Order this document
by AN1837/D

AN1837

Non-Volatile Memory Technology Overview

By Stephen Ledford
Non-Volatile Memory Technology Center
Austin, Texas

Introduction

Today's microcontroller applications are more sophisticated with application code requirements increasing in size. Code development represents a significant investment in time and resources for initial development and debug of the application. Since the code for the application is embedded in on-chip memory, this represents some challenges, especially with the debug cycle in which code bugs are identified, fixes generated, and ultimate implementation back into the memory of the microcontroller. With conventional mask ROM (read-only memory), this debug cycle could represent many weeks from the point of identification of the code issue to validation of the code fix with a new product.

In light of the debug cycle time issue with on-chip mask ROM, an emulation and development option for a microcontroller is required to enable reduction in the time required for a debug learning cycle. While RAM (random-access memory) could provide a quick and simple means of enabling quick code updates, it is not a complete replacement for mask ROM since the contents are lost as soon as power is removed.

It is this primary differentiation with the volatility of RAM and the non-volatility of ROM and the other classes of memories discussed here that

the term non-volatile memory (NVM) has become so important in the world of microcontrollers. In past product generations, the ROM memory was replaced with EPROM to enable programming of the application code, erasure with UV (ultraviolet) light, and reprogramming of the code in the debug cycle. More recent products have utilized FLASH EEPROM (electrically erasable programmable read-only memory) rather than EPROM memory to continue to improve the debug cycle even more as well as resolve some of the disadvantages presented by EPROM memory.

In addition to the use of EPROM and FLASH EEPROM memory for code development purposes, many products incorporate a smaller byte erasable EEPROM memory for use as permanent scratch pad memory, data storage, and storing unique end product characteristics, as well as many other purposes. This category of non-volatile memory offers some unique characteristics that are not usually found in FLASH EEPROM memories such as byte erasability or higher endurance characteristics.

There is a very high level of diversity in non-volatile memory application needs as is evident from the MCU products offered by Freescale. For these reasons, as well as many others, unique technologies have been developed for the major MCU (microcontroller unit) families, M68HC12, M68HC08, M68HC3xx, and MPC.

This application note describes the three major FLASH EEPROM technologies currently found in these MCUs. An explanation of the key characteristics of these memories is provided and the features that enable certain characteristics such as high density and low-power operation. Non-volatile memory also presents a broad array of new terms to describe EPROM or byte erasable EEPROM memories, and an introduction to these terms with definitions is included. Finally, EEPROM represents a similar but unique NVM category and an explanation of the basic operation and critical enabling characteristics is provided for this class of memories.

Common Terms and Definitions

The NVM landscape is extremely broad and requires that new terms be introduced to describe the behavior and characteristics of the memory as well as what differentiates one memory type from another.

ROM Read-only memory — As the name implies, the memory can only be read. The memory elements are hard coded during the wafer manufacturing process and cannot be changed or altered. This type of memory is commonly used for program code storage on a microcontroller or permanent look up tables and parameters.

EPROM Electrically programmable read-only memory — This memory type is unique when compared to ROM in that the memory is programmed electrically versus hard coding the memory elements during the wafer manufacturing process. It is similar to ROM in that after the programming event has occurred, the memory can only be read. To alter the memory, the array must first be erased by exposing the surface of the die to ultraviolet light which then permits the programming of new contents to the memory array. Since the surface of the die must be able to be exposed to UV light to perform the erase operation, a quartz windowed ceramic package must be used which can be expensive. As a result, EPROM products are sometimes packaged in a conventional plastic package which allows programming of the die but without the window to expose the surface. The unit cannot be erased and subsequently reprogrammed. In this configuration, the product is referred to as one-time programmable or OTP and offers a lower cost by sacrificing the ability to reprogram the same device.

EEPROM Electrically erasable programmable read-only memory — As the name implies, this category of memory can be electrically erased versus the EPROM operation of erasure requiring the surface of the die to be exposed to UV light. Therefore, there are no special packaging requirements of the die to take advantage of all memory features. The term EEPROM is quite commonly used by itself to refer to byte- erasable

EEPROM. The reasons for the utilization of this term in this way become apparent with the next definition.

FLASH EEPROM

This family of NVM uses the acronym EEPROM since it shares the electrical characteristics of being able to be electrically erased and programmed. However, the phrase “FLASH” has been added to describe the erase operation and distinguish it from byte-erasable EEPROM memories. The phrase FLASH refers to the manner in which an erase operation is performed. For instance, either the entire memory array or a large block of the memory is erased during one operation, which improves the throughput of the reprogramming of the memory array. There are many other differences and reasons for differentiating FLASH EEPROM from byte-erasable EEPROM, and these will be highlighted in a detailed description of each class of memory. FLASH EEPROM is commonly referred to as simply FLASH memory as a phrase that encompasses the characteristics of EEPROM while contrasting it with byte-erasable EEPROM.

Endurance

In particular, this is a critical characteristic of the EEPROM class of memories. Since the EEPROM can be electrically erased and reprogrammed, it is well suited to applications that require values to be permanently stored but yet updated on an ongoing basis in the application. As a result, the memory can be cycled between the programmed and erased data states many times. A more detailed description of the program and erase operations is provided later but these operations require high voltage to be applied to the bitcell to change data states. This high voltage degrades the electrical operation of the bitcell a small amount each time and can cumulatively reach a point after which the bitcell no longer operates properly.

Data Retention

Since a ROM is manufactured in a way that hard codes the contents of its array, the subject of data retention is not commonly applied to ROM. However, an electrically programmable memory changes data states by placing or removing charge on an electrically isolated piece of material. A detailed description of these operations is provided later. A defect in the isolating oxides surrounding the material used to store the charge

results in a leakage path for the stored charge. The bitcell data state can be changed as a result of charge loss or charge gain of the floating gate. Therefore, data retention is a parameter that defines the ability of the NVM to retain its data across the defined operating specification. Data retention applies to all classes of EPROM and EEPROM.

CHE

Channel hot electron is a commonly used programming mechanism for EPROM and some types of FLASH EEPROM. When a large bias is placed on the drain terminal of a CMOS (complementary metal-oxide semiconductor) transistor, minority carriers flow through the channel of the transistor and become heated as a result of the high electric field at the drain side of the channel which results in their energy being shifted higher. When some of the minority carriers gain enough energy, they are able to surmount the silicon dioxide energy barrier and are in turn injected over the barrier onto the floating gate of the device. One important fact to remember about CHE is that it is a one-way programming mechanism only. In other words, CHE is capable of performing the programming operation but the reverse case, erase, is not possible. A more complete description is provided in the discussion of the various memory technologies. This mechanism sometimes is also referred to as HCI or hot carrier injection.

Fowler-Nordheim Tunneling

This is an alternate form of injection for floating gate devices. The technical description of this mechanism is field assisted electron tunneling. This is different from CHE in that the mechanism is created as a result of a high electric field between the gate of the device and the source or drain. If the field is large enough, it lowers the height of the energy barrier, the silicon dioxide layer, and the electrons. Then it tunnels through the silicon dioxide and onto the floating gate. It is referred to as Fowler-Nordheim tunneling after the scientists Fowler and Nordheim who identified the case of electrons tunneling through a vacuum barrier. Lenzingler and Snow later described the case of oxide tunneling. Again, a more thorough explanation is provided in this application note as part of the discussion of the various memory technologies.

Non-Volatile Memory Operation

This section deals with the operation of the non-volatile memory (NVM).

ROM

A description of ROM is used first to lay a foundation for operation and comparison to the electrically alterable memory classes to follow. ROM can be viewed as the simplest non-volatile memory class because the memory is “programmed” during the wafer manufacturing process by taking advantage of the masking layers used during the formation of transistors. One of the most commonly used methods for performing this “programming” is utilization of the nitride layer, sometimes referred to as the active layer.

The following transistor cross sections help describe this process. These diagrams do not represent the entire wafer manufacturing process but the ones that are important to the formation of the ROM bitcell.

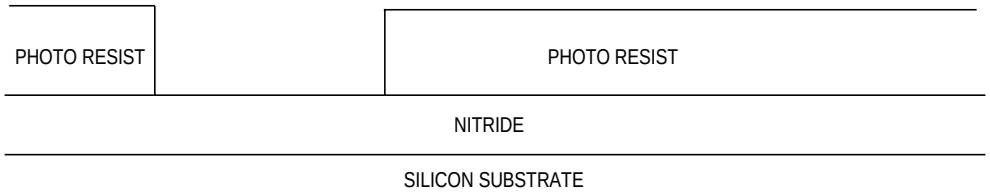

Figure 1. Nitride Photo Patterning

Early in the wafer fabrication process a layer of nitride is deposited across the surface of the wafer. A photo resist is then applied on top of the nitride and subsequently exposed in a photo-lithography process step. The result is the cross section shown in **Figure 1** where the opening in the photo resist ultimately will result in a ROM bitcell that is not “programmed” and the area where the photo resist remains, resulting in a ROM bitcell that is “programmed."

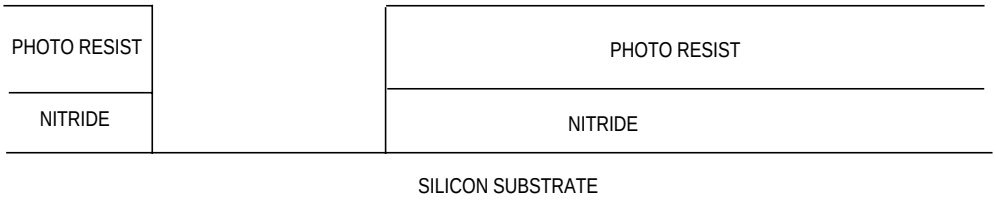

Figure 2. Nitride Etch

The next steps that take place etch the exposed nitride, leaving the silicon substrate exposed to subsequent process steps. The photo resist acts as an etch block preventing the nitride underneath the resist from being etched away.

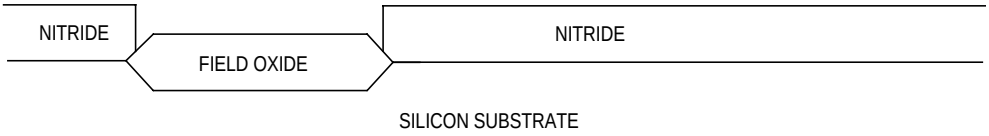

Figure 3. Photo Resist Removal and Field Oxidation

Next, the photo resist is removed, leaving the nitride exposed. The nitride is sometimes referred to as a hard mask, meaning it is a sacrificial layer used to define certain areas on the wafer. In this case, a high temperature oxidation cycle will form silicon dioxide in the regions of the wafer not covered by the nitride layer as shown in **Figure 3**. The resulting oxide is very thick and is normally used throughout the design as an isolating layer between transistors and any signal layers running across the die.

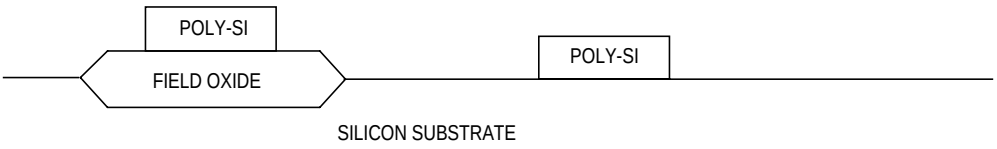

Figure 4. Nitride Removal, Gate Oxidation, and Gate Poly Formation

The cross section in **Figure 4** is the result of several process steps. The nitride has been removed and a thin layer of silicon dioxide has been grown in the regions that were under the nitride layer. This oxide growth is used as the insulating layer under the polysilicon to form the gate of a

CMOS transistor. Polysilicon is then deposited across the wafer, photo resist is applied as shown earlier with the nitride layer, the photo resist is patterned, and the extra unwanted polysilicon is etched away, leaving only the polysilicon used to form the gate of the transistor. A ROM array design has polysilicon gates in all bitcell locations as represented in this diagram. However, in the case of an unprogrammed bitcell, the polysilicon is left on top of the field oxide which results in significantly different transistor characteristics. The importance of this difference is described in the sections that follow.

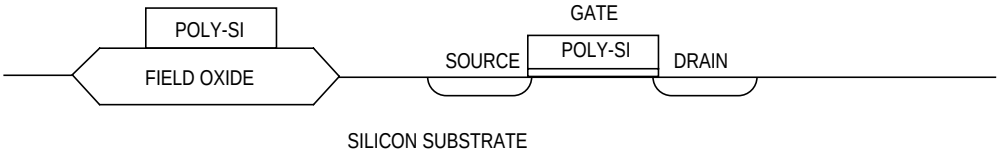

Figure 5. Source and Drain Formation

The final steps in transistor formation are the implantation of the source and drain regions. Additional wafer processing is required to define metal interconnect and routing of the transistors together. This additional processing is not represented here because it is the same for all classes of transistors.

From a circuit design perspective, the previous build up of steps for these two transistors is described in **Figure 6**.

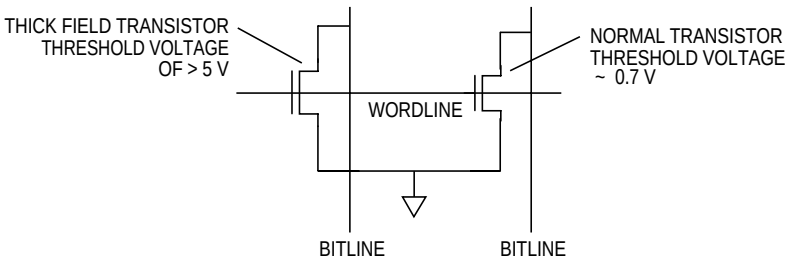

Figure 6. Bitcell Schematic Diagram

The transistor on the left is the same transistor as in the build-up diagram on the left and is referred to as a thick field transistor. Likewise, the transistor on the right is the same as in the build-up diagram and is a normal n-channel transistor. The critical difference in the two transistors is the threshold voltage of the transistors. The threshold voltage is

different due to the thickness of the oxide under the polysilicon gate. A typical transistor with standard gate oxide will have a threshold voltage in the range of 0.7 volts. Since the field oxide is very thick relative to conventional gate oxide, the threshold voltage of this transistor is very high, greater than 5 volts.

A high threshold voltage is important because to read a byte of data from the ROM array, a wordline corresponding to the selected address is raised to V_{DD}. The bitlines also corresponding to this byte are also raised to a voltage close to V_{DD}, and the sources of the transistors are grounded. These bias conditions set up a normal transistor with a threshold voltage of 0.7 volts to conduct current which is then sensed, latched, and driven out of the array. Since the thick field transistor has a threshold voltage above 5 volts, when V_{DD} is placed on the wordline and bitline the transistor cannot be turned on and will not conduct current. This state of the thick field transistor corresponds to the opposite data state of the bitcell that conducts current.

It is important to understand the basic concepts introduced by a ROM bitcell as the ability of a bitcell to either conduct a current or inhibit the conduction of current and is reused in the operation of the other electrically changeable memory types. It is also important to understand that the ROM bitcell formation is a function of a standard wafer fabrication process which is much simpler than the process required to build EPROM or EEPROM memory bitcells. Further discussion of these memory types bear this point out.

Finally, there are alternate methods of building a ROM bitcell. The description here is the most common method of ROM bitcell formation. The “programming” step uses the nitride layer which occurs early in the wafer fabrication process. This method has the drawback of a relatively long cycle time from receipt of a ROM code to the delivery of samples containing the code. As a result, there are alternate methods of achieving the same result of enabling or disabling a transistor from conducting current. All of these alternate techniques have the goal of pushing the ROM “programming” step to as late in the wafer fabrication process as possible to reduce the cycle time to turn a new code for a customer. But the end circuit operation is the same.

EPROM

EPROM was one of the first widely deployed non-volatile memory technologies, but it has been replaced gradually by FLASH EEPROM over the past few years. EPROM was used extensively in applications that required permanent storage of application code, as opposed to RAM whose contents would be lost at removal of power, but did not want to utilize a mask programmable ROM for the reasons previously discussed.

Remember from the description of the ROM manufacturing process and concept introduced of the transistor threshold voltage, or V_T, how this relates to the data state of the transistor. The same is also true for non-volatile memories but the means of achieving these states is physically much different. The diagram in **Figure 7** is useful in understanding the operation of an EPROM bitcell as well as describing the concepts reused in the other two categories, EEPROM and FLASH EEPROM.

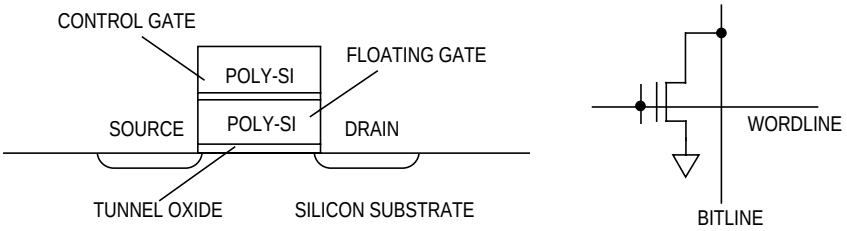

Figure 7. EPROM Bitcell Cross Section and Schematic

As can be seen from the diagram, the bitcell is constructed much differently from a typical CMOS transistor used in a ROM design. In particular, notice the extra gate in the bitcell referred to as the floating gate. It is referred to as floating because it is isolated on all sides and does not come in electrical contact with any terminal. The dielectric layer below the floating gate is commonly referred to as the tunnel oxide. The nature of this term is described further later. The top layer of the transistor is similar to the function performed by the gate of a CMOS transistor, as described for the ROM bitcell, but it also performs some additional functions for the programming operation.

Programming the bitcell is done via a mechanism referred to as channel hot electron (CHE). A large voltage, approximately 12 volts, is placed on the drain, 0 volts, or ground, on the source, and a slightly positive voltage

is placed on the gate of the device. The high voltage difference between the source and drain of the device is what heats the minority carriers in the channel. As previously described in **Common Terms and Definitions**, some of these carriers have enough energy to surmount the barrier presented by the tunnel oxide and become trapped on the floating gate. Even though the programming operation is accomplished via injection, the oxide under the floating gate is still commonly referred to as tunnel oxide, although this is somewhat of a misnomer. The programming operation requires a high programming current per bitcell to set up CHE. Because of this, most EPROM products have a high voltage supply pin, referred to as V_{PP}, that provides the high voltage used in the programming operation.

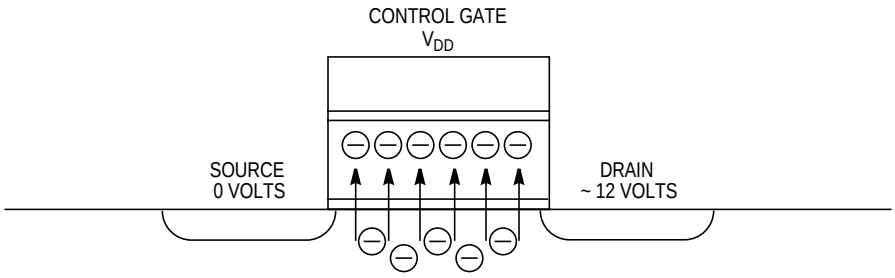

Figure 8. Channel Hot Electron Programming Mechanism

Programming does not itself refer to a particular data state of the bitcell, 1 or 0, because due to possible logical inversion in the output from the core memory array and the output to the data bus, the programmed state may correspond to either a logic 1 or 0. Therefore, it is sometimes easier when discussing the mechanics of the programming or erase operations to refer to the effect as either enabling the channel to conduct or not conduct a current. With this frame of reference, the programming operation of an EPROM bitcell injects electrons onto the floating gate of the transistor which raises the threshold voltage of the channel. A typical read operation of the memory array will place V_{DD} onto the wordline just as in a ROM read mode. As the V_T of the transistor approaches the level of V_{DD}, the channel will no longer conduct a sufficient current which will result in the bitcell being read as programmed.

One item not identified in the definition of CHE is the time to program a bitcell. Although there are some variations around these targets,

programming is typically specified to take between 20 and 50 μs per location.

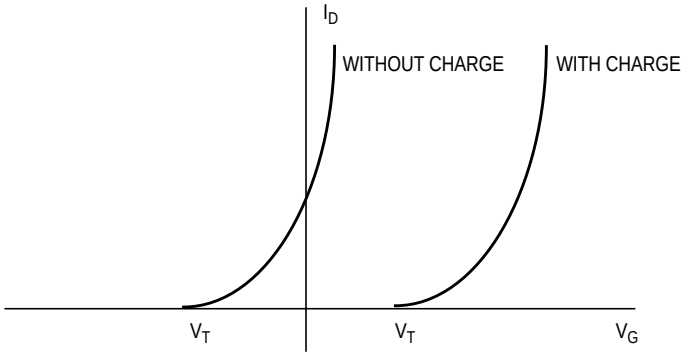

Figure 9. Effects of Programming and the V_T of a Bitcell

In the case of EPROM technology, the erase operation is performed by exposing the bitcell to UV light. UV light has the effect of removing the electrons stored on the floating gate and returning its state to what is referred to as charge neutral, which will move the V_T lower and enable the formation of a channel when the wordline is selected for a read operation. These concepts are reused in the description of the other memory technologies and form the basis for most of the NVM technology found in commercial use today.

From the description of the program and erase operations, the importance of the floating gate can be seen with respect to it acting as the storage node even if power is removed from the device. For the bitcell to remain in the programmed state, the materials surrounding the floating gate must be good insulators to prevent the loss of charge from the floating gate.

The ability of the floating gate to remain in a charge state is commonly referred to as data retention. This is where a major challenge arises in the NVM technology development. In general, the thicker the materials surrounding the floating gate, the more robust the data retention will be. However, the materials must all be thin enough to allow programming mechanisms such as CHE to take place within a reasonable program time. The challenge is finding the right balance in the manufacturing of these dielectric materials that provides acceptable programming times with robust data retention characteristics.

EEPROM

One of the primary disadvantages of EPROM is the requirement of exposing the bitcells to UV light to reprogram its contents. This can prove to be difficult, if the unit is already attached to the end application board, and requires an expensive ceramic package with a quartz window. The package cost can be reduced by placing the die in a conventional plastic package but at the sacrifice of the ability to erase and reprogram. Therefore, a solution to this problem was required that provided the ability to electrically erase the memory array. This was first satisfied with the introduction of EEPROM technology. In addition to providing the ability to erase the array electrically, the EEPROM also adds the ability to erase and reprogram individual bytes within the memory without altering the other contents.

This added functionality results in a larger bitcell when compared to EPROM. To provide the byte addressability in all modes, program, erase, and read, a second transistor must be added to the bitcell referred to as the select transistor. A cross section and schematic representation of the EEPROM bitcell is shown in **Figure 10**.

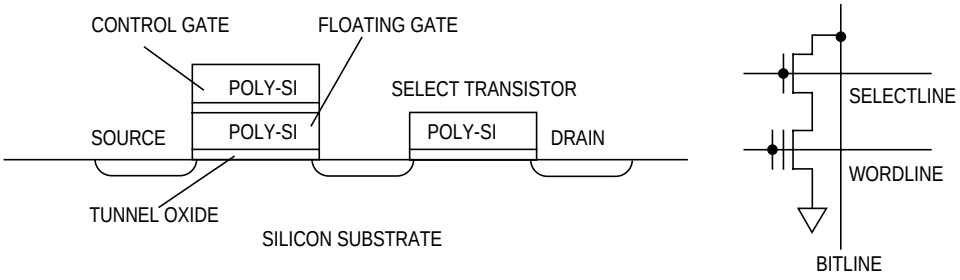

Figure 10. EEPROM Bitcell Cross Section and Schematic

Remember from the description of CHE that it is a one-way mechanism. In other words, it can move the electrons onto the floating gate, but it cannot remove them from the floating gate. EEPROM fills this role by providing an erase mechanism via Fowler-Nordheim tunneling. Since tunneling is created as a result of a high electric field between two terminals it provides a robust solution to the problem of erase. A high voltage, in the range of 18 volts, is placed on the control gate of the storage transistor while placing 0 volts on the source and drain of the transistor. The very high voltage difference between the source/drain and the control gate is what activates Fowler-Nordheim tunneling,

removes the charge from the floating gate, and can be seen from Figure 11.

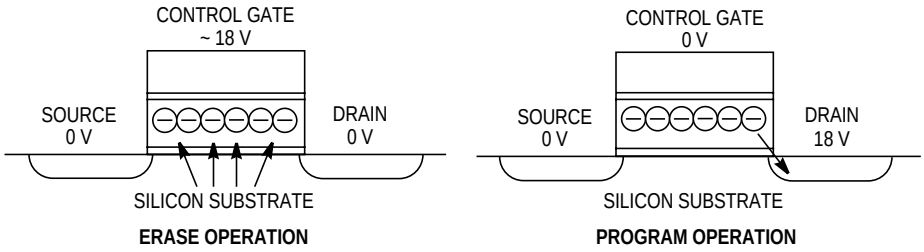

Figure 11. Fowler-Nordheim Tunneling in an EEPROM Bitcell

This diagram also shows a programming operation being performed with Fowler-Nordheim tunneling as well. As described previously, EPROM bitcells are programmed by CHE which requires an external high voltage supply pin to provide the current required to enable the mechanism. To provide full program and erase within the circuit without the need for external supplies, a lower-power programming method is required. Fowler-Nordheim tunneling requires little current from the high voltage supply used during the program or erase operations. It is primarily for this reason that Fowler-Nordheim tunneling has been adopted as the programming mechanism in addition to using it as the erase mechanism. Simple charge pumping schemes can be used to generate these high voltages on chip providing a self-contained solution without the need of supplying external high voltage supplies. One item not identified in the definition of Fowler-Nordheim tunneling is the time to program or erase a bitcell. Although some variations are around these targets, both program and erase operations typically are specified to take between 5 to 10 ms per by location.

As can be seen in Figure 11, the programming operation is performed by placing the high voltage on the drain of the storage transistor which sets up a field with the opposite polarity, as with the erase operation, and tunnels electrons onto the floating gate. Since the high voltage for programming is placed only on the drain, the electric field is set up along the drain edge of the device and it is in this region that tunneling takes place. The importance of the select transistor comes into effect when needing to selectively program only one byte. In a memory array, multiple bitcells share a common bitline, as high as 2048. If there is no

way to block this high voltage from the drain of an unselected bitcell, then the other bitcells also would be programmed. The select transistor performs the task of blocking the high voltage from reaching the unselected bitcells as can be seen in **Figure 12**.

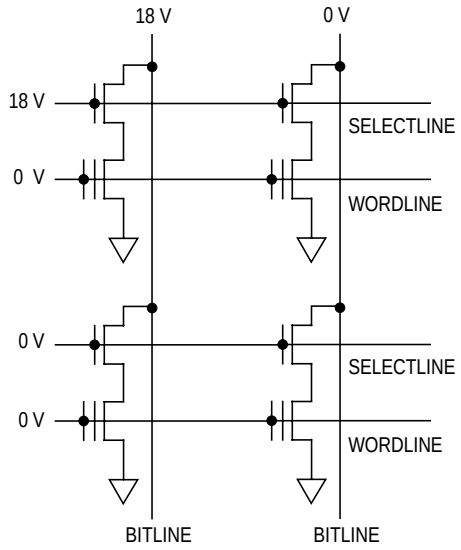

Figure 12. Selected and Unselected Bitcells during Programming

The selectline of the byte to be programmed must be raised to the same voltage, or higher, to pass the voltage placed on its drain. If 0 volts are placed on the unselected byte's selectline, then the voltage along the bitline cannot be passed to the storage device preventing it from being programmed. Since there are also many bitcells on one wordline and all of the bitcells sharing the one selectline will be able to pass the high voltage, another level of decode is required as well. As can be seen from **Figure 12**, the bitline on the right has 0 volts placed on it and hence will not be programmed since a high voltage will not be on the drain of the storage device. The bias condition of 0 volts on the bitline will be set up for all bits along the same row that will be left unchanged during the programming operation.

The astute reader would note that during erase 18 volts are applied to the wordline but this would mean that all bits along the wordline would be erased. So how is byte erase achieved with this architecture?

The answer is that the wordline is broken into byte wide or word wide pieces, depending on the bus width, and is decoded for each location.

For example, if there were 128 bits on a wordline and the array had the ability to erase eight bits individually, that would mean there would be a break in the wordline every eight bits for a total of 16 separate locations. At the point of each break, a simple pass transistor commonly is used to selectively pass the high voltage onto the control gates based on the location to be erased.

From this description it can be seen that the extra overhead in an EEPROM array is required to implement the extra features. A second transistor, the select device, must be added to each bitcell to achieve byte programmability, and the wordline must be decoded on a byte basis as well to achieve byte erasability. Both of these features add area to the memory array implementation and result in the same number of EEPROM bits consuming more area on silicon than implemented with EPROM. Due to the added overhead in both the bitcell with the select transistor and in the array with the erase decoding per byte, most EEPROM arrays are small and in general top out at several kilobytes in size.

FLASH EEPROM

The previous section covered EEPROM and the added features and capability that category of memory offers to the end application. However, these added features come at the price of larger memory arrays when compared to EPROM. For this reason, another category of non-volatile memory has been developed. FLASH memory was developed to provide the in-circuit, electrical erase offered by EEPROM but also optimizing the architecture to reduce the area for the memory array. The primary method used to achieve this is a modification of the features of the memory and in particular the removal of the byte-erase feature and sometimes also the byte program feature.

FLASH EEPROM, or sometimes referred to by just the name FLASH, is offered as a standalone memory by many companies as well as many others offering FLASH for embedded applications. As a result, many different implementations are found in the industry. A generic overview of FLASH memory and a more focused examination of the FLASH memory technologies found in Freescale products is presented here.

As mentioned, the primary difference between EEPROM and FLASH EEPROM is the removal of the ability to erase at the byte level. FLASH erases in much larger chunks of memory commonly referred to as sectors. Depending on the array size and the technology chosen, the sector size can vary significantly and therefore there is not a standard erase sector size across the industry and even within a product family. The main point to remember is that the array is erased in large pieces as opposed to byte erase found in full featured EEPROM. Almost all commercially available FLASH memories utilized Fowler-Nordheim tunneling for the erase operation.

The second major difference relates to programming and the programming size but here again there is not a clear standard across the industry. Some FLASH memories will do away with byte programming all together and will program in large sections referred to as pages. Other FLASH memories still retain the ability to program in byte wide increments. The choice in programming width is mostly determined by the throughput of erasing the memory and completely reprogramming the array. There is also some diversity among FLASH memory products with respect to the programming method. For example, some FLASH products use CHE and others use Fowler-Nordheim tunneling. As has been previously described in the EPROM and EEPROM overviews, each method has pros and cons, and it is these limitations that drive the programming size of the array.

Remember from the discussion of EPROM that CHE requires a relatively high current, especially when compared to Fowler-Nordheim tunneling. However, Fowler-Nordheim tunneling requires more time to program a memory location than does CHE. Therefore, to compensate for the longer time required per programming location using Fowler-Nordheim tunneling, the programming size is larger than that used with CHE. CHE cannot scale with respect to program size because of the high current required per bit to activate the mechanism. Although there are certainly power supplies that can supply many amps of current to a V_{PP} pin on the part, there is an issue with power distribution within the chip itself. In general, this limits the programming size when using CHE to 8 to 16 bits.

A look at each of the FLASH memory technologies found on Freescale MCUs will now be addressed.

1.5T FLASH EEPROM

This memory technology is found in the 68HC12, 68HC16, and 68HC3xx Families of microcontrollers. Development of the technology dates back to 1990 and was the first FLASH memory offered for embedded use in Freescale products. The name 1.5T FLASH comes from the arrangement of the bitcell and in this case indicates that one and a half transistors comprise the bitcell. The fact that the name implies that there is a possibility for a half transistor may seem odd, if not impossible, but in actuality indicates the construction of the bitcell to be of a category referred to as split gate. In other words, there is a gate within the bitcell that is structurally shared with another gate resulting in the half transistor. A cross section and top down view of the bitcell in **Figure 13** helps explain this concept.

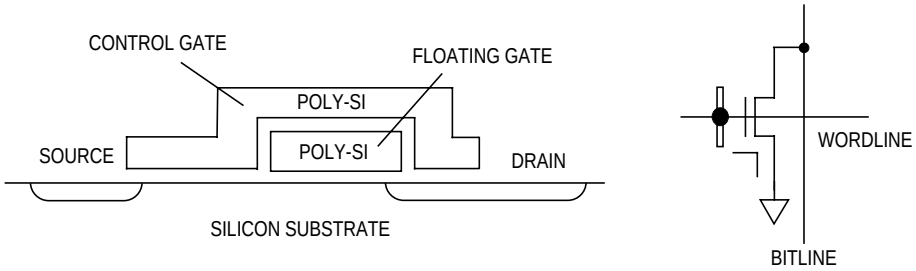

Figure 13. 1.5T FLASH Bitcell Cross Section and Schematic

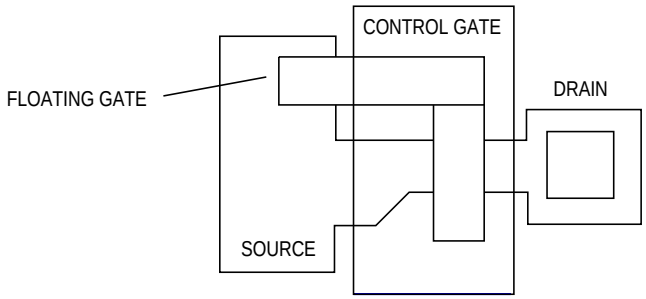

Figure 14. 1.5T FLASH Bitcell Top Down View

The half transistor in the case of 1.5T FLASH comes from the shared gate formed by the control gate overlapping the channel on the source side. This extended overlap essentially forms a select transistor on the source side of the storage node. This bitcell architecture is unique when compared to the conventional EPROM and EEPROM bitcells. In particular, the EEPROM erase operation occurs in the same channel of

the transistor as for the program operation. The 1.5T bitcell is different in that the programming occurs in the channel as a function of CHE. However, the erase operation occurs within the source region with the part of the floating gate that extends into the source. This is why this bitcell architecture is sometimes referred to by its technical name of source coupled split gate (SCSG) FLASH EEPROM. During the erase operation, the source is actually what is driven to a high voltage, approximately 12 volts, with the control gate at V_{SS}, or 0 volts, and the drain is left floating. The control gate voltage is capacitively coupled to the floating gate, and the subsequent potential difference between the source and the floating gate is what activates the erase operation.

As mentioned, the program operation is conducted with CHE. The source is driven to V_{SS}, 0 volts, the control gate is held at approximately 9 volts, and the drain voltage is approximately 7 volts. The lateral potential difference between the source and drain is what activates CHE, and the high voltage on the control gate is what forms the channel. The microcontrollers that utilize the 1.5T FLASH memory have an external high voltage pin to program and erase the FLASH memory. The externally applied voltage is typically in the range of 12 volts, and quite often the question arises about why a higher external voltage is required than what is used internally to the bitcell.

A simple explanation is that the paths from the external pin to the internal nodes are not perfect and lossless. Because switches in CMOS, and in particular high voltage switches as is necessary for this technology, experience voltage drops as a function of their threshold voltage, resistance in the channel, and other characteristics of the transistor, the external voltage applied is usually higher than the internally driven voltages. This is not to say that the internal nodes are regulated independent of the externally applied voltage. In other words, if the externally applied voltage is targeted to be 12 volts and the external circuit applies 12.5 volts, there will be an increase of the internal nodes of approximately 0.5 volts.

WARNING: *Depending on the maximum ratings of the transistors used in the path from the external pin to the internal node, an over voltage as small as 0.5 volts can result in serious damage to transistors, bitcells, and support circuitry that could permanently damage or degrade its operation.*

The formation of the select transistor on the source side of the device as opposed to the drain side for EEPROM results in some different characteristics being introduced into the bitcell operation. Keep in mind that the placement of the select transistor on the drain side of an EEPROM bitcell was done primarily to block high voltages from reaching the storage transistor, thus providing selective program and erase. Since the select transistor has been moved from the drain side to the source side, the result is that the bitcell is now subjected to data disturbs as well. Disturb comes primarily from programming another location along the same bitline of the array for this technology. A high voltage is applied to the bitline during the programming operation, and since the select transistor is on the opposite side of the device, all of the cells along the bitline will see the high voltage stress. This effect is managed through a careful and precise programming sequence, as well as proper balance of the voltages along the control gate and drain for the selected case and the rate of disturb with drain-only bias for unselected cells.

2T FLASH EEPROM

2T FLASH development started in 1994 and had several targeted features that drove a new FLASH memory technology development. In particular, the M68HC08 Family of microcontrollers had a targeted minimum V_{DD} of 1.8 volts as well as very low power operation. As was noted in the discussion of the 1.5T FLASH, this technology is capable of supporting products that operate in the range of 3 to 5 volts. Although a few circuit design methods can be used to overcome some of the limitations with respect to the ability to operate with a V_{DD} below 3 volts, virtually all of them result in higher power consumption. Obviously, with the primary objective of operating at a lower V_{DD} to reduce power consumption, pushing the 1.5T technology would not achieve the desired features of lower V_{DD} and lower power operation.

The result of the development effort is a bitcell that looks at first glance to be very similar to a conventional EEPROM bitcell, as seen in the cross section and schematic views of the bitcell. (See **Figure 15**.) However, upon closer inspection, it is noticeable that the primary difference between the 2T FLASH bitcell and a regular EEPROM bitcell is the placement of the select transistor relative to the storage transistor. Remember from the EEPROM description that the select transistor is positioned between the storage transistor and the bitline. The 2T bitcell

reverses these positions with the storage transistor now directly positioned on the bitline and the select transistor on the source side. Consequently, this technology is sometimes referred to by a more complete technical name of 2T source select, or 2TS, versus an EEPROM bitcell which is 2T drain select, or 2TD.

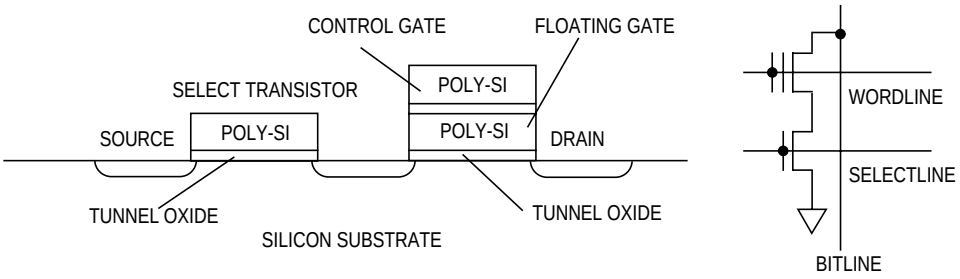

Figure 15. 2T FLASH Cross Section and Schematic

Although this simple rearrangement of the transistors in the bitcell would seem to be a small modification, it forms the critical feature that enables the bitcell to operate at both lower V_{DD} and with lower power. During the description of the 1.5T technology, there is some discussion about the circuit design methods required to switch a high voltage within a circuit. A typical logic transistor can only withstand normal operating voltages which vary by the technology generation, some as high as 5.5 volts, but this has been consistently dropping with the latest 0.25-μ generation operating with a typical V_{DD} of 2.5 volts. The gate oxide of the logic transistor is what commonly dictates this upper operating limit. Therefore, to pass high voltages like those typically found in non-volatile memory designs, another transistor type with a thicker gate oxide is typically required to withstand these high voltages. One of the common trade-offs made when thickening the gate oxide is that the dimensions of the transistor must be larger and together these two parameters generally result in the transistor being slower with respect to switching speed. In the case of program and erase operations, the slower switching speed is not a limiting factor, since the operations take many microseconds to perform. However, transistor switching speed is critical to read performance, and this is where the use of high-voltage transistors within a design create challenges to low voltage and low-power operation.

In considering the EEPROM bitcell, high voltage must be applied to the bitline for programming and to the select gate as well as control gate during erase. High voltage transistors must be used in the paths to these nodes to withstand the voltages required to perform these operations. During a normal read operation, the bitline and selectline must be driven on each access cycle at speed. However, the high voltage transistors will be slower to switch due to the thicker gate oxide and larger dimensions of the transistor which results in the memory not performing as fast as the other logic in the design. Typically, this difference is not noticeable in the voltage range of 3.3 to 5.5 volts for EEPROM and 1.5T FLASH. But as V_{DD} is lowered below 3.3 volts, the high voltage transistors cannot support the performance requirements and limit the ability of the memory to read at the rated speed.

A common design technique used to work around this problem is to use voltage boosting circuits along the selectline and bitline. The concept of a boosting circuit is to take an input voltage that is low, for example 1.8 volts, and through some charge pumps raise the voltage level. While this will, in effect, lower the operating voltage range of the memory array, it adds considerable power consumption to the read operation because of the charge pumps that must be always active. This option can be effective in applications where lowering voltage range is more important than lowering the overall power consumption of the memory array.

The 2T FLASH attacks this issue by eliminating the use of high voltages along the typical read path, the bitline, and selectline, allowing the use of standard logic transistors for these nodes. To achieve this, the first modification that must be done is to move the select transistor from the drain side of the storage transistor to the source side. Programming is performed through Fowler-Nordheim tunneling by applying –12 volts to the control gate, floating the source by turning off the select transistor, and driving 5 volts onto the bitline. With this split bias scheme, the only node within the bitcell that sees high voltage is the control gate. Remember that Fowler-Nordheim tunneling is activated by the potential difference between two nodes and the electric field created as a result. The fact that the control gate is driven below V_{SS} to a negative voltage is what sets up the large potential difference and activates Fowler-Nordheim tunneling.

Erase is accomplished in a similar fashion. The source is driven to V_{SS} through the select transistor by turning the transistor on, the control gate is driven to approximately 15 volts, and the drain is placed at V_{SS}. These biases set up the electric field required to activate Fowler-Nordheim tunneling and erase the bitcell. Again, high voltage is applied only to the control gate of the transistor while the selectline and bitline stay within normal logic operating levels. Since the critical switching nodes of the array for read mode, the bitline and selectline, always stay within the normal voltage range of the product, regular logic transistors can be used. The end result is that the memory array can now operate within the same V_{DD} range as the rest of the logic on chip. Finally, since all of the nodes have to be driven only with regular logic levels to achieve the desired performance in read mode, no charge pumps need to be on during read mode and low-power operation is also achieved.

The 2T bitcell design does not come without its disadvantages. During the discussion of 1.5T FLASH, the subject of bitcell data disturbs was introduced. Also keep in mind that the placement of the select transistor on the drain side of an EEPROM bitcell was done primarily to block high voltages from reaching the storage transistor thus providing selective program and erase. Since the select transistor has been moved from the drain side to the source side for the 2T FLASH bitcell, the result is that it is now subjected to data disturbs as well. Disturb comes primarily from programming another location within the same row of the array. The programming operation places –12 volts on the control gates of all bitcells along the row. A bitcell is selected by placing +5 volts along the bitline while an unselected bitcell has 0 volts along its bitline. The potential difference for a selected bitcell will be |17V|, but an unselected bitcell still has a relatively high difference of |12V|. This translates to the unselected bits being exposed to a weak programming event.

A good balance must be achieved between the time to program a regular bitcell and the time to disturb an unselected bitcell. Other preventative measures must be taken to manage disturbs. There is a natural level of variation for program time across the array with some bitcells programming faster than others. One method for minimizing the exposure that the unselected bitcells have is to use an adaptive programming algorithm. As a result, some new specification items must be added such as programming step size and maximum number of

program pulses. Without an adaptive programming algorithm, a programming time spec would have to be derived based upon the worst case program time across the entire array. This type of spec would eat into the operating margin for disturb and would likely render the product not manufacturable or at best very difficult to manufacture. The adaptive programming algorithm has a benefit for the user. Since the programming operation takes no more time than required for each location, the overall program time for the entire array is much faster than if a worst case value was used for all locations.

1T FLASH

One transistor FLASH, or 1T FLASH, is what is most commonly used in the industry for standalone FLASH memory. It is popular among standalone memories since bitcell area tends to dominate very large memory arrays and the select transistor takes up silicon real estate. Many details related to the bitcell and operation of the array vary among manufacturers and thus it is not possible to cover all of these in this discussion. However, a description of the 1T FLASH technology now becoming available on several Freescale microcontroller products is provided.

From a structural point of view, the 1T bitcell appears to be very similar to the EPROM bitcell discussed earlier. The cross section and schematic view of the bitcell is identical at a high level, but this is where the similarities stop. (See **Figure 16**.)

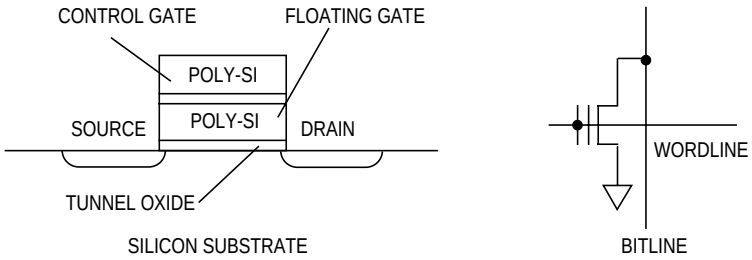

Figure 16. 1T FLASH Bitcell Cross Section and Schematic

In the case of the Freesca;e implementation of 1T FLASH for embedded products, the programming and erase operations are performed via Fowler-Nordheim tunneling. Other vendor's implementations utilize CHE for the programming operation, but Fowler-Nordheim tunneling is almost

universally used to perform the erase operation. As was the case for products with the 1.5T FLASH technology, the products utilizing the 1T FLASH have an external high voltage supply for the programming operation. The 1T FLASH design also uses an external high voltage supply pin for the program operations and is used to generate the control gate and drain voltages required to sustain the Fowler-Nordheim tunneling operation.

WARNING: *Great care must be taken to properly control this external high voltage supply just as with the 1.5T technology as internal damage can result by a small drift in this supply voltage.*

Some technical challenges that must be addressed and managed arise from elimination of a select transistor from the bitcell. Recall from the discussions of the EEPROM and 1.5T FLASH technologies that the inclusion of a select transistor provided varying degrees of protection to the bitcell during a program operation or advantages to read performance in the case of 2T FLASH. Some additional functions the select transistor performs in the array operation for those technologies have not yet been discussed here, but they become critically important in the operation of the 1T FLASH.

To read a particular bitcell for EEPROM, 1.5T and 2T FLASH, the wordline must be raised to the V_{DD} level. The voltage applied turns on the select transistor which then allows the actual storage transistor access to the bitline. The 1T FLASH bitcell does not have a separate select transistor but yet the read operation is performed in a similar fashion. This in itself does not present a problem as this replicates the read mode operation of an EPROM bitcell but comes as a result of the introduction of the erase operation to a single transistor bitcell. If the bitcell is erased too long, a condition referred to as over-erase can result. The basic explanation of this effect is that during the erase operation the bitcell V_T is moved. If the V_T is moved too far, the bitcell will operate in depletion mode, meaning that regardless of the voltage applied to its gate the transistor will always conduct current and be on. The net effect is that when a wordline is unselected, lowered to V_{SS}, the bitcell will still conduct a current and pull down the bitline resulting in all bits along the bitline reading erased. This is not a problem for EPROM bitcells since the exposure to UV light returns the floating gate to the charge neutral

state, which by design is not in depletion mode. This is not a problem for EEPROM, 1.5T FLASH, and 2T FLASH since the select transistor will prevent conduction even if the storage transistor is operating in depletion mode.

Another problem arises if a bitcell is over-erased. Not only will the bitcell cause all other bitcells along the bitline to read erased, but it also will prevent all other bitcells along the bitline from being programmed. Again, since the bitcell is operating in depletion mode, meaning always on regardless of the gate voltage, if a high voltage is applied to the bitline, which is the case for programming, the over-erased bitcell will try to pull the bitline down. Since the onboard circuits that generate the bitline voltage have a limited current supply capability designed for normal operation, the pull down presented by the over-erased bitcell will drop the bitline voltage and suppress the program operation to the point of failure. From this explanation, it can be seen that a 1T bitcell must carefully manage the erase operation, and an erase algorithm must be carefully tailored to address these issues.

Data disturb conditions also exist with the 1T FLASH bitcell for both wordline and bitline conditions. Again, the program and erase operations must be carefully designed and controlled to avoid unintentional changes to data in other sections of the array.

General Topics Related to NVM

To this point, the information in this application note has described the operation of non-volatile memory devices and has provided a detailed explanation of the operation of the NVM technologies found in Freescale products today.

However, some items have not been covered that are general to NVM operation. The final topics to be covered are program and erase endurance and data retention. This discussion provides an overview of the operation of the bitcell and impact on these parameters.

Very high voltages are utilized to perform the program and erase operations as has been seen repeatedly for each of the technologies

that have been covered here. High voltage transistors are utilized in the generation and switching of these high voltages, and these transistors are specifically developed for this purpose to withstand these conditions for extended periods. The bitcell also is developed to withstand these stresses but it has operating limits that are inherent to its operation. The principal program and erase mechanisms that have been discussed, CHE and Fowler-Nordheim tunneling, present significant stress onto the bitcell.

The challenge arises because the program and erase operations are constrained with respect to how long the operation is allowed to take. Since a relatively fast program and erase is desired, the tunnel oxide beneath the floating gate needs to be relatively thin. However, the thinner the oxide becomes, the electric field increases in strength across the oxide and hence the higher the stress. In terms of actual operation, this translates to each program and erase event degrading the oxide by a very small amount. A single operation will not result in the failure of a bitcell, but the cumulative effect on the bitcell of the program and erase operations ultimately will reach a point of failure. The failing event can be very subtle and will not always be a catastrophic and hard failure in which the bitcell ceases to be able to program or erase.

For example, the degradation in bitcell operation can result in the program or erase times needing to be longer to change the data state of the bitcell. Another case might be that the bitcell disturb characteristics degrade to the point where a longer-than-normal program event could result in data being inadvertently altered within the array. These kinds of cases and understanding their behavior are important during the development cycle for a given NVM technology. The behavior of the array must be understood across the entire operating life of the product as specified. Therefore, it is quite common for the NVM array to perform much better during the early stages of its life cycle when compared to the maximum or minimum limits in the specification. The reason for this is to account for the change in behavior and performance as a result of program and erase events over the course of normal operation and still enable the bitcell to operate at the end of life.

A specific example of this is program time. It is common to observe that the array will program faster than the stated values in the specification

during the early life of a product. However, as the unit is cycled, the program performance might slowly degrade and by the time it reaches the upper limit for cycling the array program time might be very close to the specified value. Because of this, the customer should strictly observe and implement the specified values and not become aggressive based on early life product observations. This kind of practice will result in performance issues for an application later in the life of the product and can have serious consequences.

Data retention performance is another critical characteristic of array performance. There is the obvious expectation that the data programmed into the array will remain there whether the array is programmed one time or to the upper limit of the cycles allowed. The specifications for the operation of the array play a major role in the data retention of the product through its life cycle. Subjecting the part to higher-than-specified voltages, such as V_{DD} or V_{PP}, or performing program and erase operations outside of the specification, can result in overstressing the bitcells that could effect the data retention of the product. This is another parameter that can demonstrate very good performance in the early phases of its life but if used outside of the defined specification can result in serious degradation and ultimately failure.