

A Localized Self-Resetting Gate Design Methodology for Low Power

Woo Jin Kim, Yong-Bin Kim
Electrical and Computer Engineering Department
Northeastern University
wkim@ece.neu.edu, ybk@ece.neu.edu
Boston, MA 02115

Abstract—In this paper, a modification of the traditional dynamic self-reset circuitry is introduced for low power SRAM circuit design. The reset circuitry is localized, and the negative (trailing) trigger edge of the data is used to generate the self-reset signal to avoid the problem of crowbar current from V_{DD} to V_{SS} . It is demonstrated that fanouts of $6 < F_{inter} < 10$, and $4 < F_{out} < 7$ give the best delay-product values for a $0.5\mu m$ CMOS process.

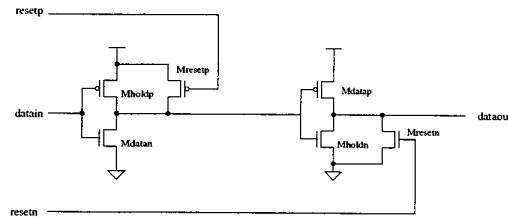


Fig. 1. Basic Block of Self-Reset Circuit

I. INTRODUCTION

Self-resetting logic is a commonly used piece of circuitry that can be found in use with memory arrays as word line drivers. It is a form of logic in which the signal being propagated is buffered and used as the precharge or reset signal. By using a buffered form of the input, the input loading is kept almost as low as in normal dynamic logic while the local generation of the reset assures that it is properly timed and only occurs when needed. The basic block of the self-reset circuit is shown in Figure 1 where the block is configured of two inverting stages. Each inverting stage consists of a data transistor (W_{datain} or $W_{dataout}$), a reset transistor ($W_{resetsp}$ or $W_{resetsn}$) and a hold transistor (W_{holdp} or W_{holdn}). When $datain$ goes high, the gates evaluate. The data devices propagate the leading edge of the data pulse, and the reset transistors return the nodes to their precharged state. Once the reset has been de-asserted, small static hold devices maintain the precharged node over the noise and leakage. In the traditional self-reset circuit[1]-[5], although there might be slight variations, the reset pulse is generated from the rising (leading) edge of the input data pulse to sequentially reset each block and its inverting stages within. The reset pulse has to be carefully controlled to assure that the reset occurs only after the data has completely passed through the corresponding stage. The reset signal for each inverting stage can be tapped from the output of an invert-

ing stage several delays down the chain if several of these blocks are cascaded, or it can be generated using a delay line external to the cascaded blocks. Thus, the delay of the reset generation chain really defines the pulse width of the data propagating through the self-reset blocks. A major design issue with this traditional scheme is the possibility of a crowbar current from V_{DD} to V_{SS} . It is necessary to design the delay between the data signal and the reset signal to avoid having both the data and the reset transistors on at the same time which would result in direct V_{DD} to V_{SS} path consuming a lot of power or, worse yet, causing a malfunction in the logic. More recently, delayed self-reset logic has been introduced[6],[7]. This scheme makes use of the trailing edge of the data input to generate the reset signal and thus is easier to avoid the problem of the crowbar current. The drawback of the scheme is that the width of the data pulse will be increased by at least the delay of an inverter and a nand gate due to the fact the trailing edge is utilized for the reset signal. When used as a global driver where several of these logic may have to be cascaded, this can become an issue as the signal will be cycle time limited.

This paper presents one way to get around this problem. Instead of using the delayed self-reset logic as a global driver, the reset circuitry can be localized and the self-reset logic used as a local word line driver. As the load will be smaller for each of the local self-reset blocks, depending on the configuration of the memory, one block will be enough

for most localized word lines. While this technique does increase the area of the basic self-reset block, it does not need the external inverter delay chains and is not subject to the RC effect due to the routing across the several stages of cascaded self-reset blocks.

II. MODIFIED SELF-RESET BLOCK

The modified self-reset circuit is very similar to the traditional self-reset circuit except that the reset circuitry is included. This technique localizes the reset generation for each self-reset block to that block only. The reset signal can only be triggered following the negative (trailing) edge of the input pulse, thus avoiding a potential crowbar current problem. As shown in Figure 2, upon receiving the rising (leading) edge of the data pulse, *reset_enable* is low and *delayed_out* is high so the nand gate keeps the reset transistors M_{resetp} and M_{resetn} off. However, once the forward going pulse starts to transition away, *reset_enable* starts to go high, causing the nand gate to activate the reset transistors M_{resetp} and M_{resetn} and restore the nodes of the two inverting stages to their precharged conditions. Once the precharging action occurs, the result is fed back to the nand gate, causing *delayed_out* to go low and deactivate the reset devices, leaving the hold devices M_{holdp} and M_{holdn} to sustain the node until the next incident data pulse. The inverters in the

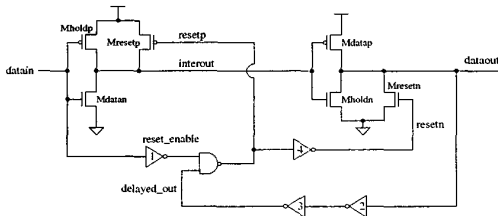


Fig. 2. Modified Self-Reset Block

reset connection are used to provide delay margins between the actions on the data nodes and their subsequent reactions on the reset nodes. The *reset_enable* inverter (inverter 1) and the nand gate ensures that M_{datan} is completely shutoff before M_{resetp} turns on, thus preventing crowbar current in that inverting stage. Also, the inverter between *resetp* and *resetn* (inverter 4) ensures that there is enough gate delay between the two signals so that M_{resetn} will not turn on before M_{datap} has finished turning off. The pulse width is determined by either the *delayed_out* delay loop ($datain \rightarrow interout \rightarrow$

$dataout \rightarrow delayed_out \rightarrow resetp \rightarrow resetn$) or the *reset_enable* path ($datain \rightarrow reset_enable \rightarrow resetp \rightarrow resetn$). While the path through the *reset_enable* offers the minimum widening of the pulse, for a more precise control of the pulse width the delay through *delayed_out* loop should be utilized. The simple timing of the signals for the minimum pulse widening are shown in Figure 3. The key concern is to size

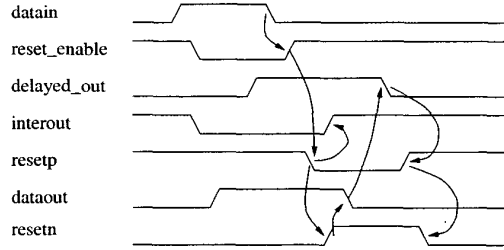


Fig. 3. Simplified Waveform of Modified Self-Reset Block

these gates to avoid an incomplete reset as the signal self terminates before it becomes fully active, but at the same time, minimize the extra loading on the *datain* and *dataout* nodes due to the local reset circuitry.

III. BLOCK DESIGN

A. Terminology

The modified self-reset circuit is simulated with HSPICE using a 0.5um CMOS technology to find the optimal sizing for the transistors. The following terminology are used throughout the simulation:

W_{inter} : the loading on the intermediate out node (*interout*) in terms of gate width

W_{datan} : the width of the data transistor M_{datan} in the first inverting stage

F_{inter} : fanout of the data transistor in the first inverting stage (W_{inter}/W_{datan})

W_{out} : the loading on the *dataout* node in terms of gate width

W_{datap} : the width of the data transistor M_{datap} in the second inverting stage

F_{out} : fanout of the data transistor in the second inverting stage (W_{out}/W_{datap})

F_{resetp} : fanout of the reset transistor M_{resetp} in the first inverting stage ($W_{inter}/(W_{resetp} + W_{holdp})$)

F_{resetn} : fanout of the reset transistor M_{resetn} in the second inverting stage ($W_{out}/(W_{resetn} + W_{holdn})$)

B. Sizing Rules

The following rules are used to size the transistors of the self-reset circuit. It is assumed that all the wire loading effects are included in the gate width (W) terms.

$$\begin{aligned} W_{datap} &= W_{out}/F_{out} \\ W_{resetn} &= W_{out}/F_{resetn} - W_{holdn} \\ W_{holdn} &= 0.05 * W_{datap} \\ W_{datan} &= W_{inter}/F_{inter} \\ W_{resetp} &= W_{inter}/F_{resetp} - W_{holdp} \\ W_{holdp} &= 0.10 * W_{datan} \end{aligned}$$

The hold transistors (W_{holdp} and W_{holdn}) need to be large enough to guarantee that the noise immunity is not reduced when the data transistors (W_{datan} and W_{datap}) are leaky. The numbers used here for the hold transistors are the values that gave reasonable results in the simulations.

$$\begin{aligned} W_{inv4n} &= W_{datap}/F_{inter} \\ W_{inv4p} &= 2 * W_{inv4n} \\ W_{nandn} &= (W_{datan} + W_{inv4n} + W_{inv4p})/F_{inter} \\ W_{nandp} &= W_{nandn} \\ W_{inv3n} &= (W_{datan} + W_{inv4n} + W_{inv4p})/F_{inter} = \\ W_{inv2n} \\ W_{inv3p} &= 2 * W_{inv3n} = W_{inv2p} \\ W_{inv1n} &= (W_{inv2p} + W_{inv2n})/F_{inter} \\ W_{inv1p} &= 2 * W_{inv1n} \end{aligned}$$

The reset path needs to be sized so that the extra loading effects due to the localized reset circuitry can be minimized, but it also needs to satisfy the requirements that the reset pulse has adequate width and that there is adequate delay between the adjacent reset pulses to avoid the crowbar current in the second inverting stage.

IV. SIMULATION RESULTS

The circuit was simulated for different values of W_{out} . From Figure 4 and Table 1, it can be seen that the best delay numbers are given for ranges of $6 < F_{inter} < 10$ and $4 < F_{out} < 7$. This data is for when $W_{out} = 400\mu m$, but is true for most of the cases. The value of the block delay, BD, is given in nano-second units.

The PDP (power-delay product:nano-second-milli-watt) decreases as the fanout number increases and tends to flatten out as shown in Figure 5. For the simulation values above, F_{inter} of 9 was used as the fanout of choice and applied to the sizing of the reset chain. Beta ratio of two was used for all the static gates in the chain. Figure 6 shows the actual simulated values of the signals given in Figure 3. The following values were used for the simulation.

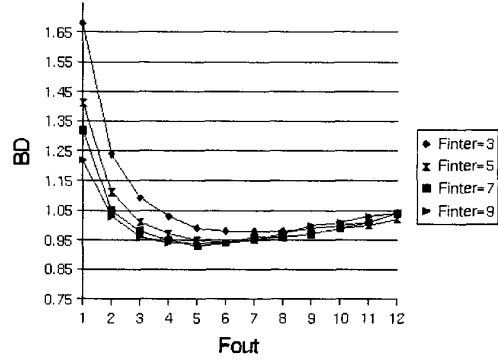


Fig. 4. BD Plots for Different Fanouts for $W_{out} = 400\mu m$

TABLE I
BD for $W_{out} = 400\mu m$ when $F_{out}=5$

F_{inter}	F_{out}	F_{resetp}	F_{resetn}	BD
2	5	1.3	10	1.06 ns
3	5	2.0	10	0.99 ns
4	5	2.6	10	0.97 ns
5	5	3.3	10	0.95 ns
6	5	3.9	10	0.93 ns
7	5	4.5	10	0.93 ns
8	5	5.3	10	0.94 ns
9	5	5.9	10	0.94 ns
10	5	6.6	10	0.95 ns

V. NOISE SENSITIVITY

Noise problem can be reduced to basically three sources. The first is the capacitive noise coupling on *datain* due to an adjacent signal line. The second is the DC ground bounce in *datain*, and the third is the leakage current to V_{SS} on *interout* node. These will combine to decrease the noise immunity of the circuit. While the ground bounce and the leakage current are overall layout and process dependent where the individual designer has no control, it is possible to minimize the effect of noise coupling. For capacitive coupling to driven logic node, the coupled noise voltage can be expressed as

$$V_n(s) = V_s(s)R_a s$$

where $V_n(s)$ is the coupled noise voltage, $V_s(s)$ is the effective noise signal, R_a is the equivalent resistance of the active output transistor of the driving gate, C_c is the coupling capacitance, and $s = j\omega$. Depending on the particular design constraints, each individual designer must determine how this translates into the actual design in terms of the transistor sizing and the routing.

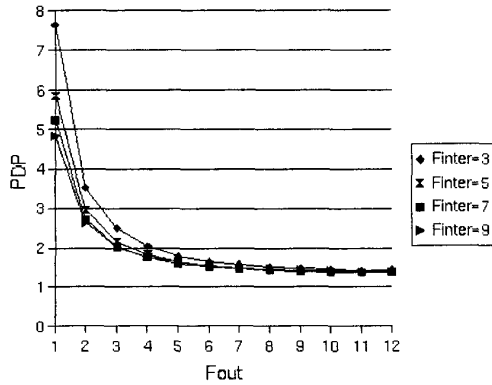


Fig. 5. PDP Plots for Different Fanouts for $W_{out} = 400\mu m$

TABLE II
Minimum PDP for $W_{out} = 400\mu m$ when $F_{out}=5$

F_{inter}	F_{out}	F_{resetp}	F_{resetn}	PDP
2	5	1.3	10	2.03
3	5	2.0	10	1.79
4	5	2.6	10	1.71
5	5	3.3	10	1.65
6	5	3.9	10	1.60
7	5	4.5	10	1.61
8	5	5.3	10	1.62
9	5	5.9	10	1.65
10	5	6.6	10	1.66

VI. CONCLUSION

Using the modified self-reset circuit, it is easier to avoid the occurrence of crowbar current that is inherent in these types of circuits. For the 0.5 μm CMOS process used, the best delay values were achieved for fanouts of $6 < F_{inter} < 10$ and $4 < F_{out} < 7$. These values are subject to the device parameters. The major design issue is to ensure that the delay of inverter 4 is large enough so that M_{datap} will turn off before M_{resetn} turns on. To reduce the penalty of added loading on the *datain* and the *dataout* nodes, minimum sized devices should be used whenever possible. Owing to the nature of the circuit where the trailing edge of the data input is

TABLE III
Transistor Sizes Used in the Simulation

transistor	size[μm]	transistor	size[μm]
W_{datan}	7.4	W_{datap}	67.0
W_{resetp}	11.9	W_{resetn}	33.5
W_{holdp}	0.7	W_{holdn}	3.3
W_{nandp}	8.0	W_{inv2p}	3.6
W_{nandn}	8.0	W_{inv2n}	1.8
W_{inv1p}	1.2	W_{inv3p}	12.0
W_{inv1n}	0.6	W_{inv3n}	6.0

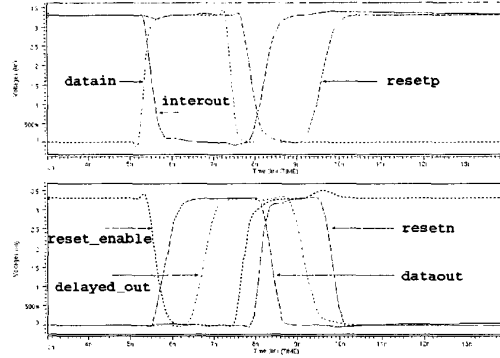


Fig. 6. Simulated Waveforms for $W_{out} = 400\mu m$

used for the reset signal generation, the data pulse width is extended by the delay of the inverter and the nand gate in the *reset_enable* path. This minimum delay was about 250ps for the 0.5 μm CMOS device parameters. This indicates that the modified version of the circuit can become cycle time limited if there are many levels of cascading. Noise sensitivity issue is not discussed in this paper, but it should be taken into account when sizing the transistors and laying out the circuit. Insertion of a latch transistor between the two inverting stages of the block will also improve the noise immunity.

REFERENCES

- [1] T. I. Chappell, et. al., "A 2-ns Cycle, 3.8ns Access 512-kb CMOS ECL SRAM with a Fully Pipelined Architecture", *IEEE Journal of Solid-State Circuits*, vol.26, no.11, pp.1577-1585, Nov. 1991
- [2] R. A. Heald, et. al., "A 6-ns Cycle 256-kb Cache Memory and Memory Management Unit", *IEEE Journal of Solid-State Circuits*, vol.28, no.11, pp.1078-1083, Nov. 1993
- [3] D. Wendell, et. al., "A 3.5ns, 2Kx9 Self Timed SRAM", *1990 IEEE Symposium on VLSI Circuits*, pp.49-50
- [4] H.C. Park, et. al., "A 833Mb/s 2.5V 4Mb double data rate SRAM", *1998 IEEE International Solid State Circuits Conference*, pp.356-357
- [5] W. Hwang, et. al., "Implementation of a Self-Resetting CMOS 64-bit Parallel Adder with Enhanced Testability", *IEEE Journal of Solid-State Circuits*, vol.34, no.8, pp.1108-1117, Aug. 1999
- [6] K.W. Mai, et. al., "Low-Power SRAM Design Using Half-Swing Pulse-Mode Techniques", *IEEE Journal of Solid-State Circuits*, vol.33, no.11, pp.1659-1671, Nov. 1998
- [7] B.S. Amrutur, et. al., "A Replica Technique for Wordline and Sense Control in Low-Power SRAM's", *IEEE Journal of Solid-State Circuits*, vol.33, no.8, pp.1208-1219, Aug. 1998