Architecture and Design of a High Performance SRAM for SOC Design

Shobha Singh, Shamsi Azmi, Nutan Agrawal, Penaka Phani and Ansuman Rout
Central R&D, STMicroelectronics, Noida 201301 INDIA
[Shobha.singh, shamsi.azmi, nutan.agrawal, penaka.phani, ansuman.rout]@st.com

Abstract

Critical issues in designing a high speed, low power static RAM in deep submicron technologies are described along with the design techniques used to overcome them. With appropriate circuit partitioning, transistor sizing, choice of a suitable Sense Amplifier, a good resetting technique and judicial use of dual Vth transistors we have achieved a high speed memory without dissipating too much power.

The Introduction gives the specifications of the memory that was our design target. In Section II, we describe the key techniques. Finally, we present the implementation on a testchip, and silicon measured results, which (we believe) is the best in class of embedded SRAM compliers available from various vendors in the world at the time of writing this paper. Also this architecture has achieved yields well over 95% in 0.18u technology.

1. Introduction

In this new era of System On Chip(SoC), there is emphasis on providing all the components on a single chip including the memories. Normally, memories occupy approximately 70% of the total chip area and hence they are always a critical factor in determining the performance and yield of a chip. Since memories are very densely designed, seeing their criticality for any application, there has always been demand for high speed, power economic embeddable memories which are stable on silicon and give very good yield which is the key factor to maintain profitability and have a competitive edge in the market. Since different designs have different size requirements of memory blocks, catering to each specific demand separately is not a time effective solution. Therefore we need a solution through which we can generate different sizes of memories quickly. A memory block is an arrangement of memory cells in rows and columns, with the Input/Output blocks and decoders block along the periphery of the memory cell matrix. Here the idea is to divide the memory block in basic leaf cells and then have a compiler to assemble them in a memory specific to the customer requirements. The variable parameters are normally words, bits, mux, drive etc. So it is a one time effort of deciding the architecture, designing and verifying the memory compiler and the advantage is that we can reduce the time to market and overall cost of design.

In this paper, we are concentrating on the architecture and design aspects of a memory compiler. We have presented here some of the key features of designing a high speed memory. A read cycle can be broadly classified into 3 categories,

i) Clock to wordline selection,
ii) Bitline discharging and data sensing,
iii) Resetting the intermediate signals for next cycle.

In order to achieve a faster word line selection, we have used core splitting and dynamic decoding. The advantage of core splitting is that without an area overhead we gain in speed, whereas the key advantage of using a dynamic decoder is that address setup time is quite less as compared to the static decoding scheme. Bitline discharge rate is determined by the memory cells and we need to have a good balancing ratio between discharge rate, write margin and noise margin for it's functionality keeping in mind the density aspect. So we need to have a fast sensing scheme to overcome the limitations of the memory cells and in here we have used a bitline decoupled latch type sense amplifier for data sensing, which is able to detect a very low swing in bitline and has a very high gain factor. A unique selftiming scheme is employed to track the bitline delay and reset intermediate signals for the next cycle. Also dual Vth technique is used to gain further in terms of speed with memory cells in high Vth and the periphery in low Vth transistors.

2. Design concepts

2.1. Circuit partitioning

A conventional memory has a single core with input/output block, control and the decoding circuit build around it. This arrangement works well for a small memory, but as the memory size increases, the load

(capacitive and the resistive) on the wordlines and bitlines also increases which in turn reduces the speed of the memory. If by any means we can reduce the resistive and the capacitive load, we can control the delay. There are various kinds of implementations worked upon till now to reduce the bitline and wordline loads. The most conventional one is the memory with a single core. But that is good for small memories only and the timings deteriorate for with increase in memory size. Another popular approach is to divide the core into number of pages and banks, this approach is good for both power and speed as the load on bitlines and wordlines is reduced, but it is not an area efficient solution as we need to have local wordlines and decoders for each page which increases the area. A very simple approach is to just split the core in two parts (see fig. 1) and use the same wordline driver for it (we can have different wordline drivers also if we can afford it). This way there will be two paths for the driver and each of the paths will see half of the capacitive and resistive loads as compared to the case in which there is only a single core.

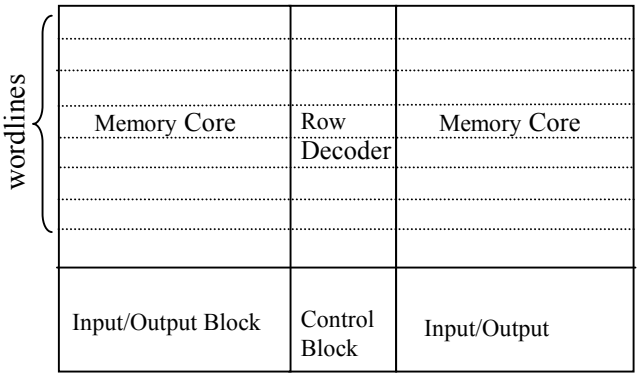

Figure 1. Block diagram of a memory with a divided core and having the same wordline driver

We can represent the load on a word line as shown in fig. 2. This pi network representation is quite close to the actual case [5] and gives us a very close approximation of signal propagation at node B in fig. 2(a). Fig. 2(b) represents the case of the split core. In case of a single core, the rising time will be governed by the time constant T=RC for fig. 2(a) and T=RC/4 for fig. 2(b) at nodes B1 and B2. The rising time for this RC network at point B can be given by the equation:

$$V(t) = V(1-e^{-t/T})$$

It is obvious from the above equation and the value of the time constant T for both the case that though the points B1/B2 and B start rising almost at the same time (there will be a small time difference owing to the reduced capacitance at node A1 and A2, but we can neglect it) there is a time gain in both the nodes reaching the required voltage level of 1 as is shown in fig. 3. So in this case without losing anything on area, we can achieve a faster signal transmission.

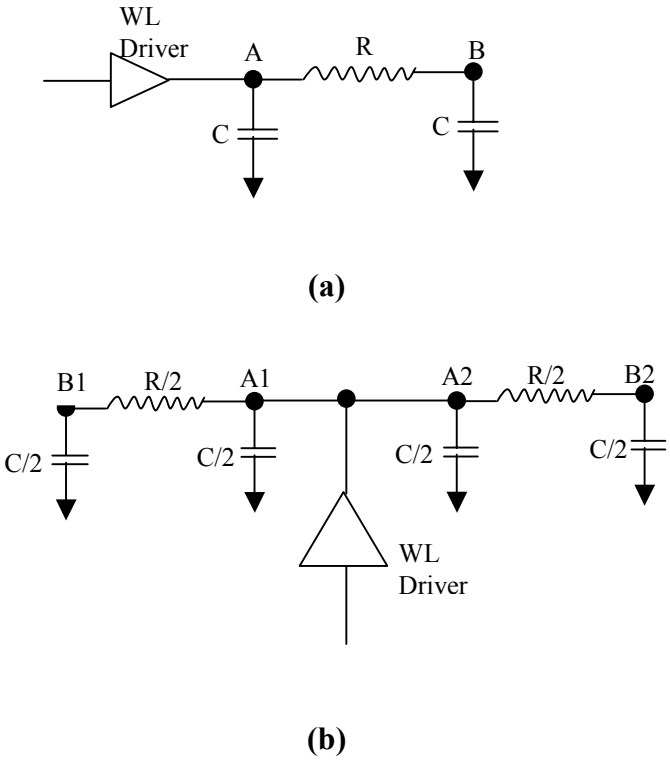

Figure 2. (a) pi network representation of a row of memory block with a single core. (b) pi network representation of a row of memory with split core and a single driver.

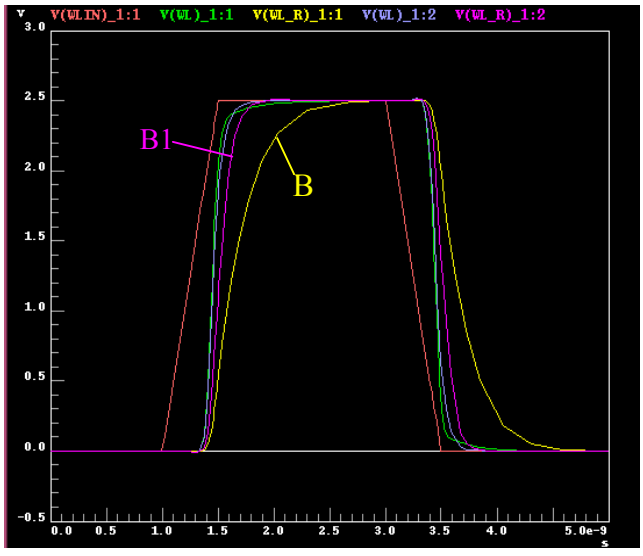

Figure 3. Eldo Simulation results for the two representations

2.2. Fast sensing and selftiming techniques

A considerable portion of the delay and the power is taken up in sensing of the data. Since the memory cells have very small transistors, the rate of discharge of the bitlines is very low and for fast data sensing we need a sense amplifier which can detect a very low swing in bitlines. This has twofold advantages, one , we will have fast access time, second, since the bitlines will be discharged by a small amount, power is saved in charging and discharging the bitlines.

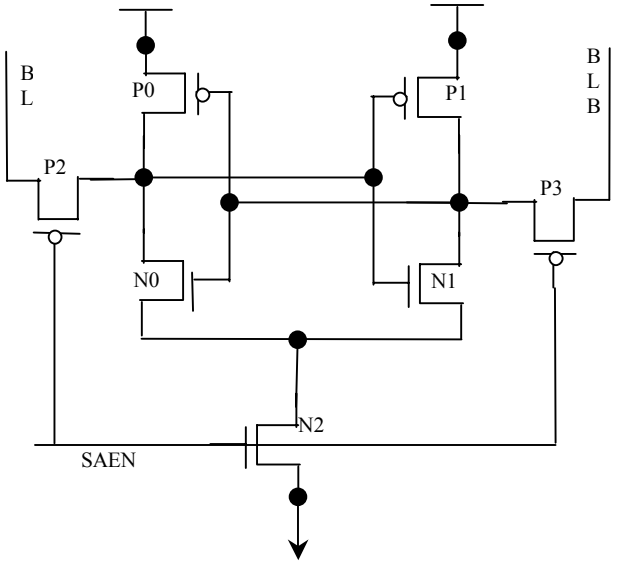

Figure 4. A bitline decoupled latch type sense amplifier

A latch type sense amplifier gives a good result in terms of speed and power [2], [4] because it is able to detect a very low bitline difference and gives a very high gain factor. But the point to be taken care in case of a latch type sense amplifier is when to switch on the sense. For this a selftiming technique is required to switch on the sense when a required bitline difference is obtained at the sense input. This selftiming technique should be such that it is able to maintain the required bitline difference in all the process, voltage and temperature conditions. In figure 4., schematic diagram of a bitline decoupled latch type amplifier is shown. As is shown in the figure, the bitlines are connected to the latch through two PMOS transistors and the signal SAEN is used for switching on the sense amplifier. SAEN is also used for activating the two pass transistors P2 and P3, hence at the time of sense activation, the bitlines are also decoupled and precharging of the bitlines can start at the same moment.

The selftiming strategy employed in this memory generator is as shown in Fig 5. It has a considerable advantage over the conventional delay chain method of implementing the selftiming and is also better than the replica technique discussed in [2]. The disadvantage in [2] is that although the replica cell is able to match the bitline delay, the SENSE ON signal and the wordline signals will have a load mismatch because of which each sense amplifier will see different bitline difference at the time of SENSE ON. If we use the dummy bitline signal after a series of inverter delays to activate the sense, the capacitive load on that signal will be different than the capacitive load on the wordline since the pass transistor sizes of a memcell will be different than the transistor sizes of the sense amplifier. So the sense activation signal will see a different load and the wordline will see a different load and even if the driver sizes are the same, there will be a difference in the signal propagation delay owing to the different loads on them. If we are able to match the load on this sense activation signal originating through the dummy bitline with the WL signal and also there driver sizes, we can ensure that all the sense amplifiers see the same kind of bitline difference at there inputs irrespective of the number of columns.

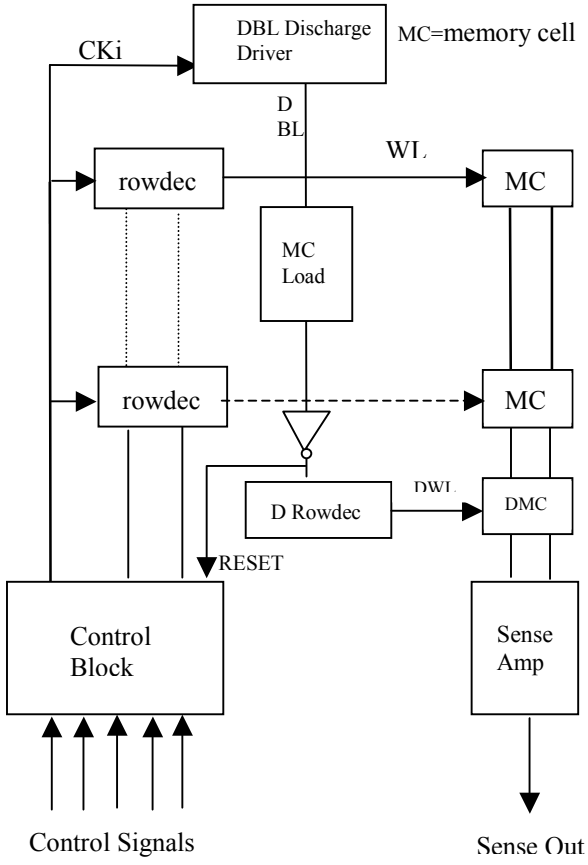

Figure 5: Block diagram of the selftiming strategy

In this implementation, we have used the clock signal which is enabling the row decoders for wordline selection to activate the dummy bitline discharge driver. The block diagram for this implementation is as shown in fig. 5. The dummy bitline should be matched with the actual bitline. In this case we need to size the dummy bitline discharge driver such that it matches the discharge rate of the actual bitline. As is shown in fig. 5, we can split the delay from Cki to SENSE ON in two paths:

Total delay = td1 + td2

Where,

td1 = delay in dummy bitline discharge

td2 = DBL_ to sense out delay.

td1 is a variable delay which will depend on the number of rows, whereas td2 is a fixed delay independent of the memory size and dependent on the process, voltage and temperature conditions. The total delay td has to be matched with the delay in producing the desired bitline difference at the sense amplifier. Below the memory core, we have used another row with a dummy memcell which matches the actual wordline. This is to ensure matching between both actual wordline WL and the dummy wordline DWL. This is done to ensure that all the sense amplifiers will be will be activated at the same bitline difference at their inputs. The RESET signal is also generated from the dummy bitline itself.

2.3. Dual Vth usage

In order to gain in speed, reducing the threshold voltage is a very effective technique [1], [3]. It is also advantageous at low voltage operations where we gain both in terms of speed and power. We have used this technique to improve the speed of our memory. Excluding the memory cells, the dummy row and the sense amplifier, rest all of the digital logic has been converted to low Vth transistors. The memory cells have been excluded since being very small transistors, the leakage current will increase whereas the gain in speed is almost negligible. The control block, the decoders and IO blocks are all in low Vth, whereas the memory cells, the dummy column and dummy row along with the sense amplifier are in high Vth. Having this kind of configuration has helped us in gaining in speed and also reducing the dynamic power consumption by a considerable amount. Figure 6. shows the comparison between a memory with high Vth transistors and low Vth transistors in the memory core periphery. All the timings shown in the said figure are the worst case timings for a 8192x16 memory size. The PVT conditions at which the timings are quoted for this memory in figure 6. are as given below:

V=2.0V

T=125 DegC

Process = Slow

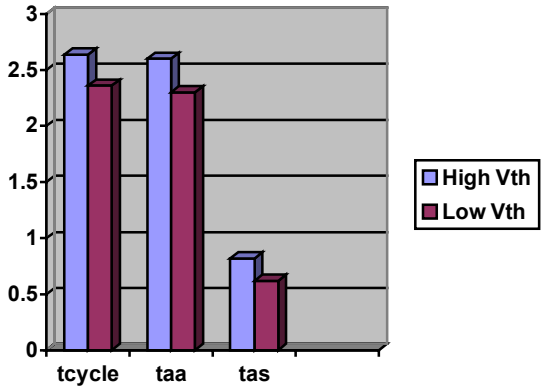

Figure 6. Comparison between Low Vth and High Vth memory for a 8kx16 cut

3. Implementation

Utilizing the above mentioned techniques, a memory compiler has been designed with words ranging from 32 to 16k and bits variance from 2 to 64. The core is split in two parts and different mux options viz. mux4, mux8 and mux16 are provided to have a balancing ratio between the wordline and bitline capacitances. The memory core is arranged such that it can have a maximum of 1024 rows with 512 columns or 512 rows with 1024 columns subject to the condition that the ratio of the numbers of rows and columns should meet the following criterion:

$$1/8 \leq \text{rows/columns} \leq 8/1$$

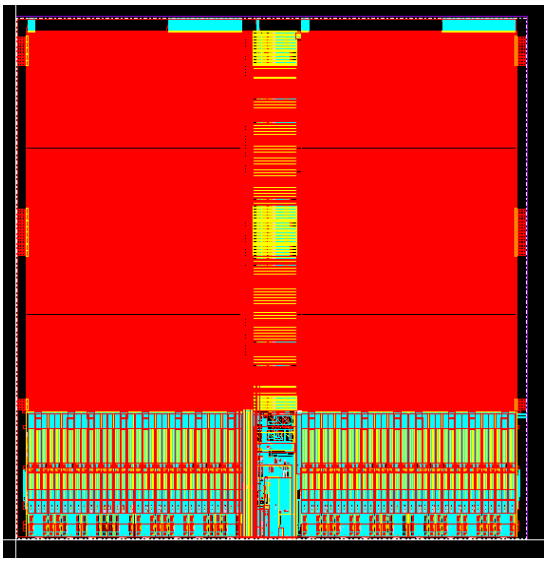

Figure 7. Layout view of a 8kx16 memory in 0.18u technology

In this memory the latch type sense amplifier is used for data sensing and the timing sequence is tuned to achieve sense amplifier operation at a bitline difference of 18mv at 1.8v and 25 DegC. In this arrangement, one sense amplifier is assigned to every 4 columns and the column decoded signals are used for connecting the correct pair of bitlines to the sense amplifier. Also depending on the number of rows, different discharge transistor sizes are chosen in order to have the optimum desired bitline difference at the time of "SENSE ON" for different sizes of memory and avoid paying penalty in terms of delay and power for any other given size.

4. Results

Using before mentioned techniques, a testchip has been fabricated with 6 memories each of different sizes and aspect ratio with built in self test and characterization circuits. Shown below in the table are the timing figures for 8kx16 cut for different voltage and temperature conditions. taa is the access time in read cycle, taaw is the access time in write cycle and tas is the address setup time whereas tcycle is the operating speed of the memory.

Table 1: Timing figure for a 8kx16 memory

Voltae & temp.	taa	Taaw	tcycle	Tas
V=1.95, T= -40	1.046	0.966	1.301	0.419
V=1.8, T= 25	1.539	1.409	1.686	0.549
V=1.5, T=125	2.602	2.368	2.647	0.827

5. Conclusion

This paper proposes a few techniques of designing a memory which gives us the best speed and is also power economic. As is shown in the results, by judiciously designing the sense amplifiers, tuning the selftime path and also deriving multiple clocks and proper synchronization of them with the main clock, we can achieve good timings. In the row decoding section, using 4 clocks anded with the address actually reduces the power consumed by the decoder block to ¼ as compared to the case in which only one clock is used.

6. References

[1] Isao FUKUSHI, et al., "A Low-Power SRAM Using Improved Charge Transfer Sense Amplifiers and a dual-Vth CMOS Circuit Scheme" Symp. On VLSI Circuits Digest of Technical Papers, pp. 142-145, 1998.

[2] Bharadwaj S Amrutur and Mark A. Horowitz, "A replica Technique for Wordline and Sense Control in Low-Power SRAM's" IEEE journal of Solid State Circuits, vol. 33, pp 1208-1219, 1998.

[3] Nobutaro Shibata and Morimura Hiroki, "A 1-V, 10-MHz, 3.5-mW, 1-Mb MTCMOS SRAM with Charge-Recycling Input/Output Buffers", IEEE Journal of Solid-State Circuits, Vol. 34, No. 6, pp. 866-877, June 1999.

[4] Simon J. Lovett, Gary A. Gibbs, and Ashish Pancholy, "Yield and Matching Implications for Static RAM Memory Array Sense-Amplifier Design", IEEE journal of Solid State Circuits, Vol. 35, No. 8, pp. 1200-1204, August 2000.

[5] Ming-Chuen Shiau and Chung-Yu Wu, "The Signal Delay in Inetrconnect Lines Considering the effects of Small Geometry CMOS invertoers", IEEE Transactions on Circuits and Systems, Vol. 37, N0. 3, pp. 420-425, March 1990.