High Speed Circuit Techniques in a 150MHz 64M SDRAM

Valerie Lines, Mammoun Abou-Seido
Cynthia Mar, Arun Achyuthan
MOSAID Technologies Incorporated Canada

Sampei Miyamoto, Yoshihiro Murashima,
Shinzo Sakuma
OKI Electric Co., Ltd Japan

Abstract

This paper outlines three methods used to decrease the access time in a 64M SDRAM. The access time from the read command, Taa, is reduced by the use of a novel column redundancy scheme with easy programming and by the use of current sensing in the data output path. The access time from the clock, Tac, is reduced by the use of a digital DLL whose functionality can be tested with on chip test functions.

1.0 Introduction

The increased die size of SDRAMs in successive generations increases the challenge of maintaining low access times. One factor that contributes to the access time from the read command, Taa, is the evaluation of the column redundancy. A scheme which eliminates this evaluation time and requires fewer fuses to be blown than a conventional scheme is described in section 2.

A fast data output path is essential to obtaining a low Taa. However, a large chip size can result in long databuses requiring circuit innovation to allow fast databus cycle times. Current sensing has been used in SRAMs[1] to reduce the interconnect delay; this paper describes the use of current sensing to reduce the busing delay in the data output path of a 64M SDRAM.

To minimize the access time from the clock, Tac, the delay between the clock bond pad and the internal clock signal used in the output buffer must be minimized. However, the pinout and pad distribution in 64M SDRAMs force the clock pad to be located far away from the majority of the data pads. A PLL or DLL must be used to generate a clock with no delay from the external clock which can be used in the output buffers. A digital DLL is used in this 64M SDRAM, as described in section 4.

These circuits have been implemented in a four bank 64M SDRAM designed jointly by OKI Electric Co. Ltd. and MOSAID Technologies Incorporated. The SDRAM operates at 150Mhz with a Tac of less than 5ns.

2.0 Column Redundancy Scheme

Column redundancy traditionally slows down the column access path in DRAMs and SDRAMs because the input address must be compared with the programmed faulty address before the decision to enable the normal or redundant column is made. This part uses a novel column redundancy scheme in which the faulty column is bypassed before the column cycle occurs. Thus there is no time penalty with this redundancy scheme, and it has the added advantage that only one fuse needs to be blown to replace a faulty column.

A block diagram representing the architecture of the column redundancy scheme is shown in Figure 1. Figure 1a shows the distribution of redundant columns along the array. The array has been divided into sections; each section consists of 8 groups of N columns, plus a redundant group of N columns. If there is a faulty column in one group of columns, that group is removed from the column decoder and all subsequent groups in the section are shifted down as shown in Figure 1b.

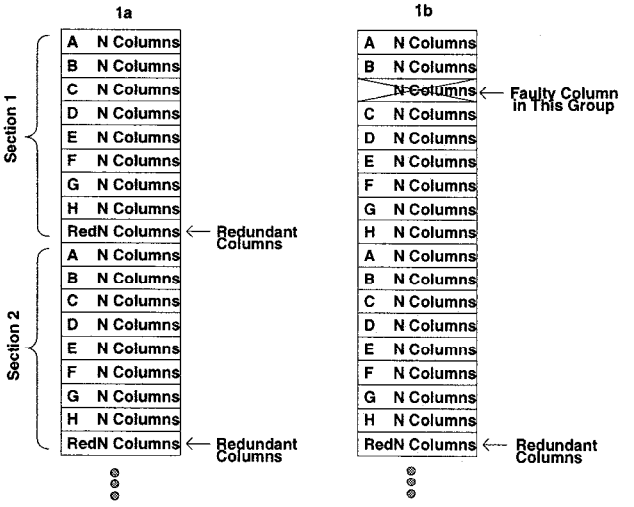

Figure 1: Column Redundancy Architecture

Figure 2 shows the circuit representation of the redundancy scheme. The fuses for each group of columns are connected in series, driven by a row address signal, PXi, (to choose which subarray to replace the column). When no fuse is blown, all $\overline{\text{SHIFT\_Ri}}$ nodes are precharged low by the low PXi signals during the row precharge cycle, but become high during a valid row cycle. When a fuse is blown, all $\overline{\text{SHIFT\_Ri}}$ nodes to the right of the blown fuse remain precharged low, causing the output of the first stage of the Y-decoder ($\overline{\text{DECi}}$) to be diverted to the adjacent secondary stage of the column decoder.

0-8186-8099-7/97 $10.00 © 1997 IEEE

Proceedings of the 1997 IEEE International Workshop on Memory Technology, Design and Testing (MTDT '97)
1087-4852/97 $10.00 © 1997 IEEE

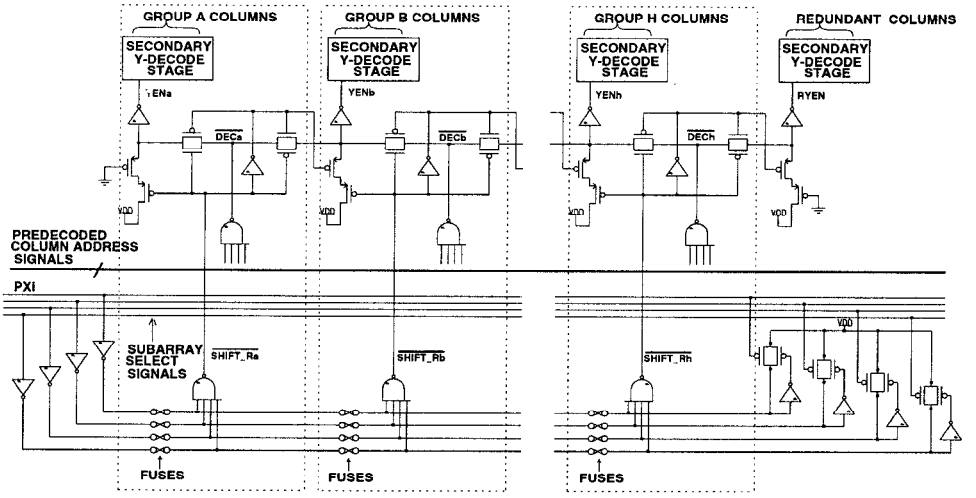

Figure 2: Column Redundancy Circuit

Thus, many fuses do not need to be blown to replace a column as in traditional circuits, and no delay is added to the column access time when the column decoder becomes reconfigured to eliminate access to the faulty columns.

3.0 Data Path

As SDRAMs become larger, one of the challenges in designing the datapath is achieving high frequency data cycles when the databuses are very long. This part uses three methods to achieve a high frequency datapath: (i) an interleaved databus scheme in the array in which the databuses operate at half the chip frequency[2], (ii) current sensing of data on long read databuses and (iii) converting the global databuses to single ended databuses operating at half the chip frequency in write cycles. The interleaved databus scheme used to bus data from the array to differential amplifiers provides adequate time to read from the array without having to prefetch data. The use of current sensing allows the output databuses to operate at the chip frequency. A block diagram of the input/output datapath is shown in Figure 3.

In a read cycle, data from the 75 MHz DB_A and DB_B pairs in one bank is sensed in consecutive cycles and driven onto the global databuses at 150MHz. Current sense amplifiers are used to evaluate the data on the global databuses (GDB, $\overline{GDB}$). The output of these first stage current sense amplifiers is multiplexed onto the read databuses (RDB, $\overline{RDB}$) with data from the other banks. Finally, the 150MHz data on the read databuses is sensed in current sense amplifiers and latched in the output buffer. Note that the global databuses and the read databuses are each more than 9mm long, but current sensing allows the databuses to be driven and precharged in a 6.6 ns cycle. The use of current mode drivers also allows easy multiplexing of data from different banks.

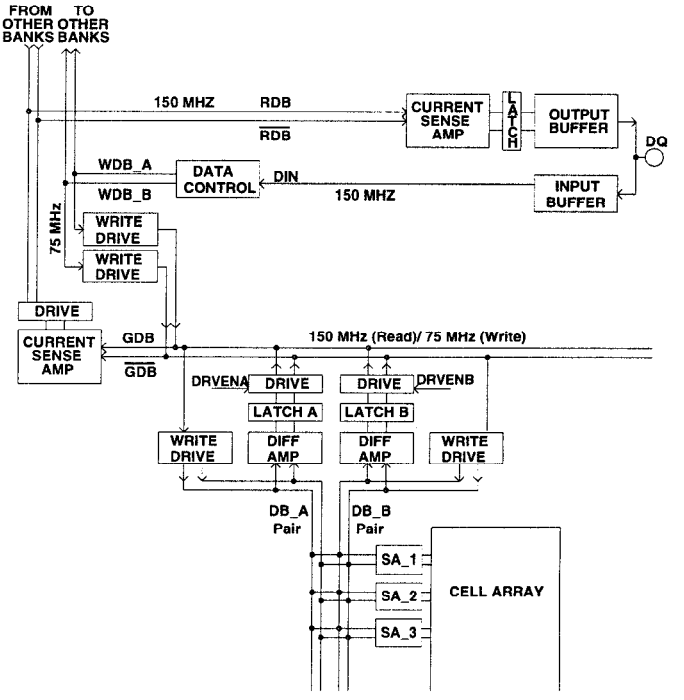

Figure 3: Input/Output Datapath

Proceedings of the 1997 IEEE International Workshop on Memory Technology, Design and Testing (MTDT '97)
1087-4852/97 $10.00 © 1997 IEEE

The current sense amplifier for the global databuses is shown in Figure 4. The voltage swing on the global databuses is less than 0.5mV.

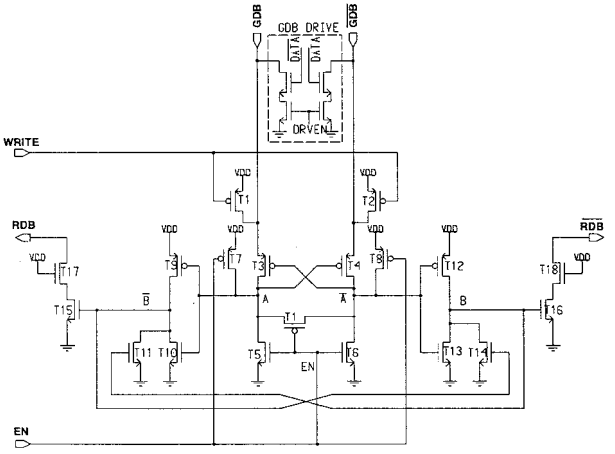

Figure 4: Global Databus Current Sense Amplifier

In a write cycle, input data is bussed to a control circuit which puts the data either on WDB_A or on WDB_B, depending on which databus pair, DB_A or DB_B is going to be used for the present clock cycle. Data is driven onto the selected WDB in one cycle and held for two clock cycles as shown in Figure 5. In the next cycle, the input data is placed on the other WDB bus. Data on WDB_A is driven onto GDB and the data from WDB_B is driven onto $\overline{GDB}$. Thus the GDB pair become single ended databuses during a write cycle, allowing adequate setup and hold time for the write drive circuits.

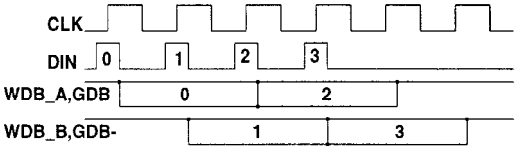

Figure 5: Input Data Timing (Cas Latency 4)

4.0 DLL

At high clock frequencies, a PLL or a DLL must be used in SDRAMs to achieve a low access time from the clock, Tac. The use of a PLL or DLL allows the delay from the external clock to the internal clock used to clock the output data to be zero. By eliminating the delay from the clock pin to the output buffer, which can be quite long in large SDRAMs, the Tac consists of only the delay through the output buffer.

A DLL is often built using an analog delay line to allow fine adjustment of the internal clock skew; however an analog delay line has a long settling time and will not hold its value in power down. A digital delay line does not have those drawbacks and if accurate models are available a digital delay line can be designed using only a digital simulator, allowing quick design. Moreover, a digital DLL does not require any off-chip components, and has good noise immunity.

A block diagram of the DLL is shown in Figure 6. It consists of a clock buffer, a DLL control block and phase detector, the delay line, a driver at the end of the line, and a reference delay feedback loop. The buffered external clock is the reference clock, CLKR. The phase of the feedback clock, CLKI, relative to CLKR is detected in the control block and the shift control signals, SHIFT_L, SHIFT_R, and CLK_SHIFT are generated.

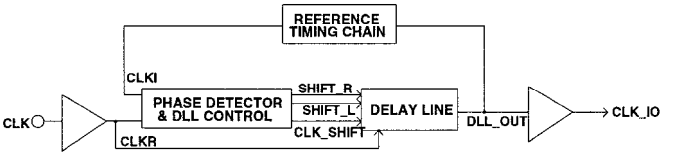

Figure 6: DLL Block Diagram

The DLL delay line consists of an array of N delay elements, each of which is controlled by a shift register element. When the delay line is initialized, there is a '1' in the first shift register element and only the first delay element is used in the delay line. When the DLL is enabled, the control circuit will give a SHIFT_R command as long as CLKI is earlier than CLKR. This shifts the '1' one stage to the right on each clock edge and adds one more delay element to the delay line. When CLKI becomes later than CLKR, a SHIFT_L command will be given to remove delay elements from the delay line. If the DLL is in a seek and track mode, the '1' in the shift register will move back and forth between adjacent elements. There is also a seek and lock mode, in which the shift clock is disabled after the first lock has occurred. The mode of operation can be programmed by the user by performing a mode register set command with a special key.

A circuit diagram of the delay element and its associated shift register element is shown in Figure 7. The output of the shift register element, Qi, determines if the input to the delay element is passed to the delay line or not. If not, the input to the delay stage is inverted and the delayed sig-

Proceedings of the 1997 IEEE International Workshop on Memory Technology, Design and Testing (MTDT '97)
1087-4852/97 $10.00 © 1997 IEEE

nal is passed to the next stage.

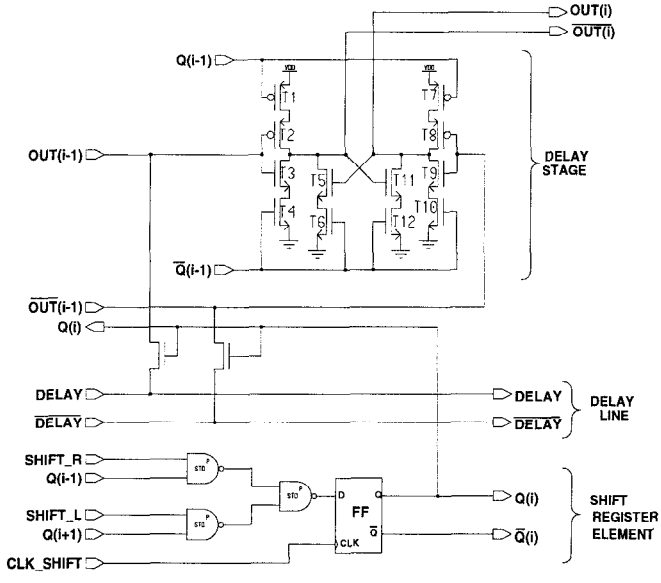

Figure 7: Delay Line Unit

The operation of the DLL can be tested by implementing test modes which are entered by a mode register set command and which generate an output flag which is sent to a DQ pin. Test modes are available to test the shift right and the shift left functionality of the shift register.

The reference timing chain in the feedback loop is used to compensate for the delay from CLK to CLKR and from the output of the DLL line, DLL_OUT, to CLK_IO. It consists of a chain of delay elements.

This DLL can have clock jitter equal to one delay element in the DLL line, t_{delay}, and a phase error equal to one delay element in the reference timing chain, t_{ref_delay}. Thus in theory, the total difference between the external clock and CLK_IO can differ by up to $t_{delay}+t_{ref\_delay}$, although the probability is low. This difference will cause slight variations in the Tac time and must be considered when determining margins for setup and hold times of internal flip-flops clocked by CLK_IO.

Note that while the DLL generated clock allows a low Tac, it also results in a short data hold time. The DLL generated clock must be adjusted to ensure that output data window is adequate.

5.0 Results

A 64M SDRAM was manufactured with the following results:

Chip Size: 9.45 x 20.44 mm

Taa: 20.4ns

Tovc (with DLL): 6.1ns

(Note: access times measured in an 8Mx8 part with a SSTL-3 interface, Cload=30pF at 80C,VDD=2.7V)

6.0 Summary

Three methods used in a 64M SDRAM to achieve high speed operation were described - the use of a redundancy scheme which eliminates delay in the Y-access, the use of current mode sensing in a high frequency output datapath, and the use of a digital DLL to minimize the access time from the clock.

References

[1] E. Seevinck et. al., "Current-Mode Techniques for High-Speed VLSI Circuits with Application to Current Sense Amplifier for CMOS SRAM's", IEEE J. Solid State Circuits, Vol 26, No. 4, pp 525-535 April 1991.

[2] T. Wojcicki, R.C. Foss, "High-Speed Circuits for Mega-Bit Synchronous DRAMs", Proceedings of the 4th International Conference on VLSI and CAD, VL5-2, October 1995.

Proceedings of the 1997 IEEE International Workshop on Memory Technology, Design and Testing (MTDT '97)
1087-4852/97 $10.00 © 1997 IEEE