Temperature self-adaptive program algorithm on 65 nm MLC NOR flash memory*

Shi Weihua(史维华)[1,†], Hong Zhiliang(洪志良)[1], Hu Chaohong(胡潮红)[2], and Kang Yong(亢勇)[2]

(1 State Key Laboratory of ASIC and Systems, Fudan University, Shanghai 201203, China)

(2 Intel Technology Development (Shanghai) Co. Ltd, Shanghai 200131, China)

Abstract: This paper presents an implementation for improving muti-level cell NOR flash memory program throughput based on the channel hot electron (CHE) temperature characteristic. The CHE I_g temperature characteristic is analyzed theoretically with the Lucky electron model, and a temperature self-adaptive programming algorithm is proposed to increase I_g according to the on-die temperature. Experimental results show that the program throughput increases significantly from 1.1 MByte/s without temperature self-adaptive programming to 1.4 MByte/s with the proposed method at room temperature. This represents a 30% improvement and is 70 times faster than the program throughput in Ref. [1].

Key words: temperature self-adaptive programming; 65 nm multi-level cell flash memory; program throughput
DOI: 10.1088/1674-4926/30/8/085012 **EEACC:** 2570

1. Introduction

As the density increases, flash memory applications demand low cost and fast program capability. Multi-level cell (MLC) storage technology is widely used to increase density and lower product cost. As a result, improvement of the program throughput has become a hot topic in NOR MLC design[2, 3].

NOR flash memory is mainly used for code storage because of its fast random access time. MLC NOR flash memory mainly uses channel hot electrons (CHE) for programming[4, 5]. With sufficient drain bias, the minority carriers cause impact ionization at the drain side; the carriers which gain enough energy from the lateral field overcome the barrier and are injected into the floating gate. CHE provides a fast program method with considerable current. However, CHE I_g exhibits temperature dependence[6], which will downgrade the performance.

This paper proposes an algorithm to improve the program throughput by using the CHE temperature characteristic. The CHE temperature dependence is analyzed with the Lucky electron model[4]. In addition, some experimental data are also provided to exhibit the CHE temperature characteristic. The implementation of temperature self-adaptive programming on both circuit and algorithm perspectives, which improves the program throughput by adjusting I_g based upon the temperature, is given. Experimental results indicate that program throughput with temperature self-adaptive programming increases significantly from 1.1 to 1.4 MByte/s at 20 °C. This represents 30% improvement and is 70 times faster than the program throughput in Ref. [1].

2. Temperature characteristic of the flash cell gate current

A compact flash cell model is shown in Fig. 1. The floating gate voltage follows Eq. (1)[4]:

$$V_{\rm fg} = \frac{1}{C_{\rm t}}(C_{\rm cg}V_{\rm cg} + C_{\rm d}V_{\rm ds} - I_{\rm g}), \tag{1}$$

where $C_{\rm cg}$ is the capacitance between the control gate and the floating gate, $C_{\rm d}$ is the capacitance between the drain and the floating gate, and $C_{\rm t}$ is the total capacitance from the floating gate to the other terminals of the transistor. Equation (1) shows that $V_{\rm fg}$ depends on the gate current $I_{\rm g}$.

Several models are used to describe the gate current caused by the CHE, such as the Lucky electron model[4], the effective electron temperature model and the physical model. The gate current is determined not only by the number of the hot electrons and their energy distribution, but also by the oxide field, which determines the fraction of the hot electrons reaching the floating gate. Due to the two-dimensional nature of CHE and its many unknown parameters, there is no analytical model for the CHE. Even so, the Lucky electron model

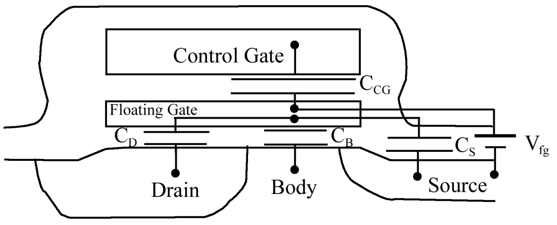

Fig. 1. Compact flash cell model.

* Project supported by the Intel Technology Development (Shanghai) Co, Ltd.

† Corresponding author. Email: whshi@fudan.edu.cn

Received 31 December 2008, revised manuscript received 27 March 2009

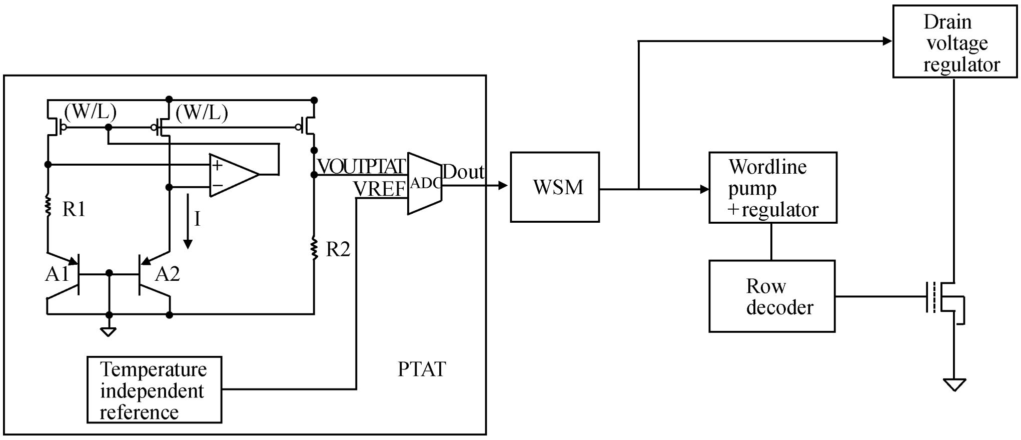

Fig. 3. Block diagram of temperature self-adaptive programming.

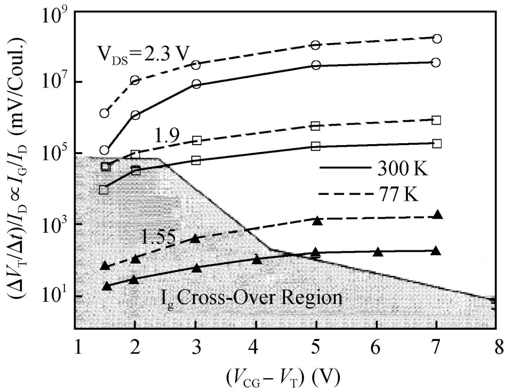

Fig. 2. Normalized $\Delta V_t/\Delta t$ as a function of $(V_{CG} - V_T)$[6].

exhibits the CHE I_g temperature characteristic[4]:

$$I_g \approx \frac{I_d}{\lambda_r}\frac{\varepsilon\lambda^{1.5}E_m^{2.5}P_2P_3}{1.15\sqrt{6}qN_s}\exp\left[-\left(\frac{\phi_B}{E_m\lambda}+\frac{2\phi_B}{3E_m\lambda}\right)\right], \quad (2)$$

where $P_2 = 1-\alpha e^{\alpha}E_1(\alpha)$ and $\alpha \equiv \frac{2kT}{q\lambda E_s} \approx \frac{6kT}{q\lambda E_{ox}}$. This expresses the temperature characteristic of I_g. With rising temperature, both I_d and P_2 decrease for N type transistors, resulting in a decrease of I_g.

Figure 2 shows experimental temperature dependence data for both I_g and I_b in Ref. [6]. It also shows that I_g will decrease when temperature increases. This is also proven on Intel 65 nm MLC technology. In this technology, with the same program efficiency, the program currents are 67, 64.6 and 60.2 μA at temperatures of 85, 40 and –25 °C, respectively. This proves that the increased I_d can compensate the decreased I_g at high temperatures.

According to the temperature characteristic of I_g, changing I_d, the slope of the word-line voltage, as well as the DOP (degree of parallelism) based on the temperature can be used to achieve high program throughput. The detailed implementation of this will be described later.

3. Implementation of temperature self-adaptive programming

In MLC NOR flash, the lowest V_t (V_{tl}) is decided by the leakage, while the high end V_t (V_{th}) is determined by the reliability, thus the range from V_{tl} to V_{th} is very limited. Taking into account cell from cell variation and cycling degradation, the range from V_{th} is even worse than the ideal case. To achieve multi-levels, MLC NOR storage requires tight V_t distribution for each level. This means the right amount of charge should be placed on the floating gate during programming. Usually accurate charge placement is achieved by the program and verification (P & V) algorithm, but this penalizes program throughput since it requires extra verification of the iteration[7, 8].

This section explains the circuit and algorithm implementation for temperature self-adaptive programming.

3.1. Circuit design

An on-chip temperature detector which is proportional to absolute temperature (PTAT) is used to detect the temperature in real time[9]. Figure 3 is a block diagram, where WSM is the write state machine. Based on the PTAT output, the WSM controls the word-line pump, regulator, and drain voltage generator to generate different output voltages depending on temperature.

3.2. Algorithm implementation

Staircase word-line voltage programming[7, 8] is used in temperature self-adaptive programming. However, the degree of parallelism (DOP), the word-line voltage and V_d will be changed based on the PTAT output. Here, DOP is the number of cells which can be programmed in parallel. Firstly, as described in Section 2, the programming I_d varies with the temperature, which leads to the fact that the DOP can be changed at different temperatures. For example, the DOP can be 48 at 85 °C while it is 50 at 40 °C . Secondly, the CHE temperature characteristic also shows that the word-line voltage can ramp faster at low temperatures. Finally, V_d is tuned to increase I_d in Eq. (2) according to the temperature. The detailed algorithm is shown in Fig. 4. Experimental results show that the increase rates of program throughput are 20% and 30% at 85 and 20 °C, respectively.

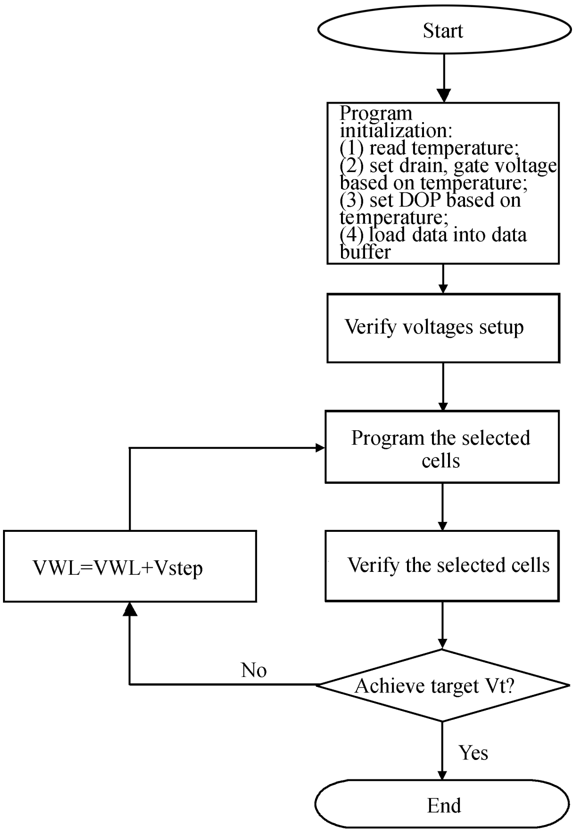
Start
Program initialization:
(1) read temperature;
(2) set drain, gate voltage based on temperature;
(3) set DOP based on temperature;
(4) load data into data buffer
Verify voltages setup
Program the selected cells
VWL=VWL+Vstep
Verify the selected cells
No
Achieve target Vt?
Yes
End

Fig. 4. Simplified algorithm with temperature self-adaptive programming.

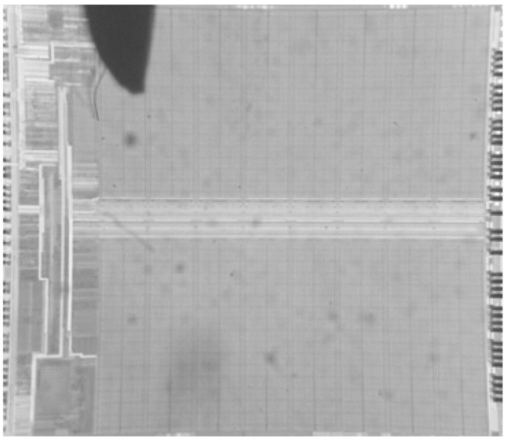

Fig. 5. Die photomicrograph with 1-Giga bits 65 nm MLC NOR.

4. Experimental results

The proposed temperature self-adaptive programming is implemented in Intel 65 nm flash technology and tested. Figure 5 is a die photomicrograph of a 1 Giga bit (GB) 65 nm 2 b/cell MLC chip. The measurement results of program throughput are shown in Fig. 6. The measured temperatures are 20, 40 and 80 °C, and 36 buffer program times are obtained under each temperature, where the buffer is 1024 bytes for the program. From the experimental results, it takes 718 μs after applying adaptive temperature self-adaptive programming instead of 916 μs with traditional programming to program one buffer at 20 °C. Compared with 0.17 MByte/s[10] and 0.8 MByte/s[11] with DOP = 256, this 65 nm MLC NOR flash has achieved 1.1 MByte/s with DOP = 50 without temperature self-adaptive programming. With the proposed programming algorithm, it can achieve 1.4 MByte/s with the same DOP.

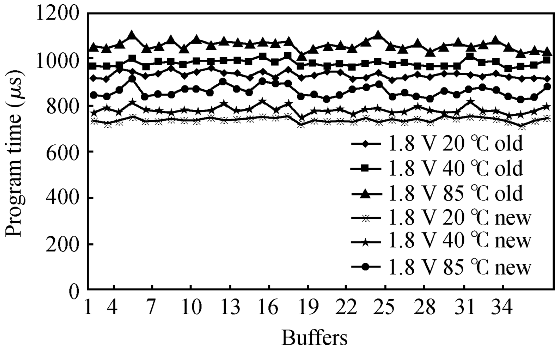

Fig. 6. Comparison of program without and with temperature self-adaptive programming for 1024 bytes.

5. Conclusion

This paper has presented the implementation of temperature self-adaptive programming on 65 nm 2 bits/cell MLC technology. The $I_{\rm d}$, the slope of the word-line voltage and the DOP are changed according to the temperature characteristic of $I_{\rm g}$. Experimental results show that the program throughput increases significantly from 1.1 MByte/s without temperature self-adaptive programming to 1.4 MByte/s with the proposed method at room temperature. This represents a 30% improvement and is 70 times faster than the program throughput in Ref. [1]. In addition, it can also be applied in other flash technologies for improving the program throughput.

In practice, as a lot of design parameters in the program algorithm can be affected by the temperature, the control state machine can be modified to further improve the program throughput.

Acknowledgement

The author would like to thank the Flash Product Group of Intel Technology Development (Shanghai) Co., Ltd. for their great support and timely assistance in accomplishment of this research work.

References

[1] Wong G. Flash memory trends. Flash Memory Summit, 2008. http: //web. njit. edu/~rlopes/5.2%20-%20Flash%20Memory%20Trends_FMS.pdf

[2] Takeuchi K, Kameda Y, Fujimura S, et al. A 56 nm CMOS 99 mm^2 8 Gb multi-level NAND flash memory with 10 MB/s program throughput. ISSCC Dig Tech Papers, 2006: 507

[3] Taub M, Bains R, Barkley G, et al. A 90 nm 512 Mb 166 MHz multilevel cell flash memory with 1.5 MByte/s programming. ISSCC Dig Tech Papers, 2005: 54

[4] Hu C. Lucky electron model of hot electron emission. IEDM Tech Dig, 1979: 22

[5] Yeargain J, Kuo K. A high density floating gate EEPROM cell. IEDM Tech Dig, 1981: 24

[6] Esseni D, Selmi L, Sangiorgi E, et al. Temperature dependence of gate and substrate currents in the CHE crossover regime. IEEE Electron Device Lett, 1995, 16(11): 506

[7] Grossi M, Lanzoni M, Riccò B. Program schemes for multilevel flash memories. Proc IEEE, 2003, 91(4): 594

[8] Grossi M, Lanzoni M, Riccò B. A novel algorithm for high-throughput programming of multi-level flash memories. IEEE Trans Electron Devices, 2003, 50(5): 1290

[9] Kim J P, Yang W, Tan H Y. A low-power 256-Mb SDRAM with an on-chip thermometer and biased reference line sensing scheme. IEEE J Solid-State Circuits, 2003, 38(2): 329

[10] Silvagni A, Zanardi S, Manstretta A, et al. Modular architecture for a family of multilevel 256/192/128/64 mbit 2-bit/cell 3v only NOR flash memory devices. IEEE Trans Electron Devices, 2001, 48: 937

[11] Versari R, Esseni D, Falavigna G, et al. Optimized programming of multilevel flash EEPROMs. IEEE Trans Electron Devices, 2001, 48: 1641