

# Variability-insensitive scheme for NAND flash memory interfaces

C.-I. Son, S. Yoon, S.W. Chung, C.-I. Park and E.-Y. Chung

A novel NAND flash memory interface (NFMI) scheme to cope with uncertainty due to process, voltage and temperature (PVT) variations is proposed. The new NFMI scheme introduces a signal called data valid strobe to replace the signal read enable bar, which is a read strobe in the standard NFMI protocol. Experimental results show that the proposed scheme is insensitive to PVT variations, unlike the existing NFMI scheme, and hence substantially increases system performance as well as reliability.

**Introduction:** NAND flash (NF) memories have been widely used in mobile applications owing to their light weight and low power consumption. One of the most critical performance parameters in such applications is the data transfer time between the host machine and the NF memory connected to it. The data transfer time heavily depends on the performance of the NF memory controller placed between the host machine and the NF memory. Most NF memory controllers are equipped with software called flash translation layer (FTL), which is necessary for effective data management and read/write performance enhancements. Several approaches have been proposed to enhance the performance of flash cards from architecture and software perspectives. In [1], the authors introduced a compression layer to save the data space occupied in NF memories and to reduce the data transfer time. However, this method requires extra compression and decompression time for write and read, respectively. Another FTL enhancement proposed in [2] was using log blocks that can exploit the locality based on previous access history.

The technique presented in [3] also aimed at improving the read/write performance using the concept of locality. User data is typically mapped on blocks, and meta-data such as the address mapping table used in FTL is mapped on pages for fast access, as is the case with caches. However, previous approaches rarely considered the performance issues from the implementation perspective. One critical factor in this regard is the clock frequency. It has become a very important issue owing to PVT variations in the deep sub-micron era, where clock frequency degradation can be problematic for reliable data transmission [4]. Moreover, there are tight constraints on the cost and form factor of flash cards owing to fierce competition in the market. For this reason, the RC oscillator (RCOSC) has been widely used for clock generation in most flash cards such as secure digital (SD) card, multimediacard (MMC), memory stick, and other devices. However, RCOSC is more sensitive to PVT variations than phase-locked loop (PLL), which is widely used in application-specific integrated circuits (ASICs) and system-on-chips (SoCs). Also, the pad used in NF memories is sensitive to PVT variations owing to the output loading capacitance and resistance. To cope with this PVT variation issue, we propose in this Letter a new NFMI scheme, especially focusing on RCOSC and pads.

**Existing NFMI scheme:** Fig. 1 shows the block and timing diagrams of the existing NFMI scheme. The clock frequency of a controller is mainly determined by the interface with the NF memory. The critical parameter is the setup time constraint of the data register, which is FF1 in Fig. 1. Suppose that the clock period is  $t_{CK}$  and the variation of RCOSC is  $\epsilon$ . The controller issues a read command and NF memory reads data from NF cells to NF registers. The controller waits until its transfer is completed and then asserts the signal REB (read-enable) to low at the positive clock edge. The signal REB appears on the wire 'REB' after  $TO_{REB} \pm \alpha$ , where  $\alpha$  is the delay variation of REB PAD and the corresponding logic circuits due to PVT variations. By lowering the signal REB, the NF memory is instructed to transfer the data in the registers to IO. This operation takes  $(TI_{REB} + TO_{IO}) \pm \beta$ , where  $\beta$  is the delay variation of the NF memory. The worst-case delay for this path is typically defined as ' $t_{REA}$ ' in an NF memory specification. The data on IO propagates to the data input of FF1. The propagation delay is  $TI_{IO} \pm \gamma$ , where  $\gamma$  is the delay variation of IO PAD and control logics due to PVT variations. To safely fetch the data even in the worst case,  $t_s$ , the setup time constraint for FF1 is given by  $(t_{CK} - \epsilon) - \{(TO_{REB} + TI_{REB} + TO_{IO} + TI_{IO}) + (\alpha + \beta + \gamma)\}$ . More details of timing parameters can be found in [5].

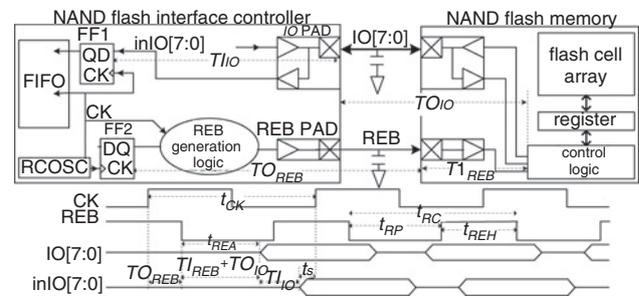


Fig. 1 Block and timing diagrams of existing scheme

- $TO_{REB}$ : REB output time ( $5 \pm \alpha$  ns)
- $TI_{REB} + TO_{IO}$ : REB input time + IO data output time (same as  $t_{REA}$ )
- $TI_{IO}$ : IO data input time ( $2 \pm \gamma$  ns)
- $t_{CK}$ : Controller clock period ( $30 \pm \epsilon$  ns)
- $t_{RC}$ : NF read cycle time (minimum 30 ns)
- $t_{RP}$ : REB pulse width (minimum 15 ns)
- $t_{REB}$ : REB high hold time (minimum 10 ns)
- $t_{REA}$ : REB access time (maximum 20 ns)
- $t_s$ : FF1 setup time

**New NFMI scheme:** The block and timing diagrams of a new NFMI scheme as shown in Fig. 2. This new scheme eliminates the signal REB and introduces a new signal called data valid strobe (DVS), which flows from the NF memory to the controller. After the read command is issued, the data is read from NF cells to NF registers, and the signal DVS and read data arrive at the controller after  $TO_{IO} \pm \theta$  and  $TO_{DVS} \pm \theta'$ , respectively. (Here  $\theta$  and  $\theta'$  are the delay variations of PAD and logic circuits due to PVT variations in NF memory.) Note that  $\theta$  is smaller than  $\beta$  of the previous NFMI scheme since the path for signal REB is eliminated. Also,  $\theta'$  will be close to  $\theta$  since the delay path of signal DVS in NF memory is similar to that of signal IO.

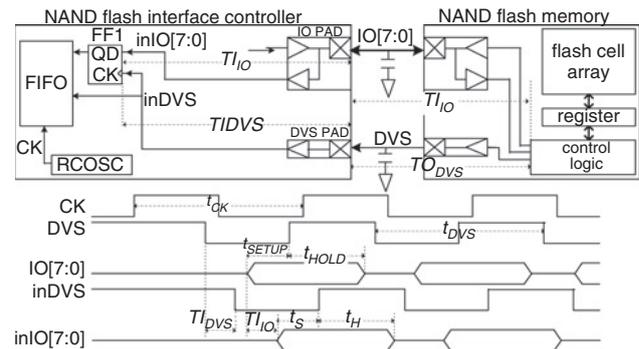


Fig. 2 Block and timing diagrams of proposed scheme

- $TO_{DVS}$ : DVS output time
- $TI_{DVS}$ : DVS input time ( $2 \pm \gamma'$  ns)
- $TO_{IO}$ : IO output time
- $TI_{IO}$ : IO input time ( $2 \pm \gamma$  ns)
- $t_{DVS}$ : DVS signal period. (30 ns)
- $t_{SETUP}$ : NF memory IO output setup time spec. (minimum 5 ns)
- $t_{HOLD}$ : NF memory IO output hold time spec. (minimum 5 ns)
- $t_s$ : setup time of FF1
- $t_H$ : hold time of FF1

The signal DVS and IO are fed into FF1 after  $TI_{DVS} \pm \gamma'$  and  $TI_{IO} \pm \gamma$ , respectively. Note that the corresponding data path for the read data is not changed in the new NFMI scheme. The only difference for FF1 is that it is clocked by the signal DVS instead of the controller clock, thus FF1 is independent of the controller clock. The signal DVS and IO use the same type of input pads and their propagation paths from the pads to FF1 are quite similar (only include wires and buffer logics), thus their delay variation will be also similar in terms of magnitude as well as direction, namely,  $\gamma' \approx \gamma$  and  $TI_{DVS} \approx TI_{IO}$ . In the worst case,  $t_s$ , the setup time constraint for FF1, is given by  $t_{SETUP} - \{(TI_{IO} + \gamma) - (TI_{DVS} + \gamma')\}$ . Also,  $t_H$ , the hold time of FF1 is given by  $t_{HOLD} + \{(TI_{IO} + \gamma) - (TI_{DVS} + \gamma')\}$ .  $t_{SETUP}$  and  $t_{HOLD}$  are given by the target NF memory. If  $\gamma \approx \gamma'$  and  $TI_{IO} \approx TI_{DVS}$ , then  $t_s \approx t_{SETUP}$  and  $t_H \approx t_{HOLD}$  which are insensitive to PVT variations, and the timing constraints of FF1 only depend on the timing specification of the NF memory. To support different clock frequencies of FF1 and the controller, a FIFO queue is used. A write operation to the queue

is clocked by DVS, whereas a read operation from the queue is clocked by the controller clock. The queue size can be easily computed by considering the difference between two clock frequencies.

**Experimental results:** We compared two interface schemes for K9F1G08U0A, which is an NF memory manufactured by Samsung Electronics [5]. Typically, the timing parameter values in NF memory data sheets are specified under the worst-case conditions (for  $\beta$ ,  $\theta$  and  $\theta'$ ), thus we only focus on the variation effects on the controller side in this experiment. In the existing NFMI scheme, the maximum clock frequency of the controller is determined by  $t_{RC}$ , which is the minimum clock period of the controller. For K9F1G08U0A,  $t_{RC}$  is 30 ns. Also the pad delays are set to 7 ns, which is measured by simulation assuming that they are designed using Samsung's 130 nm low-power process technology. The other timing parameters are shown in Fig. 1. Also, we modelled the variations of clock period ( $\varepsilon$ ) and PAD delays ( $\alpha + \gamma$ ) using the Gaussian distribution. We measured the number of setup timing violations in FF1 when the controller reads 40.96 KB data from the NF memory using both interface schemes, changing the standard deviation in the Gaussian distribution.

Fig. 3 shows the number of timing violations for each standard deviation tested. In the Figure, the mean values of clock period and pad delay are 30 and 7 ns, respectively. As shown in Fig. 3, both interface schemes do not generate any timing violations when the standard deviation is below 0.3. If the deviation becomes above 0.3, the number of timing violations increases abruptly when both clock period and pad delays are affected by the variations. If either clock period or pad delay is sensitive to variation, the violation occurrence is less serious, as expected. Note that the new NFMI scheme is insensitive to these variations as shown in its setup and hold time constraints. In the ideal case where the deviation is 0, the minimum clock period of each scheme will be identical, namely 30 ns. If the deviation becomes 1, the controller clock period of the existing scheme should be set to 33 ns to avoid timing violations, whereas the new scheme will maintain the same clock period. In this case, the read performance of the new scheme is improved by 10%, simply by increasing the clock frequency. The performance gap between the two schemes becomes larger and larger as the variations increase.

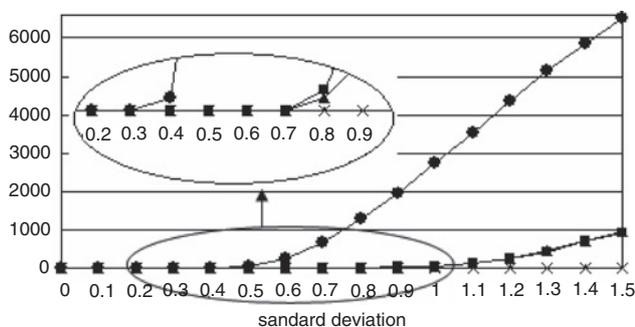


Fig. 3 Number of setup time violations in FF1 with PVT variations

- $\varepsilon$  variations in existing scheme
- ▲—  $\alpha, \gamma$  variations in existing scheme
- $\varepsilon, \alpha, \gamma$  variations in existing scheme
- x—  $\varepsilon, \gamma, \gamma'$  variations in proposed scheme

**Conclusions:** As process technology scales down, the clock frequency and pad delay of flash cards are largely affected by PVT variations. The proposed NFMI scheme can dramatically reduce these variation effects by introducing a new read strobe signal called DVS and therefore enables an NF controller to read data from NF memory with the maximum clock speed possible.

© The Institution of Engineering and Technology 2006  
18 July 2006

Electronics Letters online no: 20062239  
doi: 10.1049/el:20062239

C.-I. Son and E.-Y. Chung (School of Electrical and Electronic Engineering, Yonsei University, 134 Sinchon-dong, Seodaemun-gu, Seoul 120-749, Korea)

E-mail: eychung@yonsei.ac.kr

S. Yoon (Intel Corporation, 2200 Mission College Blvd., Santa Clara, CA 95054, USA)

S.W. Chung (Division of Computer and Communication Engineering, Korea University, Anam-dong, Seongbuk-Gu, Seoul 136-701, Korea)

C.-I. Park (Flash Memory Software Group, Samsung Electronics CO., Ltd, San 16 Banwol-Dong, Hwasung-City, Gyeonggi-Do, 445-701, Korea)

## References

- Huang, W.-T., Chen, C.-T., Chen, Y.-S., and Chen, C.-H.: 'A compression layer for NAND type flash memory systems'. 3rd Int. Conf. on Information Technology and Applications (ICITA), July 2005, Vol. 1, pp. 599–604
- Kim, S.-Y., and Jung, S.-I.: 'A log-based flash translation layer for large NAND flash memory'. 8th Int. Conf. on Advanced Communication Technology, 2006. ICACT February 2006, Vol. 3, pp. 1641–1644
- Kim, H., and Won, Y.: 'MNFS: mobile multimedia file system for NAND flash based storage device'. 3rd IEEE Consumer Communications and Networking Conf. (CCNC), January 2006, Vol. 1, pp. 208–212
- Hwang, C.G.: 'Where does memory go in the 21C?'. Symp. on VLSI Circuits Dig. Tech. Pprs, 2000, pp. 88–91
- Samsung Electronics Company: 'K9F1G08U0A 128M × 8-bit NAND flash memory data sheet V1.0', January 2006