Single-Rail Self-timed Logic Circuits in Synchronous Designs

Frank Grassert, Dirk Timmermann

University of Rostock
Institute for Applied Microelectronics and Computer Science
Richard-Wagner-Str. 31, 18119 Rostock-Warnemünde, Germany
{frank.grassert; dirk.timmermann}@etechnik.uni-rostock.de

ABSTRACT

This paper presents a self-timed scheme for dynamic single-rail logic integrated in a single phase clock design. A generalized completion detection for generation of self-timed signals from single-rail gates is described and we show a novel application of the redundancy of a SD-adder to ease the self-timed signal generation. Further we discuss an universal evaluation scheme to overcome the problem of only non-inverting functions with dynamic single-rail gates. The presented SD-adder was integrated in a synchronous scheme and combines the advantages of simple synthesis and clock distribution for synchronous designs with fastest evaluation. Self-timed schemes result in fastest latch-free structures and robustness against clock-skew. Further the single-rail scheme on gate-level yields lower power consumption and smaller circuits. The use of inverting and non-inverting single-rail gates makes the synthesis close to standard synthesis. Simulations for the redundant adder design show area and power savings of 40% and 30% compared to complementary DOMINO logic structure.

1. INTRODUCTION

Dynamic logic styles yield fast evaluation, high throughput and small latency. However, the main disadvantages are often multi-phase clocking schemes, dual-rail structures with large area and high power consumption. For highest speed, latch-free self-timed structures are used, which usually work with dual-rail logics. Single-rail logics are smaller and have less power consumption, but the generation of self-timed signals is difficult and only non-inverting functions can be used. Because the synthesis and clock distribution is much easier in synchronous designs a single phase clock is desirable and therefore, an integration of self-timed schemes in a synchronous structure combines fastest latch-free evaluation with single-clock behavior.

DOMINO logic [1] is a simple realization of the dynamic idea but requires two clocking signals in minimum for pipelined designs (figure 1). For highest speed Harris et. al. [2] published a latch-free skew tolerant realization, where a scheme of overlapping clocking signals decreases sensitivity to clock slopes and the propagation delay is only the sum of the gate delays.

Another disadvantage of DOMINO is, that only non-inverting logical functions can be realized. The clock-delayed DOMINO logic in [3] is an example of a single-rail dynamic logic, which uses dynamic nodes as inputs, but requires the detection of slowest inputs. In complex logic this requirement is hard to fulfill, because the evaluation time may depend on the input signals. Further, the problem of false discharge during precharging the inputs is not discussed. Other solutions for only non-inverting functions with DOMINO are avoiding all inverting functions in the netlist, but this often results in larger designs, or using a complementary structure (dual-rail). Here, for each signal an inverted signal exists.

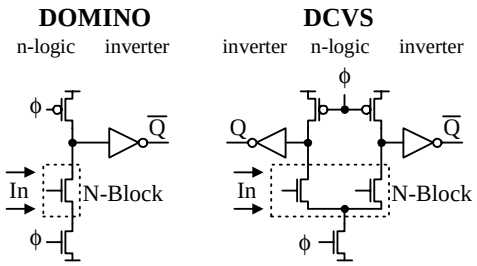

Figure 1. DOMINO, and DCVS logic; DCVSL merges the complementary logic function in the n-block; DOMINO and DCVS need more than one clock signal in pipelined structures

A different approach for highest speed is the use of dynamic logic styles in asynchronous designs [4]. In [5], a divider using a ring structure was realized and yields no delay in addition to the evaluation time. Such self-timed techniques require completion signals and, therefore, differential logic styles or dual-rail realizations are used. In [6], Differential Cascode Voltage Switch Logic (DCVSL) was introduced (figure 1). It builds the starting point for several differential logic styles and was derived from two complementary DOMINO gates with merged logic trees.

In this paper,

- the main goal is the usage of short self-timed chains in synchronous designs for fast latch-free evaluation.
- the presented generalized completion detection makes the usage of single-rail logic in self-timed schemes possible.
- the novel application of the redundancy of a SD-adder eases the self-timed signal generation.
- the presented evaluation scheme allows the use of dynamic nodes as inputs and we overcome the problem of only non-inverting outputs for dynamic single-rail logics.
- the simulated adder combines synchronous behavior with small latency and low power consumption.

Section 2 describes the basics of self-timed structures and the completion detection in single-rail dynamic logic. A possibility to use dynamic nodes as direct inputs is introduced in section 3. In section 4 the design of an asynchronous redundant adder is explained and the results are discussed in section 5. Section 6 presents the conclusions.

2. SELF-TIMED STRUCTURES

2.1 Basics

In a differential logic style the complementary functions are merged together and are not independent. This sometimes results in less transistors and, therefore, in a smaller area in comparison with dual-rail DOMINO logic. In contrast any single-rail logic can be converted to dual-rail logic by independently building the complementary logic part.

Differential logic styles are used for self-timed circuits where we need to detect whether a gate has evaluated. In dynamic logic styles both complementary output signals have the same value during precharge, but change to different values during evaluation. Therefore, the generation of completion signals is simple. This seems to be an advantage, but we have to reflect that this results from the overhead of complementary signals. Therefore, a dual-rail structure becomes more efficient in self-timed circuits.

2.2 Self-timed Schemes

Self-timed structures reduce the latency and make latches and flipflops unnecessary. In [7] we presented a scheme to include self-timed structures in a global clock system. There it is possible, that the critical path has an evaluation time of only the sum of all the evaluation times of the single gates without additional delays through latches. Therefore, the evaluation time of the critical path is the minimum clock period. However, the structure must be carefully designed and a calculation of the timing behavior is advantageous. Furthermore, such structure reduces the sensibility against clock skew and, consequently, no additional delays for a safe function is necessary.

There are two main ways to build up a self-timed scheme for dynamic logic: the gate outputs control the clocking of the previous or the following gate. The main advantage of the first structure is a simple implementation with minimum evaluation time. If the evaluated outputs of a gate have settled, a completion signal is generated and this sets the previous gate in the precharge phase (inputs are processed – start precharge). Precharged outputs set the previous gate in the evaluation phase (outputs are precharged – start next evaluation, inputs can be processed). Because the evaluation of the outputs starts only with valid inputs, every gate is waiting for valid inputs during the evaluation phase. Therefore, the evaluation time is only the sum of the gate evaluation times with no additional delays. Figure 2 shows such a dynamic dual-rail self-timed structure.

2.3 Generalized Completion Detection

The completion detection follows the principle of differentiating the states of the outputs nodes after precharge and after evaluation. A complementary structure ensures an output change on every gate, because the complementary outputs are generated independently. To generalize this behavior we can detect the completion on a set of n output signals of n different gates, if they fulfill the equation:

$$f_1(I_m) \text{ or } f_2(I_m) \text{ or } \cdots \text{ or } f_n(I_m) = 1.$$

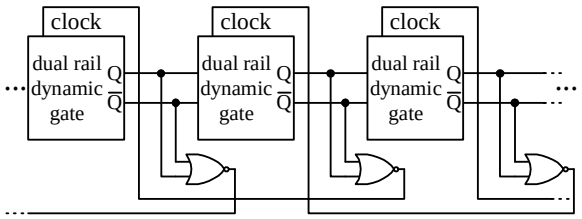

Figure 2. Self-timed structure with dual-rail dynamic logic

This assumes that all these gates f_i evaluate in a small range of time, so that completion generation does not disturb slower or faster paths. This assumption seems reasonable in most small blocks with limited logic depth, for example in data paths. Then, during precharge all outputs stay on the same value and during evaluation at least one output must change its value and, therefore, the completion detection is ensured. The completion logic is little enlarged because a 3- or more-input NOR can be necessary. The essential gate outputs can be find out by logic synthesis. If no such completion can be applied, an additional gate must be implemented exclusive for self-timed generation. The utilization of a redundant number system is a good way for easy completion generation (section 4).

3. SINGLE-RAIL LOGIC

The main goal for the use of dynamic logic styles is to use only faster n-transistors in the logic trees, to achieve higher performance and smaller area. The problem of directly connecting two dynamic stages is, that the precharged high output node will result in conducting n-transistors in the following logic tree. At the beginning of the evaluation phase this connection to ground can lead to a discharge of the following output and in an erroneous state. DOMINO logic solves this issue with an inverter, so following n-transistors are off and no discharge can occur. Therefore, single-rail logic styles like DOMINO can realize only non-inverting functions. To overcome these limitation the following conditions for evaluation and precharge must be hold.

3.1 Generalized Conditions for Evaluation

To use simple single-rail logic equivalent to static CMOS an inversion is necessary. For the following explanations we define a DOMINO output (dynamic stage plus inverter) as a non-inverted signal and the dynamic node (output of the dynamic stage) as the inverted signal. It may be allowed to directly connect two consecutive dynamic stages with the inverted signal (without the inverter of DOMINO), if it is ensured that

a) at least one series transistor exists, which is connected to a non-inverted (DOMINO) output (with position next to the dynamic node to prevent charge sharing) and
b) the non-inverted input signals arrive later than the inverted input signals.

In this case at least one transistor disjoins the output from the ground during the precharge phase and at the beginning of the evaluation. When the inverted outputs arrive first, they have sufficient time to switch off all n-transistors. The non-inverted (DOMINO) outputs arrive later and all other transistors are

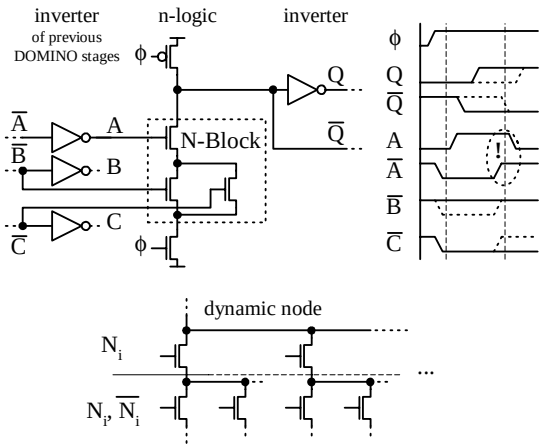

Figure 3. Structure of single-rail DOMINO and timing diagram (all negative inputs must arrive first; the timing diagram also shows the problem of false discharge during precharging the inputs) and essential structure of a single-rail DOMINO n-block; the AND-function with a non-inverted input prevents a prematurely discharge

already in their correct evaluation state and no erroneous discharging can appear. To force this function a dimensioning of the gates or a delaying of the non-inverted outputs is possible. Figure 3 shows this behavior. Input A is a non-inverted input and arrives with one inverter delay compared to input $\overline{B}$ and $\overline{C}$.

This results in a limited logic function of a gate. Figure 3 shows the resulting structure of the n-logic trees. No OR-gate is possible with at least one inverted input. However, in dynamic logic an OR-gate should be in any case implemented as a parallel path in the previous logic block, because only parallel transistors are necessary and these do not delay the evaluation. Further, AND-functions with only inverted inputs are not allowed, but this can be transformed to a NOR-function with non-inverted inputs (OR-function, where the inverted output is used).

It should be noted, that the logic style is not fixed to DOMINO logic. All dynamic logics are feasible including adapted differential styles, where the logic trees of two or more independent functions are merged.

3.2 Conditions for Precharge

As illustrated in figure 3 (dotted lines in timing diagram) an erroneous discharge can occur if the input signals are precharged during evaluation and the gate has not discharged the internal dynamic node. During precharge the inverted outputs settle to the high value first. Later the non-inverted (DOMINO) outputs settle to low value delayed by the inverter. Therefore, a path to ground can exist for a short time if all inverted nodes are already precharged but the non-inverted (DOMINO) output has not. If necessary, this problem can be solved through disconnecting the logic tree from ground, if the evaluation has finished.

3.3 Operation in Self-timed Circuits

For our self-timed structure we can solve the problem of false discharge during precharging the inputs in a simple way. After

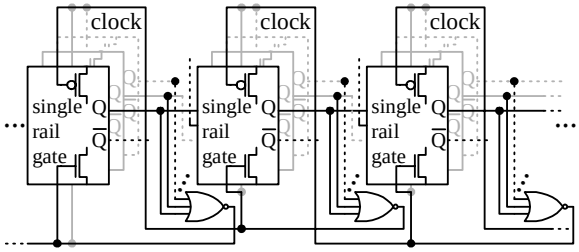

Figure 4. New modified self-timed scheme for single-rail dynamic logic; the self-timed signals were generated from the outputs of several gates

completion detection the previous gates are set to precharge. This signal can be used to disconnect the own logic-tree from ground. Then, both clock transistors in the dynamic gate are controlled separately. The n-transistor is clocked with the own completion signal, the p-transistor is controlled from the self-timed signal of the following stage. Figure 4 shows such a modified structure. Consequently, no valid inputs are allowed during precharge phase to prevent a short circuit. If this can not be guaranteed the separate use of the clock transistors is not possible and an additional n-transistor is necessary for disconnecting the logic-tree after evaluation. It should be noted, that such modifications depend on the timing behavior and they are not always necessary.

3.4 Integration in a Global Clock Scheme

An integration of the presented single-rail scheme in a synchronous single phase clock design targets a simpler synthesis, clock distribution and usage. In [7] we presented a scheme to integrate self-timed schemes in a true single phase clock scheme. The main ideas are to use short chains of asynchronous logic to keep runtime differences negligible and to clock the last gate of such a chain with the global clock.

During synchronization of the single-rail self-timed logic to the global clock it has to be ensured that the precharge conditions are not broken and that the information can be stored before the last gate. Here, an additional serial n-clock-transistor in the logic tree of the global clocked gate is necessary to ensure the function. One n-clock-transistor is controlled by the global clock, the second one is controlled with the own completion signal. Therefore, the second transistor prevents a false discharge, if the input values precharge during evaluation and the first transistor prevents a short circuit during precharge with valid inputs.

4. SINGLE-RAIL SELF-TIMED SD-ADDER

As an example implementation of the presented ideas a redundant signed digit (SD) adder is used. The redundant digits –1, 0, 1 are represented with two signals, e.g. two bit. To simplify the completion detection the digit representation shown in table 1 is used. This scheme ensures that at least one signal changes its value during evaluation. Therefore, the generalized completion detection is simple to implement. For testing purposes blocks of one bit adders were implemented. These adders generate the carry free redundant result in three stages of dynamic logic. Self-timed signals are generated for each stage, therefore, one adder contains a chain of three stages (figure 5) of DOMINO logic.

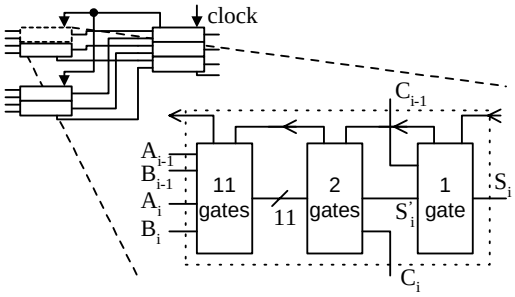

Figure 5. Structure of the self-timed SD-adder; a 1 bit cell consists of three dynamic single-rail stages

Then, we combined these two bit adders to a larger adder structure with consecutive blocks (a + b) + (c + d). This results in a self-timed chain of six stages. This structure was synchronized to a global clock with a modified AC-TSPC scheme presented in [7] (figure 5).

Table 1. SD-digit representation

Digit	Self-timed representation
-1	10
0	01
1	11
free	00 – precharge

The timing differences between parallel paths must be small. Therefore, short chains are meaningful and simpler to realize. An intensive timing calculation can predict the timing behavior. Then, critical paths can be adapted for reliable function.

5. RESULTS

The simulations were done for a 0.6 µm AMS process at 3.3V. Figure 6 shows our results in terms of power, area and speed in comparison to the same structure in complementary DOMINO logic.

An initialization period is crucial for simulating the self-timed structure. During switch-on an internal state can occur where more than one valid signals remain in the self-timed pipe. The starting point can be reached by clocking the pipeline without valid inputs. Then an 'empty' state can be achieved where the self-timed structure is reset.

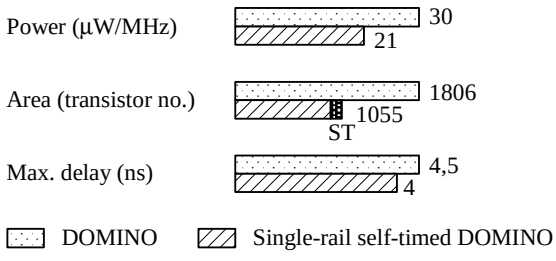

Figure 6. Comparison of the SD-adder structure in single-rail self-timed DOMINO logic and complementary DOMINO; ST marks the area for self-timed logic

The comparison shows that the area is reduced to 58%. The simple complementary DOMINO logic always needs two complementary signals. The self-timed structure requires additional logic for completion detection (11%). However, because of the higher clock load of the DOMINO structure (476 transistors) the self-timed structure saves also area in the clock tree (only 12 transistors).

The reduction in area results in a reduction in the power consumption. In complementary DOMINO logic one node is continuously discharged. The power consumption has been reduced to 70%. In self-timed logic we have additional power consumption due to the completion detection but lower power in the clock tree.

The speed of the DOMINO structure is slightly slower because of the higher loads of the dual-rail structure. Until now the simulated structures have not been optimized. Most transistors were sized to 7 for p-devices and 4 for n-devices.

6. CONCLUSIONS

This paper presents a realization of a dynamic single-rail self-timed logic integrated in a synchronous clock. Three main problems were solved: generating self-timed signals in single-rail logic, realizing inverting and non-inverting functions in dynamic single-rail logic and synchronizing short chains of self-timed gates. Additionally, the utilization of the redundancy of a SD-adder eased the self-timed generation in our example realization. The redundant number adder structure was simulated and shows the possibility of realizing single-rail self-timed circuits in dynamic logic. The self-timed structure ensures, that all signals meet the determined conditions and this results in a fast latch-free evaluation for a synchronous design. In comparison to a DOMINO realization the power and area consumption was reduced to about 70% and 60%, respectively.

7. REFERENCES

[1] R. H. Krambeck, C. M. Lee and H.-F. S. Law, "High-Speed Compact Circuits with CMOS", Journal of Solid-State Circuits, IEEE, Vol. SC-17, No. 3, Jun. 1982.

[2] D. Harris and M. A. Horowitz, "Skew-Tolerant Domino Circuits", Journal of Solid-State Circuits, IEEE, Vol. 32, No. 11, Nov. 1997.

[3] G. Yee, C. Sechen, "Clock-Delayed Domino for Dynamic Circuit Design", IEEE Transactions on VLSI Systems, Vol. 8, No. 4 , Aug. 2000.

[4] G. M. Jacobs, R. W. Brodersen, "A Fully Asynchronous Digital Signal Processor Using Self-timed Circuits", Journal of Solid-State Circuits, IEEE, Vol. 25, No. 6, Dec. 1990.

[5] T. E. Williams and M. A. Horowitz, "A Zero-Overhead Self-Timed 160-ns 54-b CMOS Divider", Journal of Solid-State Circuits, IEEE, Vol. 26, No. 11, Nov. 1991.

[6] L. G. Heller, W. R. Griffin, J. W. Davis and N. G. Thoma, "Cascode Voltage Switch Logic: A Differential CMOS Logic Family", Proceedings of International Solid-State Circuits Conference, IEEE, 1984, pp. 16-17.

[7] F. Grassert, D. Timmermann, "Dynamic Single Phase Logic with Self-timed Stages for Power Reduction in Pipeline Circuit Designs", International Symposium on Circuits and Systems (ISCAS), IEEE, May 2001, pp. IV 144-147.