A Low-Power SRAM Design Using Quiet-Bitline Architecture

Shin-Pao Cheng **Shi-Yu Huang**
Electrical Engineering Department
National Tsing-Hua University, Taiwan

Abstract

This paper presents a low-power SRAM design with quiet-bitline architecture by incorporating two major techniques. Firstly, we use a *one-side driving scheme* for the *write* operation to prevent the excessive full-swing charging on the bitlines. Secondly, we use a *precharge-free pulling scheme* for the read operation so as to keep all bitlines at low voltages at all times. SPICE simulation on a 2K-bit SRAM macro shows that such architecture can lead to a significant 84.4% power reduction over a self-designed baseline low-power SRAM macro.

1. Introduction

Static Random Access Memory (SRAM) has found its way into almost every IC as an embedded component. Traditionally, an SRAM macro is mainly formed by an array of cells consisting of four or six transistors and a number of periphery circuits such as row decoder, column decoder, sense amplifier, write buffer, etc. Information access from/to this macro consume power in both dynamic and static ways. The dynamic power involving the switching of signals is consumed in operations such as wordline decoding, bitline charging/discharging, sense amplification, etc. The static power is consumed when there is a direct path from V_{DD} to ground during memory access.

The power consumption of the address decoders and the sense amplifiers can usually be tamed quite satisfactorily. In a low-power SRAM design, the question is often how to minimize the excessive power due to the following sources: (1) wordline switching power, (2) bitline switching power, and (3) static power [4].

Divided wordline or divided bitline architecture [2][3][7] partitions the entire array into smaller pieces. The access of the memory is thereby confined to a smaller sub-array and the total capacitance that needs to be switched during a memory operation is thus reduced to save power. Nevertheless, divided architecture also requires significant area overhead for the extra decoding and controlling circuitry. Wordline pulse control [5] is an important scheme aimed at reducing the excessive static current flowing through the cells that are open for access. To achieve this goal, the wordline is only activated (i.e., raised to high voltage) for a minimal period of time. Also, current-mode access schemes [6] have reportedly achieved both high speed and low power consumption because of its instant read operation and low bitline swing nature. However, current-mode SRAM tends to be less reliable as compared to the conventional voltage-mode counterparts because of its higher sensitivity to the noise.

In this paper, we explore low-cost low-power architecture using *quiet bitlines*. By *quiet*, we mean that the voltages of the bitlines stay as low as possible at all times. The immediate reward is that all charging/discharging power associated with the bitlines can be eliminated dramatically. To achieve this goal, both the *write* and *read* operations need to be modified. For the *write* operation, a *one-side driving scheme* is devised in such a way that only a strong '0' signal is forced into the cell being accessed from one side, while leaving the other side floating. For the *read* operation, we use a *pulling scheme* that operates in four stages: bitline equalization, wordline activation, and bitline pulling, and finally sense amplification. Unlike conventional ways, we rely on low-power wide-input-range sense amplifier [1] to abandon the precharge operation to further minimize the energy loss on the bitlines.

SPICE simulation on a 2048-bit SRAM macro indicates that this design has the potential to reduce the power dissipation down to only 2.7 mW, a significant reduction as compared to the 17.3 mW of another conventional in-house macro of the same size, or the 45.9 mW of a compiler generated macro.

2. Preliminaries

In this section, we review the basic structure and operations of a conventional 6-transistor SRAM.

2.1 Basic Column Structure

The structure of one column – including a representative cell, bitline conditioning circuitry, and write buffer – is shown in Fig. 1.

Inside the cell, there are two inverters that form a cross-coupled latch for storing one-bit complementary information as *Q* and *Q-bar*. The two nMOS transistors of the inverter pair are called *n-drivers*, while the two pMOS transistors *p-drivers*. Also, it takes two nMOS transistors, called *access transistors*, to guard the passage of this latch to the bitline pair, *bitline* and *bitline-bar*.

In the bitline conditioning circuitry, it is common that there are two *precharge transistors* and an *equalizer transistor*, denoted as *EQ*. The precharge transistors connect the bitline and bitline-bar to V_{DD} at all times.

Proceedings of the 2005 IEEE International Workshop on Memory Technology, Design, and Testing (MTDT'05)
1087-4852/05 $20.00 © 2005 **IEEE**

While the memory macro is idle, these two transistors will set the bitline pair to their default voltage levels. Regarding the equalizer, it is mainly used to balance the voltage levels at bitline and bitline-bar and resolve any mismatched initial conditions before the read operation.

In the write buffer, a simple logic circuit takes as input the data bit and decides which strong *write drivers* (i.e., transistors with large sizes) should be turned on to force a complementary signal on the bitline pair. Here, the *write drivers* are multiple times stronger than the precharging transistors and thus will dominate the final voltage of the bitline if fighting exists.

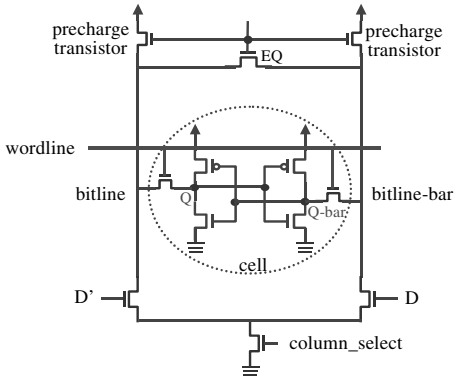

Fig. 1: Traditional 6-T SRAM column.

2.2 Basic Read Operation

When a read operation is issued, the memory macro will go through the following steps:

- (1) Row decoding: The row address is decoded to activate a selected wordline. For low-power purpose, two-stage NAND type decoder is often desired at moderate sacrifice of speed.
- (2) Bitline driving: After the wordline has been turned on, the target cell connects to the bitline and bitline-bar. The so-called *cell current* through an *n-driver* of the target cell (as indicated in Fig. 2) will discharge the voltage of either bitline or bitline-bar progressively. The process is normally complete once 100~200 mV difference has been established between the bitline and bitline-bar. During this bitline driving step, a direct current is formed from V_{DD} through a precharge transistor and the *n-driver* to the ground. Aggressive wordline pulse control will de-assert the wordline at the end of this step to suppress this static current.
- (3) Sensing: The sense amplifier is turned on to amplify the small difference voltage at the bitline pair into full-swing logic signals. There are often two stages of sense amplifiers performed in cascade to accelerate the process.
- (4) Precharging: At the end of the read operation, all bitline pairs will return to their normal default conditions through the precharge transistors and the equalizer and get ready for the next operation.

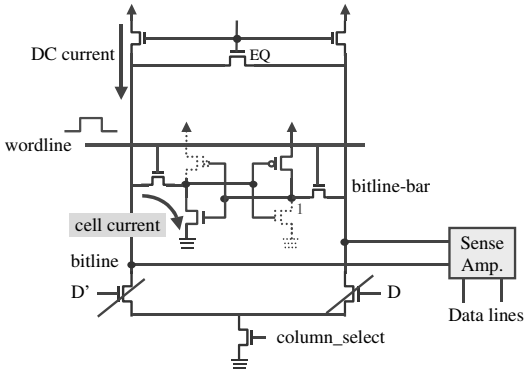

Fig. 2: Bitline discharging for the read operation.

2.3 Basic Write Operation

When a write operation is issued, the memory macro will go through the following steps:

- (1.1) Row decoding: Similar to the read operation, the row address decoding starts immediately to activate the selective wordline.
- (1.2) Bitline driving: For a write operation, this bitline driving conducts simultaneously with the row address decoding by turning on proper write buffer. After this step, the bitline pair will be forced into full-swing logic level.
- (2) Cell flipping: If the value of the stored bit in the target cell is the opposite to the value being written, then the cell flipping process will take place. Normally, the transistors in a cell are sized in a way that the cell only reacts to the '0' signal side of the bitline pair. For example, let us assume that the originally stored information in the target cell is {Q='1', Q-bar='0'}. After the access transistors have turned on, the strong '0' signal on the bitline will overwhelm the stored Q bit '1' and make it become '0'. Once this over-writing is successful on one side, internal latch will take over to force the cell flipping and finally make Q-bar to change from '0' to '1' as well.
- (3) Precharging: At the end of the write operation, all bitline pairs return to their normal default conditions through the precharge transistors and the equalizer and get ready for the next operation just like the read operation.

3. Quiet-Bitline Architecture

In this section, we will first analyze the power dissipation components and then propose our quiet-

bitline architecture.

3.1 Overall Strategy

The power of an SRAM macro is normally consumed in the row and column decoding, wordline driving, bitline charging, sense amplification, output driving, and the static current during the read or write operations.

It is known that the power dissipated on the address decoding and wordline driving can be controlled to a small amount if NAND-type decoders are used. Also, the power dissipation of the sense amplifiers can be tamed by using dynamic latch-based circuits [1]. Therefore, the dynamic power consumed in an SRAM macro is mainly on the bitline charging. This component, scaling with the number of bitlines, is the major target in our power optimization process.

For a read operation, the bitline charging power (consumed in the precharging step) can be reduced quite significantly if the aggressive wordline pulse control is used to limit the bitline swing to around 100~200 mV. However, all bitlines are usually forced to have full swings during a write operation, mainly to enable a quick cell flipping.

Our strategy for alleviating the excessive power on bitlines consists of two major tactics. First of all, we investigate a *one-side driving* scheme in which only strong '0' signal is forced into a bitline or bitline-bar for the write operation, while leaving the other side floating. Secondly, we *remove the precharging* for both read and write operations. The combination of these two tactics leads to a design with quiet bitlines. This strategy may cause speed penalty to some degree, however, the significant power reduction thereby achieved could more than compensate the side effects.

3.2 Proposed Read Operation

The proposed read operation now differs from the previously mentioned read operation in two aspects. (1) There is no bitline precharging any more. Consequently, the initial voltages of the bitline and bitline-bar before the read operation could be more diverse. Typically, the bitline pair is first equalized to a low voltage ranging from 0V to a few hundred mV as the starting condition. (2) Without the precharge transistors, the cells are the only drivers to the bitline pair after the wordline is activated. Normally, we rely on the *p-driver* to pull up the voltage of one side of the bitline pair. The voltage of the other side could remain unchanged when the initial voltage is low, or could become lower due to the discharging by the *n-driver*. In either way, there is one major advantage – there is no static power any more. In order to support such a precharge-free read operation, the sense amplifier that takes the difference voltage of the bitline and bitline-bar should be able to operate over a wide of common-mode input signal range, e.g., from 0V to 500mV.

3.3 Proposed Write Operation

The proposed write operation is similar to the basic write operation in terms of the bitline driving. The write buffer is still strong enough to force a value '0' into one side of the target cell. However, the cell flipping will proceed in a slightly different way. To be more specific, the cell flipping can be divided into two sub-steps:

- (Weak pulling): The storage node connected to the floating bitline or bitline-bar only rise slowly when the wordline is still activated. This is mainly because of the heavy loading of the bitline or bitline-bar and the relatively weak pulling strength of the *p-driver.*
- (Quick latching): Right after the wordline is de-activated, the target cell detaches the bitline pair and behaves like a static latch. Although the difference of Q and Q-bar at the beginning is relatively small, it will quick amplify them into complementary logic signal and thus complete a write operation.

In the conventional design in which the bitlines are precharged to high by default, the second sub-step, quick latching, is insignificant, because the Q and Q-bar may have reached their stable values even when the wordline is still activated.

Example 1: (New write operation) Fig. 3 shows our column architecture and simulation waveforms of signals {D, bitline, Q}. We assume that the data to be written to the target cell is D = 1, forcing a strong '0' signal into Q-bar through bitline-bar quickly, whiling making bitline floating. Once the stored signal Q-bar switches from '1' to '0', its complementary signal Q will rise only slowly as in the weak pulling stage. In the waveform of signal Q, this stage lasts from node A to node B. As the signal of D goes low, the signal Q rise quickly as in the quick latching stage, as indicated from node B to node C.

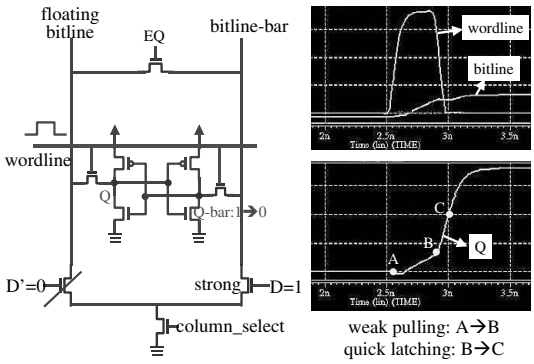

Fig. 3: A write example.

On the average, we observed that such a one-side '0'-driving scheme reduces the bitline voltage swing to only 250mV – a 14% value of the power supply voltage 1.8V.

It implies that we can save up to 86% power for just avoiding the full-swing signal '1'-side driving. Moreover, the power savings are beyond that. The extra direct current flowing through the precharge transistor, access transistor, and then to the *n-driver* has also been eliminated.

3.4 Worst-Case Scenarios

In our architecture, the read or write operations are both conducted with different initial conditions on the bitlines. Derivation of the worst-case scenarios is thus important in order to ensure the circuit's functionality.

To gauge the worst-case scenario, we investigate how high the voltages of bitlines could become under a sequence of 15 consecutive write '1' operations. The voltage waveforms of the bitline and bitline-bar are shown in Fig. 4. It can be seen that each write operation pumps certain charge into the floating bitline by the *p-driver* and cause its voltage to rise a few hundred mV. However, as the consecutive write operations proceed, the charge pumping strength gets weaker and weaker and finally the bitline voltage saturates at around 900 mV. This waveform indicates that the largest voltage a bitline or bitline-bar can get is around 900 mV. Under such a hostile situation, we apply a read and a write operation respectively to make sure that it can still function correctly. In the upper figure, we apply a read '0' operation from another cell in the same column. In the lower figure, we apply a write '0' operation. Both cases show correct response at the bitline pairs.

On the other hand, the read operation does not unilaterally increase charge on the floating bitline pair. A cell essentially pumps charges on one side of the bitline pair while discharges on the other. As a consequence, a sequence of consecutive read operation does not create hostile situation on the bitlines.

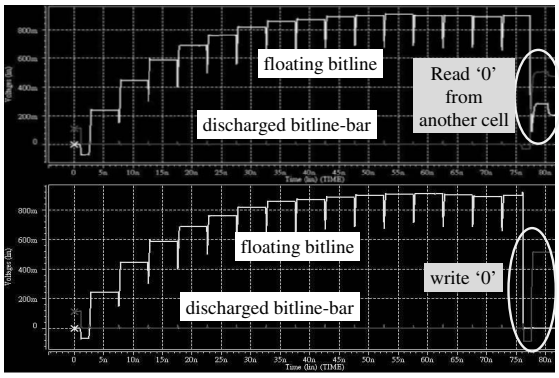

Fig. 4: Effects of 15 consecutive write operations on bitlines.

4. Experimental Results

To evaluate the effectiveness, we have implemented the proposed architecture as an SRAM macro with 2K bits. This macro is required in our in-house Viterbi decoder as the so-called survivor path memory. For comparison, we also implemented a conventional SRAM macro with the same configuration using the common low-power architecture with two-stage NAND-decoder, the wordline pulse control and low-power sensor amplifiers.

Table 1 lists the power dissipation and operating speed of these two versions and a macro generated by a commercial memory compiler. The following is the meaning of each column:

- I-read: The average V_{DD} current for performing a read operation.
- I-write: The average V_{DD} current for performing a write operation.
- Average Current: calculated by (I-read + I-write) / 2.
- Access Time: The time required to complete a read or write operation.

It can be seen that the new quiet-bitline architecture consumes only 1.5mA supply current on the average, i.e., 15.6% of the baseline architecture, or only 5.9% of the one produced by a memory compiler. Our Viterbi decoder is designed to operate at 200MHz. The access time of the new SRAM macro, 1.8 ns, is well below the clock cycle time requirement 5ns.

Table 1: Comparisons of average power consumption.

Type	I-read (mA)	I-write (mA)	Average Current (mA)	Access Time (ns)
Compiler	26.1	24.9	25.5 (266%)	1.05 ns
Conventional	6.1	13	9.6 (100%)	1.8 ns
Quiet-Bitline	1.3	1.6	1.5 (15.6%)	1.8 ns

5. Conclusion

We discovered in this work that the power dissipation of an SRAM macro could be slashed dramatically if certain performance loss can be accommodated. In many designs, the SRAM macros are not the performance bottlenecks and therefore such a strategy does not actually degrade a system's performance at all. Our low-power strategy aims to keep bitlines at low voltages at all times. We achieved this goal by two schemes – the precharge-free pulling scheme for the read operation and the one-side driving scheme for the write operation. This strategy has been preliminarily verified through a 2048-bit macro configured as 32 rows by 64 columns. SPICE simulation indicates that 84.4% power reduction out of a baseline design is possible.

Reference

[1] L. Benini, G. De Micheli, and E. Macii, “Designing Low-Power Circuits: Practical Recipes,” *IEEE Circuits and Systems Magazine*, Vol. 1, No. 1, pp. 6-25, 2001.

[2] J. S. Caravella, "A 0.9V, 4K SRAM for Embedded Applications," in *Proc. of CICC*, pp.119-122, May 1996.

[3] K. Itoh et. al., "Trends in Low-Power RAM Circuit Technologies", *Proc. of the IEEE*, pp.524-543, April 1995.

[4] M. Margala, “Low-Power SRAM Circuit Design,” *Proc. of IEEE Int’l Workshop on Memory Technology Design and Testing*, pp. 115-122, August 1999.

[5] H. Morimura and N. Shibata, “A Step-Down Boosted Wordline Scheme for 1-V Battery Operated Fast SRAMs,” *IEEE Journal of Solid-State Circuits*, Vol. 33, No. 8, pp. 1220-1227, August 1998.

[6] J.-S. Wang, W. Tseng, and H.-Y. Li, “Low-Power Embedded SRAM with the Current-Mode Write Technique,” *IEEE Journal of Solid-State Circuits*, Vol. 35, No. 1, pp. 119-124, Jan. 2000.

[7] M. Ukita et. al., "A Single Bitline Cross-Point Cell Activation (SCPA) Architecture for Ultra Low Power SRAMs," in *ISSCC Digest of Technical Papers*, pp.252-253, February 1994.