

# Bit-wise read-compare-write scheme for low power read-modify-write DRAM operation

Y.-H. Park, S. Choi and H.-J. Yoo

To save power consumption in read-modify-write (RMW) DRAM operation, a bit-wise read-compare-write (RCW) scheme is presented. Its power consumption depends on the number of bits that have to be updated by the bit-wise compare result between the read data and the modified data. If a random bit-wise update ratio is assumed, a 15.6% power saving is achieved when the proposed scheme is applied to the previous design.

**Introduction:** A low power DRAM macro is among the essential intellectual-properties (IPs), especially for mobile 3D graphic rendering operations, which need a RMW transaction [1, 2]. The partial segment activation in [1] and the single bit-line write (SBW) scheme in [3] already reduced bit-line (BL) power consumption. The proposed scheme saves data bus (DB) power consumption during the RMW transaction. Most bits of the modified data are very similar to those of the read data (or old data) except for some least significant bits because of the relation between the read data and the modified data, as shown in Fig. 1 [1, 2].

Though a small DB swing is widely applied for low power and fast read operation, a full swing DB write operation is essential in order to overwrite the previous cell data latched in the bit-line sense amp (BLSA), as shown in Fig. 1. However, the full swing DB write operation always takes place whether or not each bit of written data is the same as the corresponding bit of stored data [1-3]. The conventional scheme using a 'blind' write operation makes power consumption worse as the required bandwidth is increased. This is because the number of DB lines and the number of memory accesses are increased in order to provide the required bandwidth over several Gbit/s [1, 2]. The proposed RCW scheme reduces the overall power consumption up to 28.2% of the previous design in [1].

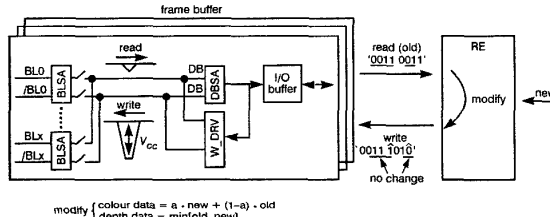


Fig. 1 RCW concept for RMW transaction

**Bit-wise read-compare-write (RCW) scheme:** The circuit and timing diagrams of the bit-wise RCW scheme are shown in Fig. 2a and b, respectively. After one CD\_Rx signal is turned on in the read cycle, a small voltage difference of about  $0.1V_{CC}$  is developed on the DB line by charge sharing. For fast and low power read operation, a pulse type CD\_Rx signal is used. After the DB sense amp amplifies the small voltage difference, the sensing data is stored in R\_LATCH. By the C2 strobe, the latched data is transferred to the modifying logic unit. In the modify cycle, the DB<sub>i</sub> line pair is pre-charged. Then, the newly written data is latched in W\_LATCH by C3 strobe. Then, the compare result between the stored data at node A<sub>i</sub> and the newly written data at node B<sub>i</sub> is available on the output node of XOR gate. Finally, the write driver and the CD\_Wx signal are either enabled or not enabled according to the compare result synchronised with the global write signal of C4, as shown in Fig. 2b. If the bit of the written data is the same as the corresponding bit of the stored data, the write driver and CD\_Wx are disabled. Then, the differential DB<sub>i</sub> remains at the pre-charged voltage level without any swing, as displayed by the dotted line in Fig. 2b. Otherwise, the BLSA is overwritten to the opposite data by a full swing DB write operation as in the conventional scheme, as displayed by the solid line in Fig. 2b. For bit-wise CD\_W control, C5 replaces C4 as shown in the CD control circuit at the lower right corner of Fig. 2a. C4 is a global write enable signal with neighbour DB blocks, but C5 is a local write enable signal depending on the bit-wise compare.

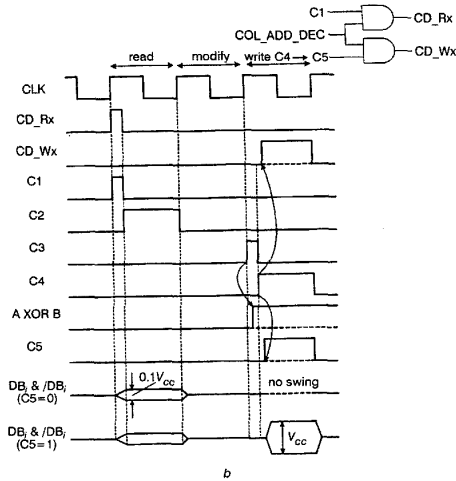
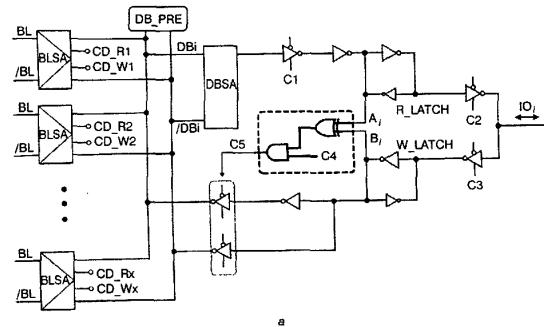


Fig. 2 Circuit and timing diagrams of RCW scheme

a Circuit diagram  
b Timing diagram

**Results:** In the RCW scheme, the total energy consumption of the DB line is expressed as  $E_{DB\_R} + \sigma_{DB} \cdot E_{DB\_W}$ , instead of  $E_{DB\_R} + E_{DB\_W}$ , where  $E_{DB\_R}$  and  $E_{DB\_W}$  are the total energy consumption of the DB line itself for the read and write operations, respectively, and  $\sigma_{DB}$  is the average bit-wise DB update ratio between the written data and the stored data. A 45.5% power saving is achieved in the DB line operation, if a random update probability of  $\sigma_{DB} = 0.5$  is assumed. This is because energy consumption by the DB read operation with a typical 10% swing of  $V_{CC}$  is only 10% of that by the full swing write operation (i.e.  $E_{DB\_R} = 0.1E_{DB\_W}$ ). Energy consumption by the compare logic is negligible compared with power consumption by the DB write operation.

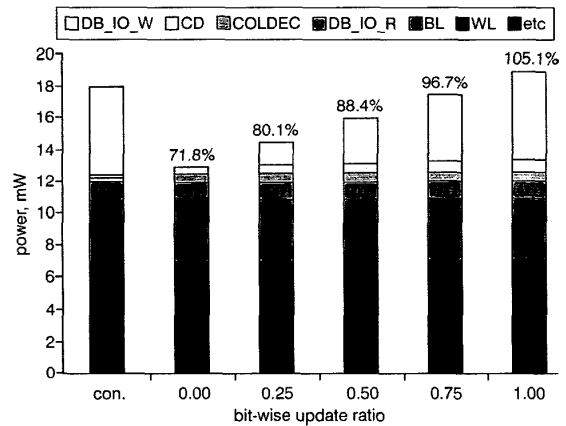


Fig. 3 Power contribution and percentage ratios according to DB update ratio ( $\sigma_{DB}$ ) when RCW scheme is applied to design [1]

Power contribution and power reduction ratios according to  $\sigma_{DB}$  are summarised as shown in Fig. 3 when the RCW scheme is applied to the

previous designs in [1]. 11.6% power saving is achieved for overall RMW operation if  $\sigma_{DB}$  and burst length are assumed as 0.5 and 1, respectively. When all data bits have to be updated (or  $\sigma_{DB} = 1.0$ ), total power consumption is increased by 5.1% of the overall power consumption. However, this is a very rare case in real applications.

Power consumption of the RCW scheme, ranging from 71.3% – 105.1% of the previous design, depends on the number of bits that have to be updated (or  $\sigma_{DB}$ ). Even though the compare unit and bit wise CD control increase the die area by 3.5% of previous design in [1], the RCW scheme is well matched with the RMW transaction without any timing penalty because the pre-fetch operation of the stored data is automatically performed in read cycle. The power saving factor by the RCW scheme is further enhanced as memory density and bandwidth are increased. This is because the increase in both DB load capacitance and burst length enlarges the portion of DB power consumption in the conventional power contribution.

**Conclusions:** The proposed RCW scheme provides a clear solution for low-power RMW operations for mobile 3D graphics or various high bandwidth DRAM macros. Its power consumption depends on the bit-wise update ratio. The power saving factor is further enhanced as memory density and bandwidth are increased.

© IEE 2002

23 October 2001

Electronics Letters Online No: 20020069

DOI: 10.1049/el:20020069

Y.-H. Park, S. Choi and H.-J. Yoo (Dept. of EE, Korea Advanced Institute of Science and Technology (KAIST), 373-1, Kusung-dong, Yusong-gu, Taejeon 305-701, Korea)

E-mail: yhpark@ceinfo.kaist.ac.kr

#### References

- 1 PARK, Y.-H., HAN, S.-H., LEE, J.-H., and YOO, H.-J.: 'A 7.1 GB/s low power rendering engine in 2D array embedded memory logic CMOS for portable multimedia system', *IEEE J. Solid-State Circuits*, 2001, **36**, (6), pp. 944–955
- 2 INOUE, K., NAKAMURA, H., and KAWAI, H.: 'A 10 Mb frame buffer memory with Z-compare and A-blend units', *IEEE J. Solid-State Circuits*, 1995, **30**, (12), pp. 1563–1568
- 3 KOOK, J., and YOO, H.-J.: 'A single bit line writing scheme for low power reconfigurable I/O DRAM macro'. IEEE European Solid-State Circuit Conference of Digest of Technical Paper, September 2000, pp. 420–423

## Static register implementation for one hot residue number systems

T. Conway

A method of implementing static registers for one-hot residue number systems is described. The method overcomes the high power dissipation problems associated with conventional flip-flops and clock distribution. The proposed design relies on the low activity factor inherent in the one hot coding structure and a hybrid clocking system that minimises the switching capacitance associated with the clock distribution.

**Introduction:** The residue number system (RNS) is a method of representing a range of integers  $0 \dots M-1$  by using their residues  $x_i$  modulo a series of relatively prime moduli  $m_0 \dots m_{N-1}$  where  $M$  is the product of the moduli. The operations of addition, subtraction and multiplication (all modulo  $M$ ) can be completed by operating on each residue independently, thus providing the potential for high-speed arithmetic [1]. The one hot RNS (OHRNS) system has been proposed as a means of accelerating the operations on each moduli by representing the individual residues in a one hot encoded manner. The resulting operations of addition, subtraction and multiplication can then be implemented by barrel shifter circuits. This encoding has been shown to achieve favourable power delay products for the arithmetic operations as well as providing practical solutions for the scaling operation [2].

However, as each residue  $x_i$ , is represented as a bus of  $m_i$  wires only one of which is high at any given time, there is a large number of storage elements required to implement a register for the OHRNS. For example, consider use of the moduli {37, 41, 43}, chosen to provide a dynamic range of 65,231 or approximately 16 bits. Each storage register would require 121 flip-flops of which only six would change in each clock cycle, i.e. three going high and three going low. Using standard static CMOS flip-flop designs would result in a significant dynamic power dissipation owing to clock switching within each flip-flop as well as providing a significant clock distribution loading effect [3]. Such an increase in power dissipation could negate the benefits in power delay products promised by the OHRNS.

This Letter describes a register architecture that takes advantage of the OHRNS attributes and overcomes the power dissipation problems associated with the one hot encoding of the residues.

**Proposed register design:** The basic flip-flop architecture is shown in Fig. 1. The storage function is achieved with a level sensitive latch which is transparent when the T/L input is high and latched otherwise. The D input is compared to the Q output by means of the exclusive OR gate. When they are the same logic level, the level sensitive T/L input is set to a logic 0 by means of the PMOS pull up and inverter and the TRIGB input is isolated from the circuit. When the D input is not the same as the Q output, then the TRIGB input is directed through the NMOS device to drive the T/L via the inverter. This TRIGB line is driven by a short pulse to logic 0 at the rising edge of the system clock. Hence, if the flip-flop D and its Q output are different at the rising edge of the system clock, then the short pulse on the TRIGB line is directed to provide a short latching pulse to the level sensitive latch which will store the input data bit.

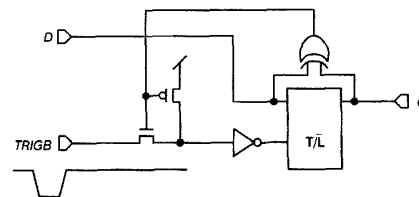


Fig. 1 Basic flip-flop architecture

This architecture has two valuable attributes. First, when there is no logic activity on the data input of the flip-flop, there is no switching activity within the flip-flop and hence there will be no dynamic power dissipation if the circuit is designed using CMOS static logic design. This will significantly reduce the power dissipation in the case of the one hot RNS system where the activity level of each one hot encoded residue is very low.

Secondly, the TRIGB input of the flip-flop drives the drain of a single NMOS device which can be made a minimum size device. When there is no new data for the flip-flop to store the input capacitance on this line will be one drain junction capacitance. Thus the clock distribution circuit will have to drive a small capacitance in all cases except when the input data has changed and needs to be stored. However, as only two data lines can change in the one hot encoded scheme, the switched capacitance on the TRIGB line will be low, leading to a low dynamic power dissipation in the clock distribution circuit (Fig. 2).

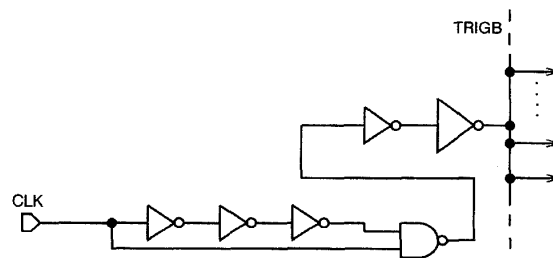


Fig. 2 Clock distribution circuit