Received 28 August 2020; revised 11 October 2020 and 5 November 2020; accepted 22 November 2020. Date of publication 30 November 2020; date of current version 28 January 2021. The review of this article was arranged by Editor M. Liu.

Digital Object Identifier 10.1109/JEDS.2020.3041088

Adaptive Pulse Programming Scheme for Improving the V_{th} Distribution and Program Performance in 3D NAND Flash Memory

ZHICHAO DU 1,2, SHUANG LI1,2, YU WANG3, XIANG FU3, FEI LIU1, QI WANG1,2,3, AND ZONGLIANG HUO 1,2,3

1 Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China
2 The University of Chinese Academy of Sciences, Beijing 100049, China
3 Yangtze Memory Technologies Company Ltd., Wuhan 430205, China

CORRESPONDING AUTHOR: Z. HUO (e-mail: huozongliang@ime.ac.cn)

This work was supported in part by the National Science and Technology Major Project of China under Grant 2017ZX02301002.

ABSTRACT For triple-level or quad-level 3D NAND flash memory, narrowing the V_{th} distribution of each state without influencing page program performance is one of the challenges. Considering this challenge, a novel adaptive pulse programming (APP) scheme was proposed. The proposed APP scheme adopted additional verify operations to separate the cells with different programming speed. It enhanced the program effect of slow cells by using increasing programming step voltage, and prevented the fast cells from over programming by using shorter programming pulse width through controlling the voltage of bitline. Compared with general incremental step pulse programming scheme, experimental results on TLC 3D NAND flash showed that, APP scheme could reduce the V_{th} distribution width of cells by around 15% and at the same time save the program time.

INDEX TERMS 3D NAND flash memory, ISPP, 3-bit per cell, V_{th} distribution.

I. INTRODUCTION

Over the past decade, due to the development of the word line (WL) stacking technology, three-dimension (3D) NAND flash memory has gained an rapidly growing market share in applications of mobile device and solid state drives (SDDs) [1], [2]. In addition, 3-bit/cell (TLC) and 4-bit/cell (QLC) NAND flash memory have become the mainstream of the market, expanding the capacity and reducing the cost of 3D NAND flash significantly [3], [4]. However, for multi-bit per cell NAND flash memories, the cell's threshold voltage (V_{th}) is limited to a certain program threshold window, normally -3V to 5V, which means that the more states the NAND flash memory device needs to store, the less margin of the V_{th} placement for each state [5]. When other disturbances and noises are further considered, the remaining read window margin becomes much worse [6], [7]. Therefore, it is highly essential for TLC/QLC NAND to shrink the voltage V_{th} distribution width of each state. Generally, incremental step pulse programming (ISPP) scheme can effectively program cells to a narrow V_{th} distribution [8], as the cell's V_{th} shift (ΔV_{th}) is almost proportional to the programming step voltage ($\alpha \bullet \Delta V_{pgm}$). Lowering programming step voltage can reduce the width of cells' V_{th} distribution, however, it results in more program and verify loops thus increasing program time. Coarse and fine programming scheme is another method to improve the cells' V_{th} distribution [9]. In addition, several options such as expanding the range of cells' voltage threshold into the more negative regions were proposed [4], [10], but the distribution margin is still limited. Despite the program voltage, for NAND flash device, programming pulse width also affects cell's V_{th}. The electrons are trapped in the storage layer at a logarithmic rate proportional to programming pulse time [11]. Study of using longer programming pulse width for the later program loops were proposed [12], it can compensate the WL RC delay influence for the cells

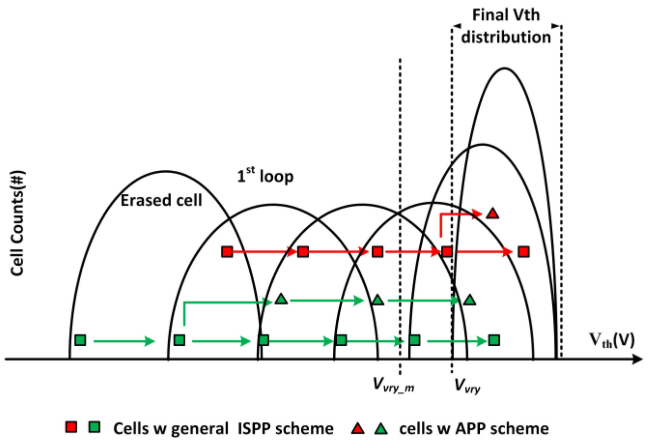

FIGURE 1. Examples of cells' V$_{th}$ movement with different programming schemes.

at the far end of WL when the voltage of program is getting higher. However, there have been few studies that utilized the method of controlling programming pulse width to shrink the V$_{th}$ distribution width. In this work, adaptive pulse programming (APP) scheme will provide a novel feasible way to obtain a narrower V$_{th}$ distribution width by controlling the programming voltage and pulse width together, and more important, ensure good program performance.

II. ADAPTIVE PULSE PROGRAMMING SCHEME AND EXPERIMENTS

In ISPP operation, a series of program phases and verify phases are executed sequentially. Due to the process variation and programming mechanism, cells' V$_{th}$ after a programming loop with a fixed programming pulse width would form an initial distribution, shown in Fig. 1. With a fixed programming step voltage and a fixed programming pulse time, all the cells, regardless of their V$_{th}$ in the previous loop, almost have the same constant V$_{th}$ shift after each step [8]. With general ISPP scheme, the cell that is programmed just slightly below the verify voltage (V$_{vfy}$) in the previous loop, will move a distance of $\alpha \bullet \Delta V_{pgm}$ from V$_{vfy}$ hence determine the distribution width during programming, as shown by the red square in Fig. 1. On the other hand, the cell that moves slowly to the verify level finally determines how many programming loops are required in the end, as shown by the green square in Fig. 1. If the cell, whose V$_{th}$ status is closed to V$_{vfy}$, could be suppressed in the subsequent programming loop, the width of the final V$_{th}$ distribution could be reduced. Meanwhile, if the cell whose V$_{th}$ is far away from the verify level could be speeded up during each programming loop, it would be highly possible that the cell could be programmed successfully in fewer loops. Based on these mechanisms, and also considering the logarithmic relationship between the V$_{th}$ shift and the programming pulse time [11], the APP scheme was proposed, which applied different programming voltages and pulse widths to the cells if their V$_{th}$ locates at different zones respectively.

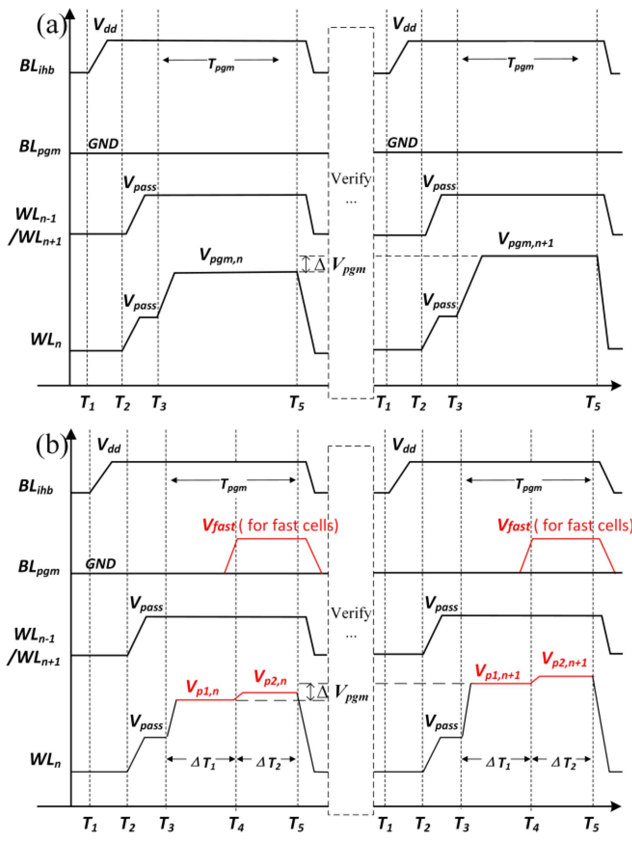

FIGURE 2. (a) Timing waveform of general ISPP scheme (b) Timing waveform of adaptive pulse programming scheme.

In order to classify the cells, an additional verify operation should be performed with a decision voltage (V$_{vry\_m}$) which is lower than the program verify voltage. In a particular loop, if a cell's V$_{th}$ is higher than V$_{vfy}$, it would be marked as "off" cell. If the V$_{th}$ is between V$_{vry\_m}$ and V$_{vry}$, it would be marked as "fast" cell, and if the V$_{th}$ is smaller than V$_{vfy\_m}$, it would be marked as "slow" cell. Once the cell is classified, adjusted programming voltage and pulse width could be applied.

Figs. 2(a) and (b) show the timing waveform of general ISPP and APP programming scheme for two programming loops, respectively. In general ISPP programming scheme (Fig. 2(a)), in the nth loop, a programming pulse with a fixed width (T$_{pgm}$) and voltage (V$_{pgm,n}$) is applied to all the cells on the selected WL, while the programming voltage will increase a constant step ΔV_{pgm} loop by loop ($V_{pgm,n+1} = V_{pgm,n} + \Delta V_{pgm}$). For the off cell, the associated bitline (BL) is biased to V$_{dd}$ so that the programming effect can be inhibited. While in the APP scheme (Fig. 2(b)), the detailed processes for fast, slow and off cells are described as follows:

1) for the slow cells, the voltage of BLs (V$_{BL}$) is biased at 0V during the whole programming pulse. In the nth loop, from the moment of T$_4$, the voltage of WL is increased from V$_{p1,n}$ to V$_{p2,n}$ ($V_{p1,n} \leq V_{p2,n}$). In

this way, the effective T_{pgm} is ΔT_1 with programming voltage $V_{p1,n}$, and then ΔT_2 with programming voltage $V_{p2,n}$. The increased voltage of WL will speed up the programming effect of the slow cells, and it will help to reduce the WL RC delay influence for the cells at the far end of WL. In addition, the programming pulse time could be decreased properly as the increasing voltage could compensate for the decreasing programming pulse time.

2) for the fast cells, the V_{BL} is first biased at 0V and then raised to V_{fast} at T_4, when the programming voltage is raised to $V_{p2,n}$. The increasing voltage of BL ($V_{BL} = V_{fast}$) inhibits the fast cells from over programming during the second part of programming pulse ΔT_2, as it rises the channel potential and reduces the effective electric intensity. As V_{fast} need to inhibit the cell, we could use $V_{fast} = V_{dd}$. In this way the effective T_{pgm} is almost only ΔT_1 with the effective programming voltage $V_{p1,n}$, which is much less than slow cells. The fixed gap voltage ($\Delta V_p = V_{p2,n} - V_{p1,n} = V_{p1,n+1} - V_{p2,n+1}$) should be approximately less than the difference between the channel potential of fast cells during ΔT_2 and the programming step voltage (ΔV_{pgm}), so that it can achieve the purpose of the scheme to prevent fast cells from over programming. For the sake of brevity, in the following text, the subscript of the loop will be omitted.

3) for the off cells, it is as the same as general ISPP scheme. The V_{BL} is biased to V_{dd} before the channel is boosted up by raising the unselected WL voltage to V_{pass}. Those cells will stay in inhibited state during the left entire program loops.

An example of cells' V_{th} movement with APP scheme is also shown in Fig. 1 by the red and green triangles. Once the cell is separated from fast and slow by sensing with V_{vry_m} at each verify loop, the fast cell's (red triangle) ΔV_{th} becomes smaller with APP scheme, meanwhile the slow cell's (green triangle) ΔV_{th} increases. A simplified flow chart of an embedded one-step program operation with APP scheme is shown in Fig. 3.

After each program loop, two verify operations for each state are implemented. As for 3D NAND chip which applied ABL (All-Bitline) structure [13], current sensing scheme was adopted, and the sensing time was around hundreds of nanoseconds. For those two verify operations, changing the selective word line voltages ($V_{vry\_m} < V_{vry}$) while keeping the sensing time constant, almost equals to changing sensing time for the two operations but keeping the same selective word line voltage (V_{vry}). More importantly, the time cost of sensing with one more different sensing time, which is less than 1µs, is much less than sensing with pre-charging the selective word line to a different level, which will take tens of microseconds. Therefore, the cost of using different sensing times for those two verify operations could be almost ignored [3].

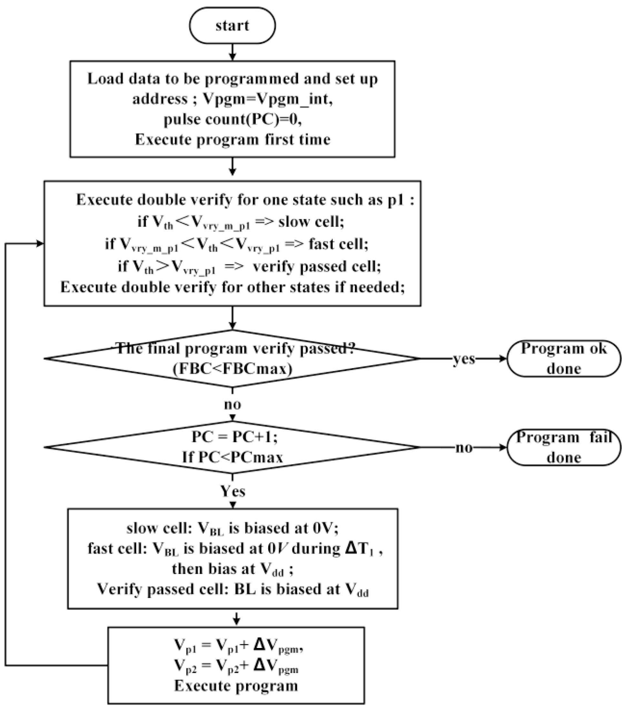
start
Load data to be programmed and set up address ; Vpgm=Vpgm_int, pulse count(PC)=0, Execute program first time
Execute double verify for one state such as p1 :
if $V_{th} < V_{vry\_m\_p1}$ => slow cell;
if $V_{vry\_m\_p1} < V_{th} < V_{vry\_p1}$ => fast cell;
if $V_{th} > V_{vry\_p1}$ => verify passed cell;
Execute double verify for other states if needed;
The final program verify passed? (FBC<FBCmax)
yes
Program ok done
no
PC = PC+1; If PC<PCmax
no
Program fail done
Yes
slow cell: V_{BL} is biased at 0V;
fast cell: V_{BL} is biased at 0V during ΔT_1, then bias at V_{dd} ;
Verify passed cell: BL is biased at V_{dd}
$V_{p1} = V_{p1} + \Delta V_{pgm}$,
$V_{p2} = V_{p2} + \Delta V_{pgm}$
Execute program

FIGURE 3. Flow chart of program operation with proposed APP scheme.

In this work, the experiments were carried out with a 64-layer 3D charge-trapping NAND flash chip on FPGA platform [14]. Both single programming step and embedded program operation were tested and compared with the proposed APP scheme.

III. RESULTS AND ANALYSIS

In order to investigate the programming pulse width influence, the V_{th} shift depending on different programming pulse width (T_{pgm}) changing is tested first and shown in Fig. 4. The details of the experiment process are described as below: first, all the cells are erased with the same erase verify voltage. Second, the whole cells are programed to the same stage to wake up by using a few incremental step programming pulse loops with a fixed width. From the n^{th} loop, the programming width is reduced accordingly. After each programming loop, every cell's V_{th} is get and averaged to obtain the final V_{th} distribution. Through this large number of read operations, RTN and other sensing noises could be suppressed [15]. Third, the ΔV_{th} between n^{th} and $n-1^{th}$ loop is measured. By repeating this erase-program-read cycles 10 times, the mean ΔV_{th} is finally obtained. Fig. 4(a) shows one example of the V_{th} distribution after each program loop of this experiment. Cells are programmed with step voltage equaling to 0.5V ($\Delta V_{pgm} = 0.5$V). From the n^{th} loop, the programming pulse width is changed from 38µs to 23µs. It can be seen obviously that the ΔV_{th} is reduced and smaller than half of the previous loop immediately.

Cases of using different programming step voltages are also tested. Fig. 4(b) shows the ΔV_{th} value between n^{th} and

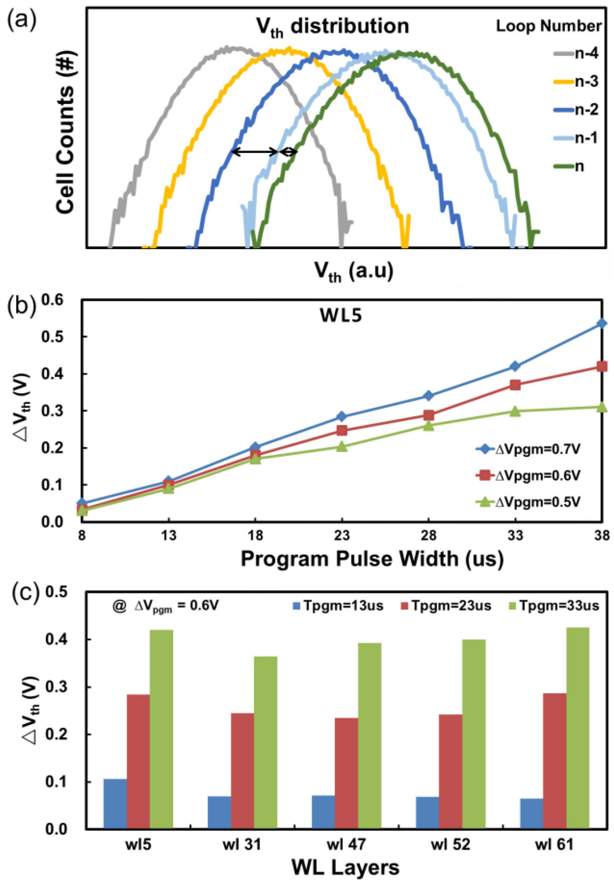

FIGURE 4. (a) V_{th} distribution after each program loop, the programming pulse width is reduced at the nth loop. (b) ΔV_{th} of nth loop with different programming pulse width of WL5 under the condition of three different programming step voltages. (c) Some typical WLs' ΔV_{th} of nth loop using three different programming pulse width when ΔV_{pgm} is 0.6V.

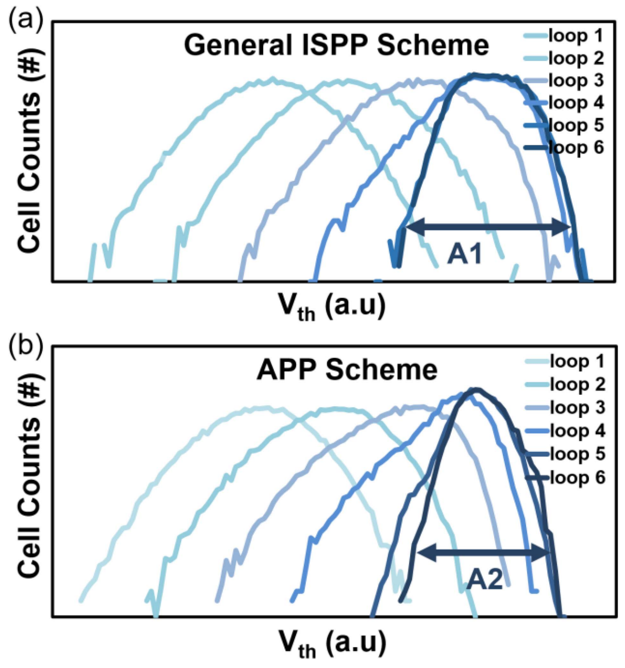

FIGURE 5. (a) the V_{th} distribution of each program loop when cells are programmed to a target state with general ISPP scheme (b) the V_{th} distribution of each program loop when cells are programmed to a target state with APP scheme.

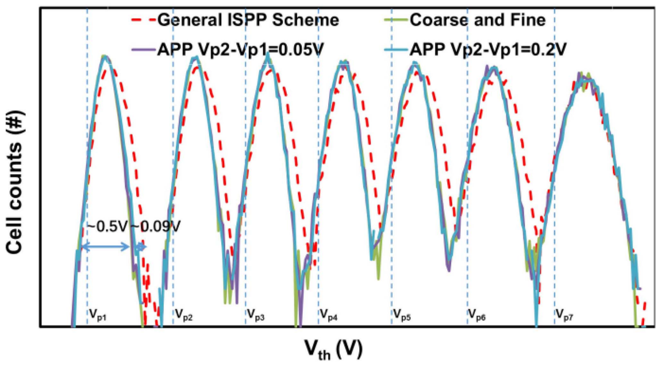

FIGURE 6. The comparison of TLC V_{th} distribution with APP scheme, coarse and fine scheme and general ISPP scheme.

n − 1th loop of WL5 with programming step voltage equaling to 0.5V, 0.6V and 0.7V respectively. The ΔV_{th} under different programming pulse width of the nth loop shows the same trend with different step voltages. However, when the programming pulse width is below 13µs, the ΔV_{th} of three cases is below 0.1V and is almost the same, which means when pulse width is below this range, the programming effect is almost regardless of the programming step voltage. When the pulse width increases more, the ΔV_{th} difference between different step voltages becomes obvious. As when the time becomes much more sufficient, the programming voltage influence on ΔV_{th} is more dominant. Finally, the ΔV_{th} will become stable and close to $\alpha \bullet \Delta V_{pgm}$. Fig. 4(c) shows the ΔV_{th} of a few typical WLs with three programming pulse width (13µs, 23µs, 33µs, respectively.) of this 64-layer NAND chip when the ΔV_{pgm} equals to 0.6V. For different WL layers, the ΔV_{th} varies slightly but the trend still exists. This variation relates to the process mechanism, as the channel hole profiles are not uniformed for different WLs.

Figs. 5(a) and (b) shows the V_{th} distribution when the cells are programmed to a target state with general ISPP and APP scheme, respectively, with verify operation. For the APP scheme, the condition is: $V_{p1} = V_{p2} = V_{pgm}$, $\Delta T_1 = \Delta T_2 = 1/2T_{pgm}$. It can be found that from the 3rd loop, the upper tail of the V_{th} distribution with APP scheme will be suppressed. And finally, the cell's upper tail of the distribution is obviously narrower than using general ISPP scheme (A2<A1), which is expected from the previous analysis.

After investigating the ΔV_{th} of single programming loop, the embedded TLC program operation is tested and compared with different schemes. Fig. 6 shows the final V_{th} distribution with three different schemes by using random data input. The APP scheme with two different conditions ($\Delta T_1 = \Delta T_2 = 1/2T_{pgm}$, $V_{p2} - V_{p1} = 0.05V$ and $V_{p2} - V_{p1} = 0.2V$, $\Delta V_{pgm} = 0.5V$) are applied, but only to the states from P1 to P6, since the upper tail of P7 state has no impact to read

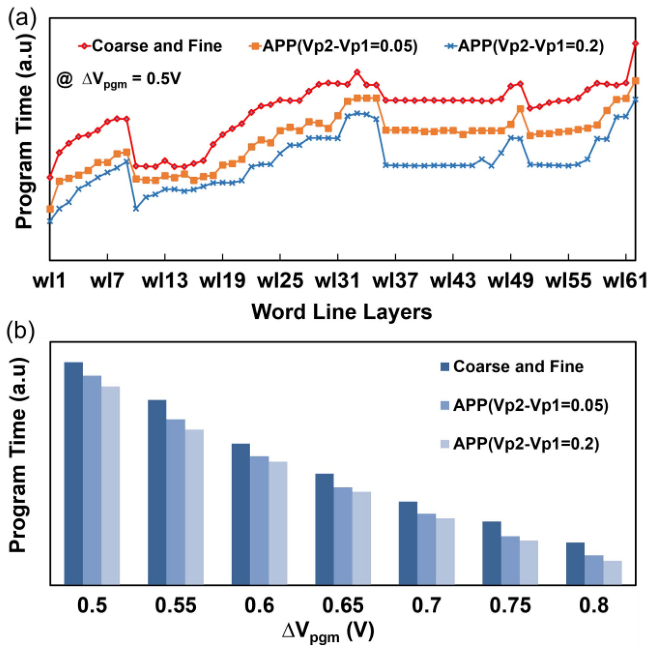

FIGURE 7. (a) Different WLs' program time of three different conditions with incremental step programming voltage equaling to 0.5V (ΔV_{pgm} = 0.5V). (b) Averaged WL program time of three different conditions with different ΔV_{pgm}.

window margin. The target verify positions of P1-P7 states are shown by the blue dash lines in Fig. 6. Since the final V_{th} distribution of a state is affected by fast charge loss effect, program disturbance and some other factors, the focus is on the difference of the distribution width between different schemes. Taking P1 state as an example, when the ΔV_{pgm} is 0.5V, the average V_{th} shift of one step is around 0.3V, while with the APP scheme (ΔT_1=ΔT_2=1/2T_{pgm}), the V_{th} shift of fast cells is around 0.17V. Setting the middle verify voltage of each state to be about 0.15V (with a margin of 20mV) less than the target level, we should see the distribution width reduced by about 0.13V (0.3V-0.17V). As shown in Fig. 6, the final state distribution width of the APP scheme is about 0.09V narrower than general ISPP scheme, which meets our expectation if some noise is considered. On average, the APP scheme reduces V_{th} distribution width by about 15% compared with general ISPP scheme while keeping similar to coarse and fine scheme. If the general ISPP scheme wants to obtain a V_{th} distribution similar to APP scheme, it is necessary to use a programming step voltage of around 0.35V, while the APP scheme is 0.5V. Accordingly, APP scheme can save program time by about 21% compared with general ISPP scheme. In addition, the larger difference between V_{p2} and V_{p1} does not widen the V_{th} distribution, since the voltage of bitline is high enough to exclude the programming effect during V_{p2} program phases for the fast cells.

The program time of two different APP cases compared with coarse and fine scheme are shown in Fig. 7. The embedded programming tests are repeated for 50 times and the program time is averaged finally. Fig. 7(a) shows the program time of different WLs with three different conditions with the incremental step programming voltage equaling to 0.5V (ΔV_{pgm} = 0.5V). It shows that almost for all WLs, the program time with APP scheme is shorter than coarse and fine programming scheme, though different WLs have some variations as the channel profiles of the 64-layer WLs vary. As shown in Fig. 4(c), the ΔV_{th} of WL31 of different programming pulse width is smaller on average than other WLs, the embedded results also show that WL31 need more time to program. For WLs around WL43, the APP scheme with 0.2V difference between ΔV_{p2} and ΔV_{p1} (ΔV_p) can save the program performance by around 7% with no loss of read window margin, while saving 3.5% with ΔV_p equaling to 0.05V. For these WLs, APP scheme can speed up the slow cells and reduce both program and verify loops. While for WLs such as WL13, APP scheme mainly reduces the verify loops as it is not sufficient to reduce the final program loops. Fig. 7(b) shows the averaged program time of all WLs while using different ΔV_{pgm} under three different conditions. It shows that with different incremental step voltages, the APP scheme also has better program performance invariably. However, when ΔV_{pgm} is larger than 0.7V, the read window margin of both coarse and fine scheme and APP scheme will lose. Taking into account other noises and disturbances, APP scheme with the condition that the programming step voltage is 0.55V, and the difference between ΔV_{p2} and ΔV_{p1} is 0.2V, can have an appropriate read window margin and save 5% performance on average of the whole WLs.

IV. CONCLUSION

In this work, the relationship between ΔV_{th} and programming pulse width was first investigated and analyzed by using single program pulse operation. On this basis, the proposed APP scheme were tested and compared with other ISPP schemes both with single state programming test and one–step TLC embedded programming test. From the experimental results, it shows that APP scheme could not only narrow the cell V_{th} distribution, but also improve the program performance.

REFERENCES

[1] D. Kang *et al.*, "7.1 256Gb 3b/cell V-NAND flash memory with 48 stacked WL layers," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2016, pp. 130–131, doi: 10.1109/ISSCC.2016.7417941.

[2] H. Maejima *et al.*, "A 512Gb 3b/Cell 3D flash memory on a 96-word-line-layer technology," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2018, pp. 336–338, doi: 10.1109/ISSCC.2018.8310321.

[3] S. Lee *et al.*, "A 1Tb 4b/cell 64-stacked-WL 3D NAND flash memory with 12MB/s program throughput," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2018, pp. 340–342, doi: 10.1109/ISSCC.2018.8310323.

[4] N. Shibata *et al.*, "13.1 A 1.33Tb 4-bit/Cell 3D-flash memory on a 96-word-line-layer technology," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2019, pp. 210–212, doi: 10.1109/ISSCC.2019.8662443.

[5] R. Micheloni and L. Crippa, and A. Marelli, "XLC storage," in *Inside NAND Flash Memories*. New York, NY, USA: Springer-Verlag, 2010, pp. 455–456.

[6] W. Hou *et al.*, “Investigation of program noise in charge trap based 3D NAND flash memory,” *IEEE Electron Device Lett.*, vol. 41, no. 1, pp. 30–33, Jan. 2020, doi: 10.1109/LED.2019.2954621.

[7] Y. Zhang, L. Jin, X. Zou, H. Liu, A. Zhang, and Z. Huo, “A novel programming scheme for program disturbance optimization in 3-D NAND flash memory,” *IEEE Electron Device Lett.*, vol. 39, no. 7, pp. 959–962, Jul. 2018, doi: 10.1109/LED.2018.2844404.

[8] K.-D. Suh *et al.*, “A 3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme,” *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, Nov. 1995, doi: 10.1109/4.475701.

[9] R. Micheloni, L. Crippa, and A. Marelli, “MLC storage,” in *Inside NAND Flash Memories*. New York, NY, USA: Springer-Verlag, 2010, pp. 285–289.

[10] G. G. Marotta *et al.*, “A 3bit/cell 32Gb NAND flash memory at 34nm with 6MB/s program throughput and with dynamic 2b/cell blocks configuration mode for a program throughput increase up to 13MB/s,” in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2010, pp. 444–445, doi: 10.1109/ISSCC.2010.5433949.

[11] W. Chen, H. Lue, Y. Hsiao, T. Hsu, X. Lin, and C. Lu, “Charge storage efficiency (CSE) effect in modeling the incremental step pulse programming (ISPP) in charge-trapping 3D NAND flash devices,” in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Washington, DC, USA, 2015, pp. 5.5.1–5.5.4, doi: 10.1109/IEDM.2015.7409635.

[12] J. Ko *et al.*, “Comparative study of WL driving method for high-capacity NAND flash memory,” in *Proc. Int. Conf. Electron. Inf. Commun. (ICEIC)*, Da Nang, Vietnam, 2016, pp. 1–4, doi: 10.1109/ELINFOCOM.2016.7562960.

[13] R. Cernea *et al.*, “A 34MB/s-program-throughput 16Gb MLC NAND with all-bitline architecture in 56nm,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, USA, 2008, pp. 420–624, doi: 10.1109/ISSCC.2008.4523236.

[14] Y. Simon, “Unleashing 3D NAND’s potential with an innovative architecture,” presented at the flash memory summit, Aug. 2018. [Online]. Available: http://www.ymtc.com/index.php?s=/cms/cate/69.html

[15] M. Jeong *et al.*, “Analysis of random telegraph noise and low frequency noise properties in 3-d stacked NAND flash memory with tube-type poly-Si channel structure,” in *Proc. Symp. VLSI Technol. (VLSIT)*, Honolulu, HI, USA, 2012, pp. 55–56, doi: 10.1109/VLSIT.2012.6242458.