shown in Fig. 6(a). In this structure, the output signal of the machine is stable after the delays of the flip-flop and the combinational circuit following the leading edge of the clock signal. This output signal can drive another state machine with a similar structure. The combinational logic is normally simple static logic in this case.

We may also form a finite state machine from precharged logic. The feedback loop must then include two precharged stages, one N-block and one P-block. Such a machine is demonstrated in Fig. 6(b).

Furthermore, we may also mix static and precharged logic. For example, static logic may be inserted between the blocks in Fig. 6(b) or a technique as depicted in Fig. 7 may be employed. The only important point to consider is that the timing of the previous latching circuit must fit the timing of the following latching or precharged circuit. In Table I a set of rules covering this problem is given.

When latches are used as storage elements, the block diagram of the state machine may take the configuration shown in Fig. 8. In the combinational logic circuit both

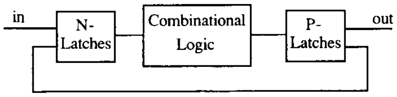


Fig. 8. Block diagram of the system with latches.

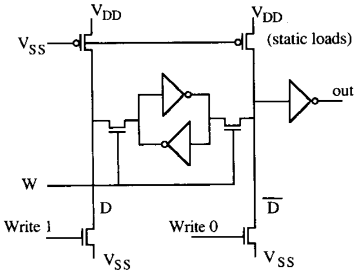


Fig. 9. Structure of a six-transistor RAM cell.

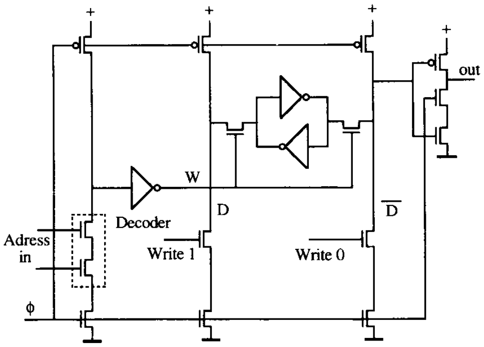


Fig. 10. Single-phase dynamic RAM circuit.

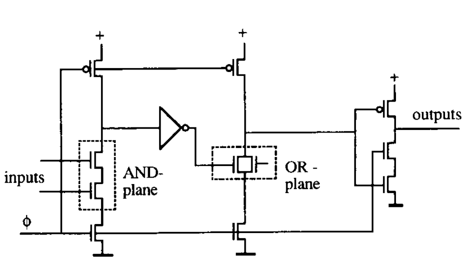


Fig. 11. Single-phase PLA circuit.

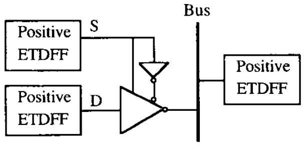


Fig. 12. Circuit for bus structures.

precharge p-units and static circuits can be included. This output signal can drive any system with a similar structure or systems with ETDFF's as storage element.

B. Single-Phase Memory and PLA Structures

The circuit of a normal CMOS RAM cell [2] is shown in Fig. 9. Note that this circuit is not a true CMOS circuit, since the circuit includes a static load (pseudo-nMOS). The memory cell is accessed by making the word select line W high. Often, the address decoder, generating the signals on W, is also designed in pseudo-nMOS in order to save transistors.

Now it will be shown how this memory cell and the address decoder can be designed in a two-stage domino mode with true CMOS (i.e., no static loads) and with the same standard single-phase clock as used above. This is demonstrated in Fig. 10. As shown, the address decoder is the first stage in a domino circuit and the memory cell and the bus circuitry are operated as a second stage. Both stages are then included in an N-block. As in any N-block (see Fig. 5(a)) the read data are latched when the clock signal goes low.

This proposed scheme again gives a block with the same timing behavior as an N-block (see Fig. 5(a)). Both address-in and write 1 and write 0 signals are then considered as input signals. The memory or the register stack thus fits exactly into the same timing scheme as any other block.

In addition, a ROM is easily designed in the same scheme. For a ROM we need only one data bus D and the memory cell includes only one transistor, connected between ground and D with its gate connected to W [2]. Data are stored by omitting some of the transistors in the array (see also Fig. 11).

The structure of a PLA also fits into the above scheme. Again, the AND plane can be considered as the first stage of a domino logic and the OR plane as the second stage (see Fig. 11). If some of the outputs are to be fed back to the inputs of the PLA to form a state machine structure, we may use the principle of Fig. 6(b). Note that the evaluation transistor (the clocked n-transistor) can be omitted in the second stage as the signal from the first stage in the domino chain always is zero during precharge. Again, the whole PLA has the same timing as an ordinary N-block.

It is also quite possible to use other variations of circuits to implement RAM's, ROM's, or PLA's. In all cases we may, for example, replace the two-stage domino principle with an N-block followed by a P-block (or vice versa). The latter principle will require more transistors but will give rise to higher speed due to less logical depth.

C. Bus Structures

The problem of selectively delivering data from flip-flops or latches to a bus, or receiving data from a bus, can give rise to timing problems [1, ch. 5]. In our structured single-phase clocking scheme we would prefer to keep the clock signal intact and not gate it or in other ways introduce an extra delay to it. The problem can be solved as shown in Fig. 12, where we access the bus with a tristate inverter, controlled by a signal with the same timing protocol as the data. The bus data will then also follow the same timing protocol and the whole system fits into the scheme already described.

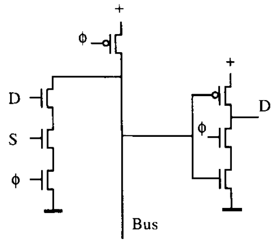


Fig. 13. Precharged bus system.

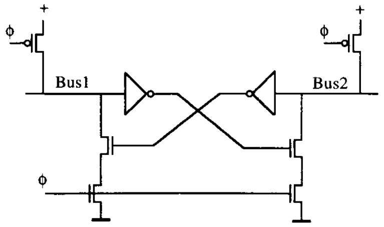


Fig. 14. Bidirectional repeater on a precharged bus.

A very useful bus structure is the precharged bus [1]. It is not only used internally on a chip but it may also be used externally, between chips. Again, we find that the single-phase clocking scheme can be applied. The precharged bus may use the same structure and timing as the internal node in a precharged circuit (Fig. 5(a)). This is demonstrated in Fig. 13, where we show a precharged bus with one input circuit and one output latch.

The input circuit is equivalent to the logic part of a precharged gate, where data and select are treated equal (and follow the same timing protocol). This circuit is thus equivalent to an open tristate when select is low. The input circuit may also include logic. If more than one input is active, the bus will perform a wired-OR function and still give valid data. The bus data can be captured either by an output latch as shown in Fig. 13 or by using it directly as input data to a second domino stage. The role of the output latch is to convert the data timing protocol from a precharged form to a normal form.

By using a precharged bus, it is also possible to introduce a bidirectional repeater circuit (which is not possible in a normal bus, see Fig. 14) [8]. This circuit is useful in long buses with large capacitive loads. It can also be used to minimize the effect of the *RC* delay of the line [9]. During precharge, both bus parts are precharged high. During evaluation, both bus parts will stay high unless one or both are made low, then both will become low. The bidirectional repeater is thus transparent both for data and for the wired-OR function between many input datas.

V. Timing of the Single-Phase Clocking Scheme

In this section timing considerations of the single-phase clocking scheme are covered. For this purpose we use the circuits of the finite state machine. After a brief discussion about the clock signal, clock skew, and logic delays, we will consider both ETDFF's and latches as storage elements and derive the necessary constraints between the clock signal and different delays to guarantee proper operation of the machine.

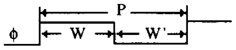


Fig. 15. Parameters of a single-phase clock signal.

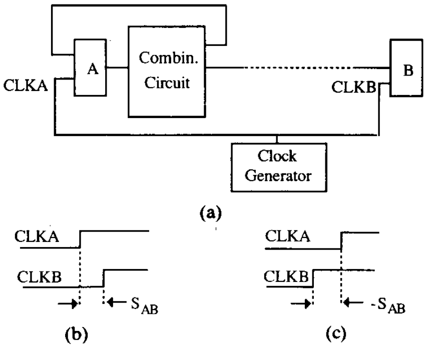


Fig. 16. Demonstration of clock skew in a single-phase system: (a) system diagram, (b) negative skew, and (c) positive skew.

The clock signal is specified by its width and period (see Fig. 15). Transition points of the clock signal, i.e., where the signal goes high or low, are also important and may be called reference edges.

In a synchronous system it is ideal if these reference edges occur at the same moment of time at all synchronizing points of the system, that is, where storage elements are placed. In reality, however, clock signals which propagate along routes with variable delays or signals may reach the same point from different paths. This variation of the delays results in clock skew. In a circuit two types of skew are encountered. They may be called self-skew and relative skew [10]. To explain the difference between these two, consider Fig. 16(a). The self-skew of *CLKA* is the delay variation of the path from the clock generator to register *A*. Delay variations are caused by imprecision in processing, variation of temperature and power supply, variability among individual storage elements in the register, slopes in the clock pulse edges, etc. The effect of self-skew on timing is insignificant.

The relative skew between the clock signals *CLKA* and *CLKB* is the difference between the delays of the paths from the clock generator to registers *A* and *B*. *CLKA* and *CLKB* timing may then take the form of Fig. 16(b) where we define the skew between clock *A* and clock *B*, S_{AB}, as the time lag of *CLKB*'s clock edge after *CLKA*'s corresponding clock edge. Note that S_{AB} may take both positive and negative values. We will further use S_{ABM} for the maximum value of skew and S_{AB_m} for its minimum value. Relative skew is evident in the same clock period and is present in nonlocal communications or when different flip-flops in a register receive clock signals from different paths. In estimating clock skew one may use worst-case

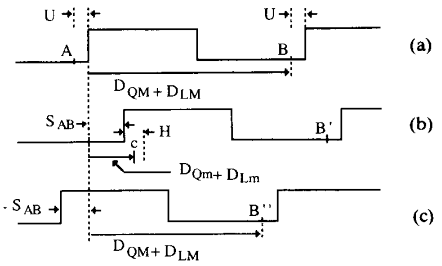


Fig. 17. Timing diagram for finite state machine with edge-triggered flip-flop.

analysis or statistical analysis [11]. In the worst-case analysis the skew is the difference between the worst delay (slowest) and the best delay (fastest) of the paths. This is assumed in Fig. 16. Statistical analysis usually results in lower skew but at the cost of a nonzero failure rate. In the following we assume that the skew associated with the leading edge and the trailing edge of the clock signal are equal. This assumption is reasonable since clock signals pass through several inverters before they reach the registers.

The combinational logic circuit included in the state machine structure can also be specified by two delay parameters. The maximum and minimum propagation delays through all circuit paths connecting any input to any output will be called D_{LM} and D_{Lm}. Flip-flops and latches are also characterized by maximum and minimum delays. We now consider the timing of single-phase clocking with edge-triggered flip-flops, latches, and precharged circuits.

A. Edge-Triggered Flip-Flop Timing

The aim of timing calculation is to specify a minimum period for the clock signal such that the state machine circuit operates correctly. For this purpose consider Fig. 16(a) and assume that registers A and B are edge-triggered flip-flops. Now consider a data stored at the input of the flip-flop A. For deterministic behavior of the state machine, this data when triggered in must have enough time to trip round the feedback loop and reach the input of the same flip-flop before the setup time of that device for the next clock cycle. This condition is graphically depicted in Fig. 17(a). An input data at point A in the current clock cycle must reach point B for the next cycle. In this case:

$$P \geqslant D_{QM} + D_{LM} + U \tag{1}$$

where D_{QM} is the maximum delay of the flip-flop. Furthermore, the data must not trip around the machine too fast, or to be more exact, it must not arrive at the flip-flop input before the end of the hold time for the present clock edge:

$$D_{Qm} + D_{Lm} \geqslant H. \tag{2}$$

As mentioned above, H is normally very close to zero, hence this condition is always fulfilled.

Now consider a situation where one of the outputs of the machine is feeding a nonlocal flip-flop, say, flip-flop B in Fig. 16(a). In this case the relative position of the clock signals at the site of flip-flop A and B may be as depicted in Fig. 16(b). Following the same principle stated above and referring to Fig. 17(b):

$$P \geqslant D_{QM} + D_{LM} + U - S_{ABm}. \tag{3}$$

The other situation must be considered. The minimum delays of the flip-flop and the combinational circuit may be such that the output signal reaches flip-flop B before the hold time of this device (see point C in Fig. 17(b)). Therefore, to avoid violation of the hold time of flip-flop B, we must have

$$D_{Lm} + D_{Qm} \geqslant H + S_{ABM} \tag{4}$$

where D_{Qm} is the minimum delay of the flip-flop. In other words, one may say that the maximum allowable clock skew is

$$S_{ABM} \leqslant D_{Lm} + D_{Qm} - H. \tag{5}$$

In the case when the skew is negative, Fig. 17(c), the clock period will increase and is obtained from the relation (3) with the sign of the skew changed. On the other hand, according to (5), the maximum positive skew is limited to the minimum delay of the flip-flop and combinational logic. This class of circuits is thus quite sensitive to positive clock skew [12], [13]. In a general system, (5) and (3) must be used for the maximum allowable skew and minimum clock period, respectively.

It should be mentioned that when a positive ETDFF drives a negative ETDFF, the minimum clock-pulse width, rather than the clock period, must fulfill the above expressions for the clock period. In this case the system will be much less sensitive to positive clock skew.

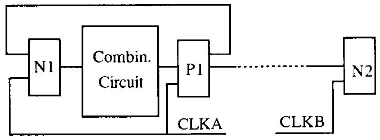


Fig. 18. Block diagram of single-phase finite state machine with latches as storage elements.

B. Data Latch Timing

The argument followed here is of the same form as used above. That is, the clock signal width and period are so determined that if data propagates along different paths and round the feedback loop, the setup and hold times of the latches must not be violated. We assume that the input data to a latch do not wait for the enabling clock edge. Rather, latches are open before input data become valid [10]. With this measure the system becomes faster and timing analysis becomes more clear. Fig. 18 shows a block diagram of the single-phase system considered here.

Now consider a data at the input of the latch $N1$. The latest arrival time of this data is the point A in Fig. 19. This data should have enough time to propagate through

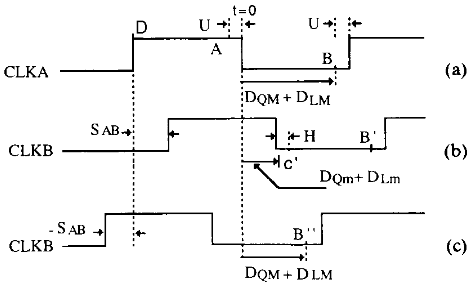


Fig. 19. Timing diagram for the state machine of Fig. 18.

the latch and the combinational circuit and reach the input of the $P1$ latch before the setup time of this latch (point B in the same figure). For the sake of clarity, we assume that the maximum and minimum delays and setup and hold times of the $N1$ latches are equal to that of the $P1$ latches. Therefore, for the above condition to hold,

$$W' \geqslant D_{QM} + D_{LM} + U \tag{6}$$

where W' is the low part of the clock signal. If another combinational circuit were included in the feedback loop between the $P1$ and the $N1$ latch, a similar relation would apply for the width of the clock signal. In the case of Fig. 18:

$$W \geqslant D_{QM} + U \tag{7}$$

where D_{QM} is the maximum delay of the $P1$ latch. Now consider a situation when one of the outputs of the $P1$ latches is sent to a remote latch $N2$ instead of to $N1$. Relative position of the $CLKA$ and $CLKB$ may then be as in Fig. 19. In this case:

$$W' \geqslant D_{QM} + D_{LM} + U - S_{ABm}. \tag{8}$$

Now we consider the effects of minimum delays of the combinational logic circuit and latches. The minimum delays of the latch and the combinational circuit must be such that the output of the $N1$ latch does not arrive to the $N2$-latch input before the end of the hold time of this latch for the present clock period (point c' in Fig. 19(b)). Therefore, to avoid violation of the hold time of the $N2$ latch, we must have

$$D_{Lm} + D_{Qm} \geqslant H + S_{ABM} \tag{9}$$

where D_{Qm} is the minimum delay of the latch. In other words, one may say that the maximum allowable clock skew is

$$S_{ABM} \leqslant D_{Lm} + D_{Qm} - H. \tag{10}$$

In the case when the skew is negative, Fig. 19(c), the W' will increase and is obtained from relation (8) with the sign of the skew changed. Again, in a general system, (10) and (8) and (7) must be used for the maximum allowable skew and minimum clock period, respectively.

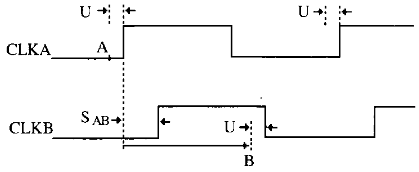


Fig. 20. Timing diagram for precharged circuits.

Traditionally, in single-phase clocking using latches, only one latch is used in the feedback loop of the state machine circuit [6]. This is equivalent to taking out latch $P1$ from Fig. 18. In this case the minimum delay of the combinational circuit and latches must be so long that a data arriving at the $N1$-latch input at point D in Fig. 19(a) cannot race through the circuit and reach the $N2$-latch input before point c'; that is:

$$D_{Lm} + D_{Qm} \geqslant W + S_{ABM}. \tag{11}$$

Fulfilling this condition makes the design very expensive and difficult. This constraint on the minimum delay of the combinational circuit and latches is a major drawback of traditional single-phase clocking with latches. This forced the designers to appeal to two-phase clocking (four phases in CMOS) in order to create two reference edges within the same clock period. Then two latches of the same kind but clocked by different phases are used in the feedback loop of the state machine.

In the single-phase scheme considered here, however, instead of using two phases we use two different kinds of latches: one that is transparent during the high part of the clock signal and the other that is transparent during the low part of the clock signal. In fact, the $P1$ latch in Fig. 18 is a barrier that stops the propagation of the data up to point $t = 0$ in Fig. 19(a). In this respect this single-phase scheme is equivalent to a two-phase scheme with no overlap or gap between the phases. Here it is only necessary to guarantee that a data released at point $t = 0$ by the $P1$ latch will not reach the $N2$-latch input before the hold time of this latch (see relation (9)). In Section VI we propose a technique that relaxes this constraint even further.

C. Precharged Circuit Timing

Here again we use the same argument as above to determine the minimum clock width and period. Assume that an N-block clocked by $CLKA$ is communicating with a P-block clocked by $CLKB$ (see Fig. 20). The latest arrival time of a data to the input of the N-block is denoted by point A in this figure. This data must have enough time to be evaluated by the N-block and reach the input of the p-block before the setup time of this block (point B in the figure). Therefore

$$W \geqslant D_{LM} + U - S_{ABm}. \tag{12}$$

W is also constrained by the precharge time T_p, so that $W \geqslant T_p$. Following the same discussion that led to (5), the

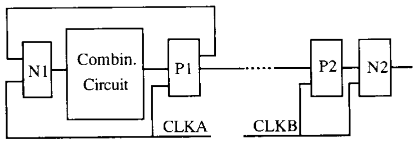


Fig. 21. Circuit technique to cancel the effects of clock skew. An extra latch is inserted on the site of the destination latch.

maximum allowable skew is

$$S_{ABM} \leqslant D_{Lm} - H. \tag{13}$$

Again, for this kind of circuit, $H \approx 0$. If there is a skew, (12) shows that a negative skew will increase the clock period. On the other hand, according to (13) the maximum positive skew is limited to the minimum delay of the block. This class of circuits is thus sensitive to positive clock skew unless some special technique is used (see [4] and [12]).

VI. Signal Resynchronization

It is usually expensive and difficult to satisfy the constraints between the clock skew and the minimum delay of the combinational logic and latches (relation (10)). To reduce the clock skew, great efforts must be exerted, for example, to hand-tune the delays in clock distribution paths, and to control other related factors. On the other hand, to increase the minimum delay of the combinational logic circuit, it is usually necessary to add extra delays (e.g., inverters) at the latch outputs as well as to increase the clock period. In the following, a method is proposed that when used can handle up to half a clock period skew with no constraint on the minimum combinational logic delay.

If we compare Fig. 19(a) and (b), it is evident that the problem is raised because when the data are transmitted by the $P1$ latch at $t = 0$, $CLKB$ is still at high state. But, the data should reach the $N2$ latch first after the $CLKB$ has gone to low state to avoid setup violation of the $N2$ latch. One, for example, can use a negative edge-triggered flip-flop instead of an $N2$ latch that receives data from the remote block. In this way, however, the setup time of the edge-triggered flip-flop may be violated. To solve this problem one can insert an extra P-latch before the $N2$ latch (see Fig. 21). By these methods a clock skew as long as half a clock period can be handled safely without putting any constraint on the minimum delay of the circuit elements (see also [14]). This same technique can also be used in association with edge-tgriggered flip-flops to relax relation (5).

When the clock skew is negative, the above racethrough problem is not present but the clock period will be longer, in accordance with relation (10).

VII. Conclusions

In this paper a unified single-phase clocking scheme was proposed. It was shown that this scheme is general, safe, and simple. The generality makes it possible to use the scheme consistently throughout a digital VLSI system. It provides a structure that can contain all components of a digital VLSI system including static, dynamic, and precharge logic as well as memories and PLA's. This also makes the CMOS circuits and systems well suited for design compilation. Timing expressions for circuits with different storage elements were developed. It was shown that a negative clock skew will just increase the clock period (this is also equivalent to signal delay) and that positive skew may give rise to failure. The problems associated with traditional one-phase clocking using latches are avoided. Techniques were proposed that relax the constraints between the clock skew and the minimum delays of the combinational logic circuits and storage elements. Using the scheme and techniques proposed in this paper should provide a simple and safe clocking for high-speed and compact VLSI digital systems. It also makes the visualization of the system simple and resembles the traditional digital systems where only edge-triggered devices were used.

References

[1] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980.

[2] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. Reading, MA: Addison-Wesley, 1985.

[3] Y. Ji-ren, I. Karlsson, and C. Svensson, "A true single phase clock dynamic CMOS circuit technique," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 899–901, 1987.

[4] J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *IEEE J. Solid-State Circuits*, vol. 24, pp. 62–71, 1989.

[5] D. L. Dietmeyer, *Logic Design of Digital Systems*. Boston: Allyn and Bacon, 1971.

[6] S. H. Unger and C. Tan, "Clocking schemes for high-speed digital systems," *IEEE Trans. Comput.*, vol. C-35, pp. 880–895, 1986.

[7] M. Penney and L. Lau, *MOS Integrated Circuits*. Princeton, NJ: Van Nostrand, 1972.

[8] K. Chen and C. Svensson, private communication, 1989.

[9] M. Afghahi and C. Svensson, "A scalable VLSI synchronous system," in *Proc. IEEE Symp. Circuits Syst.* (Espoo, Finland), 1988, pp. 471–474.

[10] G. G. Langdon, *Computer Design*. San Jose, CA: Computeach, 1982.

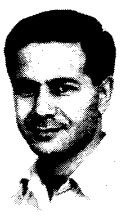
[11] M. Afghahi and C. Svensson, "Calculation of clock path delay and skew in VLSI systems," in *Proc. European Conf. Circuit Theory Design* (Brighton, England), 1989, pp. 265–269.

[12] I. Karlsson, "True single phase clock dynamic CMOS circuit technique," in *Proc. IEEE Symp. Circuits Syst.* (Espoo, Finland), 1988, pp. 475–478.

[13] M. Hatamian and C. L. Cash, "Parallel bit-level VLSI design for high-speed signal processing," *Proc. IEEE*, vol. 75, pp. 1192–1202, 1987.

[14] C. Svensson, "Signal resynchronisation in VLSI systems," *Integration, The VLSI J.*, vol. 4, pp. 75–80, 1986.

[15] R. H. Krambeck, C. M. Lee, and H. S. Law, "High-speed compact circuits with CMOS," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 614–619, 1982.



Morteza Afghahi (S'68–M'88) was born in Twiserkan, Iran, in 1950. He received the B.Sc. degree in electrical engineering from the Technical University of Tehran in 1972. He received the Ph.D. degree also in electrical engineering from the University of Newcastle upon Tyne, England, in 1980.

From 1980 to 1984 he was a Project Manager at MonenCo-Iran, where he was engaged in electrical system design. In 1983 and 1984 he was a Visiting Lecturer at Tehran Polytechnic. Since

1985 he has been working at Linköping University, Linköping, Sweden, first as a Guest Scientist and now as a Research Assistant. His main research fields are VLSI implementation of DSP algorithms, VLSI systems and circuit design, and MOS sensors.

Christer Svensson was born in Borås, Sweden, in 1941. He received the M.S. and Ph.D. degrees in electrical engineering from Chalmers University of Technology, Gothenburg, Sweden, in 1965 and 1970, respectively.

He was with the Research Laboratory of Electronics at Chalmers University from 1965 to 1978, where he performed research on MOS transistors, MNOS memory transistors, Pd-MOS gas-sensitive transistors, and the physics and chemistry of the MOS system. During 1972 he was a Guest Scientist at the Jet Propulsion Laboratory, Pasadena, CA. Since 1978 he has been with the Department of Physics and Measurement Technology, Linköping University, Linköping, Sweden, first as a lecturer and since 1983 as a Professor of Electronic Devices. At Linköping he has continued his research on MOS physics and chemical sensors but also started a new research group on integrated circuit design. He is now performing research on high-performance analog and digital CMOS circuit techniques, on timing verification and optimization of CMOS circuits, and on integrated electronic and sensor systems.