MULTILEVEL CELL STORAGE IN FLASH MEMORY & DIFFERENT SENSING MECHANISMS

JAMES SKELLY

ECG721: MEMORY CIRCUIT DESIGN

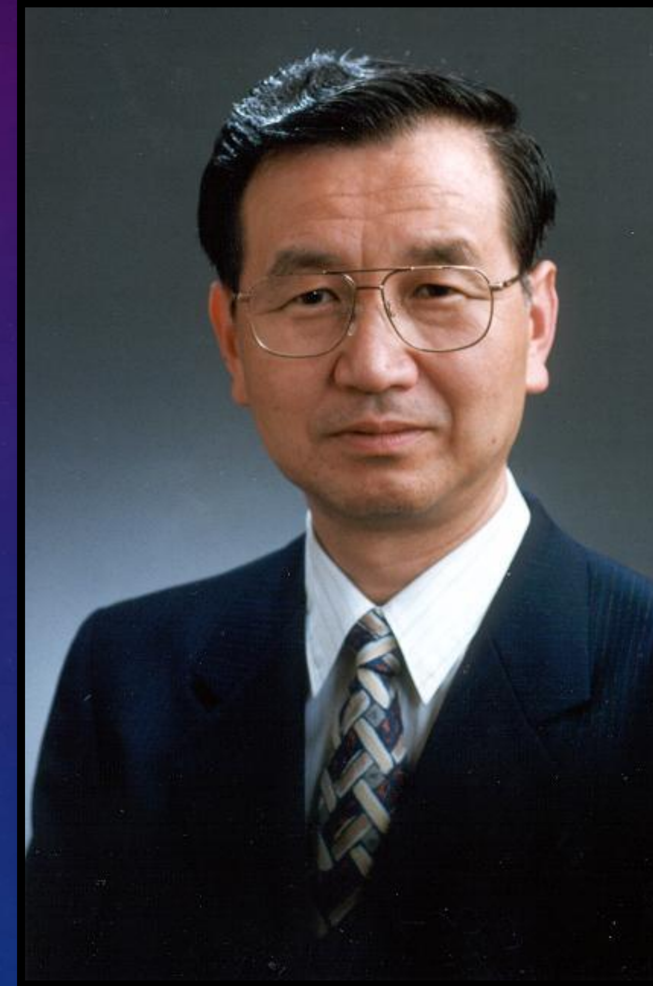
DR. R. JACOB BAKER

PART 1: INTRODUCTION TO FLASH MEMORY

- In this section, the concept of flash memory is introduced and discussed. Some points of emphasis in the introductory portion are as follows:
    - The invention and evolution of flash memory technology
    - Flash memory vs. floating gate memory
    - Origin of the name (flash)
    - Write operations
        - Channel Hot-Electron injection
        - Fowler-Nordheim Tunneling
    - Read operations and sensing

ABOUT FLASH MEMORY

- Flash memory was first invented in the 1980s by a Japanese electrical engineer named Fujio Masuoka while he was working for Toshiba.
- The late twentieth and early twenty-first century has seen enormous advancements in flash technologies with the arrival of multi-level cell technology in 1996.
- In the past eleven years, companies such as Toshiba and SanDisk have successfully produced flash memories with up to four bits stored in a single cell.
- Today, flash memory is found in a variety of devices, such as cameras, SSDs (solid-state drives), and USB flash drives.

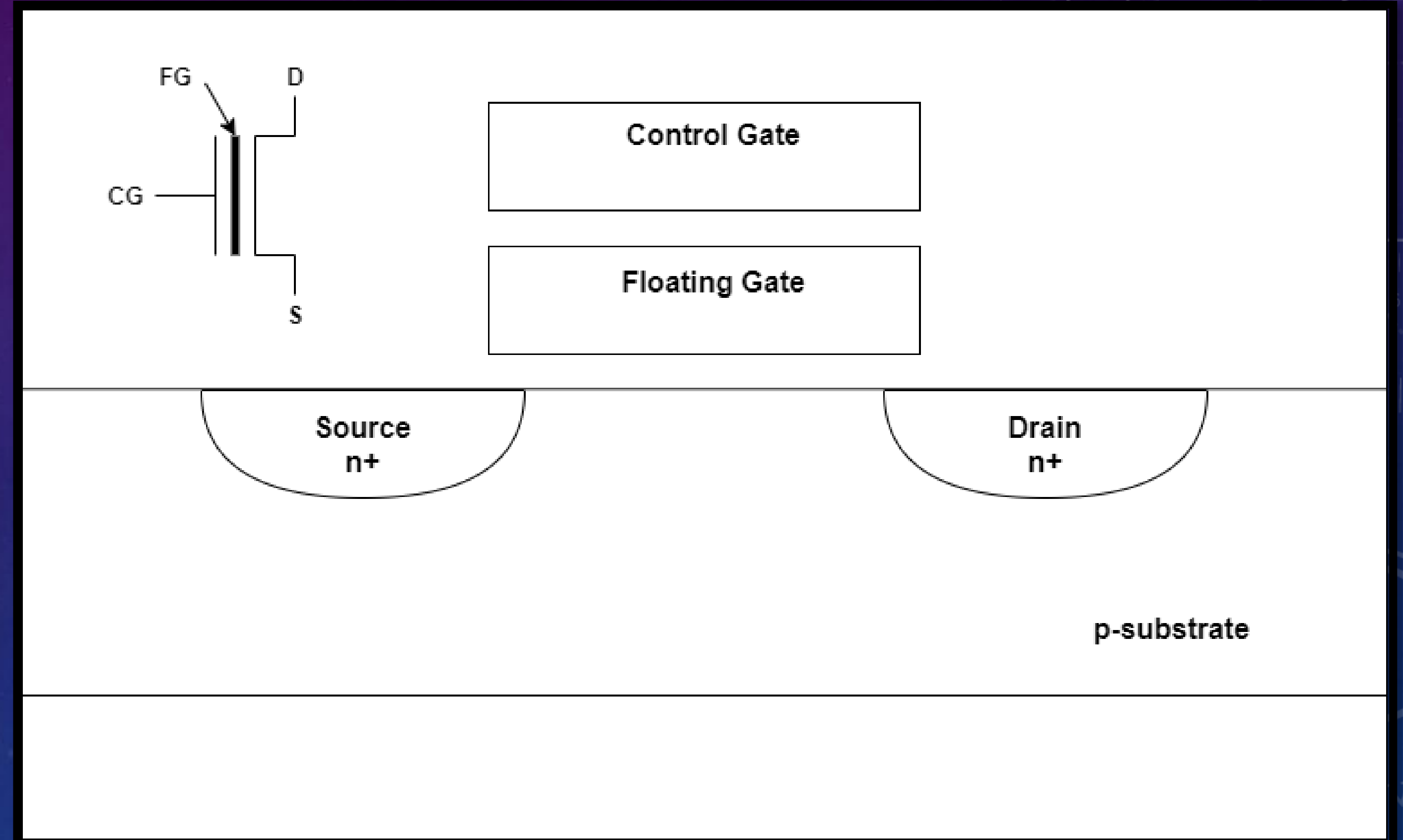


https://ethw.org/Fujio_Masuoka [3]

(Figure 1.1) Fujio Masuoka, credited with the invention of flash memory in the 1980s.

ABOUT FLASH MEMORY

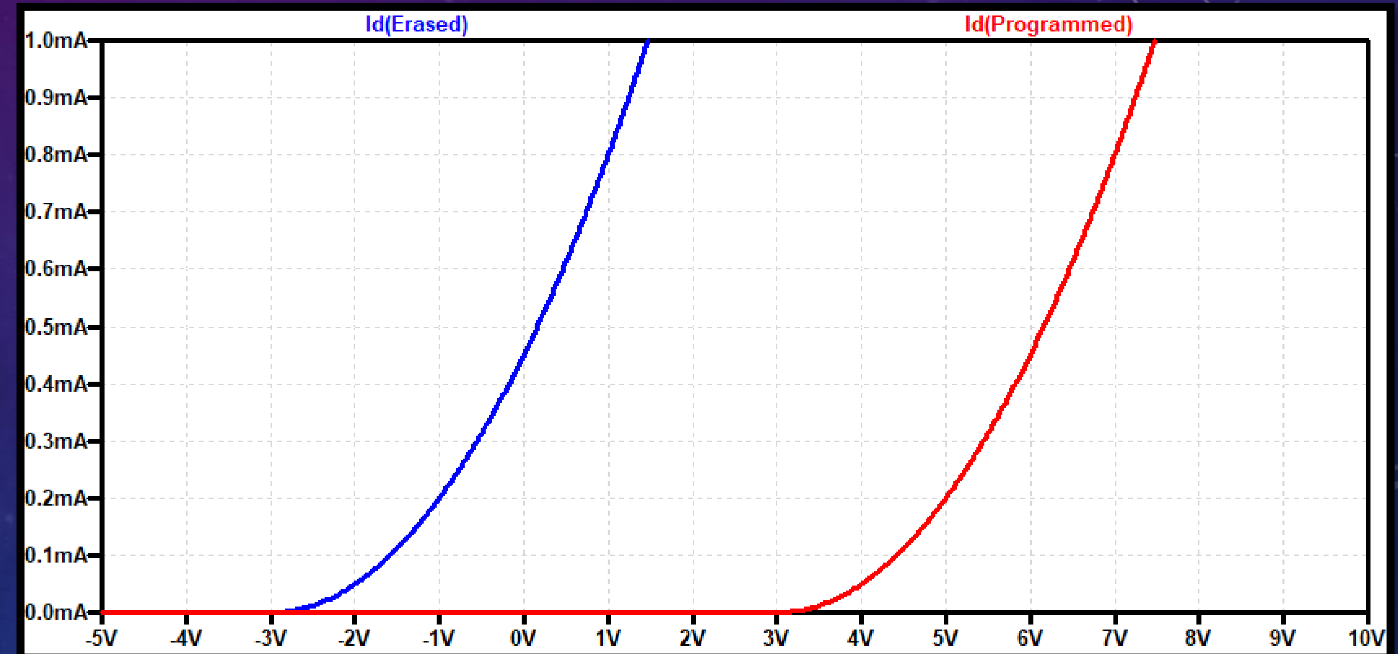
- **Flash memory** (also called EEPROM, meaning Electrically Erasable Programmable Read Only Memory) is a type of floating gate memory that can be both electrically programmed and erased.
- Floating gate memory is memory in which electrons are trapped on or pulled off a “floating” piece of polysilicon (the floating gate of a floating gate MOSFET) in order to change the threshold voltage, and ultimately, the state, of the cell. The state of a cell (programmed or erased) is determined by sensing the drain current of the MOSFET when applying a read voltage to the control gate (connected to the word line).
- Since EEPROM is oxymoronic, a different name (flash) was derived from the contrast of faster erasing speeds to a more primitive method of erasing, a slow process which required exposure to UV light, allowing charge to leak off the floating gate over time and move the cell into an erased state.



(Figure 1.2) Symbol and cross-sectional view of a typical floating gate MOSFET.

FLASH OPERATION AND WRITING TO MEMORY

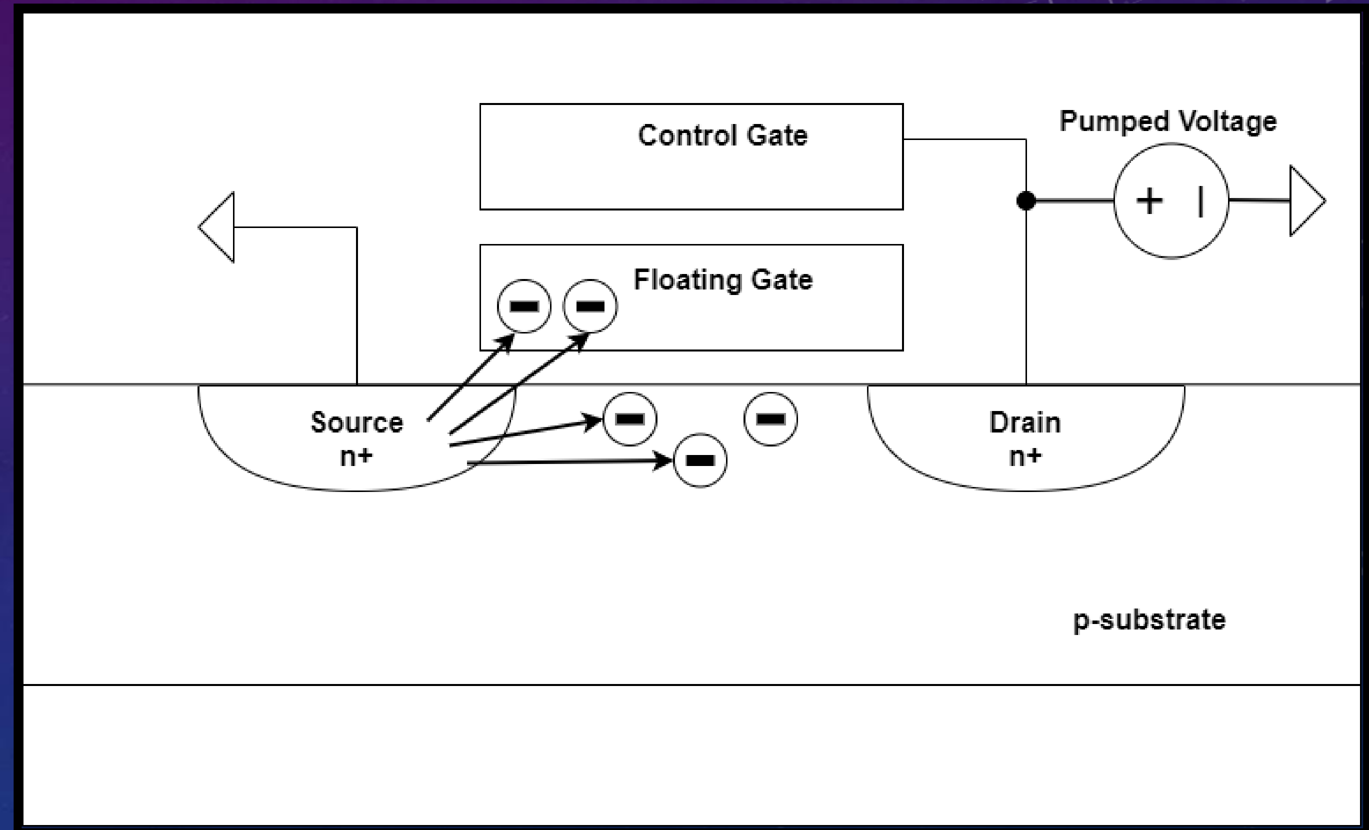
- A typical (binary) flash memory cell consists of a single floating gate MOSFET. As mentioned previously, the state of the cell (programmed, “logic 0”, or erased, “logic 1”) is dependent upon the threshold voltage of the floating gate device.
- This means that in order to change the state of or “write to” a memory cell, the threshold voltage of the device needs to be changed.
- The two main approaches used to change the threshold voltage of a floating gate device are:
    - Channel Hot-Electron (CHE) injection for programming.
    - Fowler-Nordheim Tunneling (FNT) for programming and/or erasing.



(Figure 1.3) Drain current vs. VGS in programmed and erased states of a flash memory cell.

CHANNEL HOT ELECTRON (CHE) INJECTION: PROGRAMMING

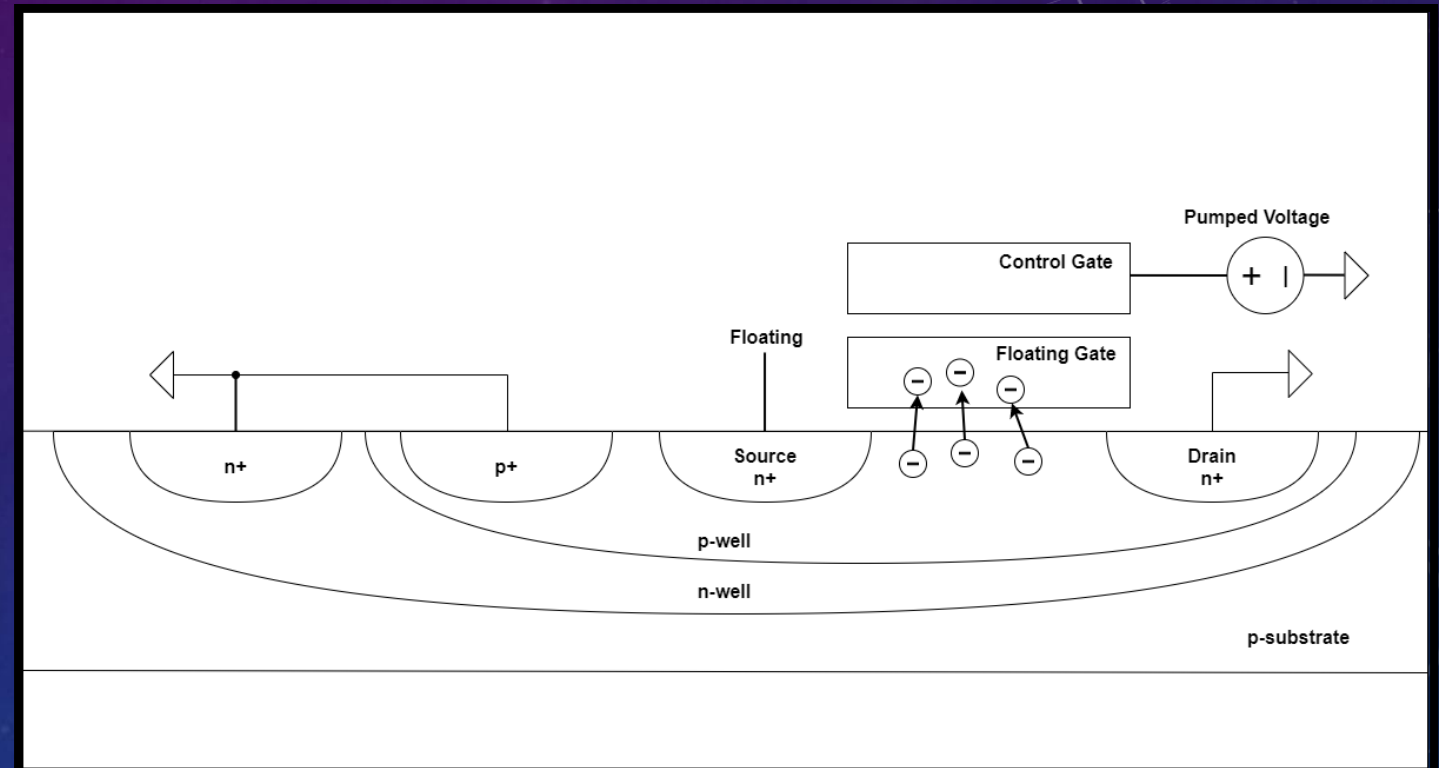
- CHE is a method of **programming** performed by applying a high voltage (much higher than VDD) to both the drain and control gate of the floating gate MOSFET and grounding the source so that **hot electrons** flow in the channel beneath the floating gate.
- Since there is a large potential difference between the source and drain of the device, the “hot electrons” that flow in the channel are highly energized. The large potential on the gate causes some of the electrons to be “injected” through the oxide (between the floating gate and the channel) and wind up trapped on the floating gate.
- This buildup of negative charge on the floating gate requires a higher potential difference for a channel to form between the source and drain of the device. This means that the threshold voltage of the device has increased, or the device is **programmed.**



(Figure 1.4) Channel Hot-Electron injection to program a cell in a flash memory.

FOWLER-NORDHEIM TUNNELING (FNT): PROGRAMMING

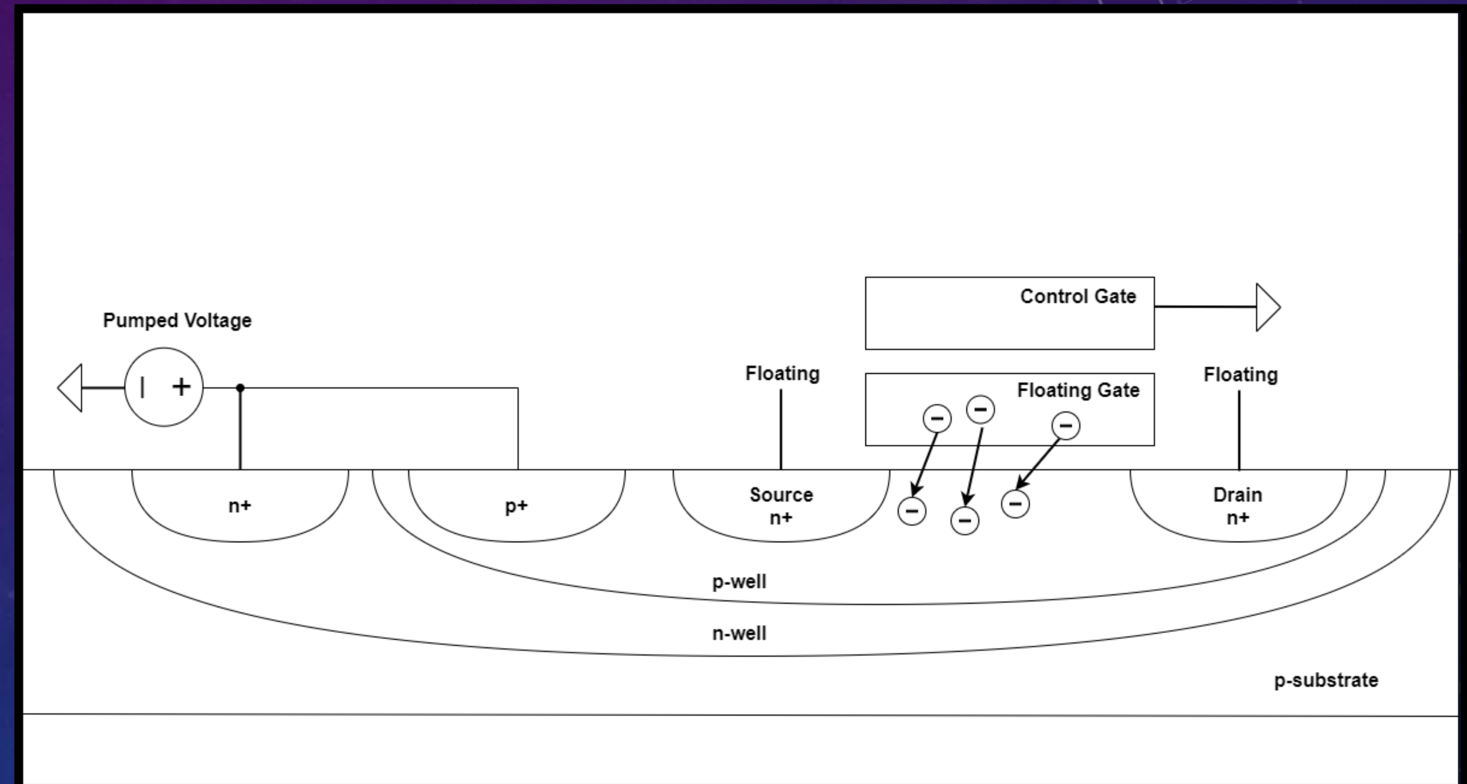
- FNT can be used for both programming and erasing a flash memory cell. Note that a double well process is necessary in a flash memory because FNT erasing varies the NMOS body potential. A double well is used to avoid forward-biasing of parasitic p-n junctions and to avoid changing the substrate potential on chip during FNT erasing.
- Programming using FNT is performed by driving the control gate to a pumped voltage, letting the source float, and grounding the drain and well contacts. The large potential on the control gate attracts electrons and causes some of them to tunnel through the oxide and build up on the floating gate.
- Again, this buildup of negative charge on the floating gate requires a higher potential difference for a channel to form between the source and drain of the device. Thus, the threshold voltage of the device has increased, or the device is **programmed.**



(Figure 1.5) Programming using Fowler-Nordheim Tunneling in a flash memory.

FOWLER-NORDHEIM TUNNELING (FNT): ERASING

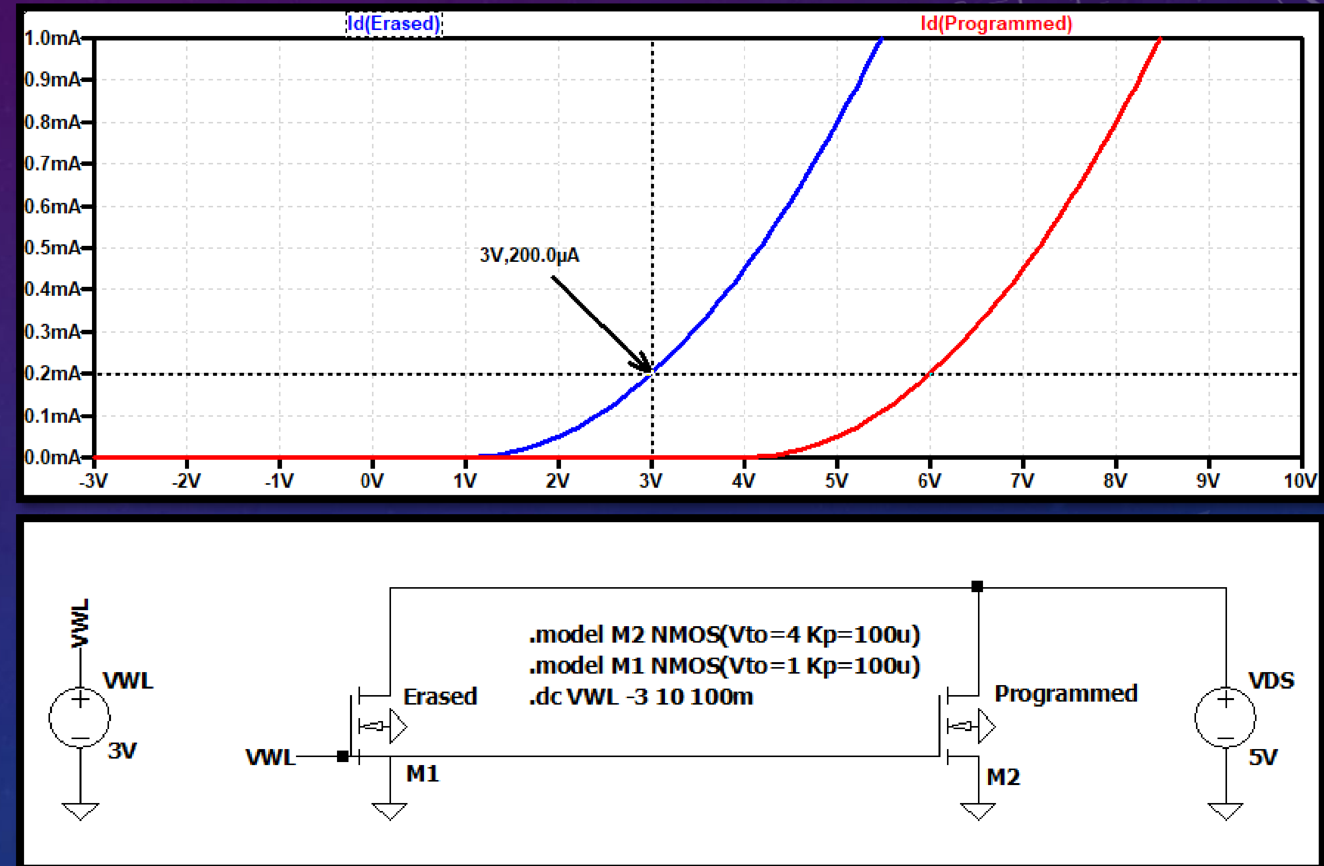
- Erasing using FNT is performed by driving the wells to a pumped voltage, letting the source and drain float, and grounding the control gate. The large potential on the well contacts pulls electrons off the floating gate and causes them to tunnel through the oxide toward the well contacts.
- The need for a double well can be seen clearly from this diagram. If a high potential were applied directly to the NMOS body and there were no n-well, the substrate all over the chip would be connected to a high potential. The source and drain are left floating to avoid forward-biasing of p-n junctions between the p-well and the source/drain.
- The removal of charges from the floating gate results in a decrease in the device’s threshold voltage, leaving the cell in an **erased** state.



(Figure 1.6) Erasing using Fowler-Nordheim Tunneling in a flash memory.

READING IN FLASH MEMORY: SENSING

- Reading or **sensing** in flash memory is performed by determining the current flowing through a floating gate MOSFET.
- When a voltage is applied to the word line for a read, an erased cell, which has a lower threshold voltage, will have a much larger drain current than a programmed cell, which has a higher threshold voltage.
- Figure 1.6 shows two MOSFETs in SPICE modeled with different threshold voltages. The erased NMOS has a threshold voltage of 1V, while the programmed NMOS has a threshold voltage of 4V.
- We can see that when a word line voltage of 3V is applied, the erased device drain current (blue trace) is 200µA, and the programmed device has no current flowing through it. This difference in current can be **sensed** so that we can read out the contents of the cell.



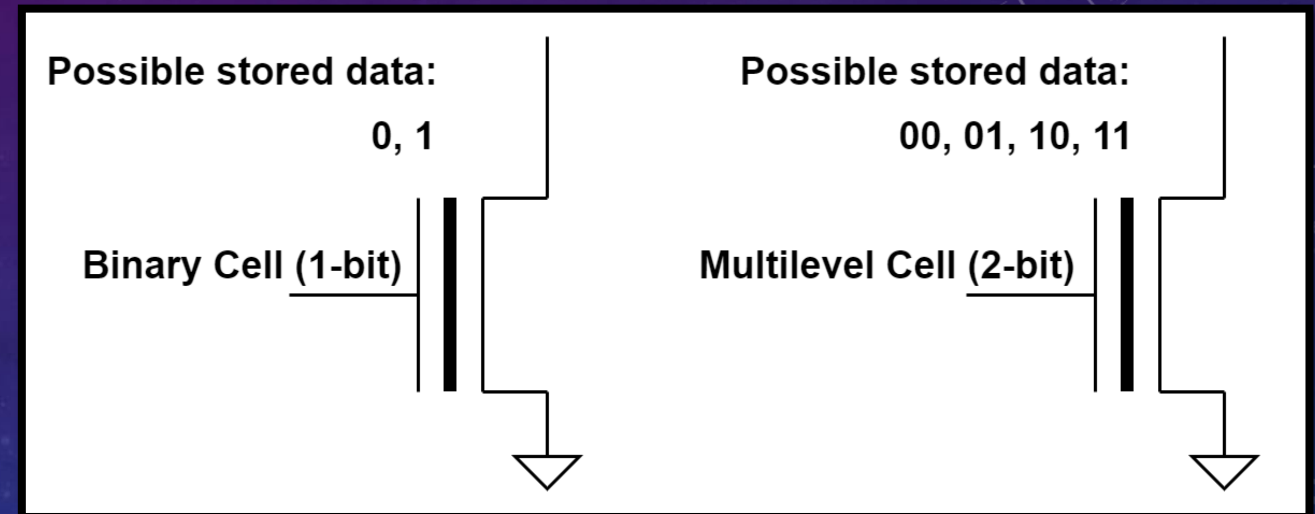
(Figure 1.7) Modeling difference in current flow in erased and programmed devices.

PART 2: MULTILEVEL CELL STORAGE IN FLASH MEMORY

- This section introduces and explains the concept of, and reason for, multi-level cell storage in flash memory. Points of emphasis in this section are as follows:
  - Operation of multi-level flash
  - Considerations for design and reduced reliability contributors
    - Total Voltage Window (TVW)
    - Programming time, accuracy
    - Charge/data retention
    - Read disturb
    - Sensing accuracy

MULTILEVEL CELL STORAGE

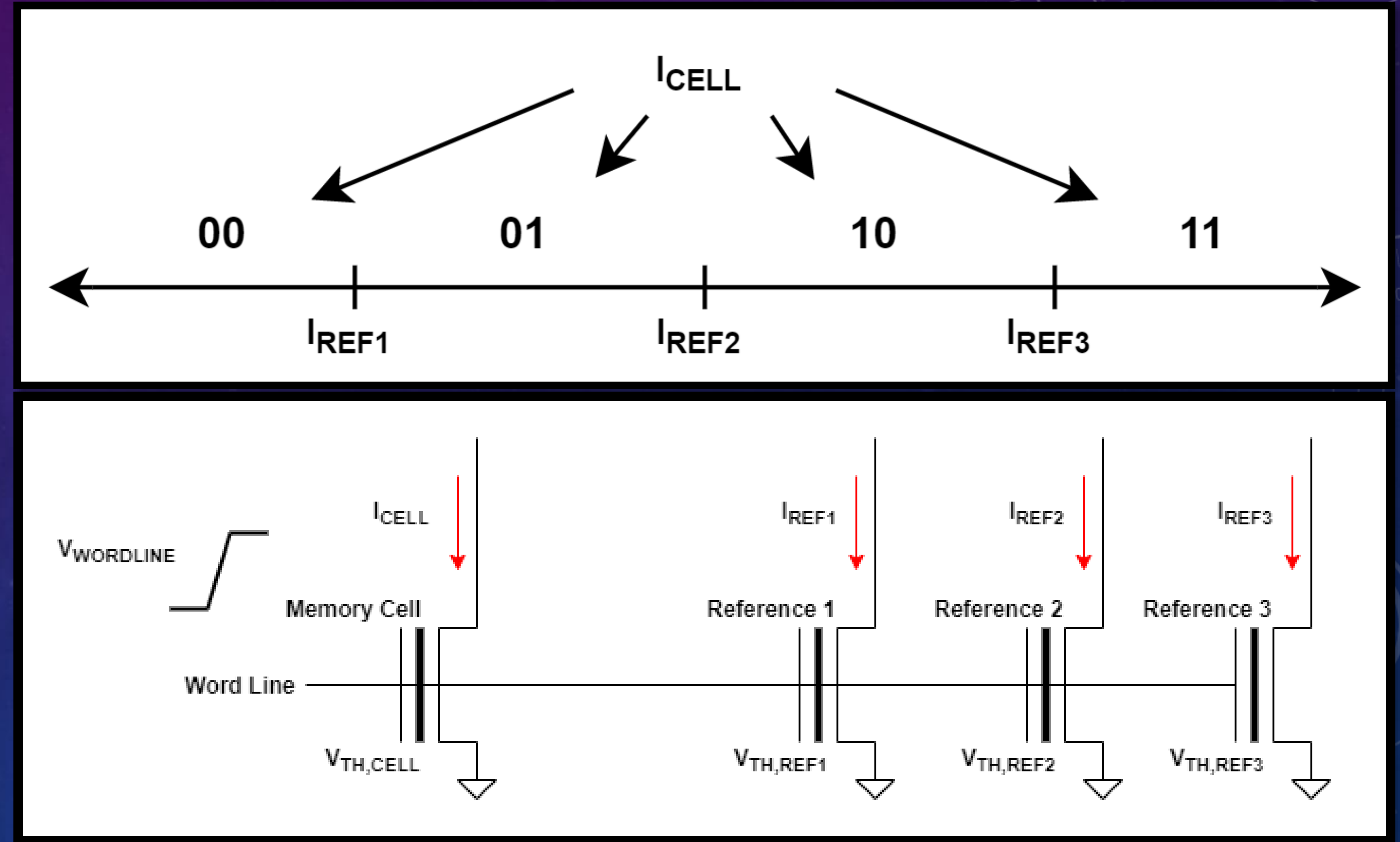
- Flash memory is priced in terms of dollars per megabyte, or $/MB. With this being the case, it is desirable to minimize the area per bit of a flash memory architecture.
- Up until now, we have only seen flash memory cells that are either programmed or erased, meaning they only store either a “logic 1” or a “logic 0” (one bit of data per cell). This is because we have only considered two possible threshold voltages for any floating gate MOSFET in the memory to have.
- If instead of only having two possible threshold voltages, we have multiple threshold voltages to compare our memory cell threshold voltage against, we can store n bits in a single cell by comparing the contents of that cell to the contents of 2^{n-1} reference cells.
- A flash memory cell that can store multiple bits is called a **multi-level cell.** To store multiple, say two, bits in a single cell nearly cuts the cost per bit in half of a flash memory without any change in process or device sizing.



(Figure 2.1) The concept of identical cells storing different amounts of data.

STORING MULTIPLE BITS IN A SINGLE CELL

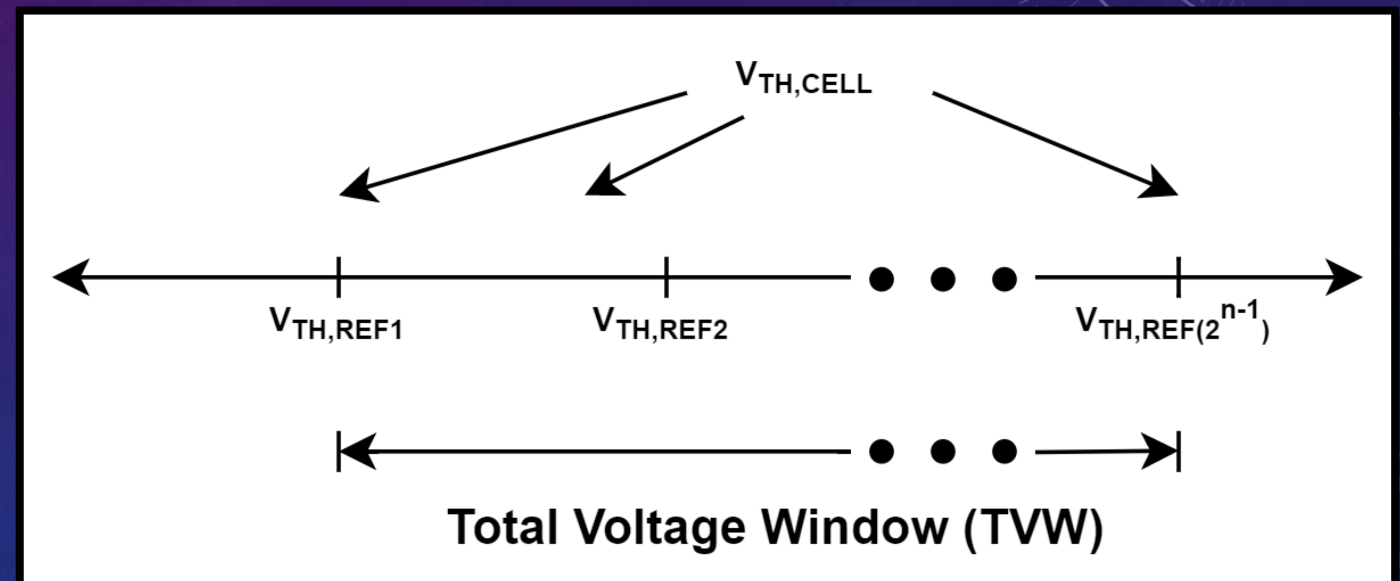
- In order to reduce layout area (and effectively also reduce cost per bit) of a flash memory architecture, multiple bits can be stored in a single cell.
- From the figure, we see that if the memory cell current is less than the current in each of the three reference cells, and using proper sensing, we would read out the bit sequence “00”. If the cell current is greater than each of the three reference currents, we would read out the bit sequence “11”. The bit sequences “01” and “10” are read out if the cell current magnitude falls between two of the reference current magnitudes.
- Note that it is desirable for the threshold voltage of the memory cell to differ from all three of the reference cell threshold voltages so that there is always an adequate difference between cell current and reference current during a sense.



(Figure 2.2) Showing how two bits can be stored in a single memory cell.

CONSIDERATIONS FOR MULTILEVEL FLASH: TOTAL VOLTAGE WINDOW (TVW)

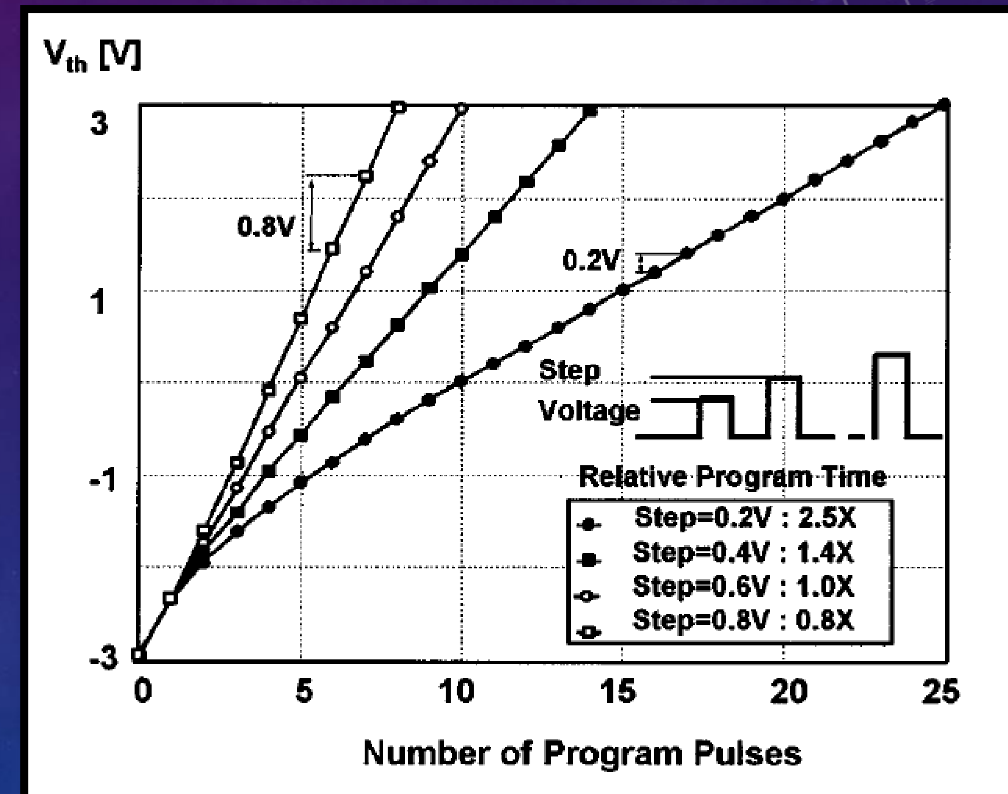
- For every n bits stored in a flash memory cell, 2^{n-1} distinct and precisely programmed reference cells are required with adequately and equally spaced out threshold voltages. This produces 2^n intervals between reference threshold voltages.
- As processes continue to get smaller, and with low-voltage operation being desirable, a multilevel flash memory property known as the total voltage window (TVW) gets smaller.
- The total voltage window is defined as the difference between the highest and lowest value of threshold voltage allowable in a multilevel flash memory.
- With on-chip noise and power consumption at the forefront of all design considerations, it is important for flash memory designers to consider maximizing the TVW while minimizing power consumption.



(Figure 2.3) Total voltage window for an n-bit multilevel flash memory.

CONSIDERATIONS FOR MULTILEVEL FLASH: PROGRAMMING TIME, ACCURACY

- Generally, programming in flash memory is done using what is known as a “program and verify” or P&V algorithm.
- P&V is a method of precise programming which uses a sequence of small programming steps to increase device threshold voltage iteratively, where each programming step is followed by a read operation to determine if another programming step is necessary.
- The accuracy of the programming depends on the size of the P&V steps (smaller steps results in greater accuracy).
- A major design consideration or tradeoff is that of programming speed for accuracy. Slower programming speeds result in great accuracy, while less accurate program operations are much faster and require far less P&V steps.

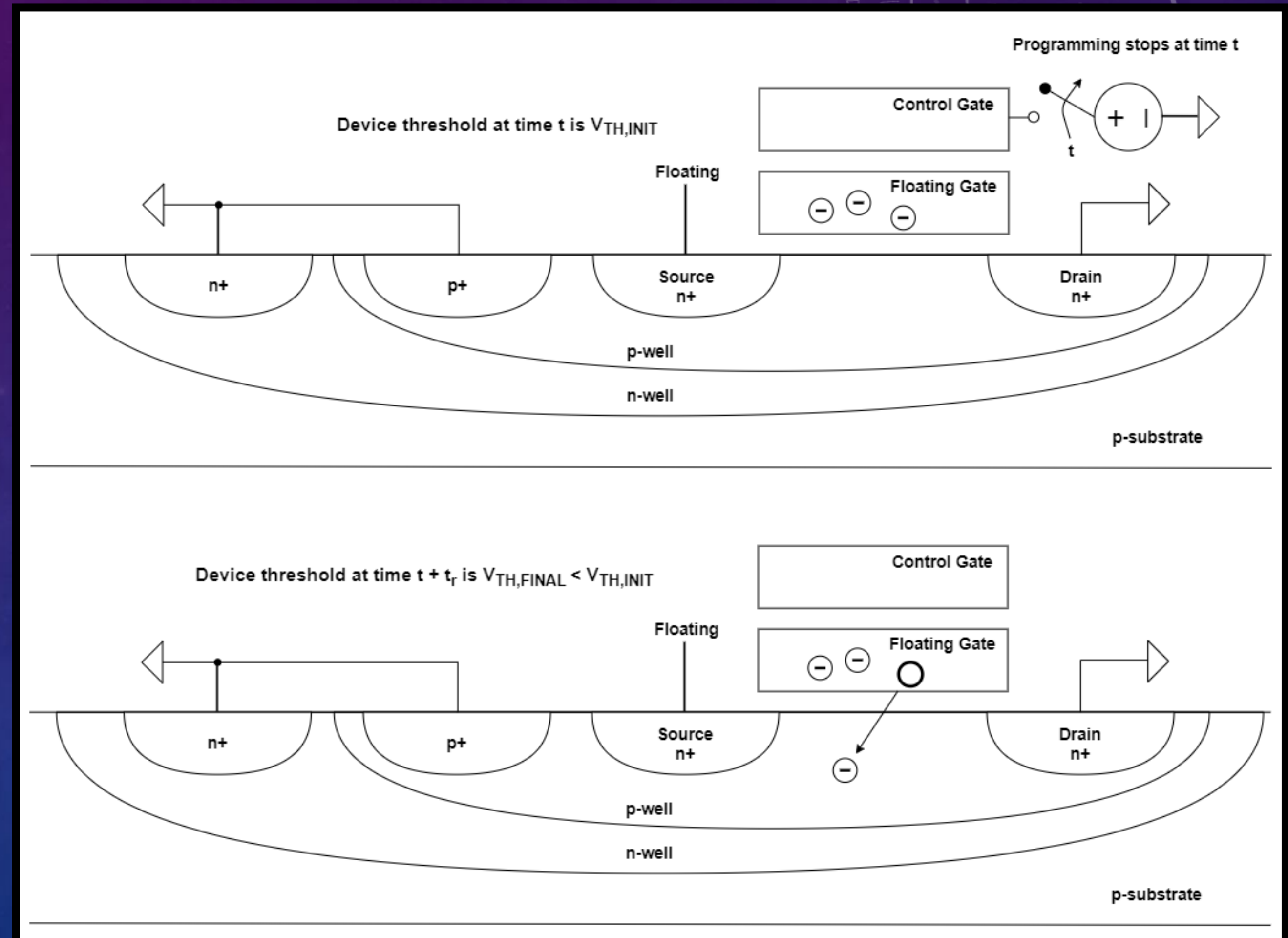


Tae-Sung Jung et al., "A 117-mm/sup 2/ 3.3-V only 128-Mb multilevel NAND flash memory for mass storage applications," in IEEE Journal of Solid-State Circuits, vol. 31, no. 11, pp. 1575-1583, Nov. 1996. [5]

(Figure 2.4) Plot from IEEE JSSC article showing how program time increases with accuracy.

CONSIDERATIONS FOR MULTILEVEL FLASH: DATA RETENTION

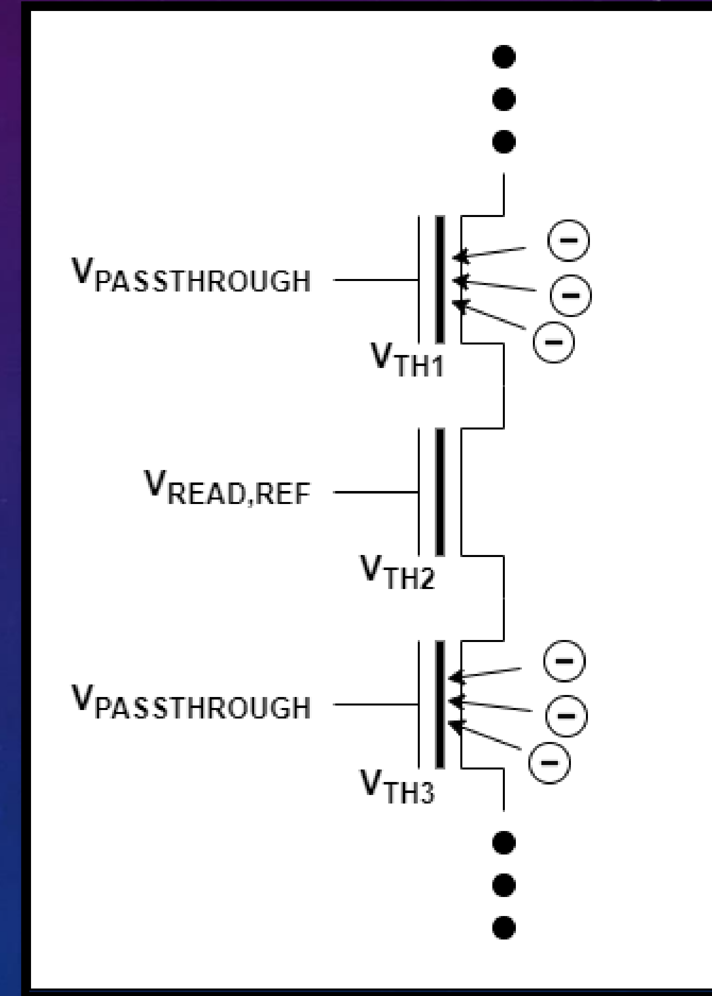
- The number one source of errors in flash memory is data retention errors caused by the leakage of charge off the floating gate over time, altering a device's threshold voltage.
- A flash cell can be characterized at any point by its *retention age*, which is the amount of time since the cell was last programmed.
- In the figure to the right, let the retention age of the cell be time t_r. At time t, the device threshold voltage is at the desired value, since programming has just finished.
- At time $t+t_r$, the device threshold voltage has decreased from $V_{TH,INIT}$ to $V_{TH,FINAL}$, a new value less than that of the initial threshold voltage value. In cases where the TVW is small, this is a huge issue and causes many errors if not resolved.



(Figure 2.5) Showing how leakage of charge changes device threshold voltage over time.

CONSIDERATIONS FOR MULTILEVEL FLASH: READ DISTURB

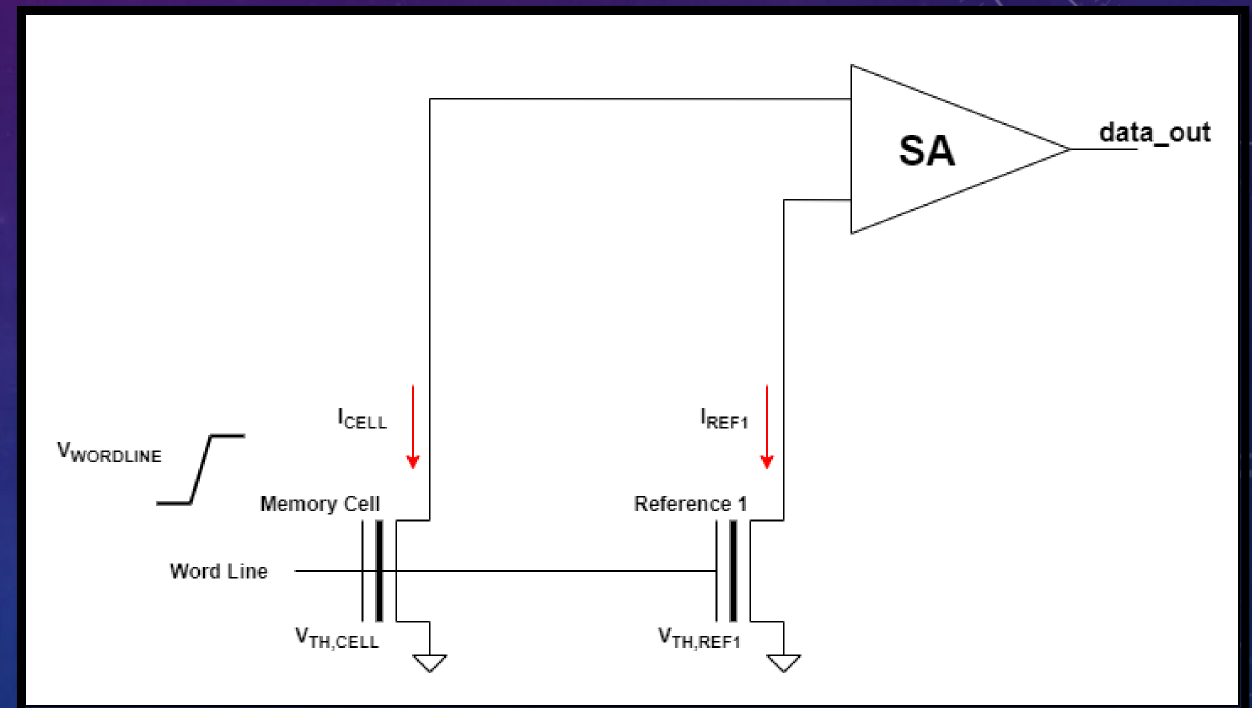
- Another source of errors, a phenomenon which causes reduced reliability in flash memory, is known as read disturb. Read disturb is when a read operation of a particular row causes alterations in the threshold voltages of a different (unread) row in the same memory block.
- The NAND flash architecture (cells are connected in series) is especially susceptible to read disturb errors. During a read, all cells in a block that are not being read need to be turned on so that the output signal can propagate out of the block. The voltage applied to keep all these transistors on for a read, the pass-through voltage, needs to be higher than any threshold voltage to assure that all devices are on.
- This voltage on the gate causes unwanted tunneling which can change the threshold or "disturb" the cell contents of cells that are not being read out. This effect becomes greater and greater as transistor size gets smaller.



(Figure 2.6) Read disturb occurring during a read operation in a NAND flash block.

CONSIDERATIONS FOR MULTILEVEL FLASH: SENSE ACCURACY

- The key to reliable flash memory is accurate sensing. As the number of bits in a cell increases, the accuracy of the sense becomes more and more important.
- We discussed previously how the number of segments in the TVW increases with increase in the number of bits stored in a multi-level cell. This means that for the same TVW, as n increases, the size of the segments in the TVW decrease significantly.
- Though other contributors mentioned previously play a huge role in the flash memory reliability, an accurate sensing circuit or **sense amplifier (SA)** is an essential component in a reliable flash memory architecture.



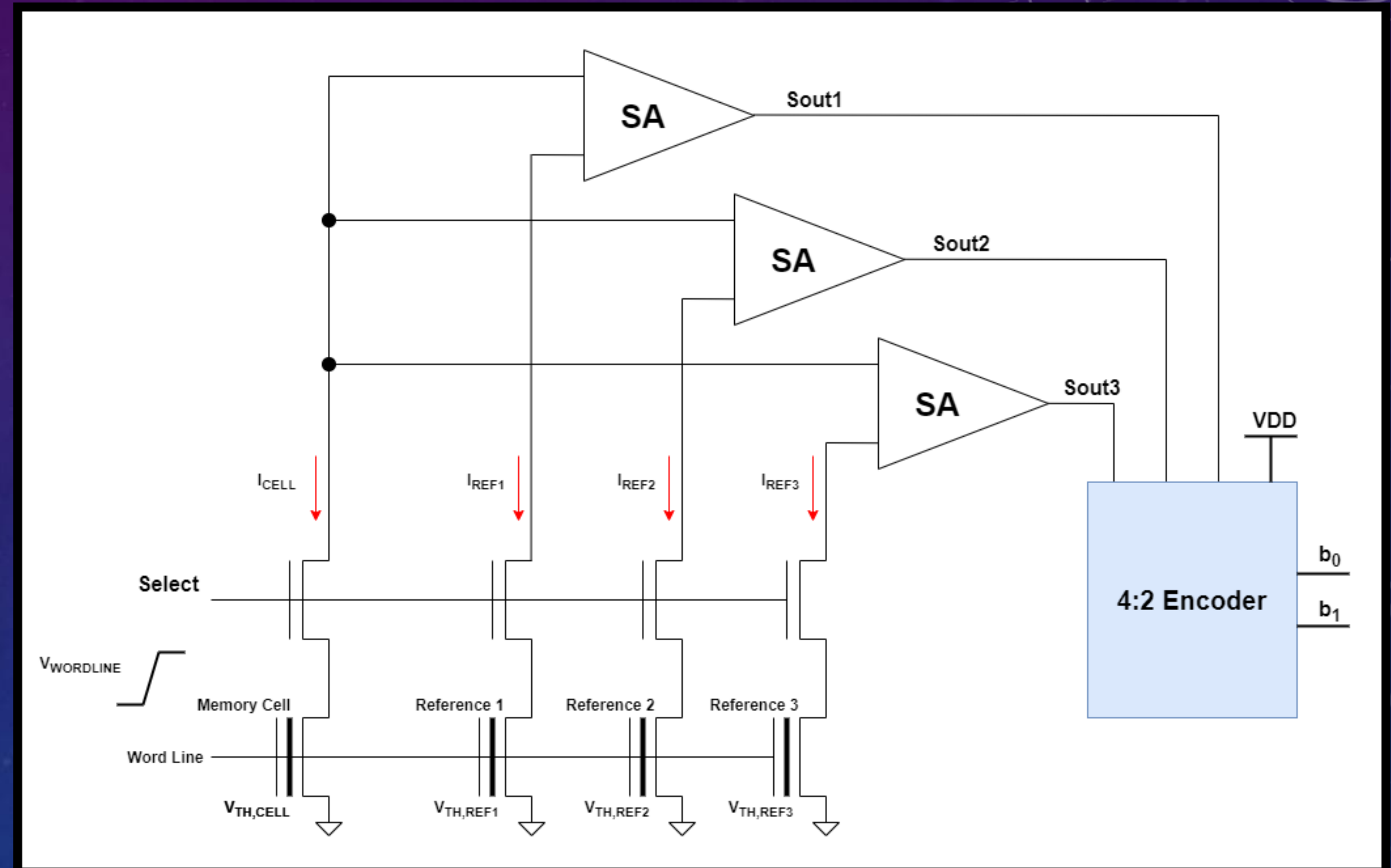
(Figure 2.7) Sensing to determine the contents of a flash memory cell.

PART 3: SENSING MECHANISMS IN FLASH MEMORY

- This section discusses different sensing mechanisms that can be used in multi-level cell (MLC) flash memory and compares each mechanism to the others in terms of design tradeoffs. Types of sensing that will be discussed include:
  - Parallel sensing
  - Serial sensing
  - Serial-parallel sensing

PARALLEL SENSING

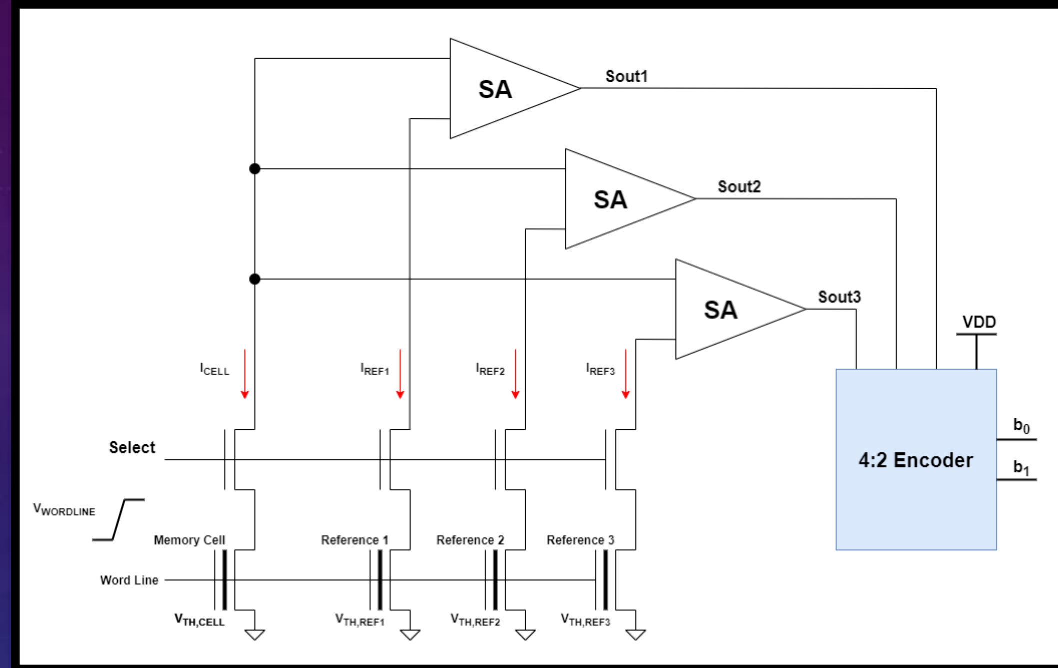
- The first method of sensing in multi-level flash to be discussed is parallel sensing, where the n-bit memory cell and 2^{n-1} reference cell contents are compared simultaneously. For the examples in this section, let n=2.
- The currents through each device can be converted to voltage through I-V converters (not pictured here) and these voltages can be compared by sense amplifiers.
- Each sense amplifier will have an output of either a “logic 0” or “logic 1” depending on which cell current (or voltage after conversion) was greater, that of the memory cell or the associated reference cell.
- The outputs of the sense amplifiers are fed into a 4:2 encoder, where one of the four inputs is a “don’t care” and is tied high in this example. The outputs of the encoder are the memory cell contents. A truth table is on the page to follow.



(Figure 3.1) Parallel sensing circuit for MLC flash where a cell stores two bits of data.

PARALLEL SENSING

- We can observe the operation of the circuit from the truth table inputs and output, and by comparing the truth table to the schematic of the parallel sensing circuit.
- When the memory cell threshold voltage is less than the threshold voltage of all three references, the output of the system is “11”.
- When the memory cell threshold voltage is greater than the threshold voltage of all three references, the output of the system is “00”.
- When the memory cell threshold voltage is less than that of reference 3, but greater than that of references 1 and 2, the output of the system is “01”.
- When the memory cell threshold voltage is less than that of references 2 and 3, but greater than that of reference 1, the output of the system is “10”.

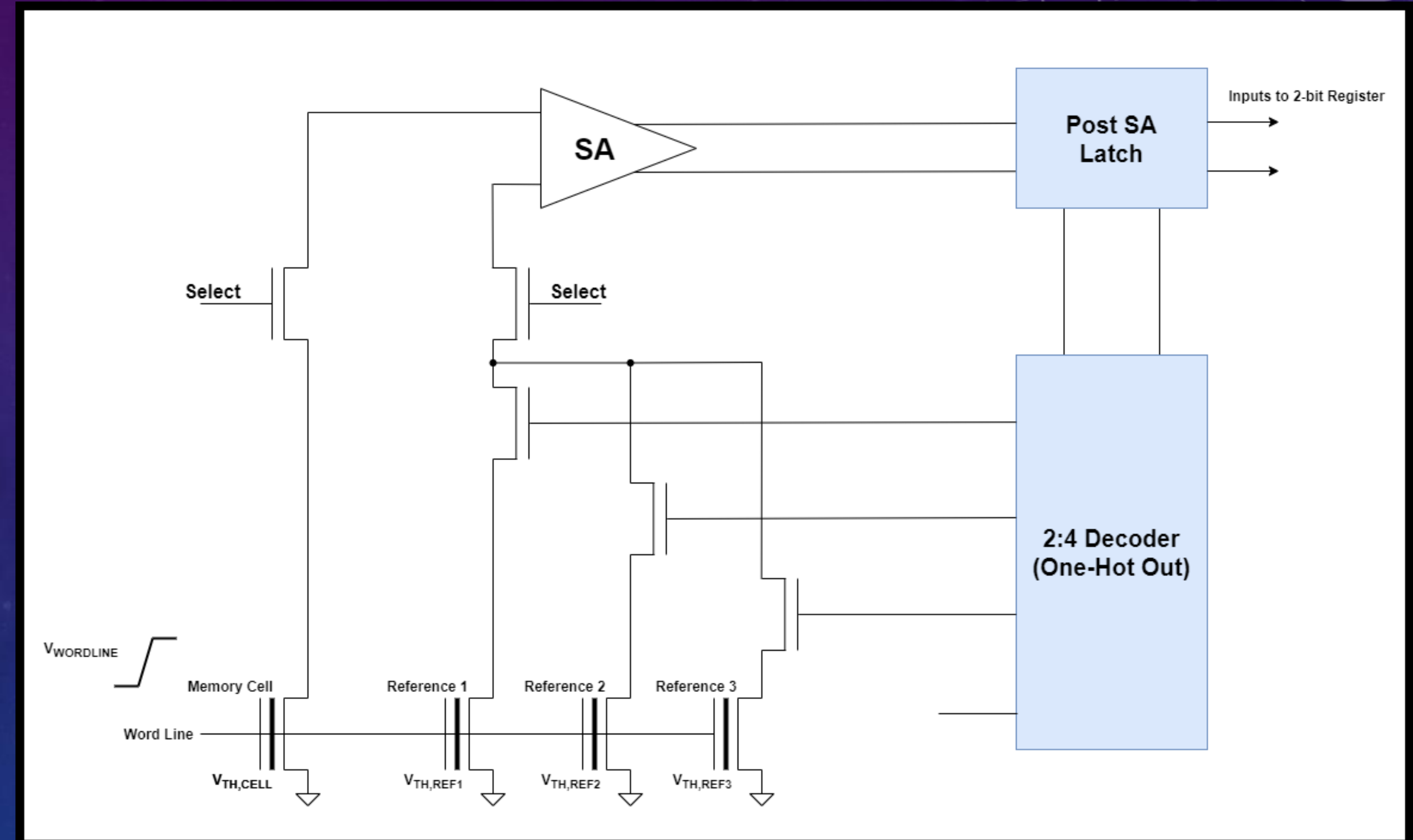


Sout0 (DNE)	Sout1	Sout2	Sout3		b_0	b_1
X	0	0	0		0	0
X	0	0	1		0	1
X	0	1	1		1	0
X	1	1	1		1	1

(Figure 3.2) Parallel sensing circuit and its truth table.

SERIAL SENSING

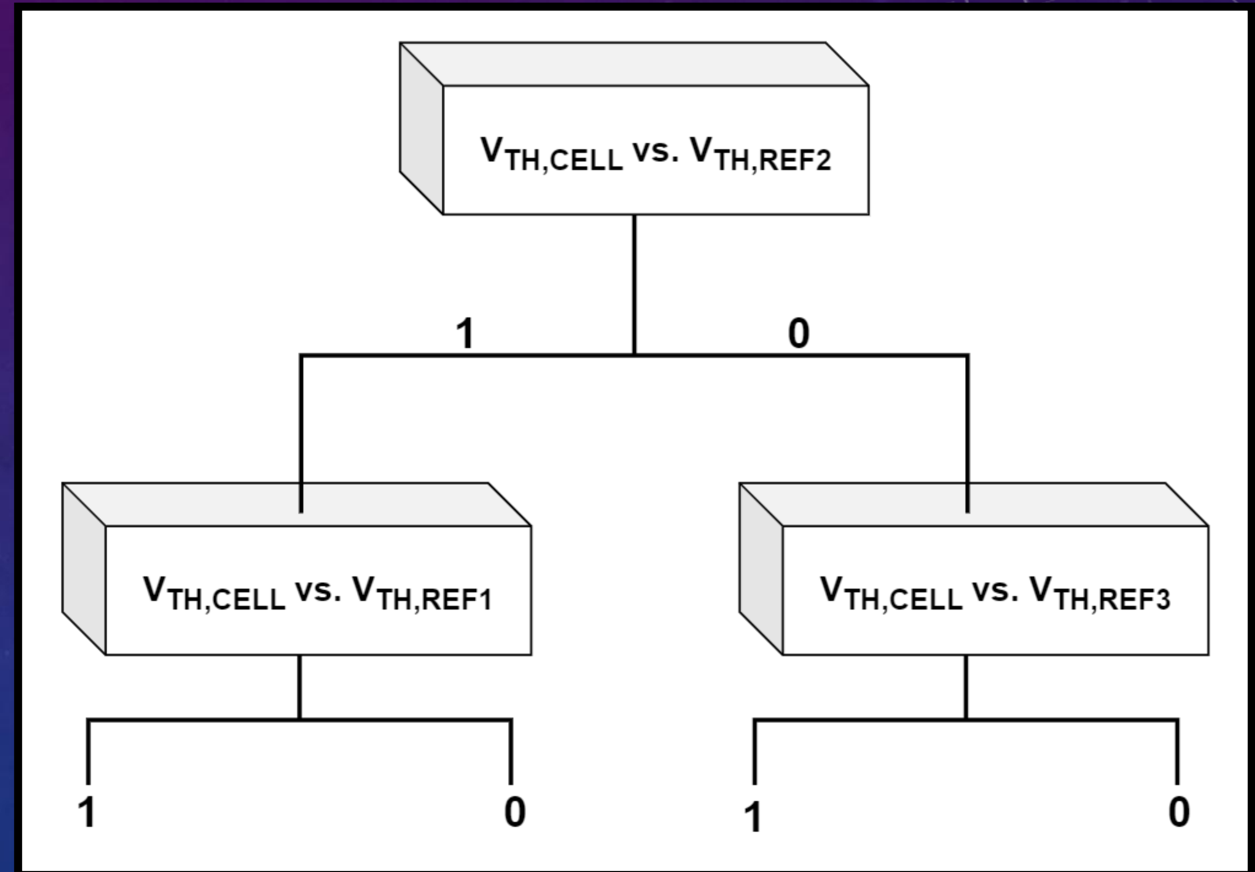
- An alternative method of sensing multi-level cell flash memory is serial sensing, where all sensing is done using a single sense amplifier.
- The serial sensing circuit starts as the one-hot decoder selects the midpoint reference device, which in this case is reference 2. The midpoint reference is the threshold voltage that is in the middle of the TVW.
- It is important to note that the threshold of reference 1 is greater than that of reference 2, and the threshold of reference 3 is less than that of reference 2.
- By selecting the midpoint device first, the output of the sense amp yields the MSB of the memory cell. Values from the post SA latch are fed into the decoder telling it which cell to select next.



(Figure 3.3) Serial sensing circuit for MLC flash where a cell stores two bits of data.

SERIAL SENSING

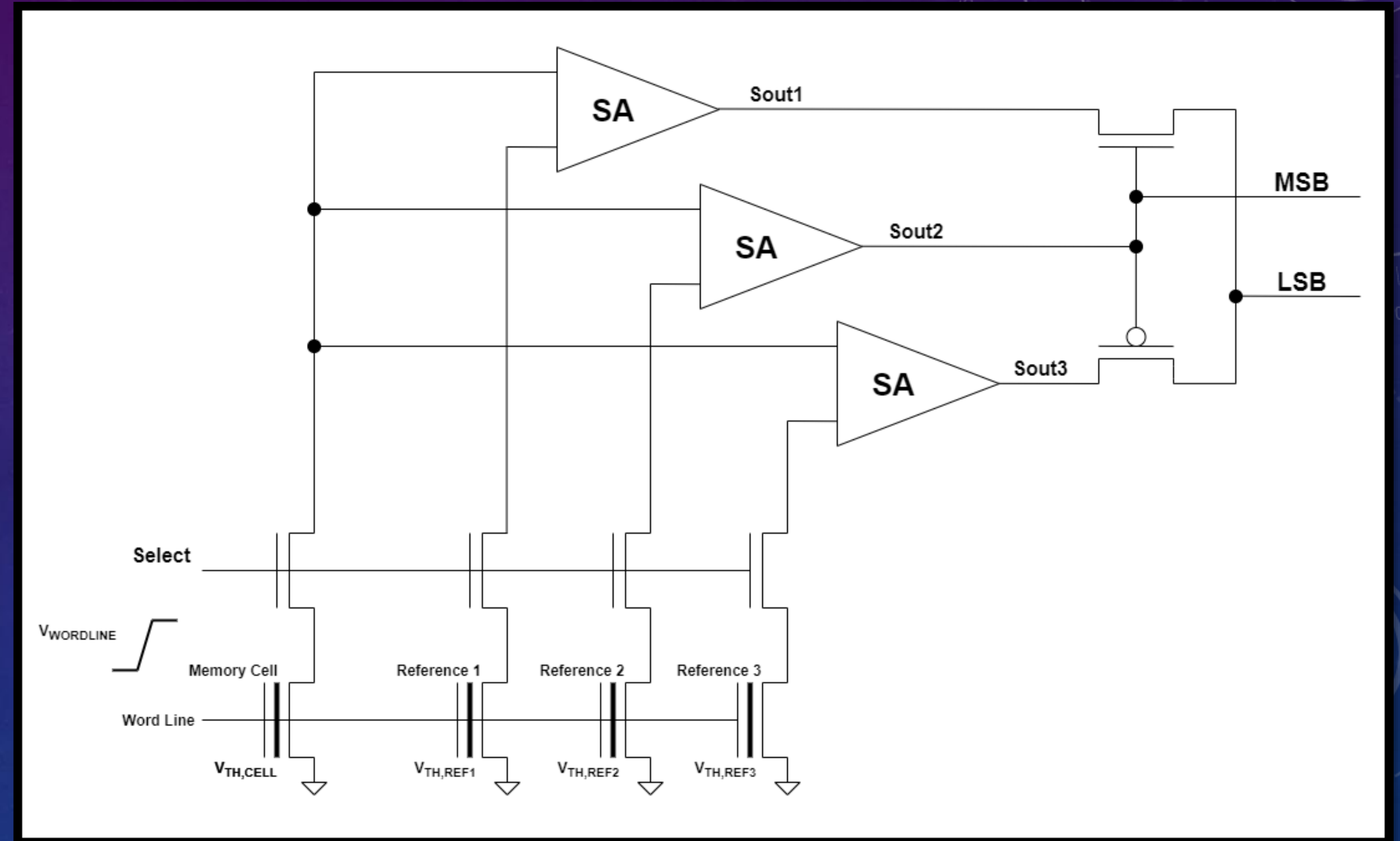
- The first comparison is that of the memory cell threshold to the second reference device threshold. The cell with the lesser threshold will result in a larger drain current and determines the output of the sense amp. This sense amp output becomes the MSB and determines which reference cell will be compared next. The MSB is sent to a hold register (see figure 3.3).
- The next reference cell is selected by the decoder and the process repeats, except this time, the output of the sense amp yields the LSB. This value is also sent to a hold register, where the complete two-bit value of the memory cell is now stored.
- The important takeaway in serial sensing is the fact that the midpoint reference is compared first for the following reason. If the memory cell threshold is greater than the midpoint reference, it is also greater than the lower reference. If it is less than the midpoint reference, it is also less than the higher reference. This limits the maximum number of cycles through the system to two and optimizes sensing speed.



(Figure 3.4) Flow diagram showing how serial sensing results in a two-bit readout.

SERIAL-PARALLEL SENSING

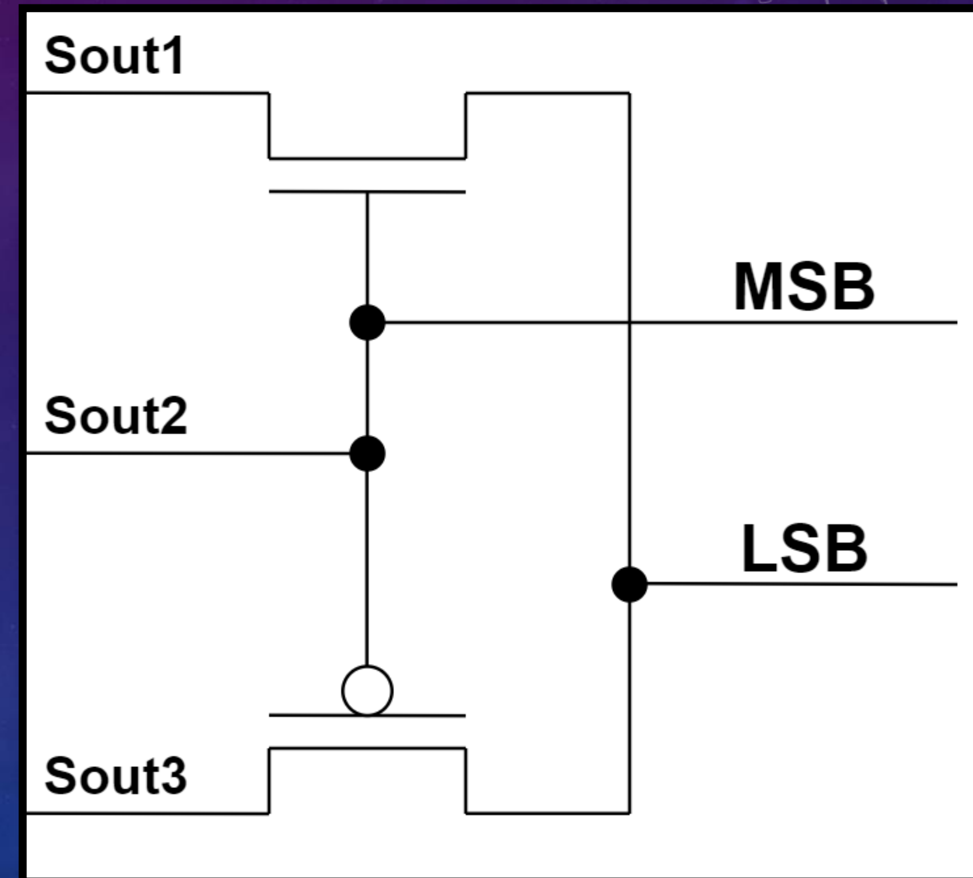
- The serial-parallel mechanism for sensing in multi-level flash memory aims to maintain the benefits of each topology while eliminating the drawbacks. Right away, we spot similarities to both previous methods of sensing.
- We can see that, similarly to the parallel method, we have three sense amps, and all three references are compared with the memory cell simultaneously.
- We can also see that similarly to the serial method, the output of the second sense amplifier is the MSB and determines which sense amp output will yield the LSB using a pair of complementary MOSFETs.



(Figure 3.5) Serial-parallel sensing mechanism with two-bit readout.

SERIAL-PARALLEL SENSING

- Let's examine the operation of the output stage of the serial-parallel sensing circuit. The node connected to the gates of both MOSFETs is the output of the second sense amp and is taken as the MSB from the memory cell.
- The MSB will either turn on the NMOS device, letting the output of the first sense amp through, or turn on the PMOS, letting the output of the second sense amp through. Whichever value passes through is the LSB of the memory cell.
- It is important to recall that NMOS devices are good at passing "logic 0" and bad at passing "logic 1", while PMOS devices are good at passing "logic 1" and bad at passing "logic 0." For this reason, extra circuitry (likely an output buffer or string of inverters) is necessary on the outputs of the serial-parallel sensing circuit to restore full logic levels for the MSB and LSB of the data during a readout.



(Figure 3.6) Taking a closer look at the output stage of the serial-parallel sensing circuit.

SENSING CIRCUITS OVERVIEW AND COMPARISON

Topology:	Parallel	Serial	Serial-Parallel
Benefits:	• Fast sensing; sensing time is equal to the time of one sense operation plus encode time.	• Use of a single sense amp makes for desirably smaller layout area.	• Faster sensing, smaller layout area, and lower power consumption than parallel sensing topology.
Drawbacks:	• Need for one sense amp per reference cell results in very large layout area.	• Relatively slow sensing; sensing time of at least twice that of the parallel architecture.	• Larger layout area than serial topology, need to restore full logic levels at outputs.
Comments:	• Tradeoff of speed for larger layout area is an important consideration when using parallel sensing.	• It is important to note that the need for more peripheral circuitry increases layout area.	• Seemingly the optimal topology for sensing of three aforementioned topologies.

(Table 3.1) Benefits and drawbacks of the three MLC flash memory sensing mechanisms discussed.

SUMMARY

- Flash memory is a type of floating gate memory, also known as EEPROM, because it can be electrically programmed and erased.
- Flash memory cells can be binary or multilevel, with multilevel cell storage being more desirable, since it allows for more data to be stored in roughly the same amount of layout area.
- As the number of bits stored in a single cell increases, reliability of the memory can be compromised by contributors such as programming inaccuracy, data retention issues, the read disturb phenomenon, and sense inaccuracy.
- There are different methods that can be used to sense in a MLC flash memory:
  - Parallel sensing
  - Serial sensing
  - Serial-parallel sensing
- The sensing mechanism which optimizes speed and layout area is the serial-parallel sensing scheme, which aims to maintain the benefits of both the serial and parallel schemes while minimizing the drawbacks of each.

REFERENCES

[1] A 125MHz Burst Mode 0.18um 128Mbit 2 Bits Der Cell Flash Memory, H.A. 'Caatro, S. Monaaa, 'M. Goldman, B. Srinivasan, T. Biessie, R. Haque, S. Chandramouli, I. Sharif. K. he, M. Ishac. G. Christensen, B. Li, T. Ly, K. Pan, M. Srwarc, G. Vadlamudi, K. Ramamunhi, S. Balasuhrahmanyam, S. Saini, M. Dayley. A. Rahman, D. Elmhursl, V. Viajedor, R. Rajagopal, R. Zeng, A. Sayed, F. Marvin, B. I'athak, J. Kreifels, R. Melcher, R. Nambiar, M. Khandaker, Q. Ngo, K. Augustine, R. Padilla, J. Keilman and C. Haid, Intel Corp., Folsom, CA

[2] Baker, R J. *CMOS : circuit design, layout, and simulation*. Hoboken, N.J: IEEE Press/Wiley, 2010. Print.

[3] “Fujio Masuoka.” *ethw.org/Fujio_Masuoka*. Engineering and Technology History Wiki, 25 Jan. 2016. Web.

[4] M. Grossi, M. Lanzoni and B. Ricco, "Program schemes for multilevel flash memories," in Proceedings of the IEEE, vol. 91, no. 4, pp. 594-601, April 2003.

[5] Tae-Sung Jung et al., "A 117-mm/sup 2/ 3.3-V only 128-Mb multilevel NAND flash memory for mass storage applications," in IEEE Journal of Solid-State Circuits, vol. 31, no. 11, pp. 1575-1583, Nov. 1996.

[6] Wikipedia contributors. "Flash memory." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 18 Apr. 2020. Web.4 May. 2020

[7] X. Gao, Y. Wang, Y. He, G. Zhang and X. Zhang, "An innovative sensing architecture for multilevel Flash memory," 2012 IEEE 11th International Conference on Solid-State and Integrated Circuit Technology, Xi'an, 2012, pp. 1-3.

[8] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai and O. Mutlu, "Data retention in MLC NAND flash memory: Characterization, optimization, and recovery," 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), Burlingame, CA, 2015, pp. 551-563.

[9] Y. Cai, Y. Luo, S. Ghose and O. Mutlu, "Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery," 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Rio de Janeiro, 2015, pp. 438-449.