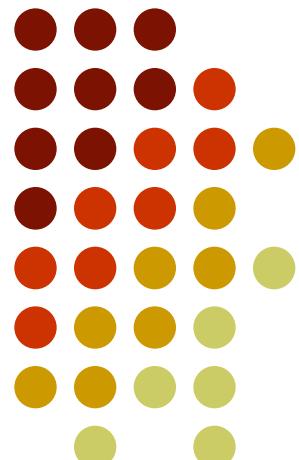
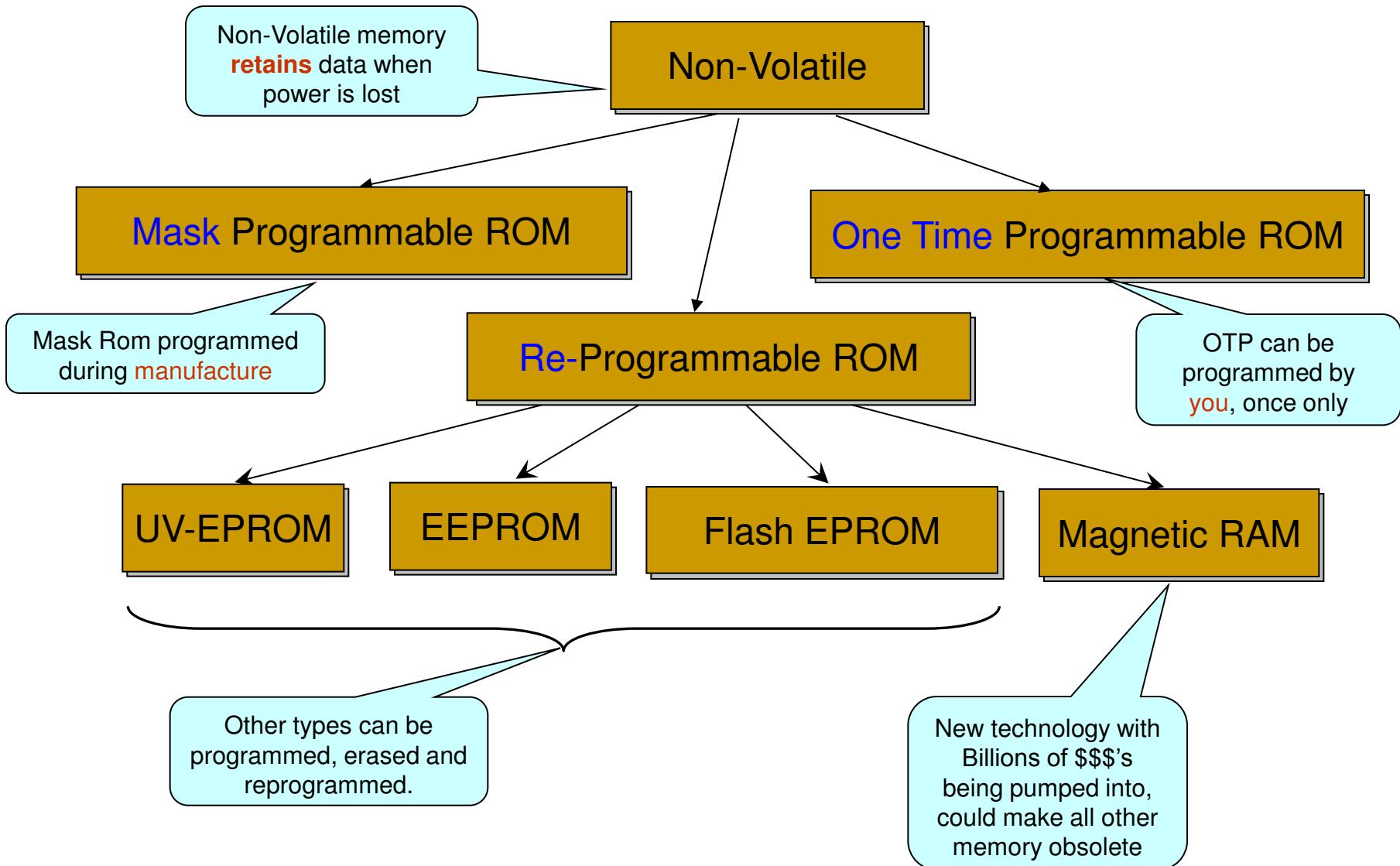


Non-Volatile Memory Technology

- Mask Programmed Read-Only Memories (Roms)
- One-Time Programmable Roms (Proms)
- Ultra Violet Erasable Proms (UV-Eproms)
- Electrical Erasable Proms (EEProms)
- Flash Erasable Proms
 - NOR vs NAND Technology
- Magnetic Ram (MRam)
- Intel & Micron's new 3D XPoint Memory

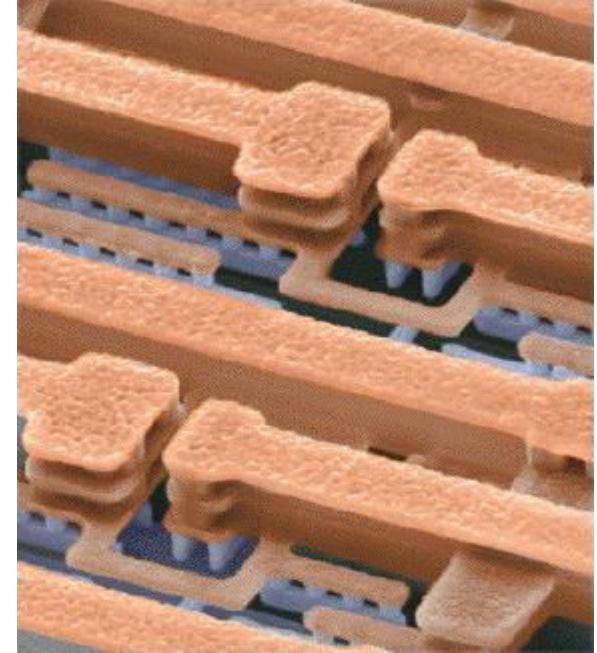


Classification of non-Volatile Memory based on Technology

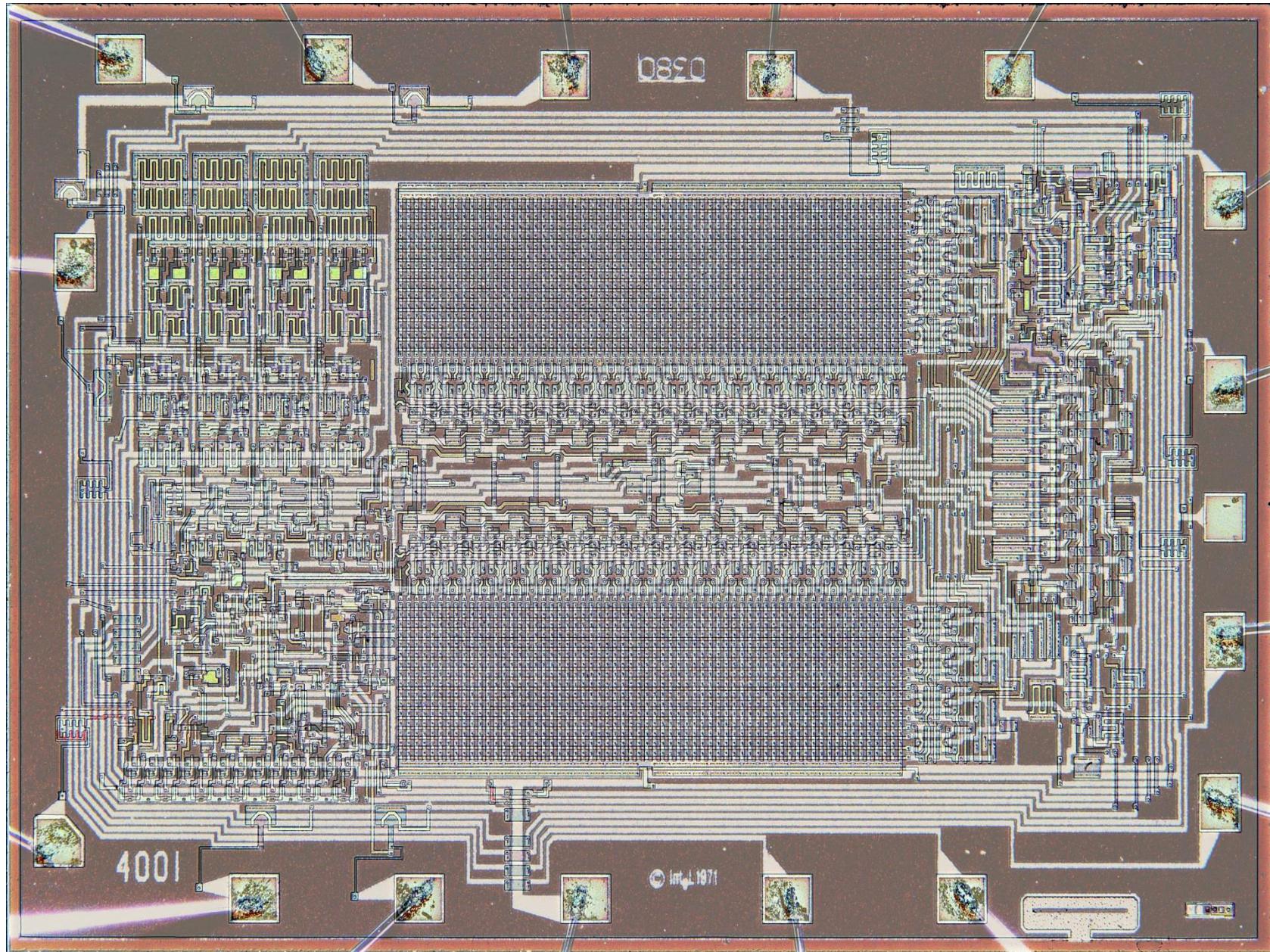


Mask Programmed ROM

- Mask programmed ROM technology was once a very popular technology.
- It was based on the idea that the chip was physically “programmed” with its data by **hard wiring** the data as electrical connections between the transistors on a chip during manufacturing.
- These hard wired connections were made during the application of the final ‘**metallization**’ layer during production and as such each ROM amounted to virtually a ‘**semi-custom**’ design.
- The drawback to this type of chip is that once manufactured it was impossible to change the data in the chip and a change, even a **minor one**, required a complete re-design of the metallization layer at significant cost, so it got used only in applications where the data/program code that was **never ever** going to change, e.g. dictionaries, language translators etc.
- The illustration above shows example of three layer copper *interconnect metallization* in IBM's new faster CMOS integrated circuits



Early Intel Mask Programmed ROM



Mask Programmed ROM

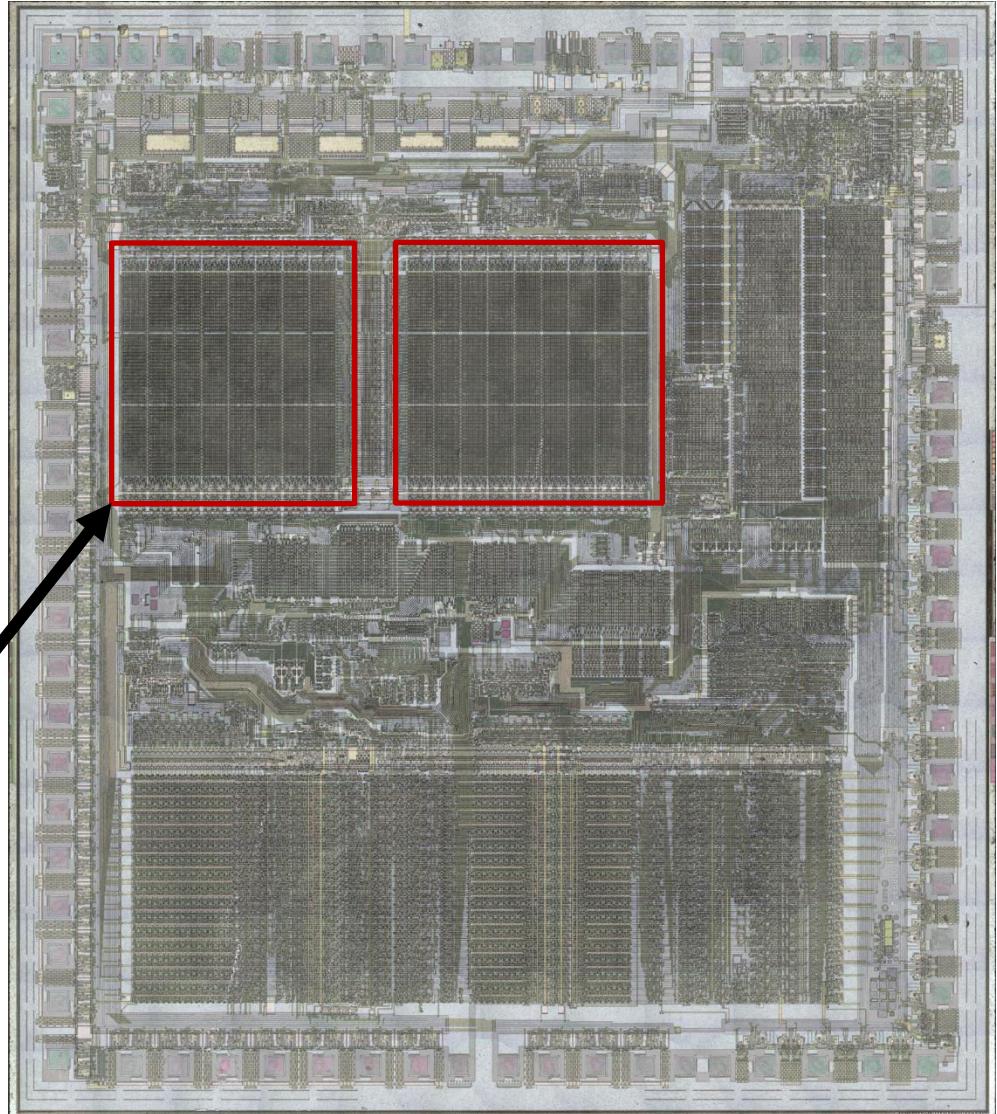
- One attraction of this kind of technology is that it is not possible to accidentally erase or corrupt the data which is possible with most other kinds of ROMs.
- A further benefit was that the chips could be used **straight out of the box from the manufacturer** i.e. there is no need to employ someone to physically ‘program’ blank devices, thus saving time and money on the production line.
- Example applications of the time, included **children's toys, dictionaries, synthesized voice/sound effects, consumer electronics, firmware, i.e. microcontrollers with very well defined software**, things not obviously looking like computers with no possible way to upgrade.
- These days Mask ROM Technology has largely fallen out of favour for *program development purposes*, since other technologies are more flexible and cheaper, but these newer technologies are simply *evolutions* of Mask Rom, i.e. they share a common architecture.



*1st ever computer toy has 128k bytes
(circa 1978 - Texas Instruments)*

Mask Programmed ROM

- Mask Rom is mostly used today inside CPUs to replace large amounts of *glue logic*.
- For example, any *truth table* can be replaced by a *look up table* implemented as a Read only memory (Rom) i.e. input data goes to address, rom supplies outputs. This also gives consistent delays regardless of boolean equation complexity.
- The illustration opposite shows two blocks of Mask Rom being used to replace fixed logic in the instruction decode and execution unit of the **68000 CPU**.
- Quartus can automatically replace synthesized logic with Rom, if you tick the right options at compile time.



1 Possible Architecture of a 2 Location x 4 Bit Wide Mask ROM Chip

Connection from Data
Transistor to load
transistor allows data
transistor to pull D0 to
logic 0 when selected

Vdd

Load transistors, (1 per data line)
generating a logic 1 on data pins.
Acting like resistor but current limited

4 Data transistors

Cell Select 0

4 Data transistors

Cell Select 1

No connection from
data transistor to load
transistor. Result: D3
pulled high by the load
transistor

D0

D1

D2

D3

Ground Connection

4 Bit Data Out

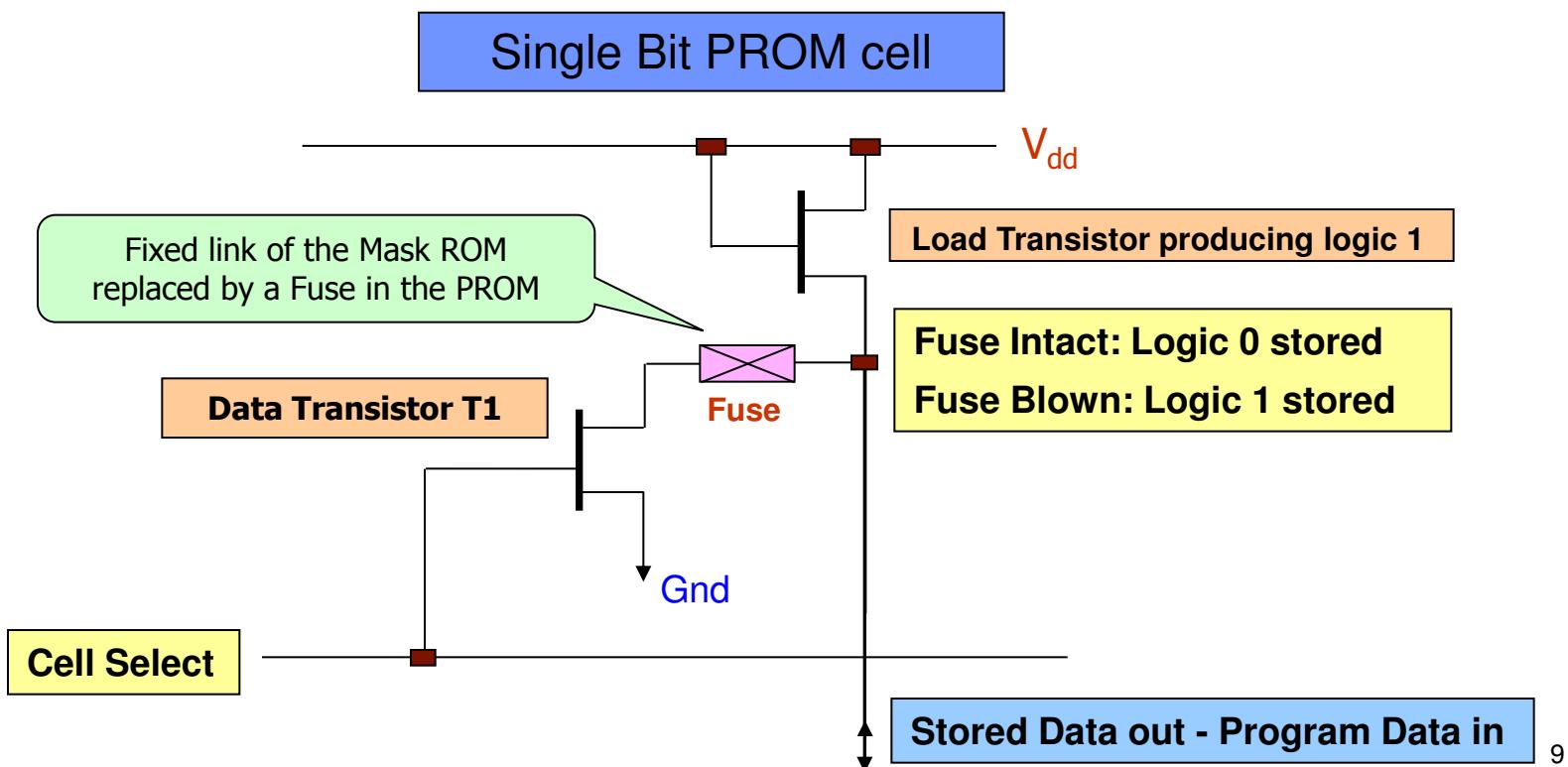
- The two **cell select lines** are generated internally from **row** and **column** decoders within the chip.
- 1 load transistor** for each data line, attempts to create a **logic 1** on its associated data line.
- By physically wiring the drain of the Data transistor, (1 for each bit or storage in the chip) to the data line, we can create a **logic 0** on the data line. By leaving the drain unconnected we retain a logic 1. Connection is defined during the metallization process, i.e. a connection is **made** or **not made** during production.

Mask Programmed Rom Characteristics

- Interface characteristics very similar to static ram (*without the ‘R/W’ line*)
- Very high bit density: Roughly 1 transistor per bit.
- 3-6 week delay in getting first sample from manufacturer. In effect each version of the programmed data needs a new semi-custom chip design, i.e. a redesign of the final ‘mask’ that applies the metallization layer.
- Large “up-front” cost of producing first production sample.
- Extremely low cost when purchased by the hundred thousand especially so when you factor in the labour savings because the user does not have to bother programming them.
- If the program is wrong or needs to be changed, you have to throw it away and start again incurring major cost and delay.
- Not so popular as a separate memory chip technology these days as other technologies are more flexible, offering erasability but manufacturers still produce them when the high volumes justify the initial development costs.

One-Time Programmable ROM (PROM)

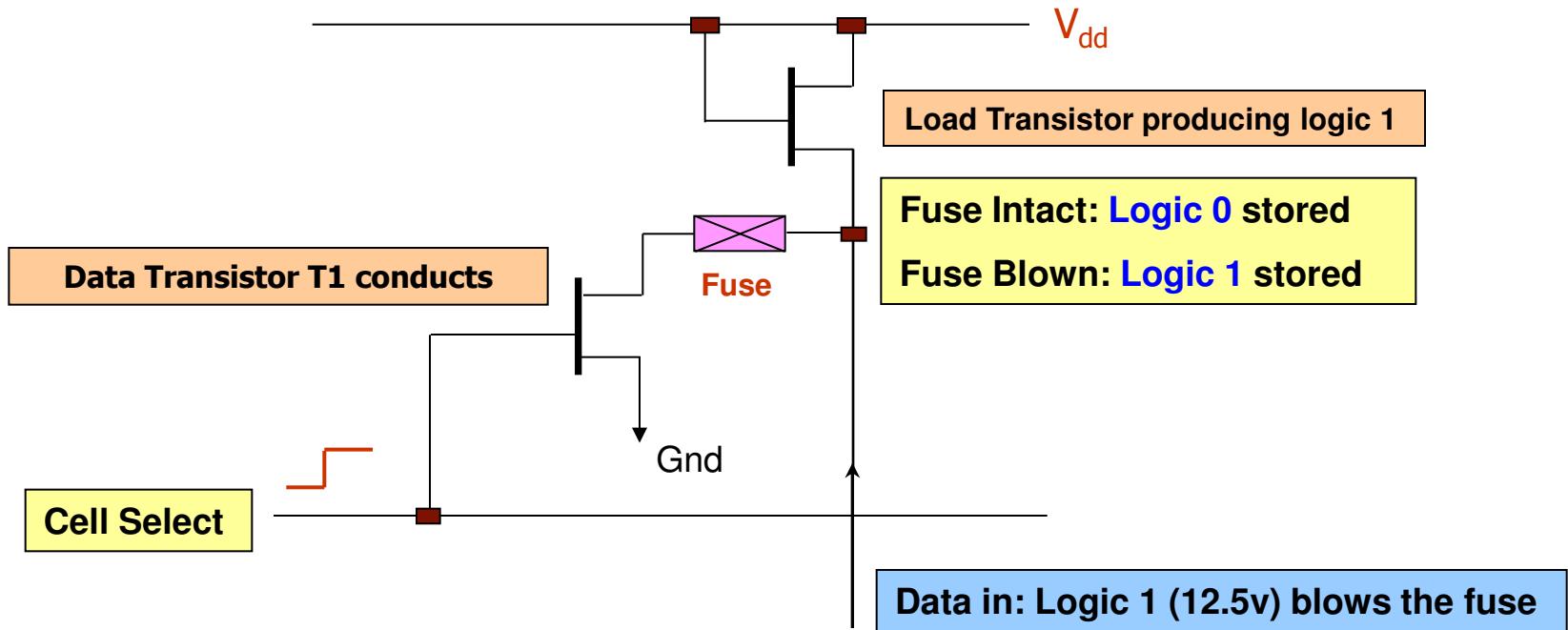
- To solve the problems of **cost** and initial design **delay** associated with Mask technology, manufacturers and users looked for ways in which a ROM could be programmed with its data after production of the chip. This lead to the development of **PROM** technology.
- Architecturally a **PROM** is identical to a **Mask Rom**, except that all the data transistors (**T1** below) initially have their drain connection “**wired**” to the load transistor, via a **small fuse**.
- With the fuse **intact**, **T1** is able to pull the Data line to **logic 0** when the cell is selected.
- With the fuse **blown**, the data line would remain at **logic 1**, because of the load transistor.



One-Time Programmable ROM (PROM)

Programming a PROM

- The device is programmed by plugging the chip into a special PROM Programmer which raises the normal operating voltage V_{dd} from 5v to say 12.5v
- Data is then driven into the data line by the Prom programmer.
- An address is presented to the chip by the Prom programmer, which then selects the cell(s) at the specified location and turns on the data transistor T1, shorting one end of the fuse to Gnd/0v.
- If the data to be programmed is a '0', then the voltage either side of the fuse is 0v, thus no current flows through the fuse and the fuse remains intact.
- If the data to be programmed is a '1', one end of the fuse is held at 12.5v and the other (via T1) at ground. The resulting high current flow, blows the fuse "creating" a logic 1.



One-Time Programmable ROM (PROM)

- To program the device, the developer downloads their data to the Prom programmer typically using a serial **RS-232**, **Parallel**, **USB** or **Network** link.
- Various formats for a data download exist and generally include **checksums/CRC** to ensure data integrity.
- **Intel Hex** and **Motorola 'S' records** are common download formats for all programmers.
- Prom programmers range in price from a **few hundred \$'s** for slow occasional use to many **tens of thousands of \$\$\$'s** for programmers used in high volume computer production.



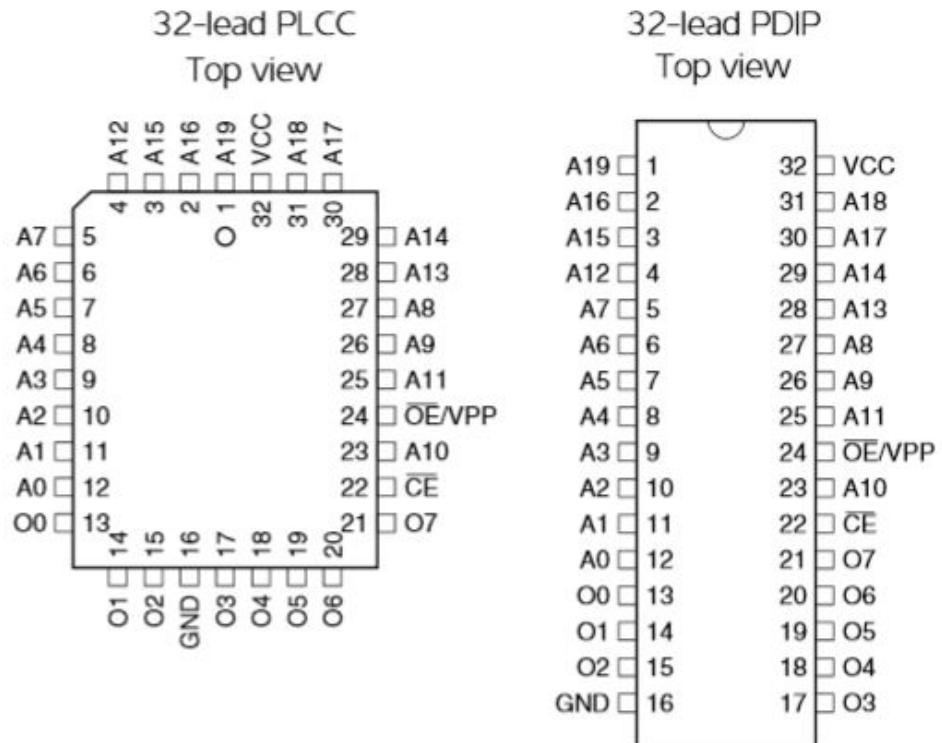
Prom Programmer

One-Time Programmable ROM (PROM)

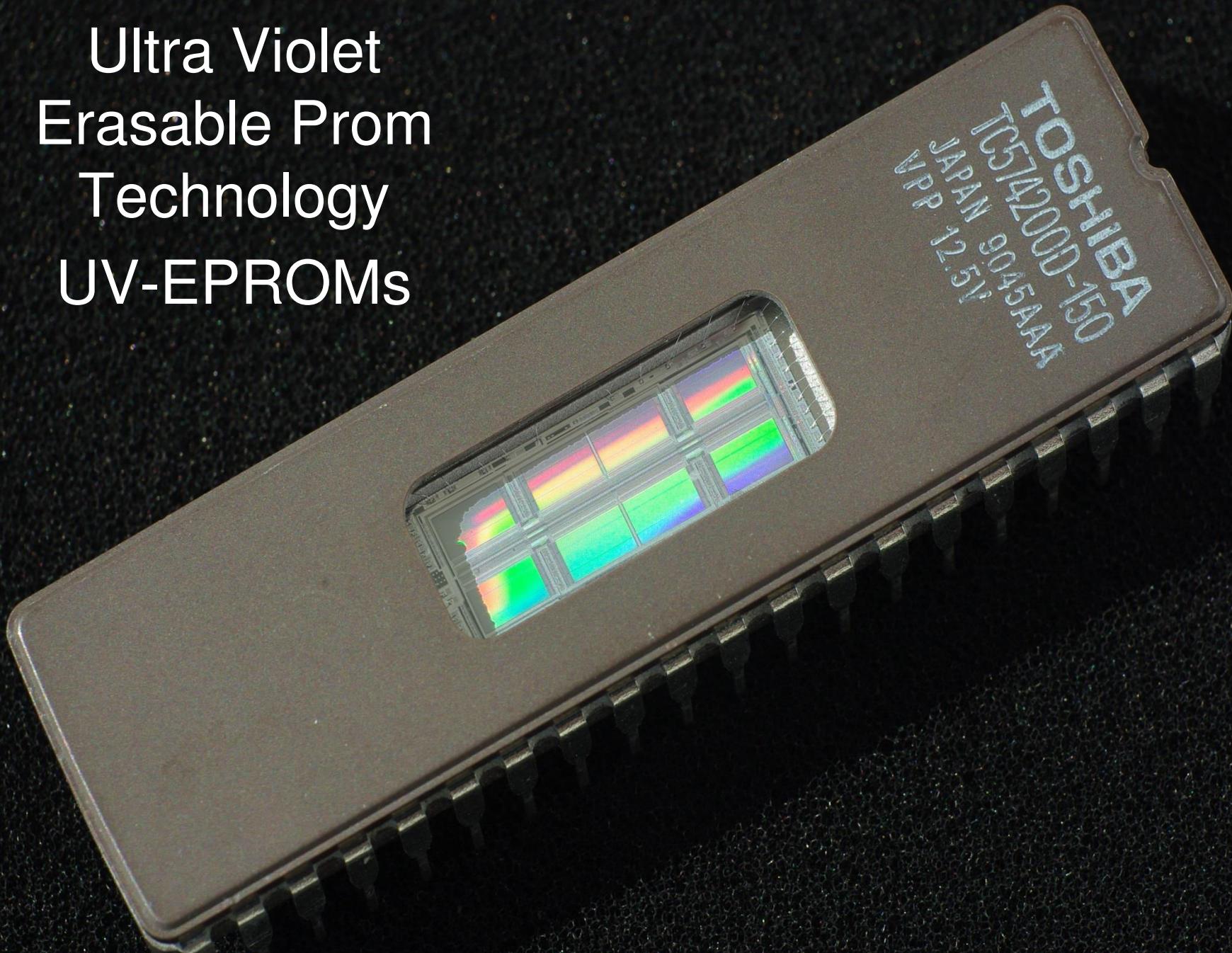
- Atmel 27C080 – 1MByte Prom (~\$13 from Digikey)

Pin configurations

Pin name	Function
A0 - A19	Addresses
O0 - O7	Outputs
CE	Chip enable
OE/VPP	Output enable/Program supply

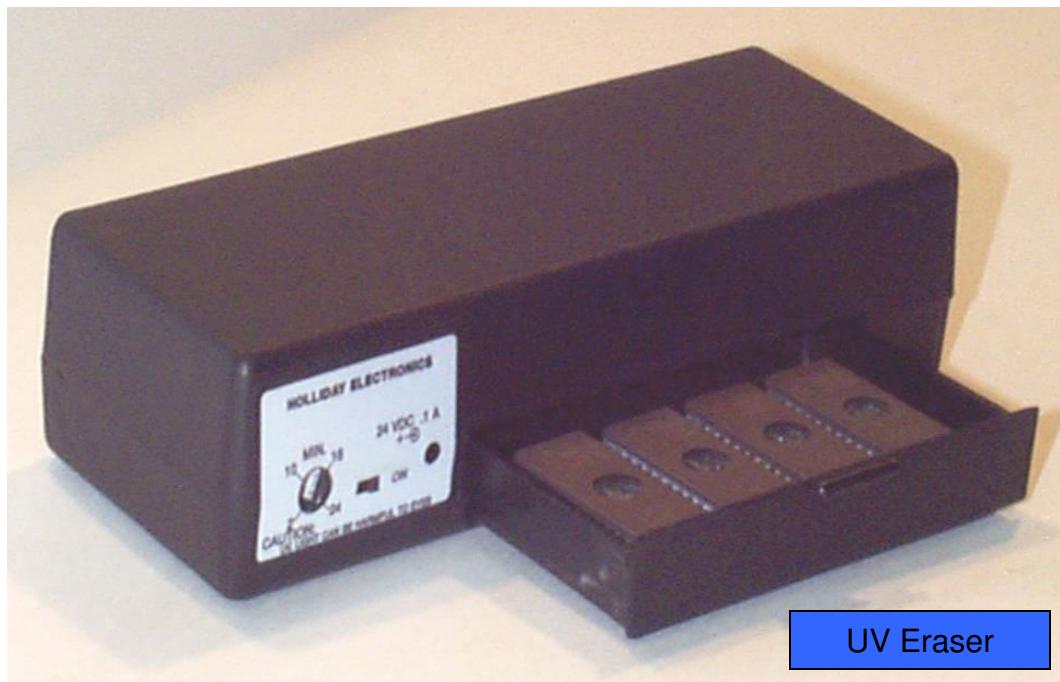


Ultra Violet Erasable Prom Technology UV-EPROMs

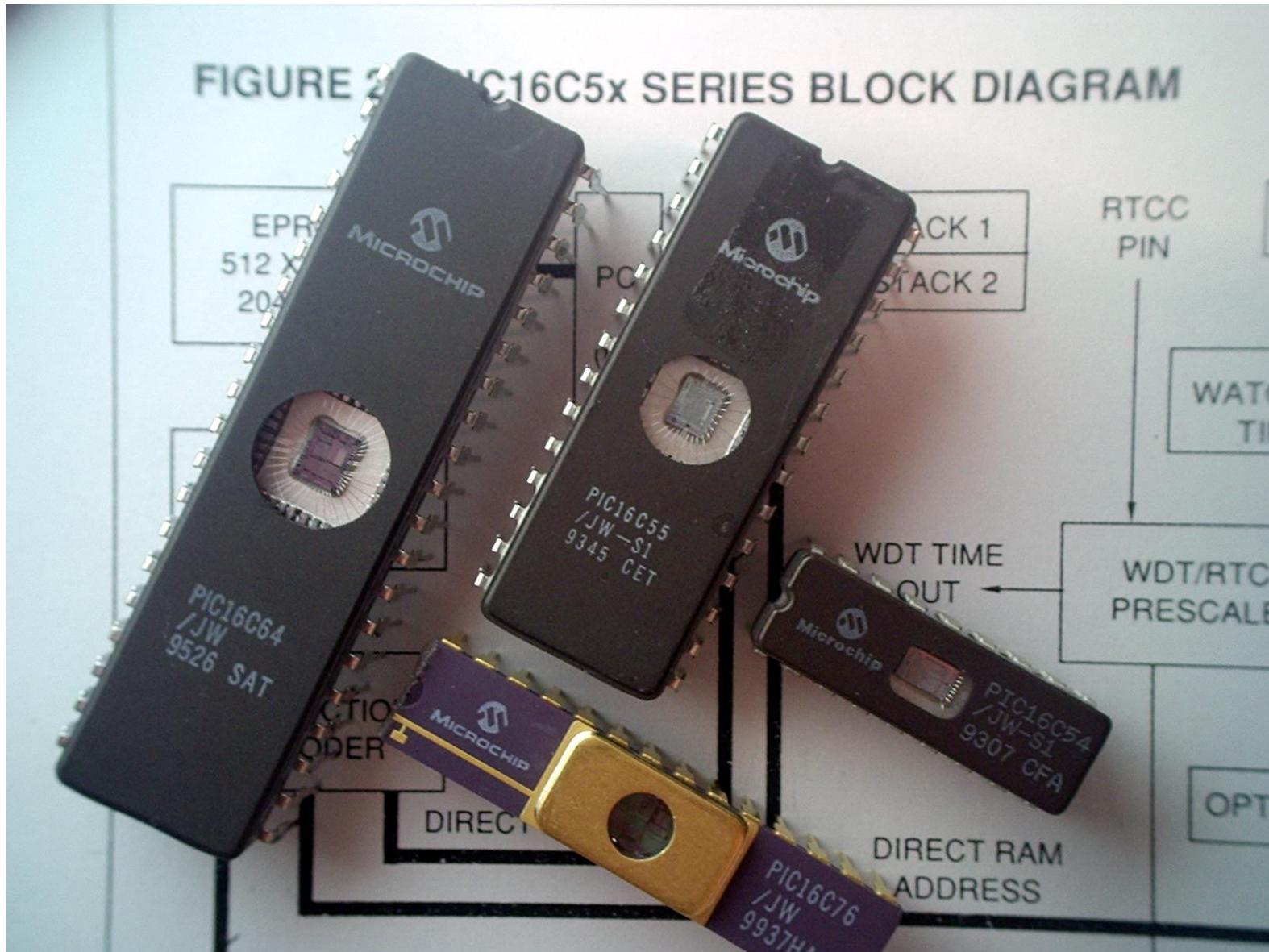


Ultra Violet Erasable Prom (UV-EPROM)

- The problem with **PROMs** for **development** purposes is that it only takes one mistake and a new chip is needed. A problem solved by the **UV-EPROM** which can be **erased** in about **15-20 mins** by exposure to a certain wavelength of UV light which also means they cannot be **accidentally** erased.
- **UV-EPROM's** thus have a small quartz glass window to allow the UV to expose the chip.
- **UV-EPROM's** are usually inserted into **sockets** on a circuit board to allow easy extraction for erasing and reprogramming purposes (which is costly).
- A small UV eraser is shown below with a **30 min timer**, and a draw for erasing up to **4 devices** at a time. They are programmed like Proms using a dedicated programmer device.

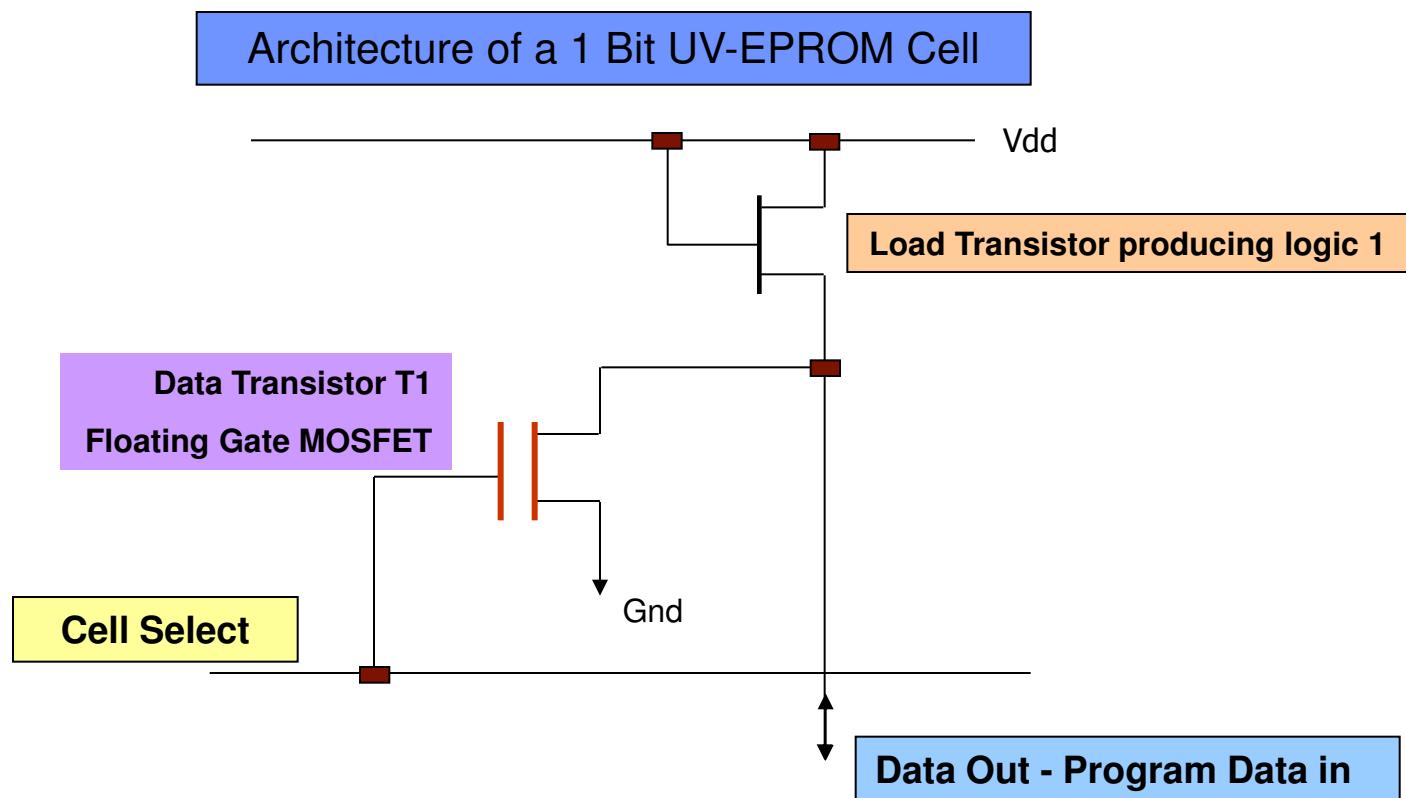


Early Microcontrollers with UV-EPROM for Firmware



Ultra Violet Erasable Prom (UV-EPROM)

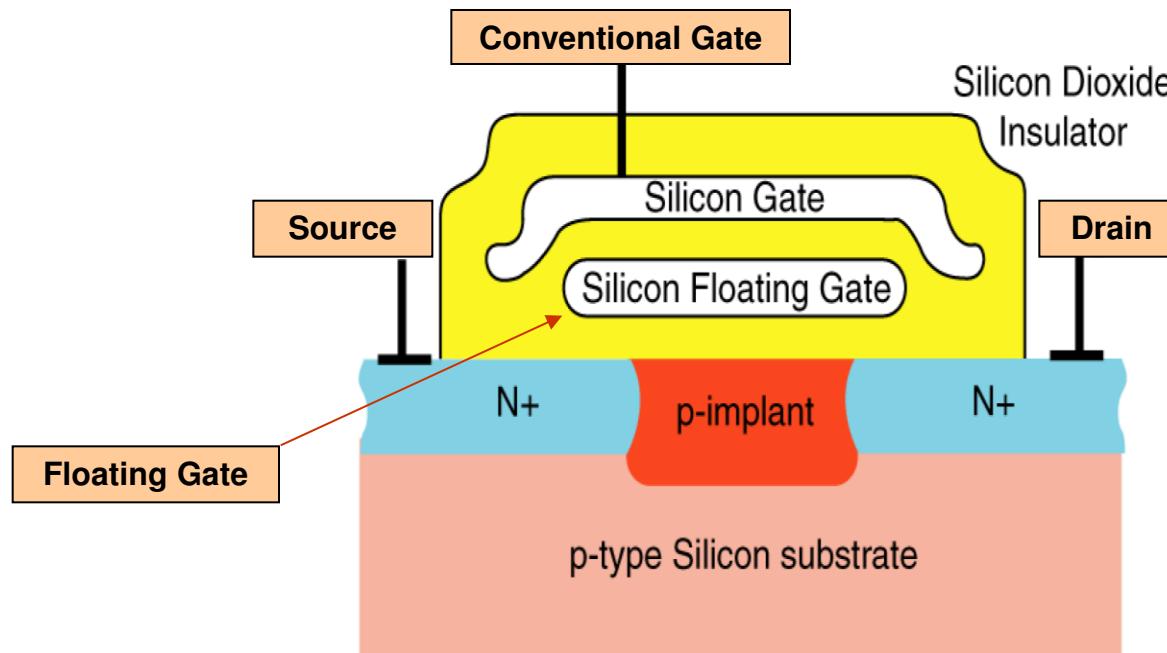
- The basic architecture of the UV-Eeprom is the same as the Prom/Rom except that the combination of **Fuse** and **Data transistor** has been replaced by a special **Floating Gate Mosfet**.



UV-EPROM Cell

Floating Gate MOSFET

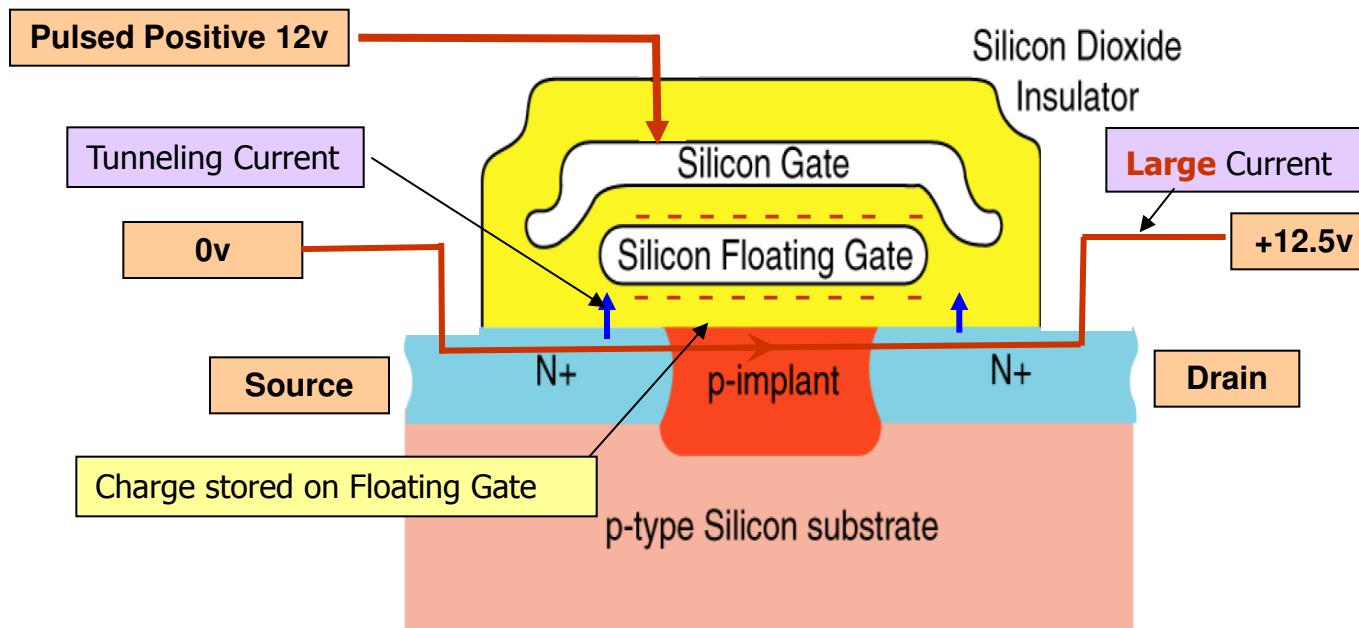
- The illustration below represents a cross section of a modified *n*-channel enhancement mode floating gate MOSFET controlled by two gates.
- This MOSFET represents the Data transistor in our previous slide. It consists of
 - A Floating Gate completely insulated by a “thin” Silicon Dioxide layer and with no electrical connection to the outside.
 - A conventional Control Gate placed above the floating gate, which is used to control the device externally.



UV-EPROM Cell

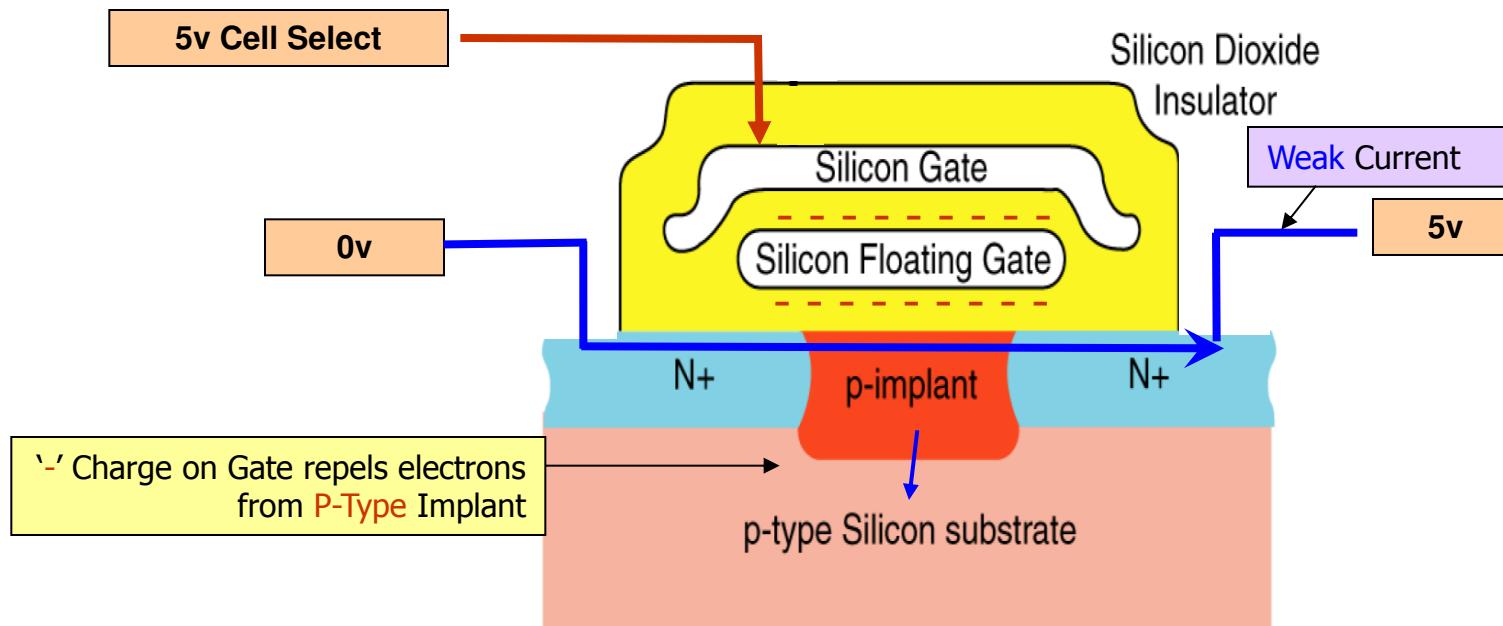
Programming a cell to store a Logic ‘1’

- To program a cell with a logic ‘1’, the Prom Programmer raises the Drain voltage for the cell to **12.5v** creating the potential for a large flow of electrons from source to drain.
- The Control Gate is pulsed **positive for ~50us** by pulsing **Chip Select** in conjunction with an address supplied to select the cell, (*both performed by the Prom Programmer*).
- Some electrons flowing between **source** and **drain** are attracted by the **+ charge** on the control gate and have sufficient **energy** to tunnel through the silicon dioxide insulator where they get trapped on the floating gate, a process known as “**Hot Electron Injection**”.
- After removal of the programming voltages, a charge is trapped on the floating gate.



UV-EPROM Cell - Reading

- After programming, the negative charge stored on the **floating gate** causes electrons to be **repelled** from the surface of the **P-type** implant.
- Because of the **depletion** of electrons in the P-Type implant, a conduction channel between Source and Drain is now **much harder to form** when a normal (5v) gate voltage is applied and only a **weak current** will flow as the device is only partially conducting.
- This means that the cell, acting as the data transistor, is **not** able to fully ‘pull down’ the load transistor to ground and thus a logic 1 is seen on the Data Output pin.
- A cell with **no** charge stored on its gate would fully conduct in the presence of a 5v signal applied to the **control gate** and **would** be able to short the load transistor to ground, creating a logic 0 at the data output pin.



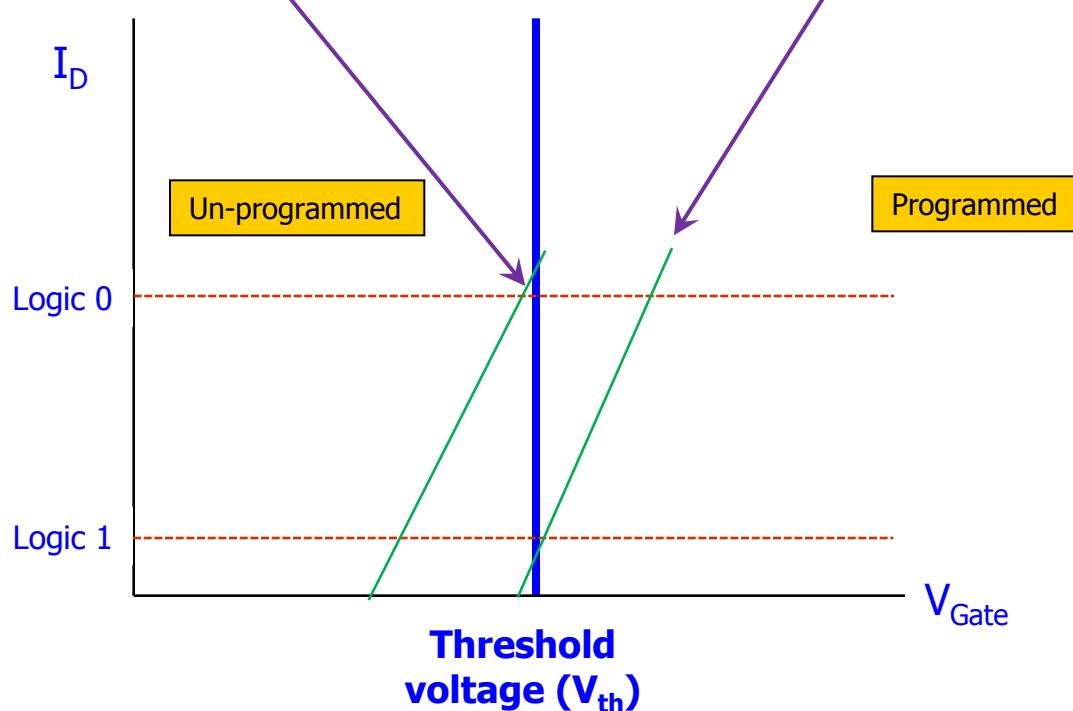
- In effect, the charge stored on the Floating Gate has altered the conduction threshold of the device, making it harder to turn the MOSFET on as shown in the two graphs below.

When **un-programmed**, device fully conducts when gate voltage = V_{th} .

Data Transistor is thus **able** to short load transistor to ground **Result = Logic 0** at data out pin

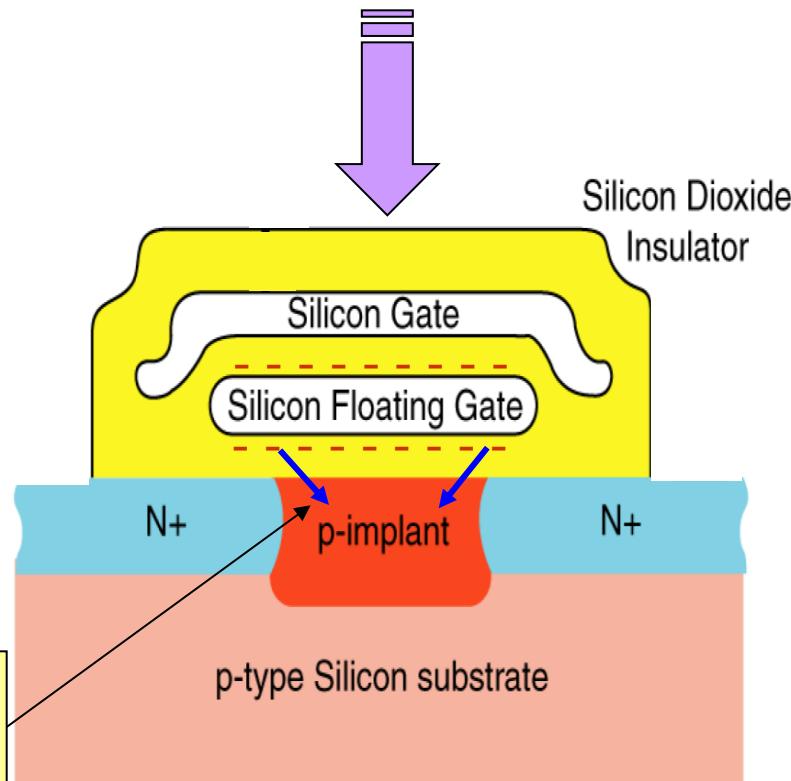
When **programmed**, device does not fully conduct when gate voltage = V_{th} .

Data Transistor is thus **unable** to short load transistor to ground **Result = Logic 1** at data out pin



UV-EPROM Cell - Erasing

UV Light (15-20mins)



Charge stored on Floating Gate
energised by UV-Light and able to
'jump' across insulator attracted
by P-type Implant

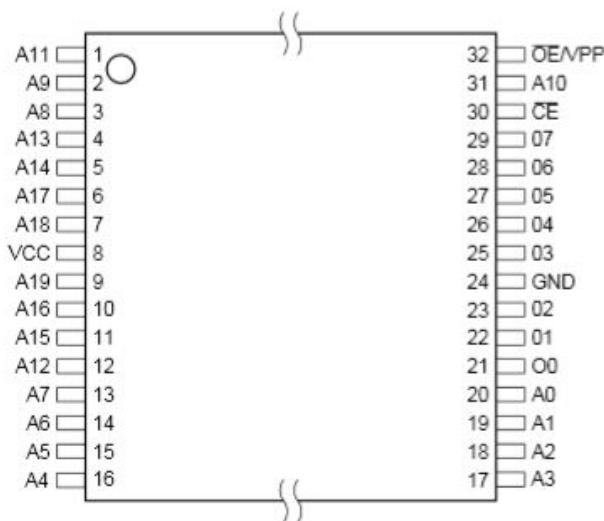
Ultra Violet Erasable Prom (UV-EPROM)

- **Atmel 27C080 – 1Meg x 8 bit UV-Eeprom (~\$9 today)**

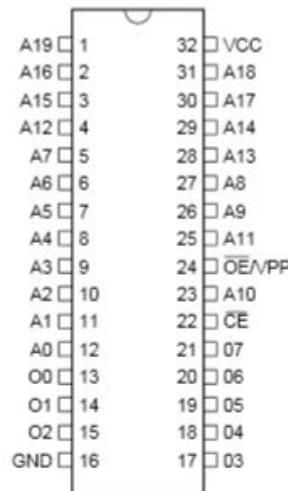
Pin Configurations

Pin Name	Function
A0 - A19	Addresses
O0 - O7	Outputs
CE	Chip Enable
OE	Output Enable

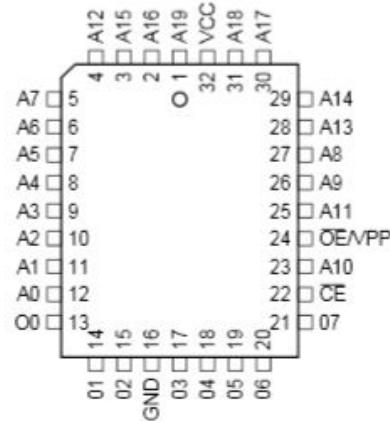
TSOP Top View
Type 1



CDIP, PDIP, SOIC Top View

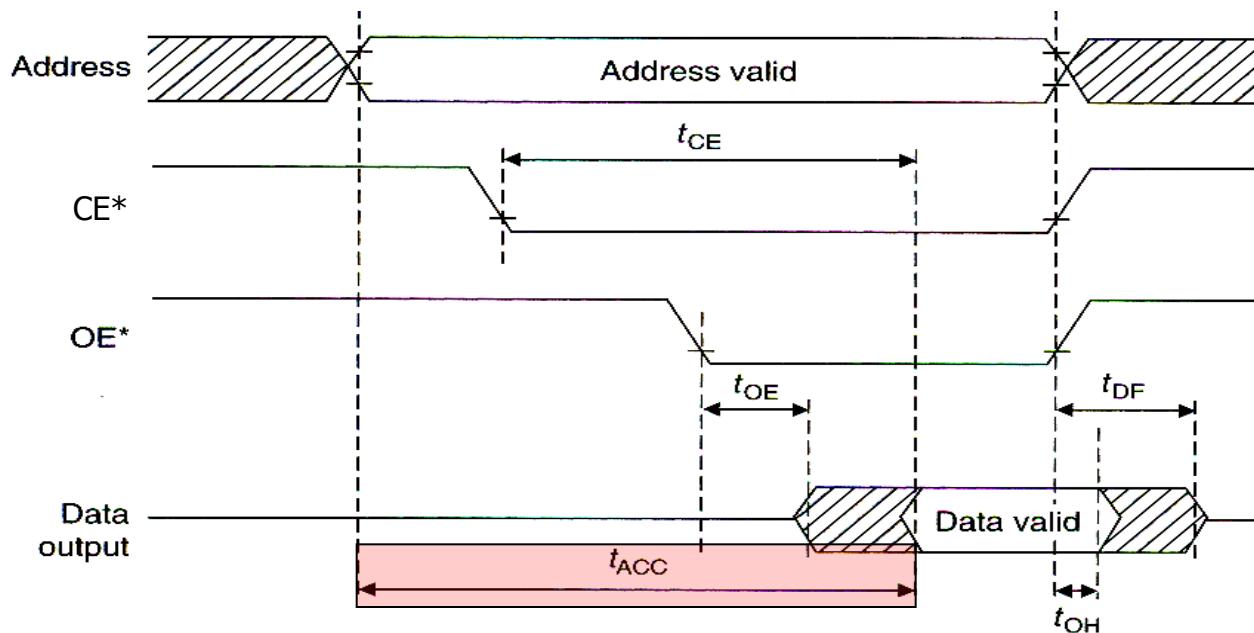


PLCC Top View



UV-Eeprom Read Cycle Timing

- Very straightforward.
- Identical Interface to a Static Ram (no R/W line though)
- Typical read access times for a 27080 are 90ns or better



Symbol	Parameter	27C080-90
t_{ACC}	Address valid to output valid	90 maximum
t_{CE}	CE* low to output valid	90 maximum
t_{OE}	OE* low to data bus floating	40 maximum
t_{DF}	OE*/CE* high to data bus floating	35 maximum

Electrically Erasable Prom Technology

Electrically Erasable PROM (EEPROM, E²PROM)

Introduction

- The big problem with **UV EPROM's** is that they needed to be **removed** from the circuit board to be erased and re-programmed and required special programmer/erasure equipment.
- In this day and age, where updates for device **firmware** are frequently available over the “**Net**”, a new generation of EPROM was needed that could be **programmed** and **erased** “**in-Situ**” (*also saving the significant cost of a **socket**.*)
- **EEProm (or E²PROM)** uses similar technology to UV EPROM i.e. a **floating gate MOSFET** to store a charge, but with an additional **single transistor per cell** able to create a sufficiently **large electric field** that the electrons on the floating gate could be **discharged**, i.e. cells can be erased without the need for **UV-Light** or removal of the chip from the PCB.

They also contain “on-chip” circuitry to generate the larger voltages required for programming **~15v**.

- Many small embedded microcontrollers have several hundreds of Kbytes of **EEProm**.

Electrically Erasable PROM (EEPROM, E²PROM)

EEProm Operation

- Can be erased and programmed “**in-situ**”, thus behaving more like RAM than PROM.
- A big advantage of **EEProm** (*over UV-Eeprom and even Flash Today*) is that **individual locations** within the chip can be selectively erased and programmed – no need to erase the whole chip making small updates faster.
- Can be hardware or software **protocol protected** to avoid accidental **erasure**.
- **Chip Erasure** involves a simple protocol of **writing commands** to the chip using a pattern of **abnormal signal assertions** (*i.e. different signaling than would normally be used during read/write operations*).

Electrically Erasable PROM (EEPROM, E²PROM)

Writing

- During a write cycle, internal circuitry **selectively erases** the individual **addressed cell(s)** prior to re-programming them with the new data, allowing selective **byte by byte** erase/reprogramming of each location.
- Write cycle times are of the order of **1 – 10 mSec** per location/address so re-programming a **complete device** can take **several minutes**.
- Chips may have some form of external **status pin** that can be **tested** to determine when a write operation has been completed, while some can be interrogated by **polling** the chip's internal **status register** in software.
- In essence your computer needs to run a program to *actually program* the chip, you cannot just write data to the chip at full speed like you would for a static ram – it takes much longer and is more complicated than that.

Electrically Erasable PROM (EEPROM)

Erasing

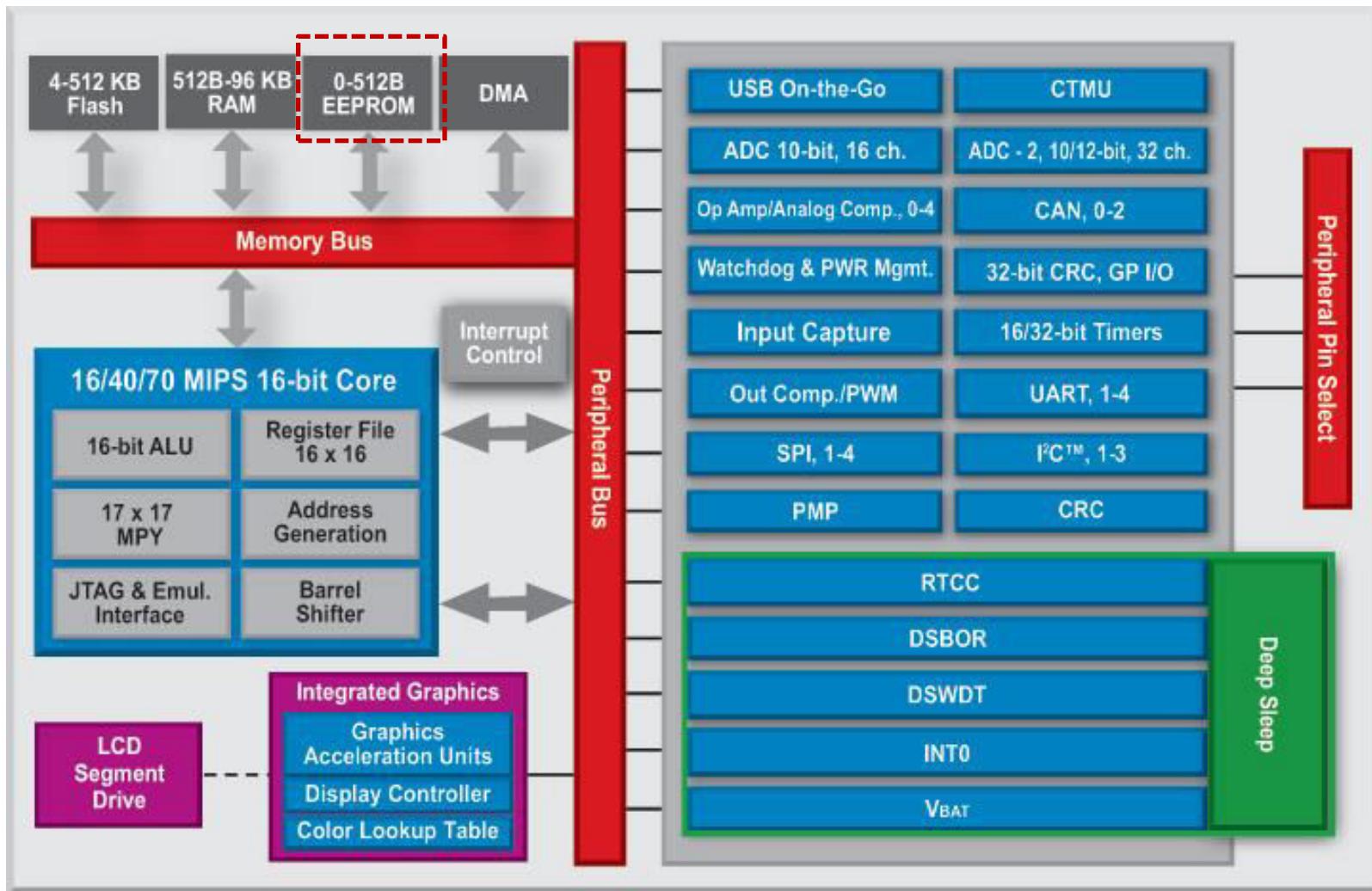
- Erase/reprogram cycles are limited to typically 10,000 – 100,000 operations per cell and, because of the high voltages and “hot electron injection” programming techniques, damage slowly occurs to the delicate floating gate and surrounding insulator i.e. they stop working.

Data Density vs UV Eprom

- Because selective, individual cell erasure requires extra circuitry, i.e. an additional 1 transistor per cell, data storage densities of the 1st generation EEPROMs were roughly half those of the previous generation UV-EPROM's fabricated using the technology of the day.
- Thus, with the introduction of EEPROM, storage densities *initially* went backwards.

Microcontroller with Electrically Erasable PROM (EEPROM)

- Microchip - PIC24 16-bit MCUs



Microchip 128k Byte Serial EEPROM on I²C bus (or SPI)



English

Search Microchip

Search Data Sheets

PRODUCTS APPLICATIONS DESIGN SUPPORT TRAINING SAMPLE & BUY ABOUT US Contact Us myMicrochip Login

24AA1025 In Production

Buy it Now



Documentation & Software

Pricing & Samples

Development Tools

Similar Products

Quick Links

24AA1025/24LC1025/24FC1025 1024K I²C CMOS Serial EEPROM Data Sheet (08/09/2013)

The Microchip Technology Inc. 24AA1025 is a 128K x 8 (1024K bit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.7V to 5.5V). It has been developed for advanced, low-power applications such as personal communications or data acquisition. This device has both byte write and page write capability of up to 128 bytes of data.



Features

- **Reliable EEPROM Memory**
 - 128K x 8 (1024 Kbit)
 - 128-Byte Page Write Buffer
 - Page Write Time 5 ms Max.
 - Hardware Write-Protect Pin
 - Factory Programming Available
- **Low Power**
 - Operating voltage 1.7V to 5.5V
 - Read current 450 uA, max.
 - Standby current 5 uA, max.
- **2-Wire Serial Interface, I²C™ Compatible**
 - Cascadable up to Four Devices
 - 100 kHz and 400 kHz Clock Compatible
- Pb-Free and RoHS Compliant

Parameter Name

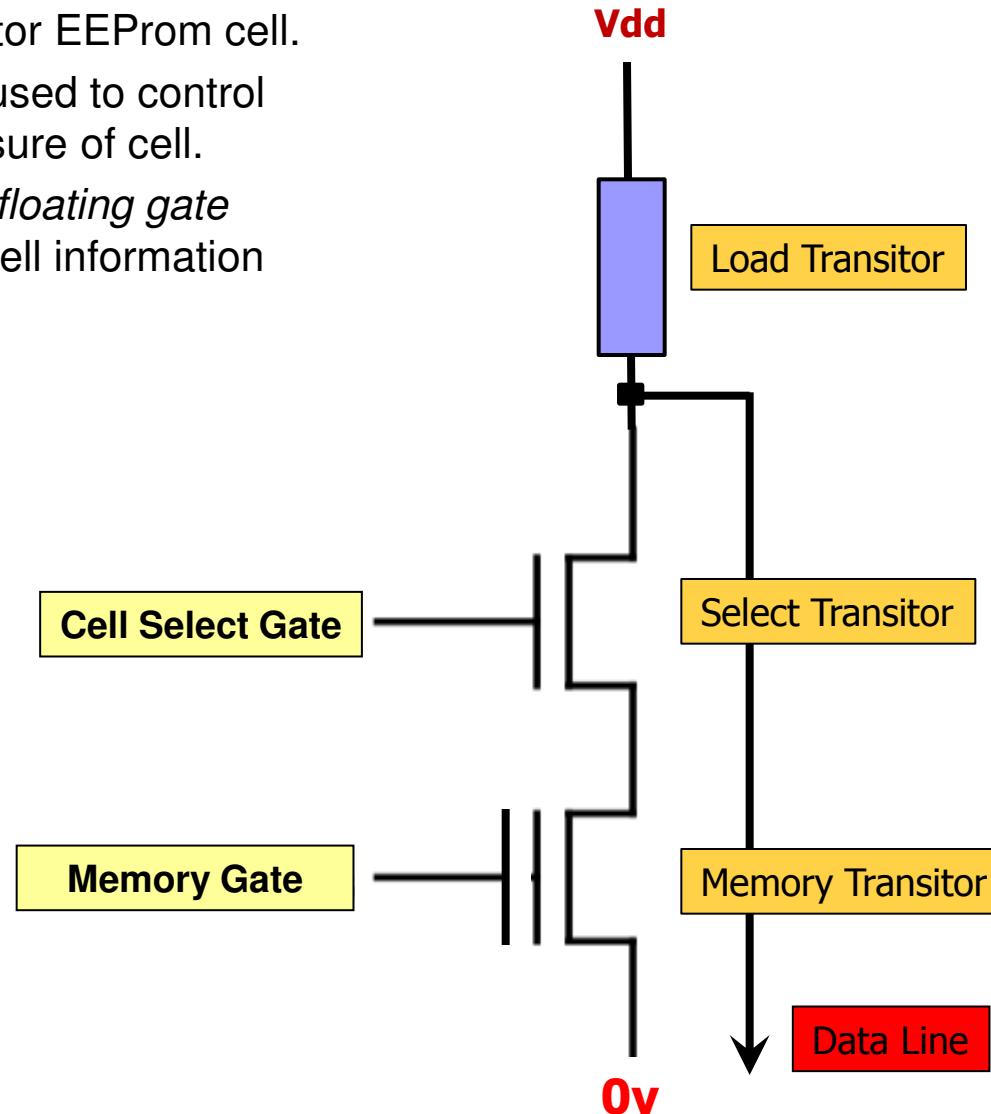
Parameter Name	Value
Density	1024K bit (x8)
Op. Volt Range (V)	1.7 to 5.5
Max. Clock Freq.	400 kHz
Page Size (bytes)	128
Write Protect	Full Array
Temp Range (°C)	-40°C to +85°C
Endurance	1,000,000
Data Retention (Years)	200

cost: \$4.00

Electrically Erasable PROM (EEPROM)

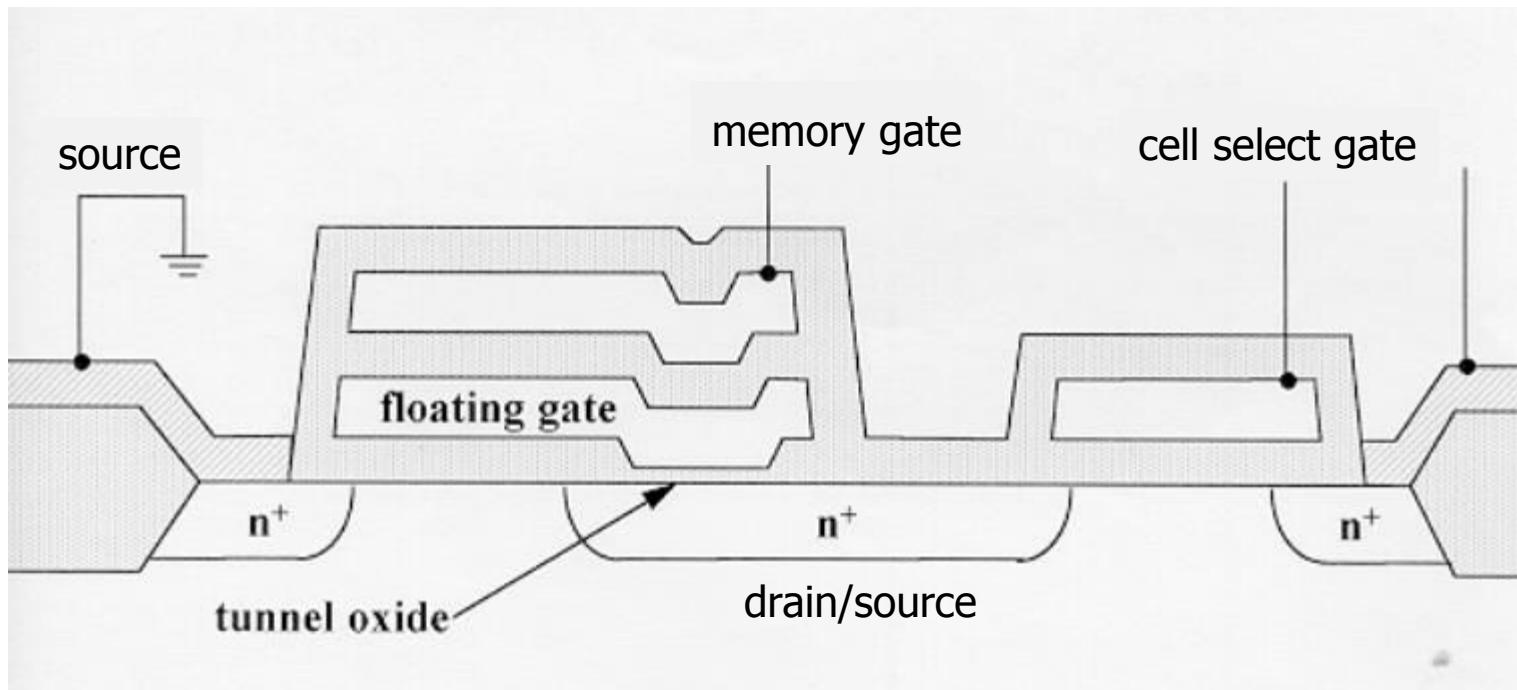
Architecture of an EEPROM Cell

- Illustration of 2 transistor EEPROM cell.
- A **Select** transistor is used to control programming and erasure of cell.
- A **Memory** transistor (*floating gate MOSFET*) holds the cell information (0 or 1)



Electrically Erasable PROM (EEPROM)

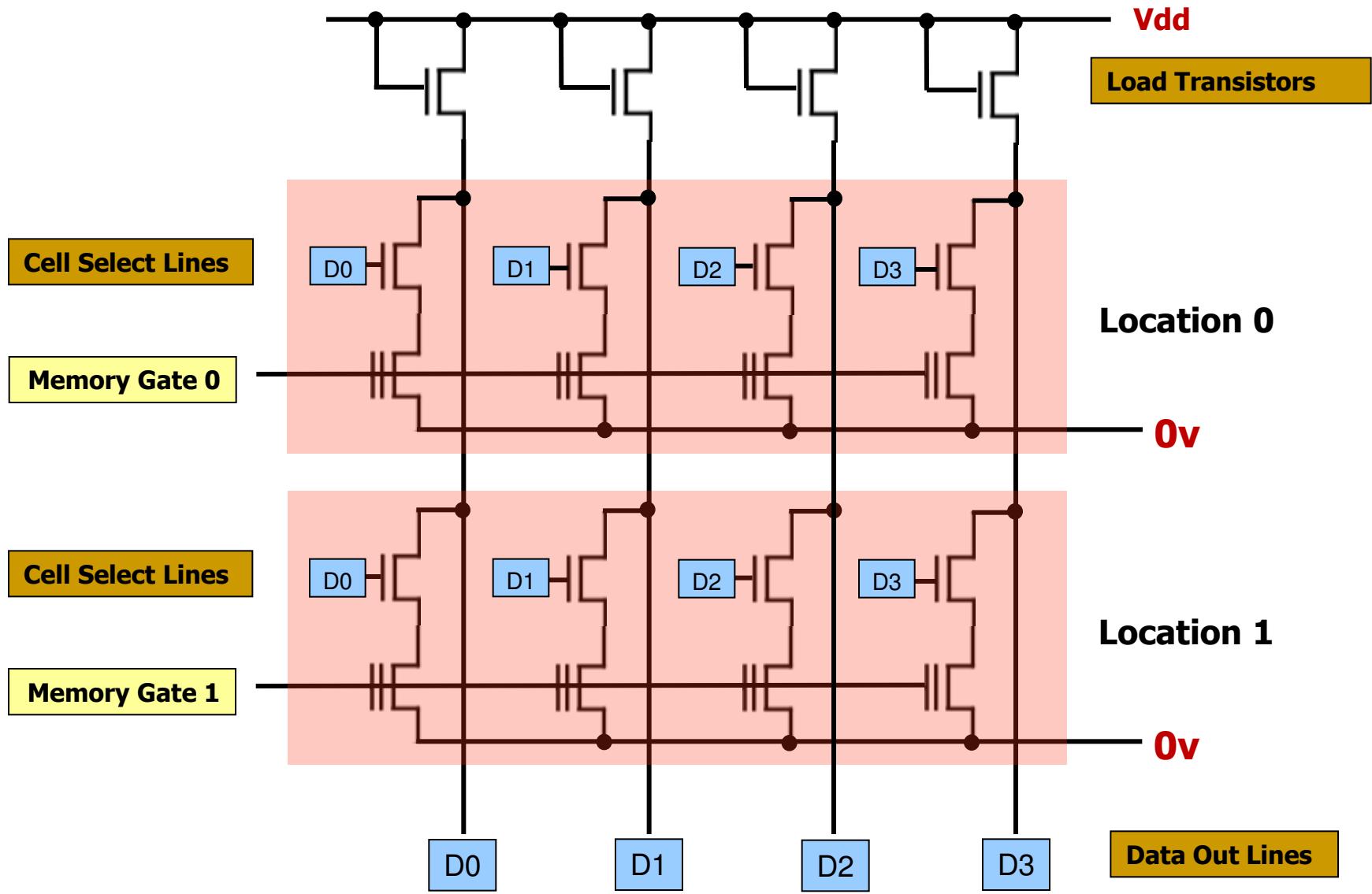
- Cross section of an EEPROM cell showing the small tunneling gap between Drain and floating gate.



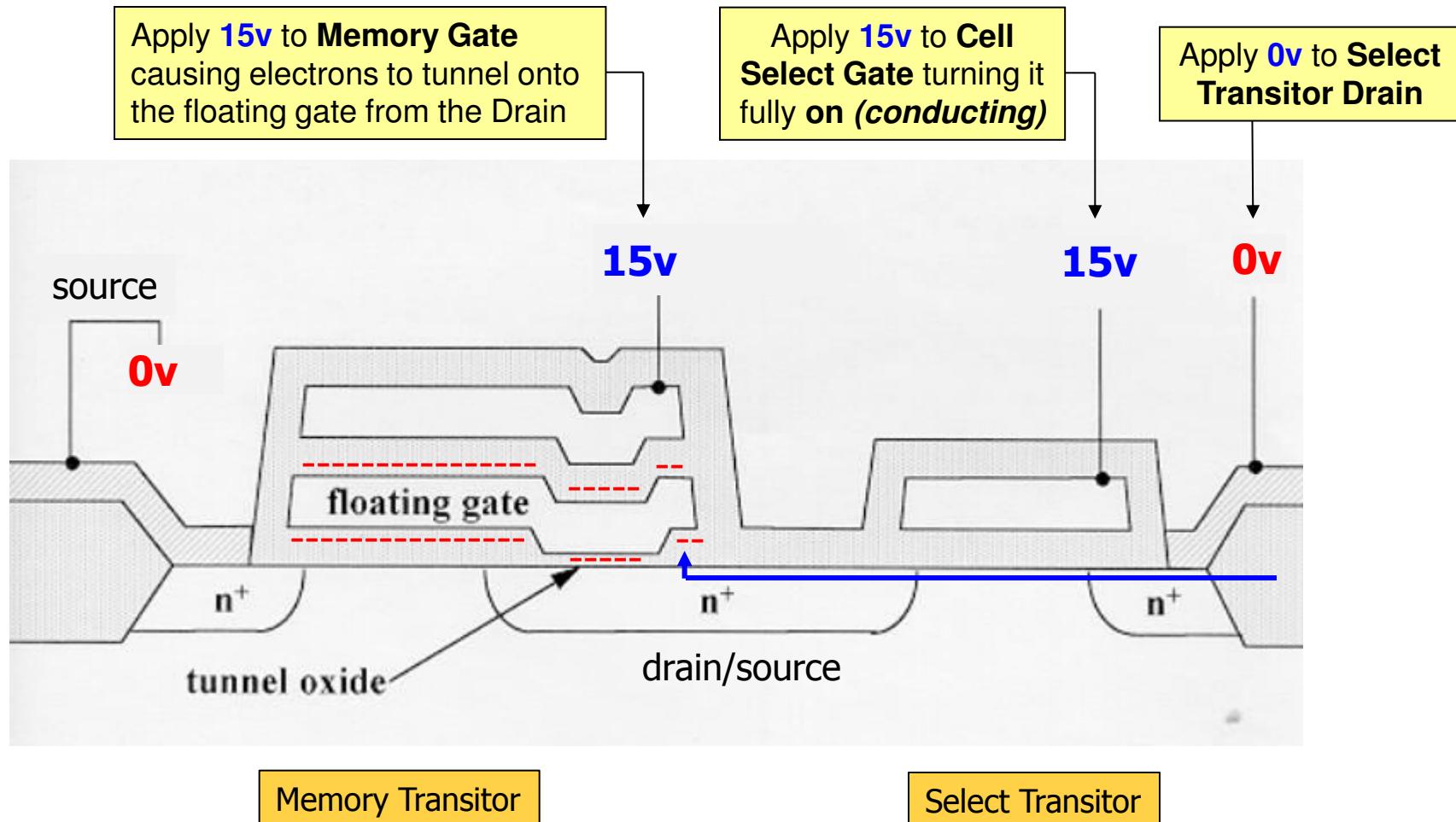
Memory Transistor

Select Transistor

Architecture of 2 x 4 Bit Electrically Erasable PROM (EEPROM)

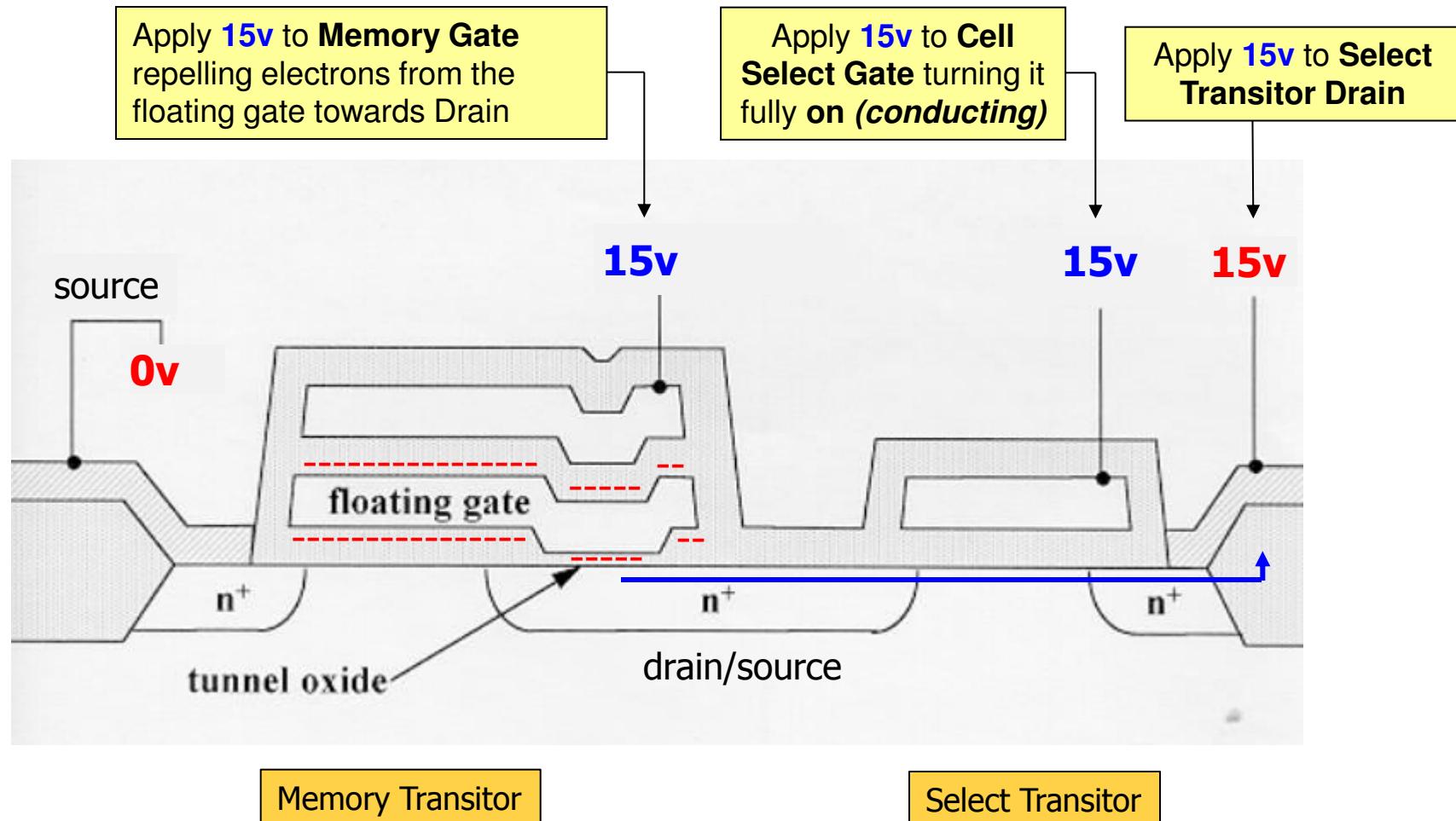


Programming Step 1 – Erase the EEPROM Cell (to Logic 1)



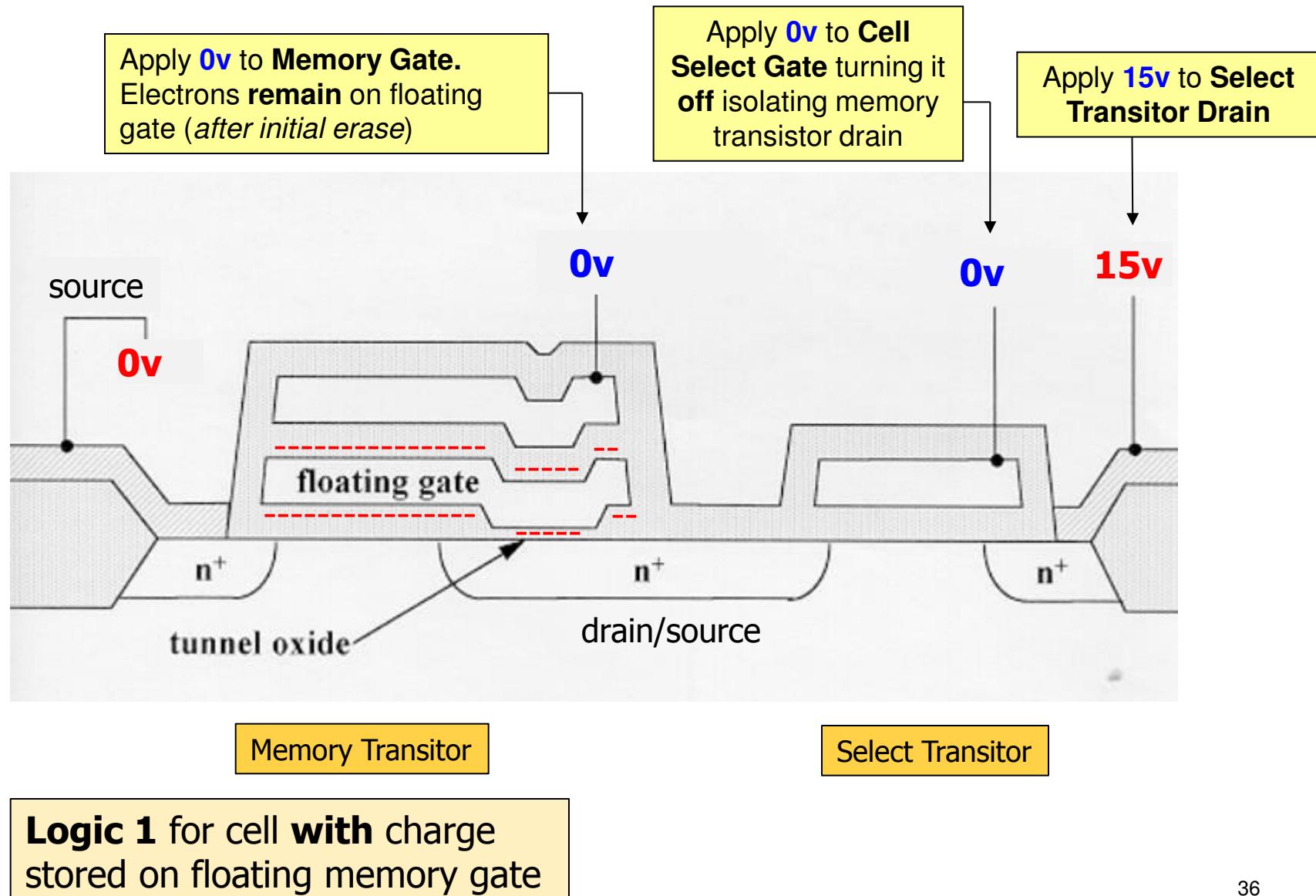
Logic 1 for erased cell when charge is stored on floating memory gate

Step 2 - Programming an EEPROM Cell (to Logic 0)

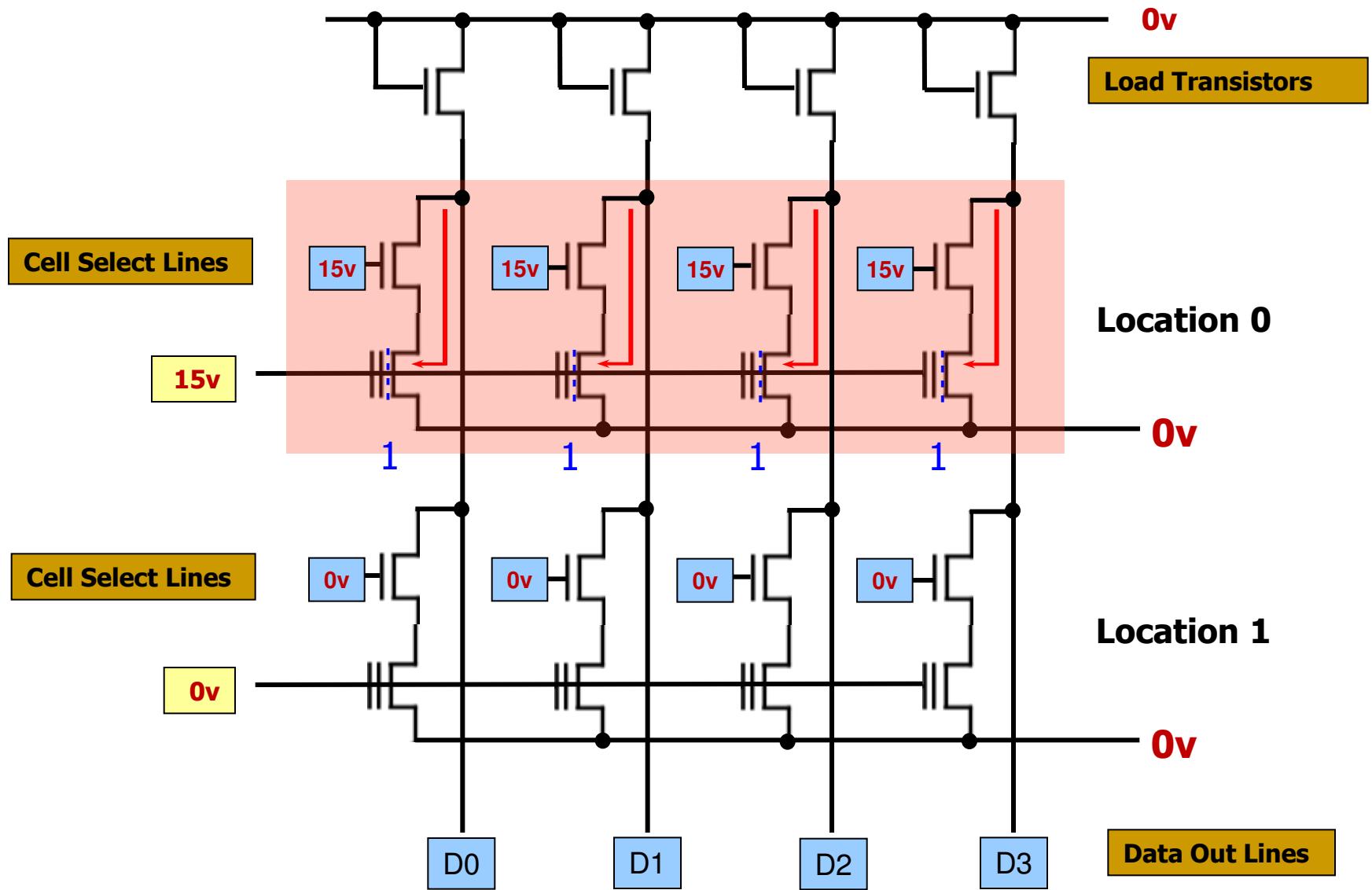


Logic 0 for cell with **NO** charge stored
on floating memory gate

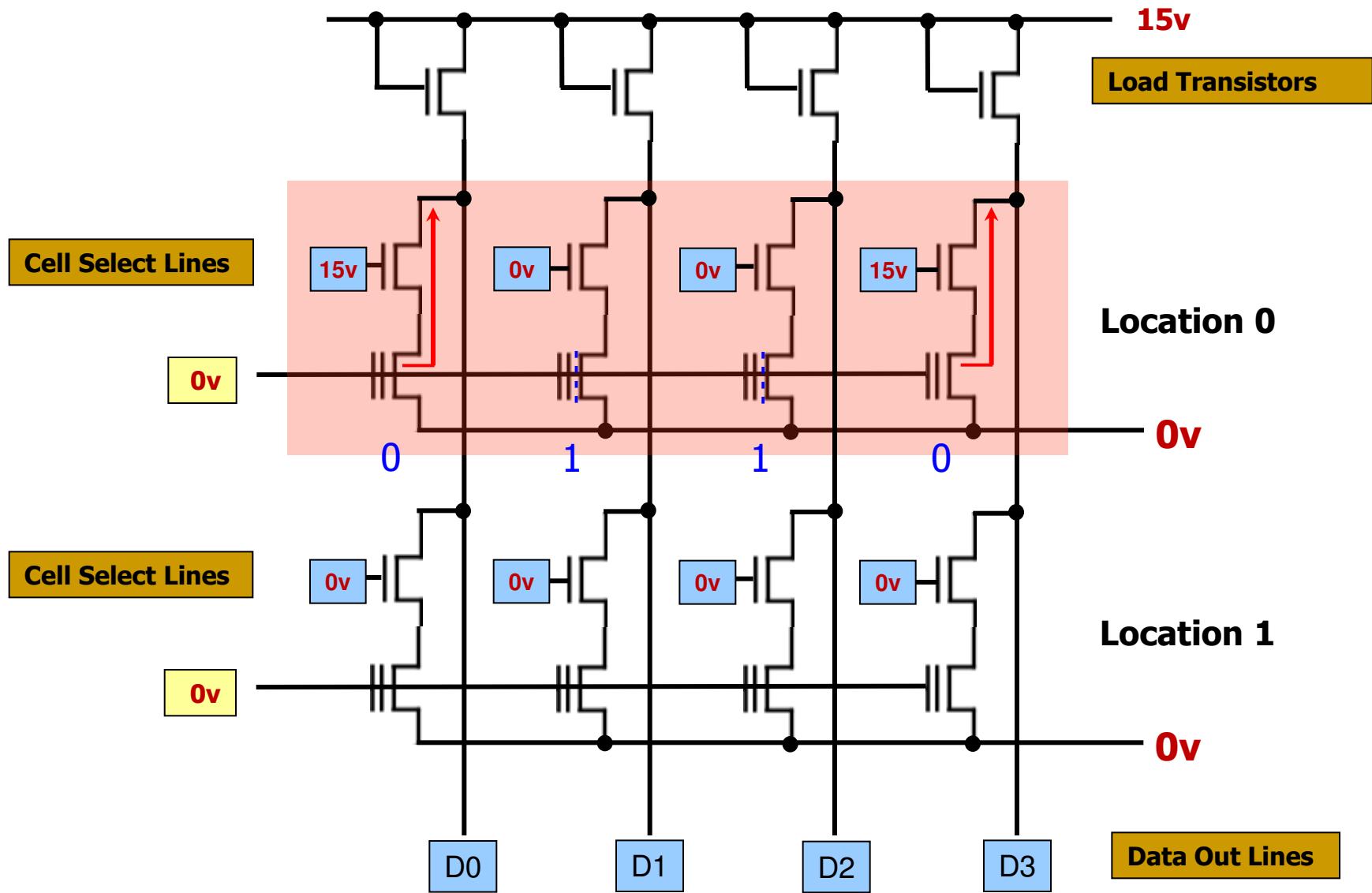
Step 2 - Programming an EEPROM Cell (to Logic 1)



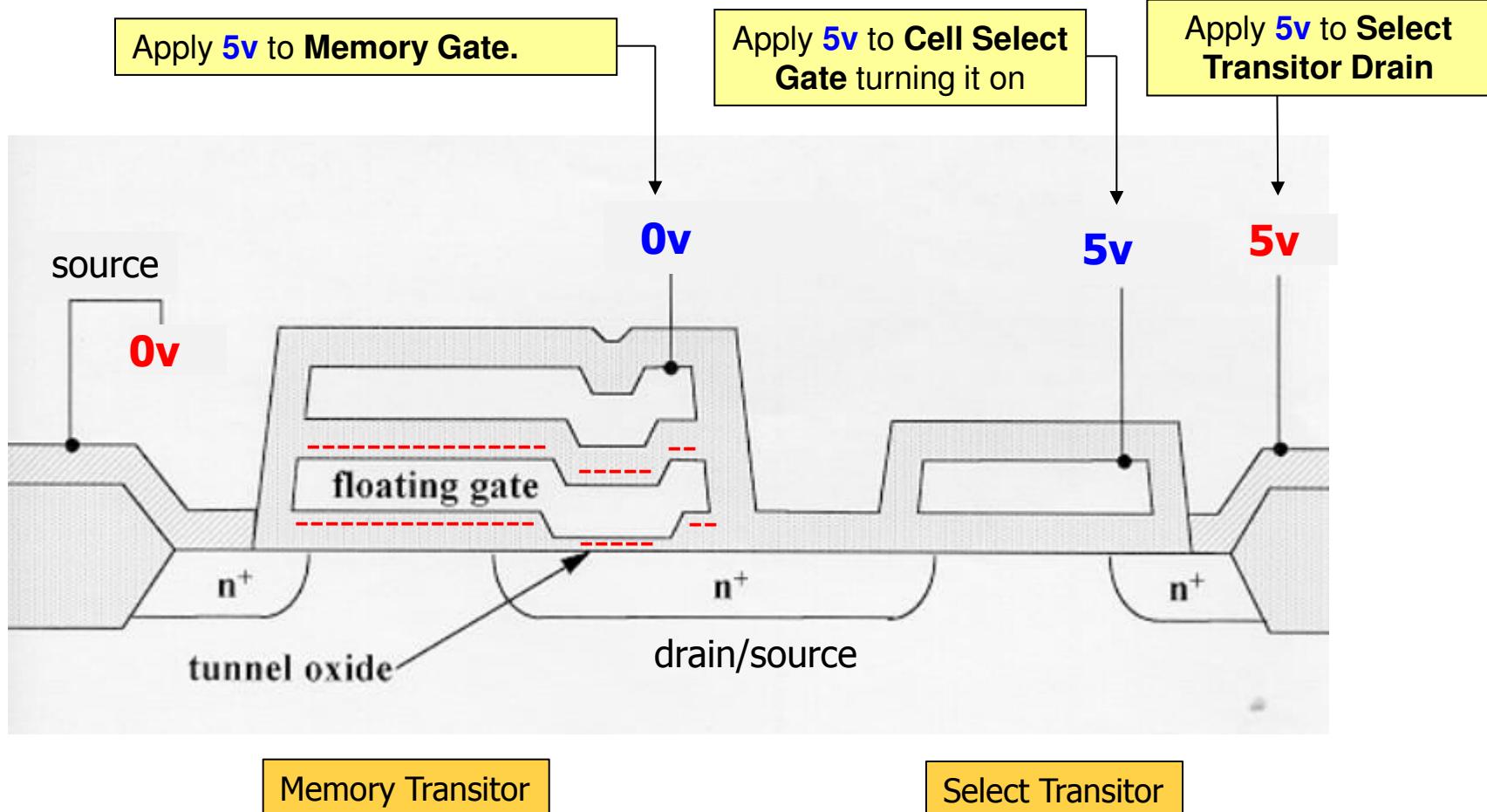
Erasing an EEPROM Location



Programming an EEPROM Location with “0110”

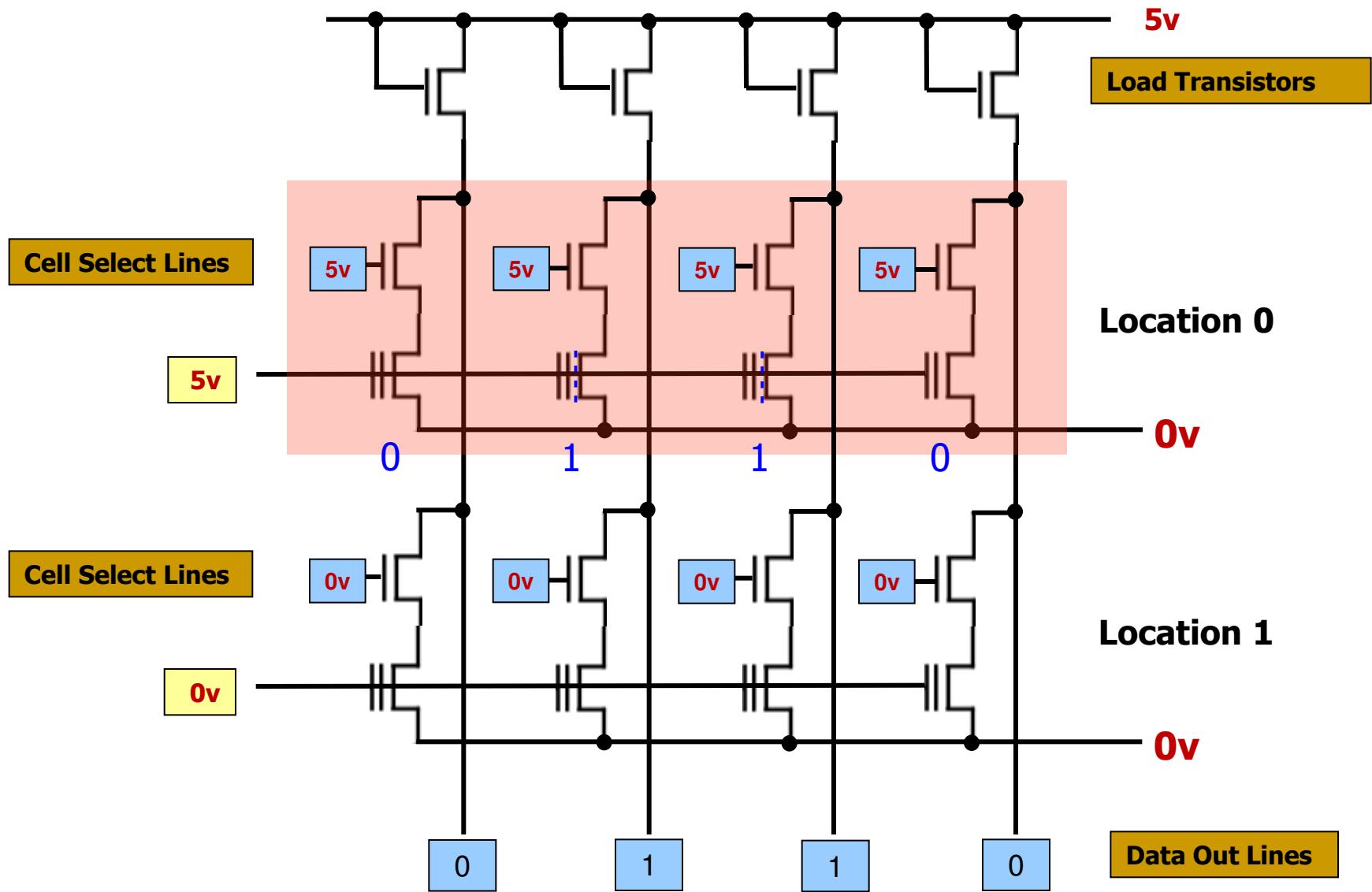


Reading the EEPROM cell



Logic 0 for cell with **NO** charge stored on memory floating gate.
Logic 1 for cell **with** charge stored on memory floating gate.

Reading an EEPROM Location



Flash (*ah aaaahh*) Technology



Flash EPROM

Flash Memory

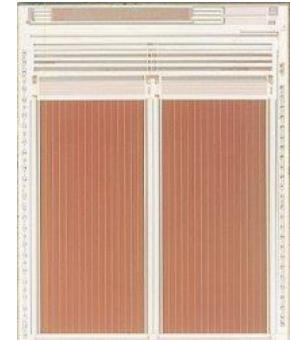
- A further refinement to conventional UV and E²PROM technology.
- Changes from E²PROM to Flash included restructuring the devices internally into separate addressable banks or sectors (*only relevant during programming and erasure, not during reading where they are still randomly accessible*).
- Still electrically erasable and programmable in-situ but the individual erase transistors of E²PROM were moved to the edges of chip where they now controlled the erasure of a whole bank of cells rather than individual cells
- This reduced the transistor count per bit of storage within the chip, leading to practically doubling the bit density over previous generation EEPROM.
- As a consequence, write operations require that a whole Block/Sector first need to be erased, and then re-written as a block. Thus it is not possible to just selectively erase an individual cell (*something that was possible with EEPROM technology*.)
- Flash still has a finite number of erase/write cycles (10^4 - 10^6 per block).

Flash EPROM

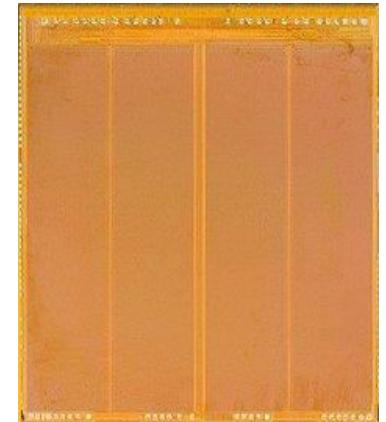
Competing E²PROM Technology: NOR vs. NAND.

- NOR invented by Intel and NAND by Toshiba.
- Differences relate to
 - Layout of memory cells in relation to each other on the chip – in *series* or *parallel* which in turn governs the bit **density** and hence **cost** per Mbyte of that technology.
 - NAND flash chips with memory cells connected in series have higher bit densities and thus on any given day, NAND flash is **more dense** than NOR.
 - Most important difference relates to the external interface the chip present to the CPU reflecting their different use and applications: *random* access (NOR) vs. *sequential* access (NAND).

See http://en.wikipedia.org/wiki/Flash_memory



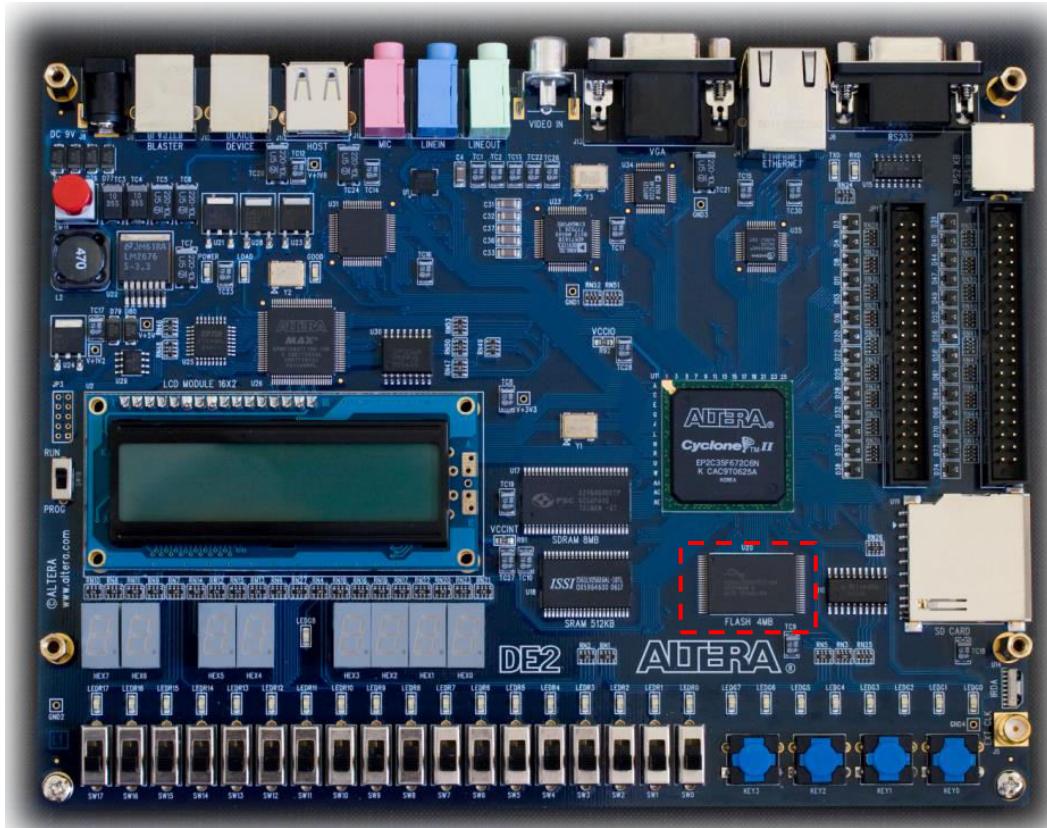
32 MB Nand Device



16 MB Nor Device

NOR Flash EPROMs

- NOR flash memory technology dominates the embedded firmware market as used for example on the LCP2294 ARM based microcontroller from Phillips EEPROMs.
- Low read latency compared to Nand flash means they are frequently used for storing BIOS in PC motherboards or firmware driver code in intelligent peripherals such as DVD-Writers etc. That is the CPU can run code directly from Nor flash device
- Current state of the art for this technology is ~8Gbits per chip



4MB NOR Flash EPROM
on the DE2 used to hold
Boot code or firmware

Example Microcontroller with Flash Memory

XMC4000 Family – (2012) 32bit ARM based microcontroller family from Infineon



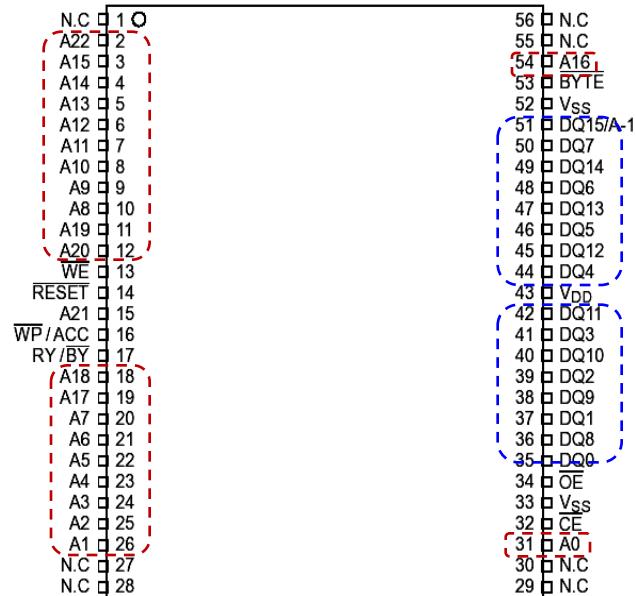
Chip	Performance				Timers				Signal processing		
	Clock	Flash	RAM	Cache	POSIF	CCU4 (4ch)	CCU8 (4ch)	High-res PWM	ADC 12-bit	Delta-sigma demodulator	DAC
XMC4100	80 MHz	128 kB	20 kB	4 kB	1	2	1	4	2		2
XMC4200	80 MHz	256 kB	40 kB	4 kB	1	2	1	4	2		2
XMC4400	120 MHz	512 kB	80 kB	4 kB	2	4	2	4	4	4	2
XMC4500	120 MHz	1 MB	160 kB	4 kB	2	4	2		4	4	2

NOR Flash EPROMs

Characteristics and Operation of NOR Flash devices

- Provides a conventional **memory/CPU interface** with **address** and **data buses** and thus interface directly to a CPU like conventional Rams and EPROM's.
- Good for storing embedded firmware that can be **executed directly** from the chip. (**Random Access Read**)
- Read access times *comparable* to conventional Static Ram technology allowing CPU to execute code directly from flash with no speed penalty (*important in small CPUs with no instruction caches*).
- Block erase** followed by **Random Access Write**.
- Write performance is relatively slow (~ 100 - 300uS per **location**), hence writing a complete chip may take several minutes but in the area of firmware upgrades, programming speed is not usually an important issue as anyone who has upgraded their PC BIOS will tell you.
- Less efficient *architecture* compared to **NAND** technology (i.e. bigger transistors wired differently) means the equivalent storage capacity **NOR** device will occupy **considerably more silicon** area than **NAND**.
- Guaranteed** by manufacturer to store data **reliably**.

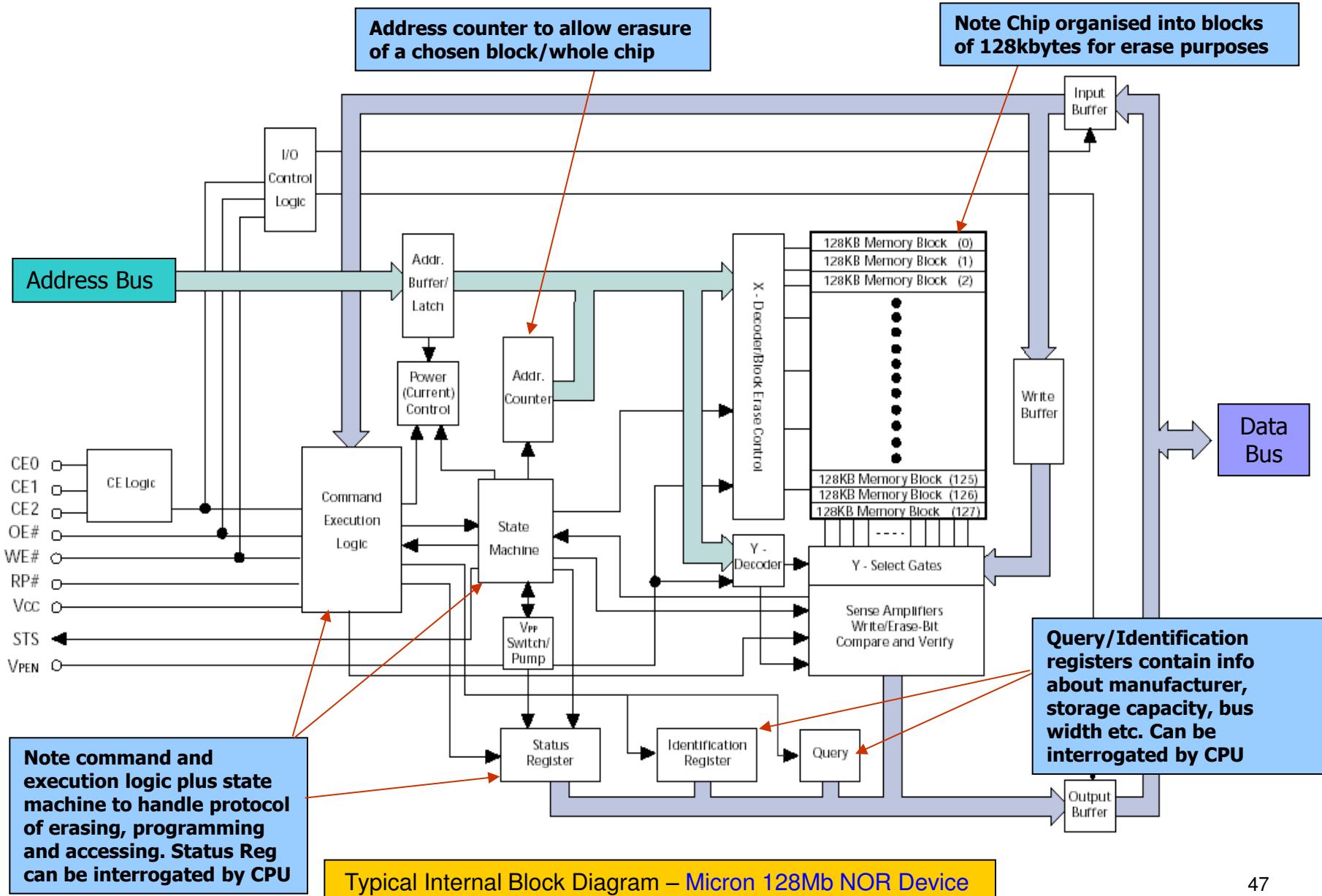
TC58FVM7(T/B)2AFT(65/80)



Toshiba 4M x 16 NOR CMOS
FLASH MEMORY

A-1, A0~A22	Address Input
DQ0~DQ15	Data Input/Output
\overline{CE}	Chip Enable Input
\overline{OE}	Output Enable Input
\overline{BYTE}	Word/Byte Select Input
\overline{WE}	Write Enable Input
$\overline{RY/BY}$	Ready/Busy Output
RESET	Hardware Reset Input
$\overline{WP/ACC}$	Write Protect / Program Acceleration Input
V _{DD}	Power Supply
V _{SS}	Ground

Internal Architecture of a NOR Flash Device



Controlling a NOR Flash Device

- A **Common Flash Interface (CFI)** standard exists which describes operating parameters and data pertinent to that specific chip/manufacturer. Your computer can read that data directly from the chip with a **CFI Query** command followed by **read** operations.
- The device operation is controlled by writing **commands** to the device.
- The value on the **data bus** defines the **command**, while the **address (if relevant)** refers to a **block** (or sector) within the chip (**SA**) and/or **physical address (PA)** or cell within the device.
- Multiple write commands are frequently required to avoid accidental command recognition. These commands follow a strict protocol of address and data otherwise it is abandoned.
- The protocol below is taken from the data sheet for the **Spansion** chip on the DE2 board.

Default 'read' mode on power up or write 'FF' to the device

Program requires 4 bus cycles.

Chip and Sector erase operation requires 6 write cycles.

Command Sequence (Note 1)	Cycles	Bus Cycles (Notes 2–5)											
		First		Second		Third		Fourth		Fifth		Sixth	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Read (Note 6)	1	RA	RD										
Reset (Note 7)	1	XXX	F0										
Autoselect (Note 8)	Manufacturer ID	4	555	AA	2AA	55	555	90	X00	01			
	Device ID, Model 03	4	555	AA	2AA	55	555	90	X01	22F6			
	Device ID, Model 04	4	555	AA	2AA	55	555	90	X01	22F9			
	Secured Silicon Sector Factory Protect, Model 03 (Note 9)	4	555	AA	2AA	55	555	90	X03	8D/0D			
	Secured Silicon Sector Factory Protect, Model 04 (Note 9)	4	555	AA	2AA	55	555	90	X03	9D/1D			
	Sector Protect Verify (Note 10)	4	555	AA	2AA	55	555	90	(SA)X02	(Note 10)			
Enter Secured Silicon Sector Region	3	555	AA	2AA	55	555	88						
Exit Secured Silicon Sector Region	4	555	AA	2AA	55	555	90	XXX	00				
CFI Query (Note 11)	1	55	98										
Program	4	555	AA	2AA	55	555	A0	PA	PD				
Unlock Bypass	3	555	AA	2AA	55	555	20						
Unlock Bypass Program (Note 12)	2	XXX	A0	PA	PD								
Unlock Bypass Reset (Note 13)	2	XXX	90	XXX	00								
Chip Erase	6	555	AA	2AA	55	555	80	555	AA	2AA	55	555	10
Sector Erase	6	555	AA	2AA	55	555	80	555	AA	2AA	55	SA	30
Erase Suspend (Note 14)	1	XXX	B0										
Erase Resume (Note 15)	1	XXX	30										

Controlling a Typical NOR Flash Device

Polling Device Status

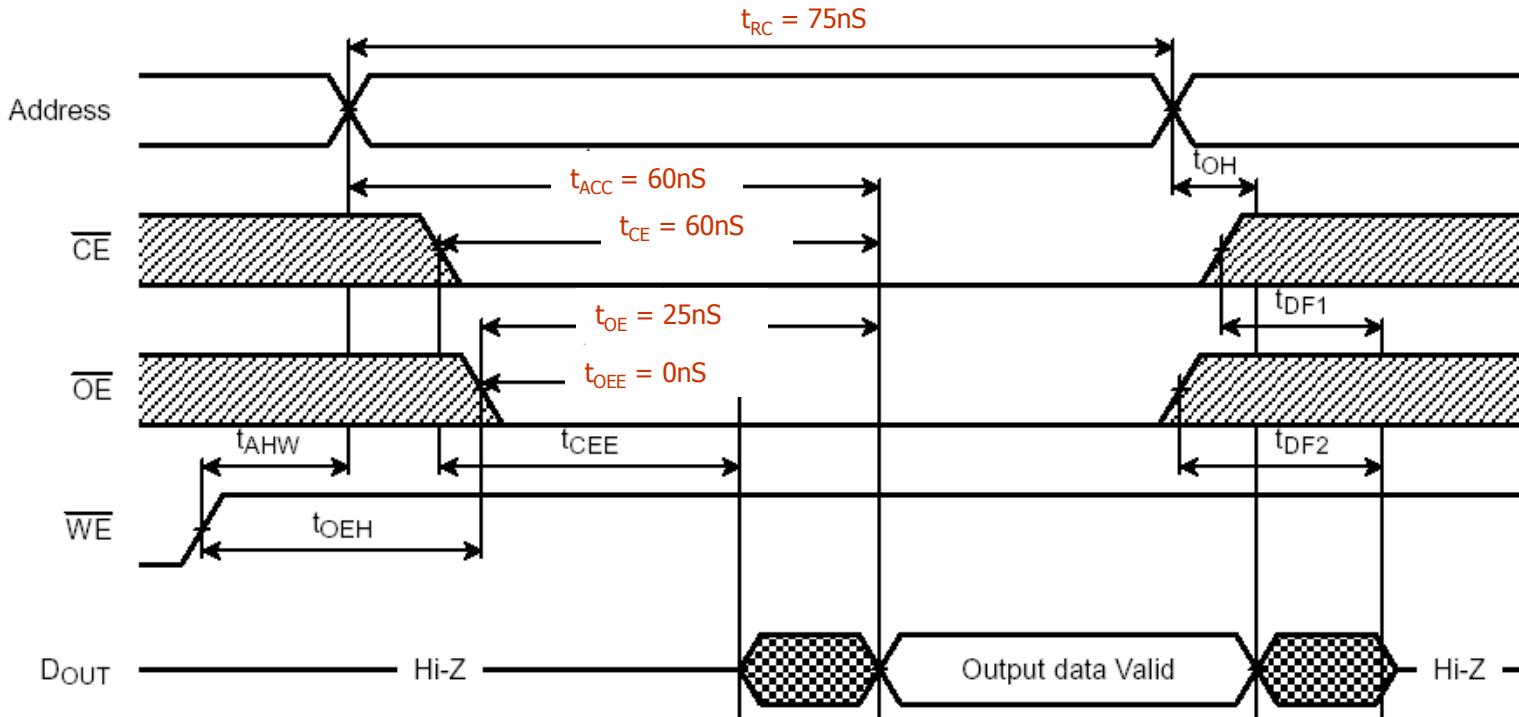
- Status interrogation of the chip (*for instance during programming or erasing etc*) is performed in different ways depending upon the chip.
 - In the example opposite, the chip has a status register (*see schematic - page 27*) which can be read at any time to give an indication of different status conditions in the chip.
 - The most important bit in the status register is **SR7** which indicates if the device is busy or idle.
 - For some devices you determine if a write has been completed by **reading back** from the same address.
 - If the chip gives you the same data you wrote in bit 7 it has finished, if not it is busy. Check **individual** data sheets.

ISMS	ESS	ECLBS	PSLBS	VPENS	PSS	DPS	R
7	6	5	4	3	2	1	0
HIGH-Z WHEN BUSY?	STATUS REGISTER BITS					NOTES	
No	SR7 = WRITE STATE MACHINE STATUS (ISMS) 1 = Ready 0 = Busy					Check STS or SR7 to determine block erase, program, or lock bit configuration completion. SR6–SR0 are not driven while SR7 = 0.	
Yes	SR6 = ERASE SUSPEND STATUS (ESS) 1 = Block Erase Suspended 0 = Block Erase in Progress/Completed						
Yes	SR5 = ERASE AND CLEAR LOCK BITS STATUS (ECLBS) 1 = Error in Block Erasure or Clear Block Bits 0 = Successful Block Erase or Clear Lock Bits					If both SR5 and SR4 are "1s" after a block erase or lock bit configuration attempt, an improper command sequence was entered.	
Yes	SR4 = PROGRAM AND SET LOCK BIT STATUS (PSLBS) 1 = Error in Programming or Setting Block Lock Bits 0 = Successful Program or Set Block Lock Bits						
Yes	SR3 = PROGRAMMING VOLTAGE STATUS (VPENS) 1 = Low Programming Voltage Detected, Operation Aborted 0 = Programming Voltage OK					SR3 does not provide a continuous voltage level indication. The ISM interrogates and indicates the programming voltage level only after block erase, program, set block lock bits, or clear block lock bits command sequences.	
Yes	SR2 = PROGRAM SUSPEND STATUS (PSS) 1 = Program Suspended 0 = Program in Progress/Completed						
Yes	SR1 DEVICE PROTECTSTATUS (DPS) 1 = Block Lock Bit Detected, Operation Aborted 0 = Unlock					SR1 does not provide a continuous indication of block lock bit values. The ISM interrogates the block lock bits only after block erase, program, or lock bit configuration command sequences. It informs the system, depending on the attempted operation, if the block lock bit is set. Read the block lock configuration codes using the READ IDENTIFIER CODES command to determine block lock bits status. SR0 is reserved for future use and should be masked when polling the status register.	
Yes	SR0 = RESERVED FOR FUTURE ENHANCEMENTS						

NOR Flash - Read Cycle Timings

Typical Read Cycle Timing Characteristics - Toshiba 128MByte NOR device

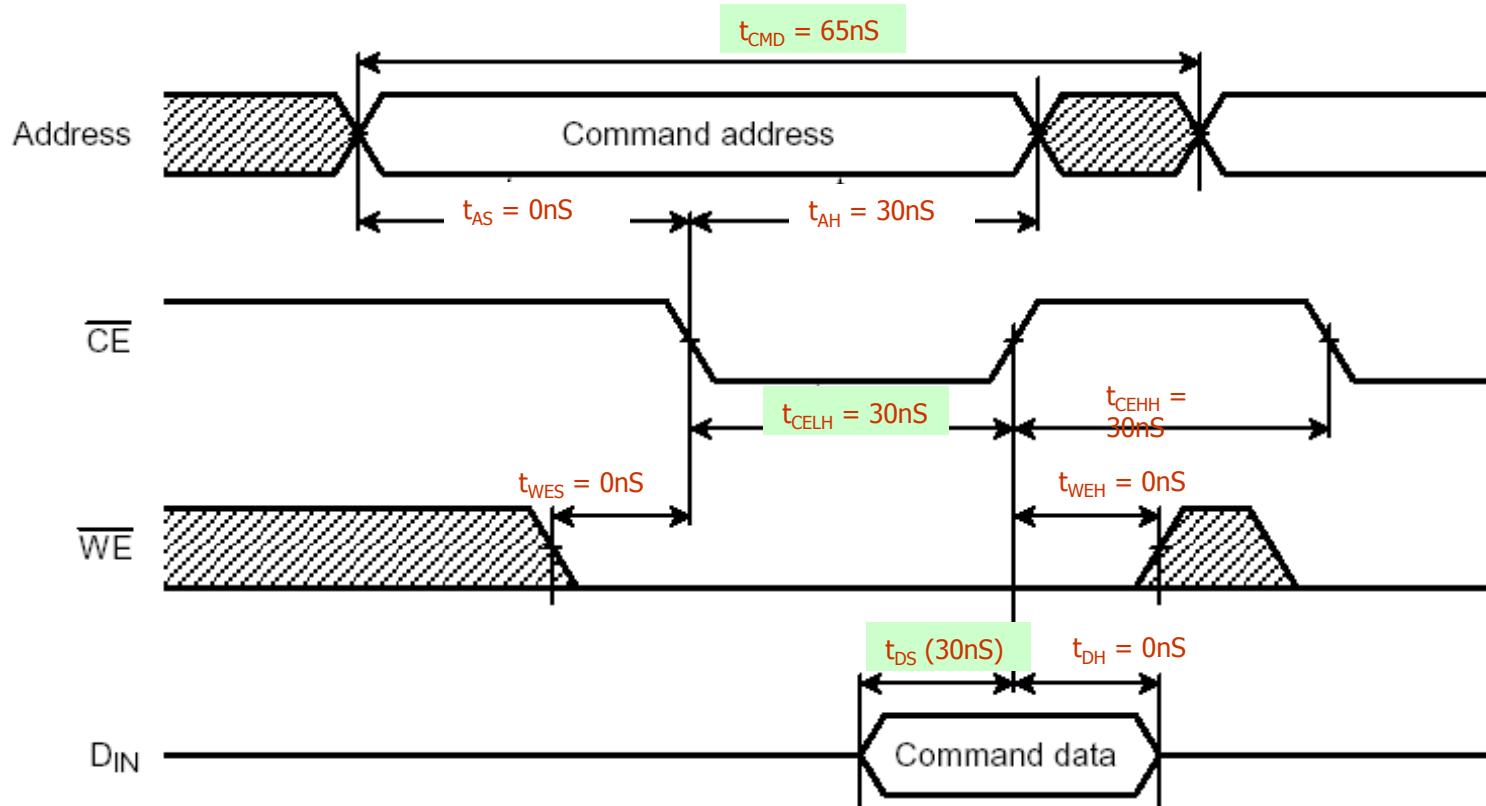
- Similar to Slow Static Ram and Eeprom
- Read access time not *quite* as fast as Sram at **60nS**



NOR Flash - Write Cycle Timings

Typical Write Cycle Timing Characteristics – Toshiba 128MByte NOR device

- Again similar to Static Ram and Eprom
- Min Write pulse width is **30nS** (chip takes *longer* than this to do actual write)
- **Data** set-up time is **30ns**, hold time is **0ns**
- **Address** setup time is **0ns**, hold time is **30ns**
- **Note** internal write takes longer than this – use **polling** to check for completion.



NOR Flash - Program and Erase Characteristics

Typical Program and Erase Characteristics – Toshiba 128MByte NOR device

PROGRAM AND ERASE CHARACTERISTICS

SYMBOL	PARAMETER	MIN	TYP.	MAX	UNIT
t _{PPW}	Auto-Program Time (Byte Mode)	—	8	300	μs
	Auto-Program Time (Word Mode)	—	11	300	μs
t _{PPAW}	Auto-Page program time	—	45	2400	μs
t _{PCEW}	Auto Chip Erase Time	—	184	2630	s
t _{PBEW}	Auto Block Erase Time	—	0.7	10	s
t _{EW}	Erase/Program Cycle	10^5	—	—	Cycles

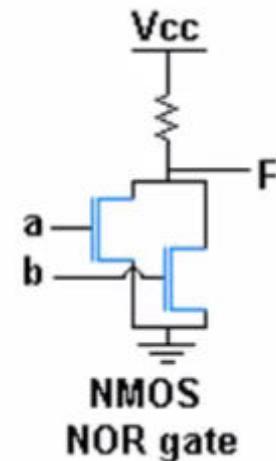
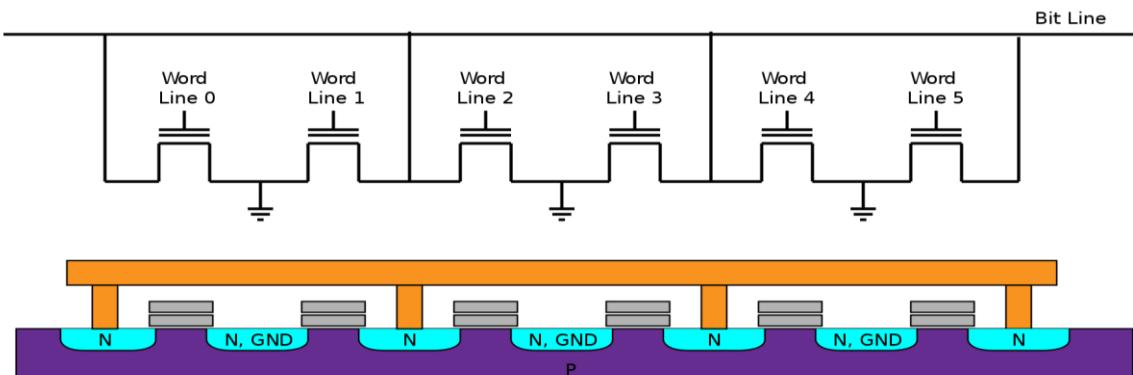
Note internal write time

Note limited erase and
reprogram life for a cell

NOR Flash EEPROM

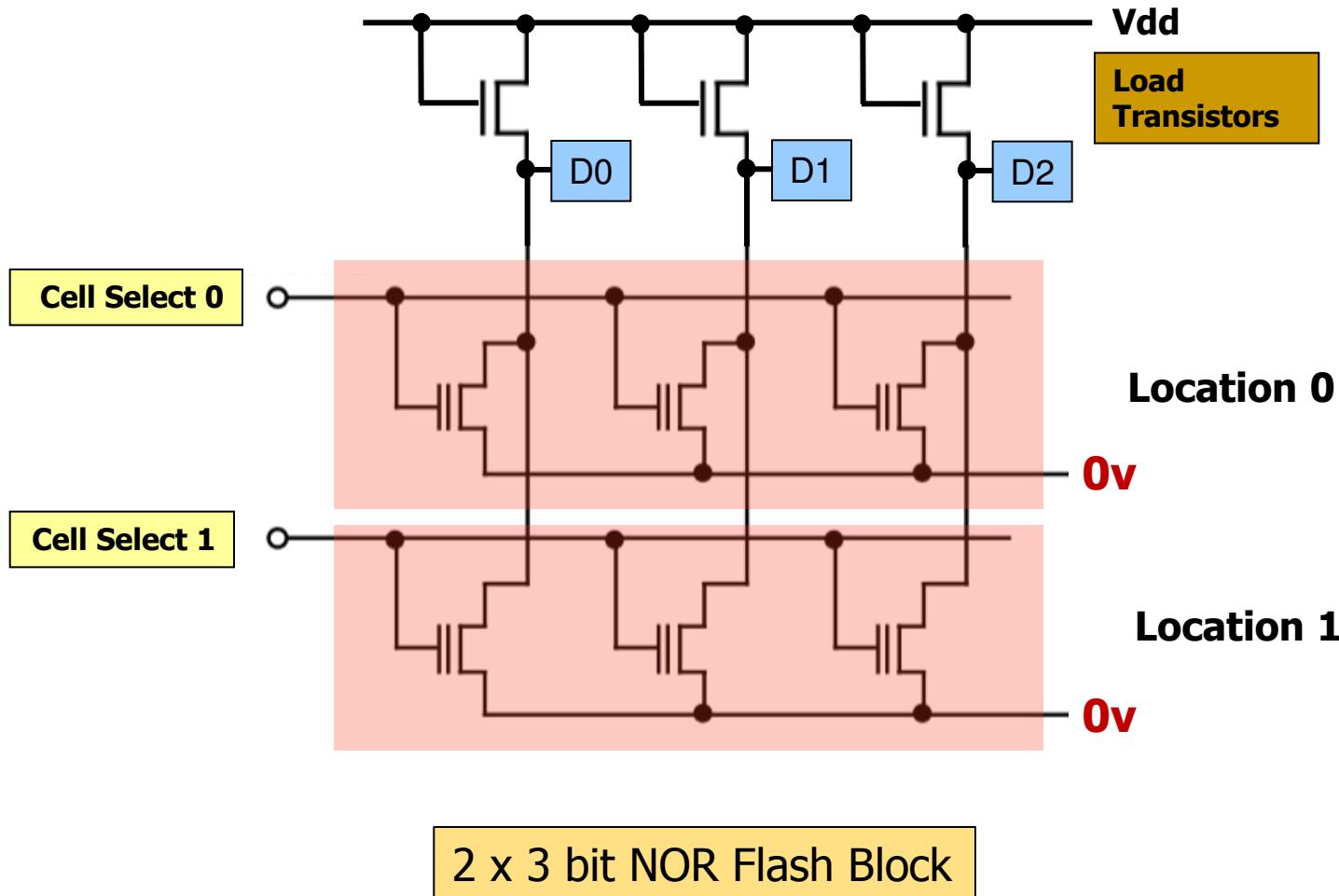
NOR Flash Layout and Operation

- MOSFETs wired in **parallel**, resembling a **NOR Gate** (*hence the name*).
- To **read** data from a cell **turn on** the mosfet you want to read (*via the word line*). The current that flows between the **bit line** and **ground** indicates the state of the charge on the floating gate. If the cell can conduct enough to pull the load transistor (*resistor*) down to ground, the cell stores a ‘0’.
- Fabricating a **NOR** flash device means each memory cell needs a **ground connection** (*which occupies space*) lowering the density of cells on the chip



NOR Flash EEPROM

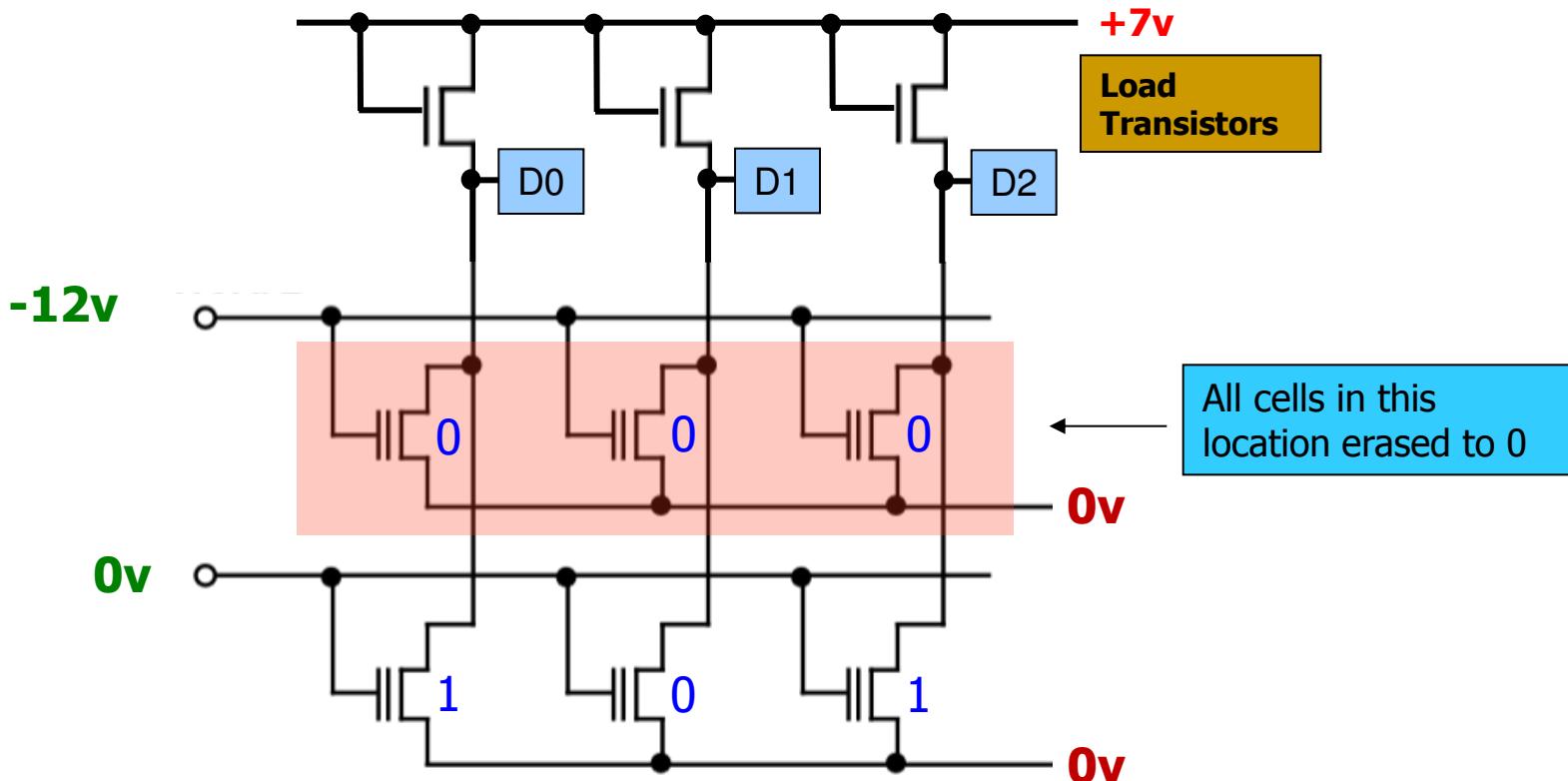
- Architecture designed to behave much like conventional rom, except erasing is done on a block by block basis.



NOR Flash EEPROM

Erasing NOR flash one location/byte at a time

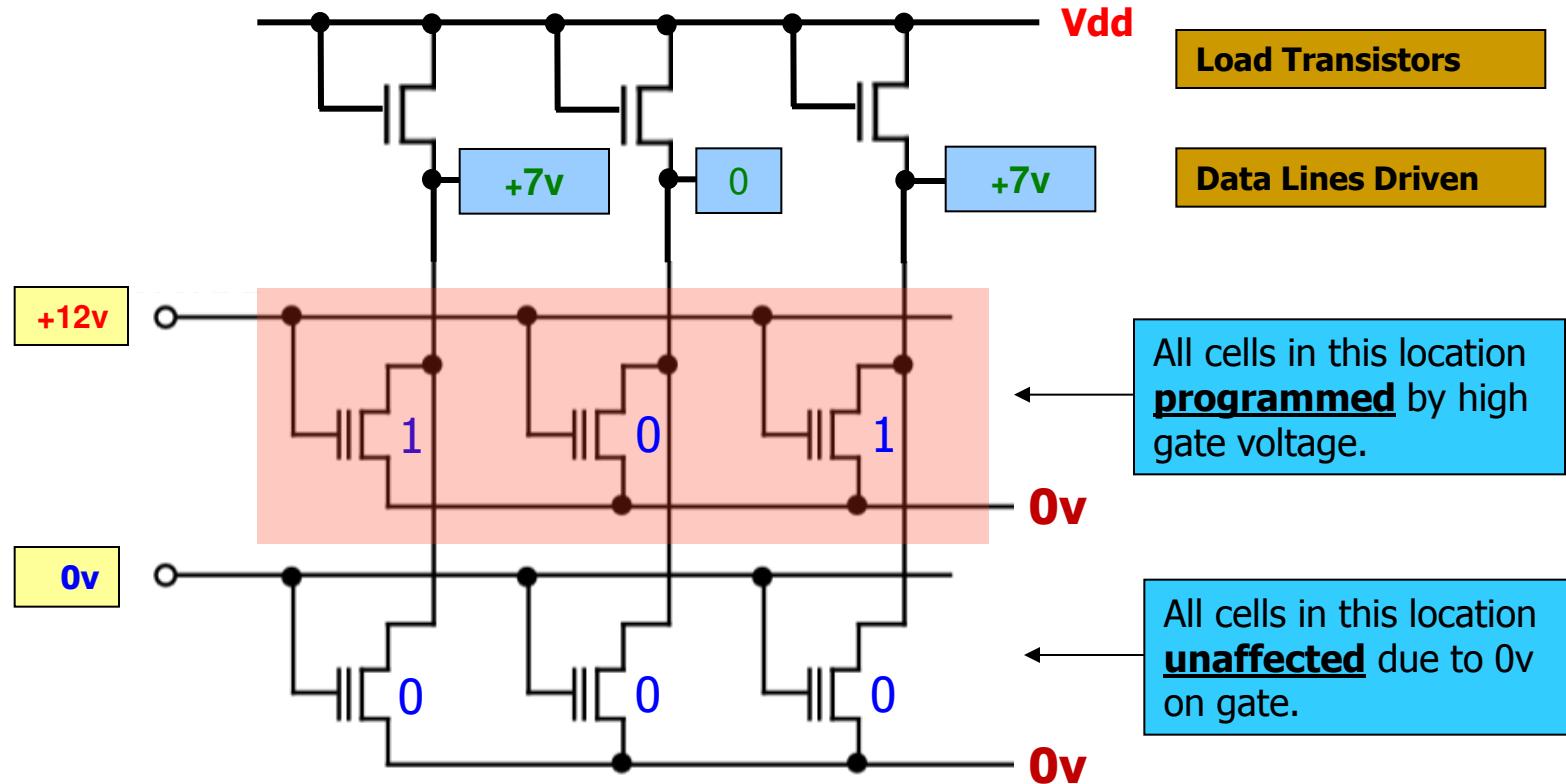
- Raise V_{dd} to a higher than normal voltage e.g. +7v
- Pulse the gates of all cells in the location to a **high negative voltage** e.g. -12v.
- This “**repels**” any electrons stored on the floating gate and causes them to tunnel across the silicon dioxide insulator to the channel and are eventually taken away.
- This creates a cell that can register a **logic 0** on data line during a **read**.



NOR Flash EEPROM

Programming a NOR Flash Memory Location

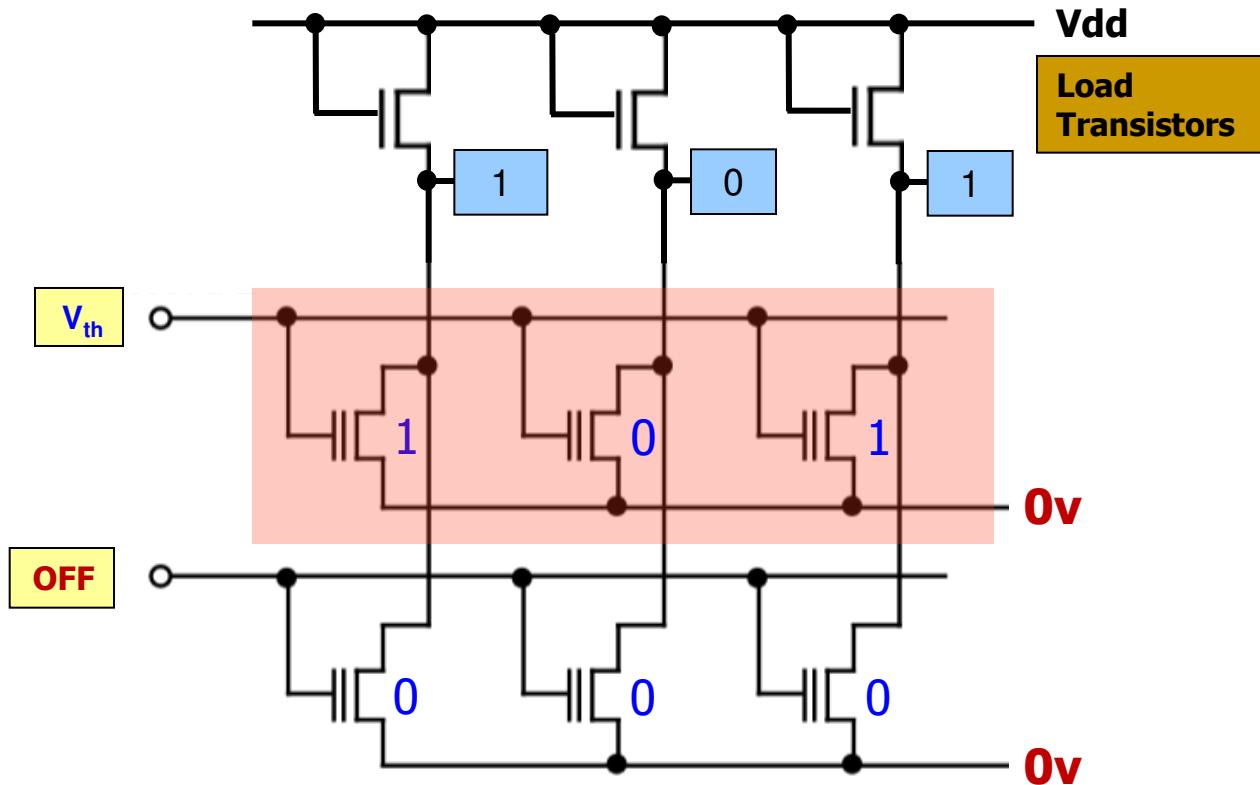
- Similar process to UV-Eeprom.
- Select the cells to be programmed by applying **12V** to their gates.
- To store a **logic 1**, drive cell **data line** to about **7v** creating a **large channel current**.
- Electrons will then *tunnel* across the insulator (*via process known as hot electron injection*) attracted by **12V** on the gate.
- For cells with no gate voltage or have **0v** on their Data line, no injection takes place.



NOR Flash EEPROM

Reading a NOR Flash Cell

- Apply an *intermediate* threshold voltage V_{th} to the cell select of the location to be read.
- If the cell has a **charge** on the floating gate, it will **not** be able to pull down the load transistor, thus a **logic 1** is read on the data line.
- If the cell you are reading has **no charge** on the floating gate, it will fully conduct with an intermediate gate voltage V_{th} , and will be able to pull down the load transistor and thus a **logic 0** is read on the data line.



Nand Flash Technology

NAND Flash EPROMs

Characteristics and Operation of NAND Flash devices

- NAND Flash devices dominate the **high storage capacity** market due to cost effectiveness and small packaging. Typical sizes up to **32GByte** (see [Click Here](#)).
- Usually they have a **sequential** external interface which means that individual bytes can be read by **writing** a internal **start address** to the chip (*over the data bus*) and with each **RE*** strobe you get data from next location.
- Random access is pretty much ruled out by this scheme due to need to write the **address** each time, but they were never intended for that mode of operation anyway. (*Use NOR if you must have random access*).
- Most devices have the same **48pin pin-out** regardless of storage capacity, due to the sequential data stream. This permits easy upgrades to larger capacity devices without redesigning PCBs.
- **Not** as reliable as **NOR** technology due to smaller mosfet size (to increase storage density), may need to be used with error correction circuitry (ECC).

TH58NVG1S3A

NC	1	○	48	NC
NC	2		47	NC
NC	3		46	NC
NC	4		45	NC
NC	5		44	I/O8
GND	6		43	I/O7
RY/BY	7		42	I/O6
RE	8		41	I/O5
CE	9		40	NC
NC	10		39	NC
NC	11		38	NC
Vcc	12		37	Vcc
Vss	13		36	Vss
NC	14		35	NC
NC	15		34	NC
CLE	16		33	NC
ALE	17		32	I/O4
WE	18		31	I/O3
WP	19		30	I/O2
NC	20		29	I/O1
NC	21		28	NC
NC	22		27	NC
NC	23		26	NC
NC	24		25	NC

Toshiba 256MB CMOS FLASH MEMORY

I/O1 to I/O8	I/O port
CE	Chip enable
WE	Write enable
RE	Read enable
CLE	Command latch enable
ALE	Address latch enable
WP	Write protect
RY / BY	Ready / Busy
GND	Ground Input
Vcc	Power supply
Vss	Ground

NAND Flash Operation

Operation Mode: Logic and Command Tables

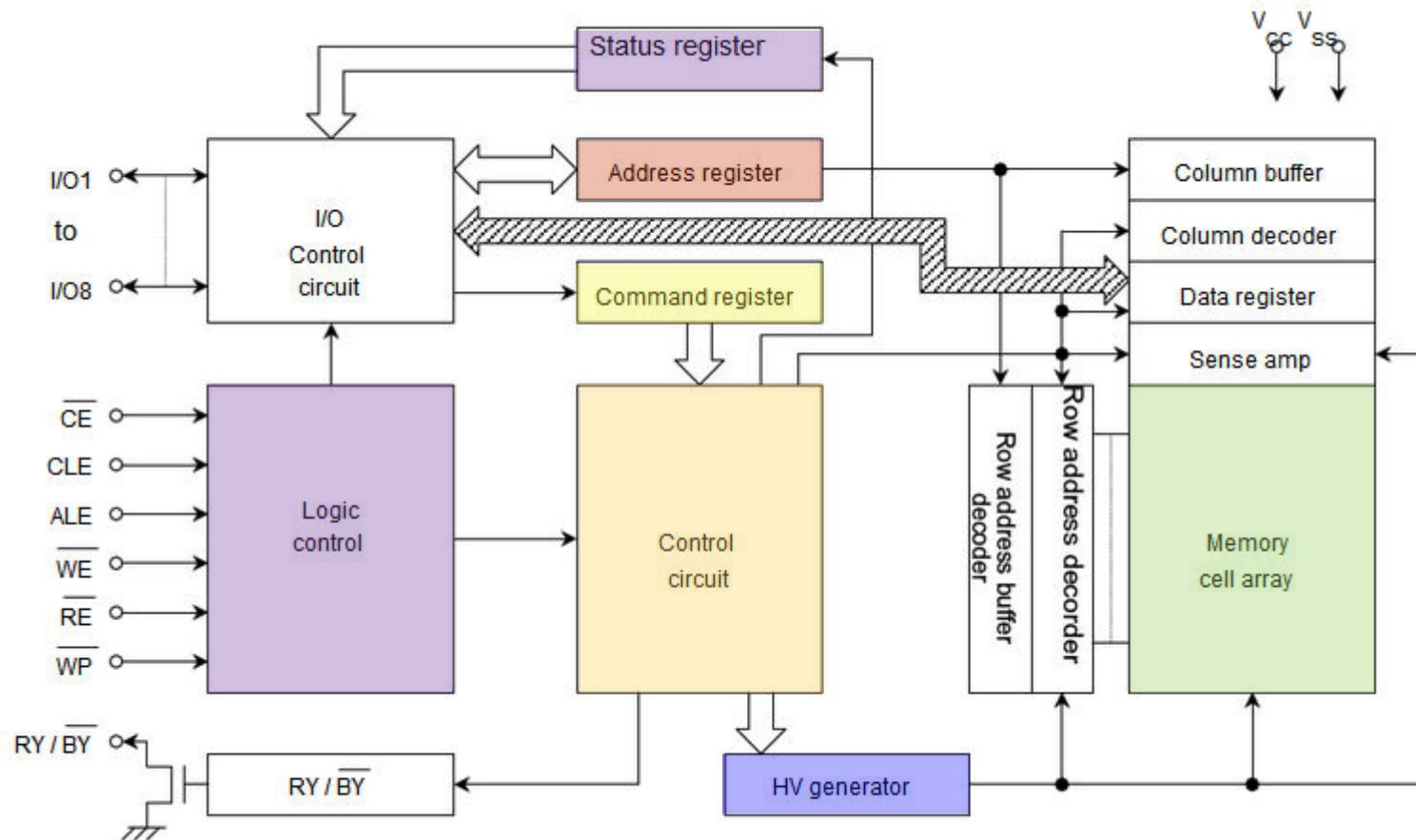
The operation of the device, such as **Program**, **Erase**, **Read** etc. are controlled by the signals **CLE**, **ALE**, **CE**, **WE**, **RE** and **WP** signals, as shown in Table 2. Commands are written over the data bus.

Table 2. Logic Table

	CLE	ALE	CE	WE	RE	WP ^{*1}
Command Input	H	L	L	↑↓	H	*
Data Input	L	L	L	↑↓	H	H
Address input	L	H	L	↑↓	H	*
Serial Data Output	L	L	L	H	↑↓	*
During Programming (Busy)	*	*	*	*	*	H
During Erasing (Busy)	*	*	*	*	*	H
During Reading (Busy)	*	*	*	*	*	*
Program, Erase Inhibit	*	*	*	*	*	L
Standby	*	*	H	*	*	0 V/Vcc

NAND Flash EPROMs

Block Diagram Toshiba 256MB Nand Flash device



NAND Flash Killer Applications

- Replacing hard disks with Solid state disks (SSD) is the big driving in Nand flash market right now. Obvious benefits are :-
 - Smaller, lighter, faster (*no seek time or rotational latency issues. Data is available instantly*),
 - More reliable (*no moving parts*) and quiet.
 - Cost-effective up to 1T Byte.
 - Block erase and sequential read/write concept of NOR flash a good match for file-system type interface provided by the operating system but much faster data transfer.
- Packaging into more robust products have placed them in the hands of the public.
- Applications outside traditional computers have been **digital camera**, **cell phones** (for address books), **MP3 Players** etc.



Solid State Disk with Serial ATA interface ~\$400



Digital Camera

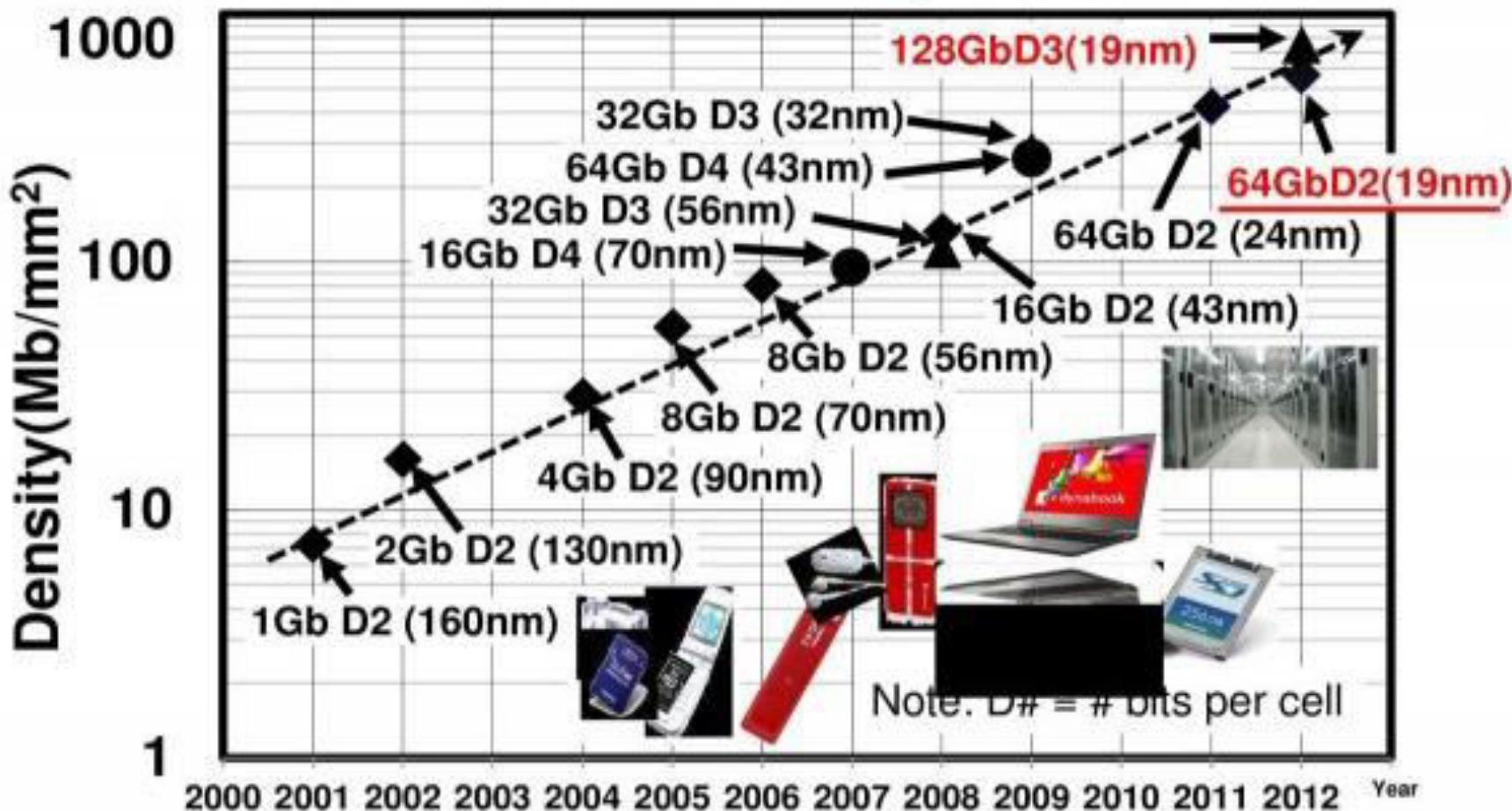


512 GByte SD Card

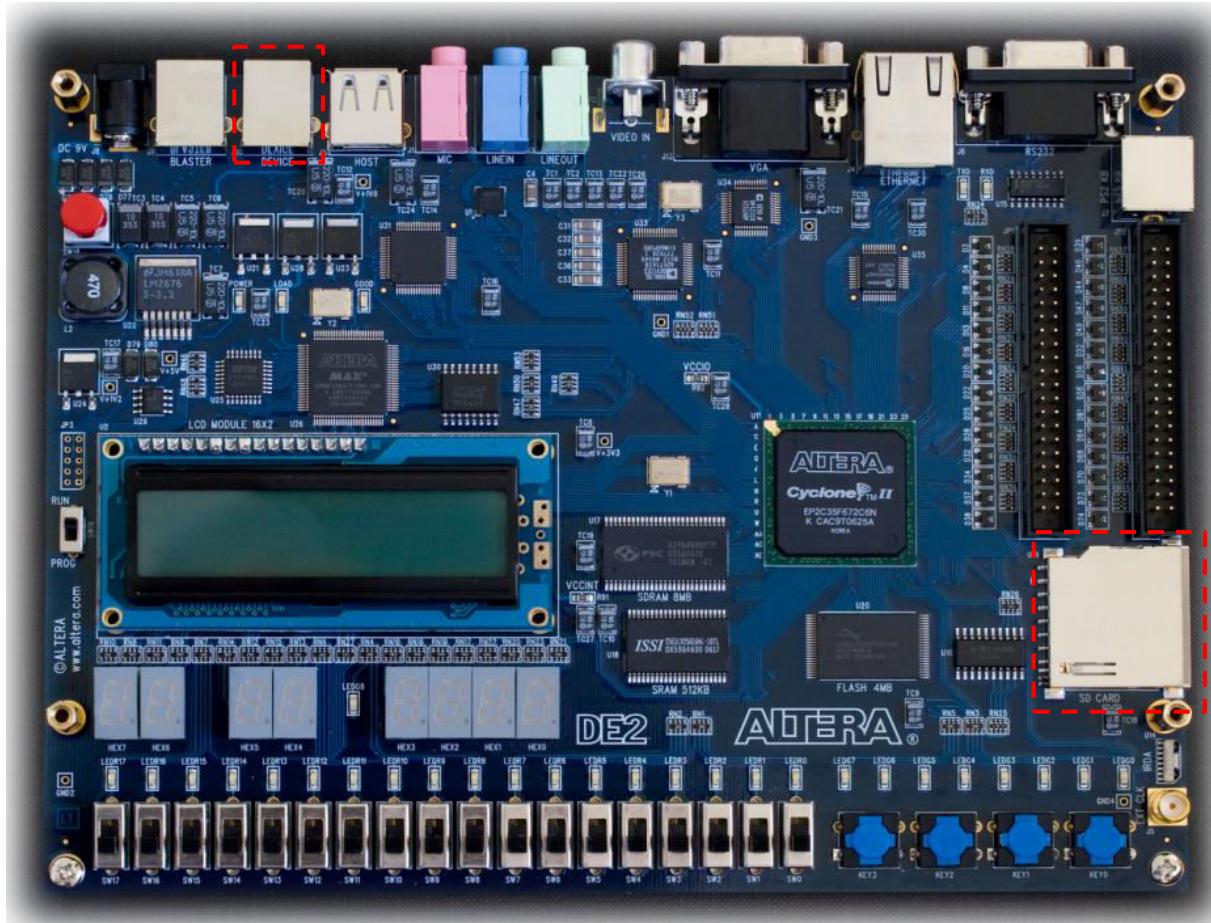


256GByte USB
Thumb Drive

NAND Flash Density Trend



NAND Flash on the DE2



*NAND flash can be
plugged into SD-
Card Port or via
USB*

NAND Flash EEPROM

NAND Flash Layout and Operation

- Nand Memory MOSFETs are wired in **series**, resembling a **NAND Gate**.
- There is a **source select** transistor at one end and a drain select transistor at the other end of the series connections.
- Storage density of **NAND** devices is higher than **NOR** due to fewer ground connections, thus more of the chip's space can be used for MOSFETs.

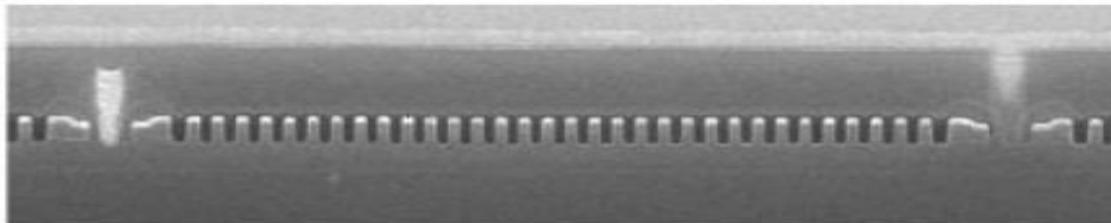
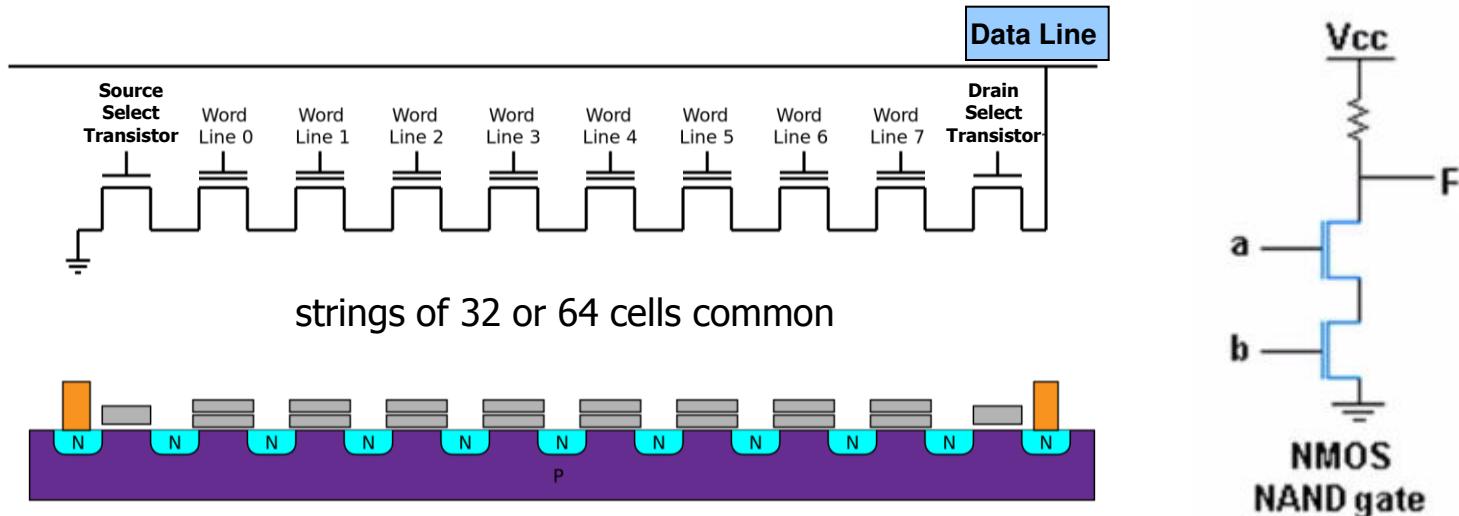
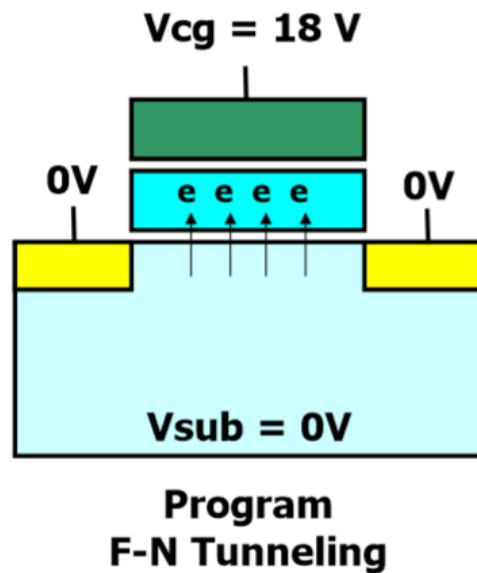


illustration of
real silicon

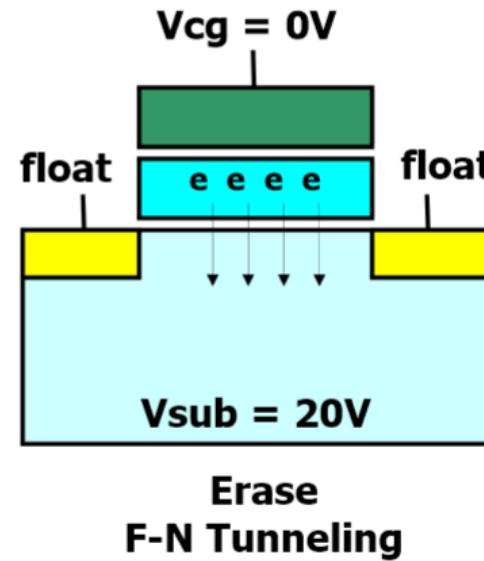
NAND Flash EEPROM

Program and Erasing

- Cells are organized into **blocks** within the chips e.g. **128kByte** blocks and an erase command to the chip causes all locations in one block to be **erased**.
- Program and erasing a cell is performed via the simple act of biasing the substrate and control gate to opposite values.
- Bias one way you get **erasure** to ‘logic 0’ (*electrons tunnel onto the floating gate*), the other way you get **programmed** to ‘logic 1’ (*electrons tunnel to the substrate*).



Logic 1 stored



Logic 0 stored

NAND Flash EPROM

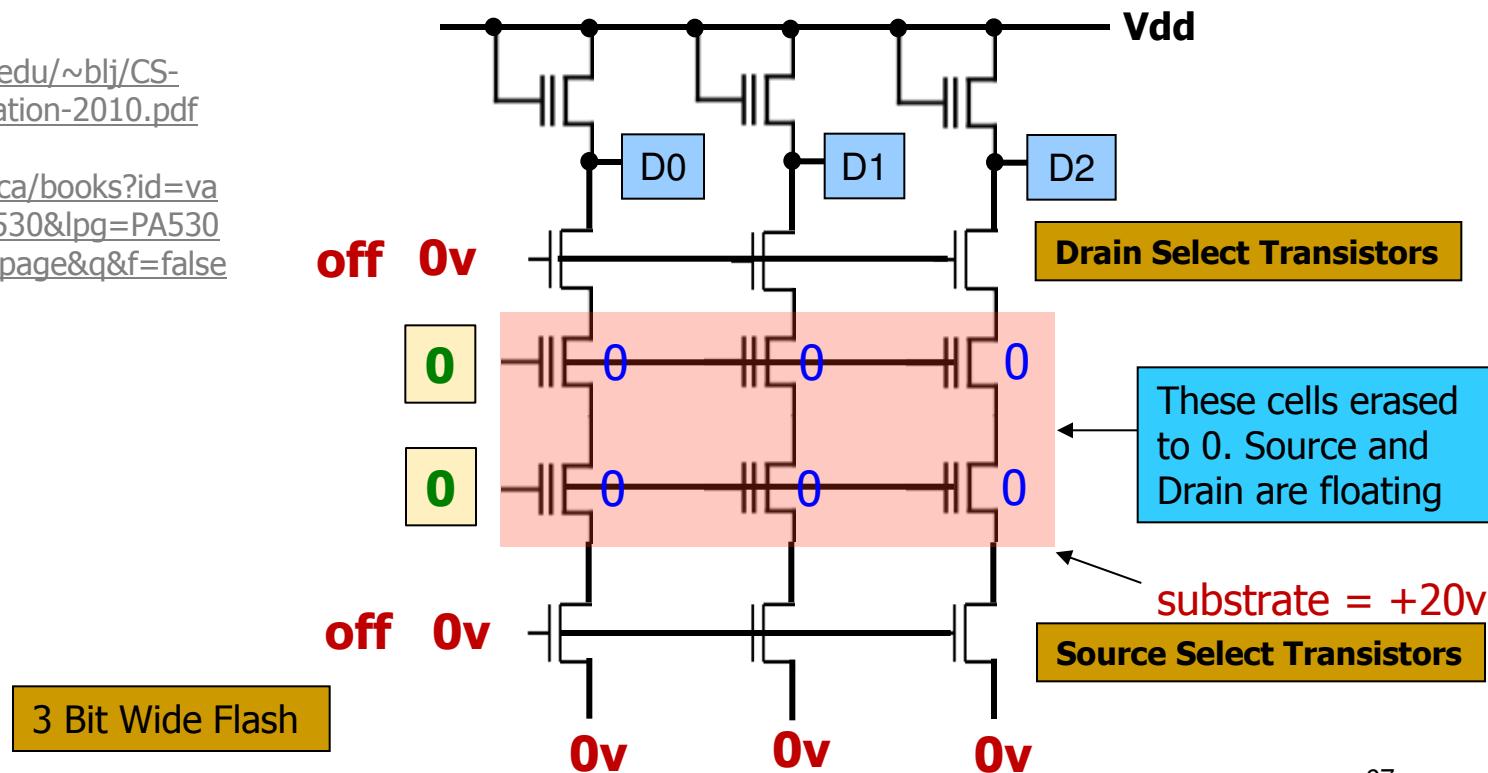
Erasing NAND flash one block at a time

- To erase cells, the substrate beneath the transistors is raised to **+20v**.
- The gates of all memory cells in the block to be erased are set to **0v**
- The electrons on the floating gate are attracted to the substrate and cause them to *tunnel* (via *Fowler-Nordheim tunneling*) across the silicon dioxide insulator to the channel and eventually are taken away, creating an erased cell that stores a logic 0.

References

<http://www.ece.umd.edu/~blj/CS-590.26/nand-presentation-2010.pdf>

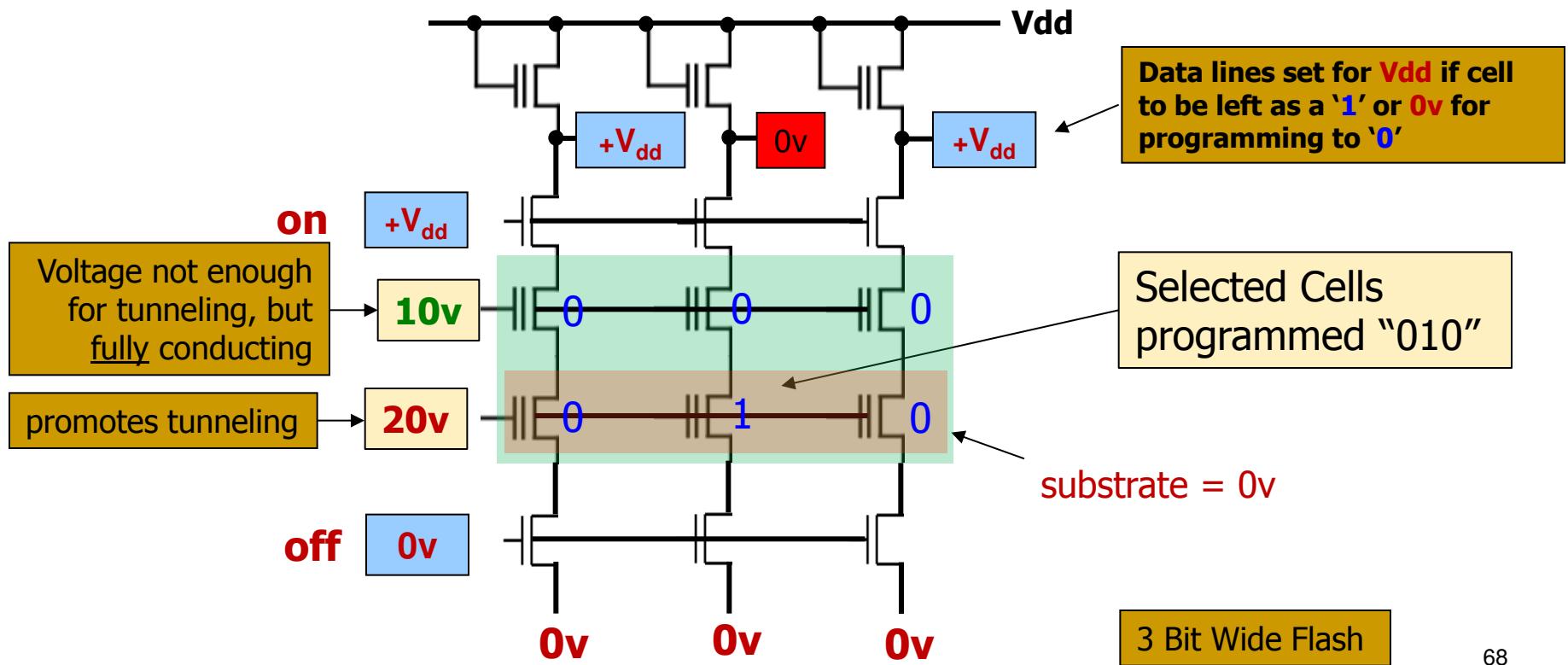
https://books.google.ca/books?id=vaq11vKwo_kC&pg=PA530&lpg=PA530&redir_esc=y#v=onepage&q&f=false
– see page 25



NAND Flash EEPROM

Programming a NAND Flash Memory Cell

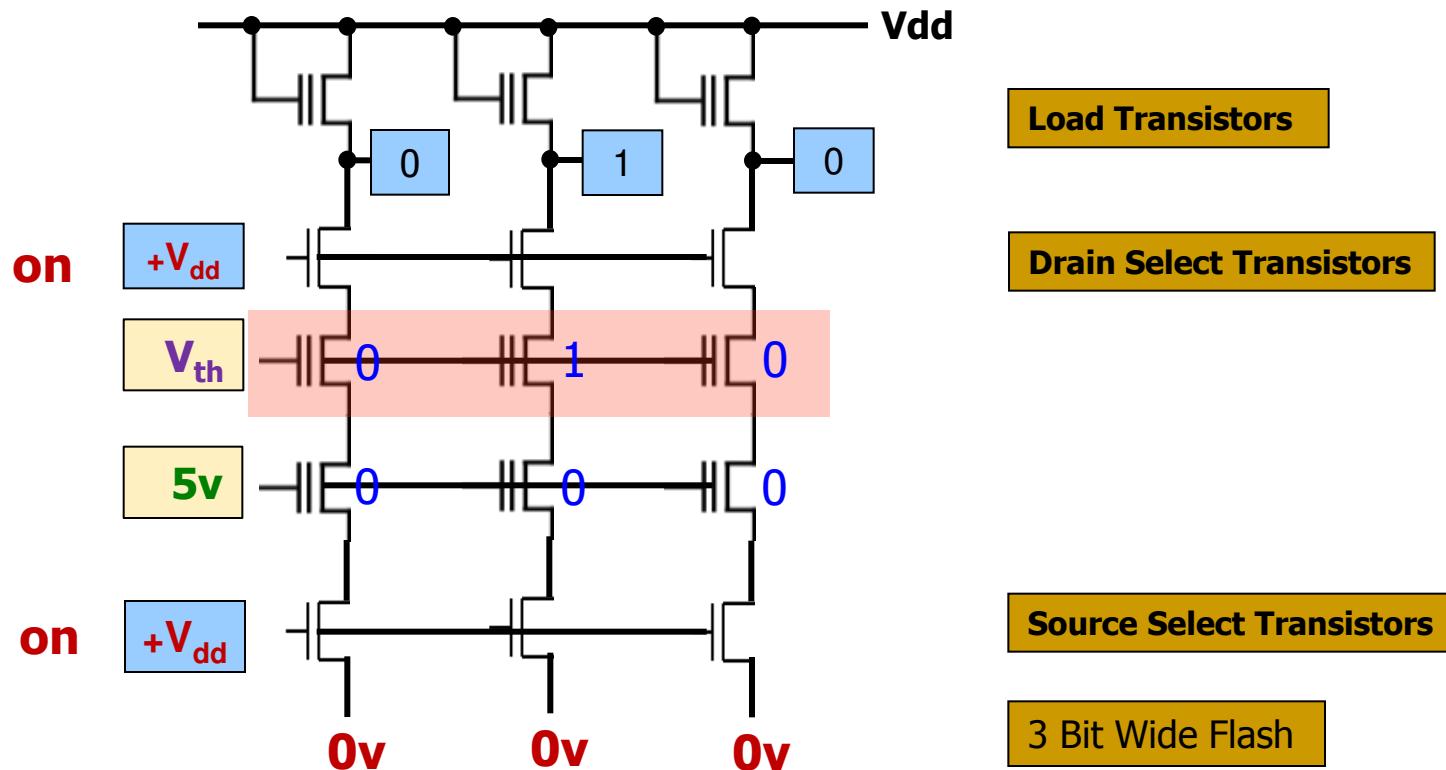
- Because of the serial connections, NAND cells have to be addressed and programmed 1 location at a time. Cells required to store '0' (i.e. erased) **can skip programming**.
- The voltages required to program a '1' into a selected cell are shown below.
- These voltages cause electrons to be "pulled" from the channel onto the floating gate. The charge on the gate then alters the conduction threshold for the cell meaning it is able to pull down the data line during reading thus registering a 1.



NAND Flash EPROM

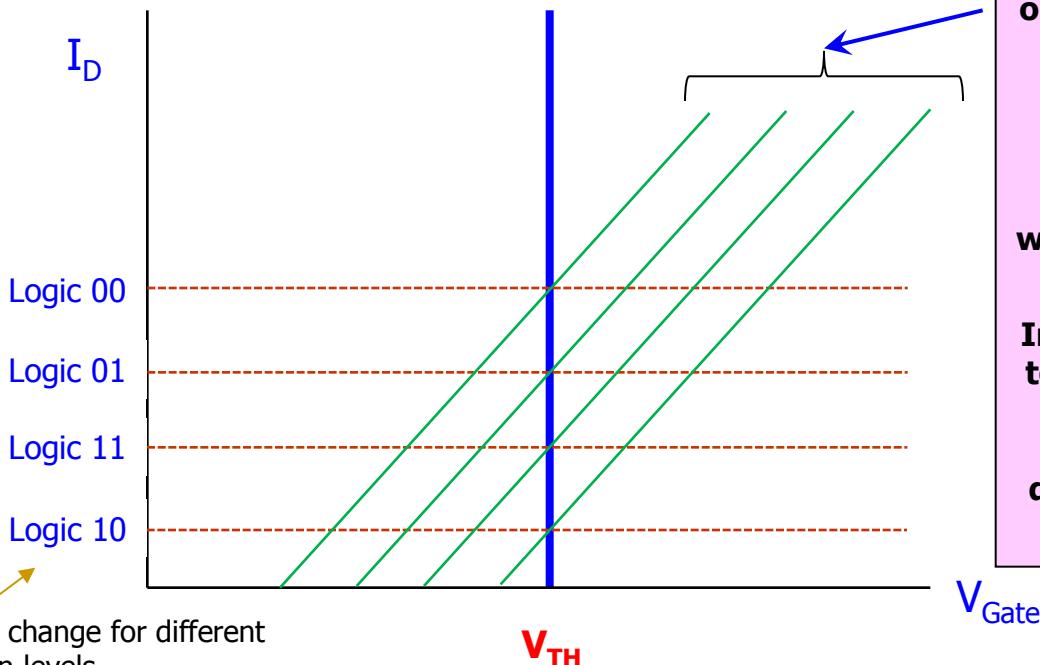
Reading a NAND Flash Memory Cell

- Turn on both **select** transistors
- For the cell that you want to **read**, apply an intermediate threshold voltage V_{th}
- For all other cells set their Gate voltage to **5v**. This causes those cells in the line to conduct and act like a piece of wire.
- If the cell to be read is storing a logic 0 (*i.e. no charge on its floating gate*), it will fully conduct and pull down the load transistor to creating a logic '**0**' on the Data Line. Otherwise a logic '**1**' is created.



Multi-Level Nand Flash Cell Technology (MLC and TLC)

- By carefully controlling the charge deposited on the floating gate during programming of a flash cell, it is possible to store up to **2 bits per cell/data** transistor as shown below.
- This is the basis of **Multi-Level Cell** technology (**MLC**) rather than **Single-Level Cell** (**SLC**) used in some high density low cost Flash chips (see later).
- **MLC** is less reliable and may need error detection and correction circuitry to make it reliable. May also have limited number of re-program cycles, especially as size of mosfets shrink e.g. < **20nM** means perhaps just **1000 erase-reprogram** cycles.
- State of the art in **2014** is **3 bits per cell (TLC)** i.e. use **8 different charge levels**.
<http://searchsolidstatestorage.techtarget.com/definition/TLC-flash-triple-level-cell-flash>



4 different charge levels can be deposited on the floating gate leading to 4 different conduction thresholds/curves.

During Reading, the application of a threshold gate voltage V_{TH} will thus result in 1 of 4 different drain currents which can be interpreted as either 00, 01, 10, 11, i.e. two bits per cell.

In theory this scheme could be expanded to say 3 or 4 bits per cell, but the minute differences in charge required to differentiate between say 8 or 16 different conduction levels makes it just too unreliable.

Microchip 8M Byte Serial Flash on SPI bus

MICROCHIP

English

Search Microchip

Search Data Sheets

PRODUCTS APPLICATIONS DESIGN SUPPORT TRAINING SAMPLE & BUY ABOUT US Contact Us myMicrochip Login

SST26VF064BA In Production

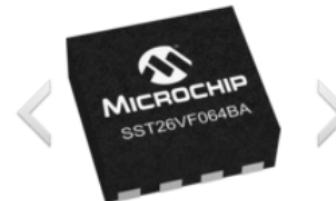
SST26VF064B / SST26VF064BA, 3.0V 64 Mbit Serial Quad I/O (SQI) Flash Memory (02/27/2015)

The SST26VF064BA Serial Quad I/O (SQI) flash device utilizes a 4-bit multiplexed I/O serial interface to boost performance while maintaining the compact form factor of standard serial flash devices. SST26VF064BA also support full command-set compatibility to traditional Serial Peripheral Interface (SPI) protocol. Operating at frequencies reaching 104 MHz, the SST26VF064BA enables minimum latency execute-in-place (XIP) capability without the need for code shadowing on an SRAM. The device's high performance and reliability make it the ideal choice for Network Appliance, DSL and Cable Modems, Wireless Lan, Computing, Digital TV, Smart Meter, Server, Set Top Box, Automotive and other Industrial applications. Further benefits are achieved with SST's proprietary, high-performance CMOS SuperFlash® technology, which significantly improves performance and reliability, and lowers power consumption for high bandwidth, compact designs.

The **SST26VF064B** default at power up is with **WP# and HOLD pins enable and SIO2 and SIO3 pins disable** allowing for SPI protocol operations without register configuration.

The **SST26VF064BA** default at power up with **WP# and HOLD pins disable and SIO2 and SIO3 pins enable** allowing for Quad I/O operations without register configuration.

Features	Parameter Name	Value
<ul style="list-style-type: none">■ Serial Interface Architecture- Nibble-wide multiplexed I/O's with SPI-like serial command structure: Mode 0 and Mode 3,■ x1/x2/x4 Serial Peripheral Interface (SPI) Protocol and SQI protocol■ Burst Modes- Continuous linear burst, 8/16/32/64 Byte linear burst with wrap-around■ Page-Program- 256 Bytes per page in x1 or x4 mode■ Flexible Erase Capability- Uniform 4 KByte sectors, Four 8 KByte top and bottom parameter overlay blocks, One 32 KByte top and bottom overlay block, Uniform 64 KByte overlay blocks■ Software Write Protection- Individual Block-Locking: 64 KByte blocks, two 32 KByte blocks, and eight 8 KByte parameter blocks■ Low Power Consumption: Active Read current: 15 mA (typical @ 104 MHz), Standby Current: 15 µA (typical)■ Packages Available: 8-contact WDFN (6mm x 5mm), 8-lead SOIC (208 mil), 16-lead SOIC (300 mil), 24-ball TBGA (6mm x 8mm)■ Serial Flash Discoverable Parameters (SFDP)	Density	64 Mbit
	Op. Volt Range (V)	2.7 to 3.6
	Max. Clock Freq.	104 MHz
	Page Size (bytes)	256
	Write Protect	Full Array
	Temp Range (°C)	-40°C to +85°C
	Endurance	100,000
	Data Retention (Years)	100
	Bus Modes	SPI,SDI,SQI

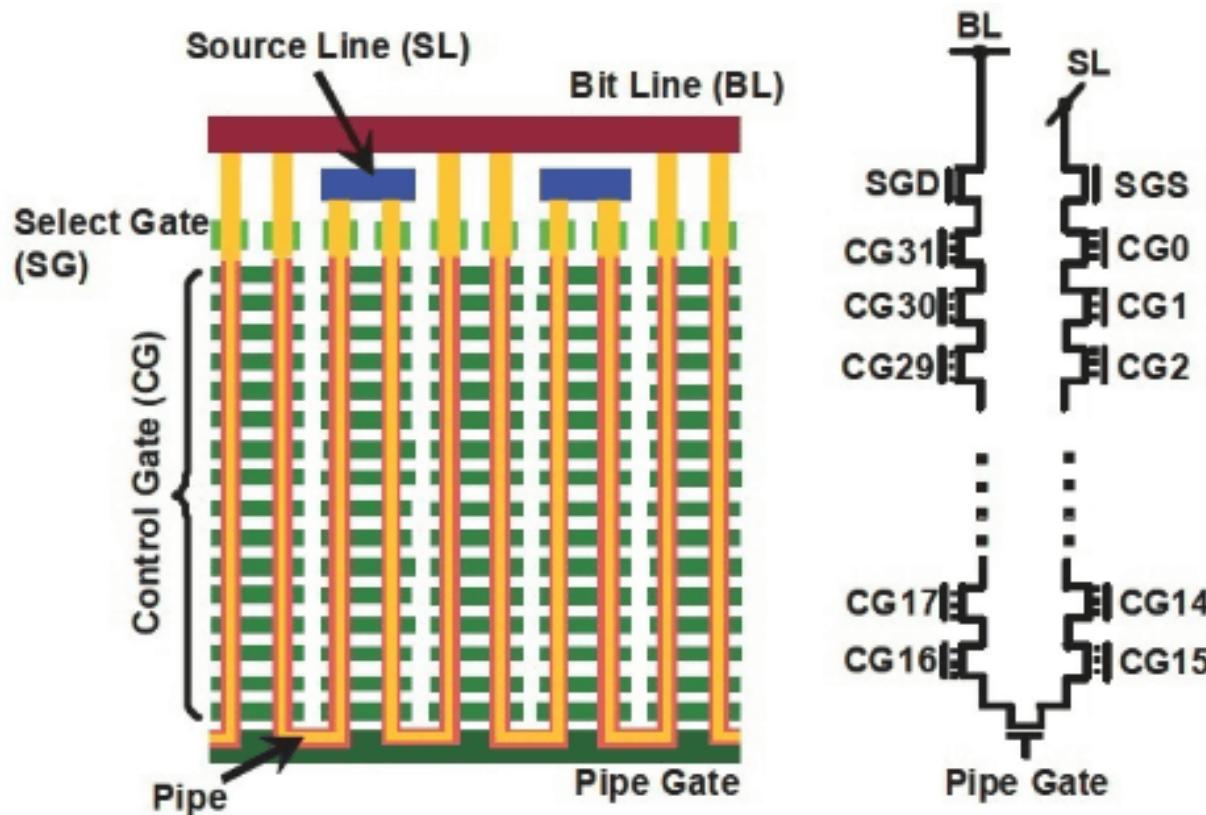


cost: \$2.00

Flash EPROM

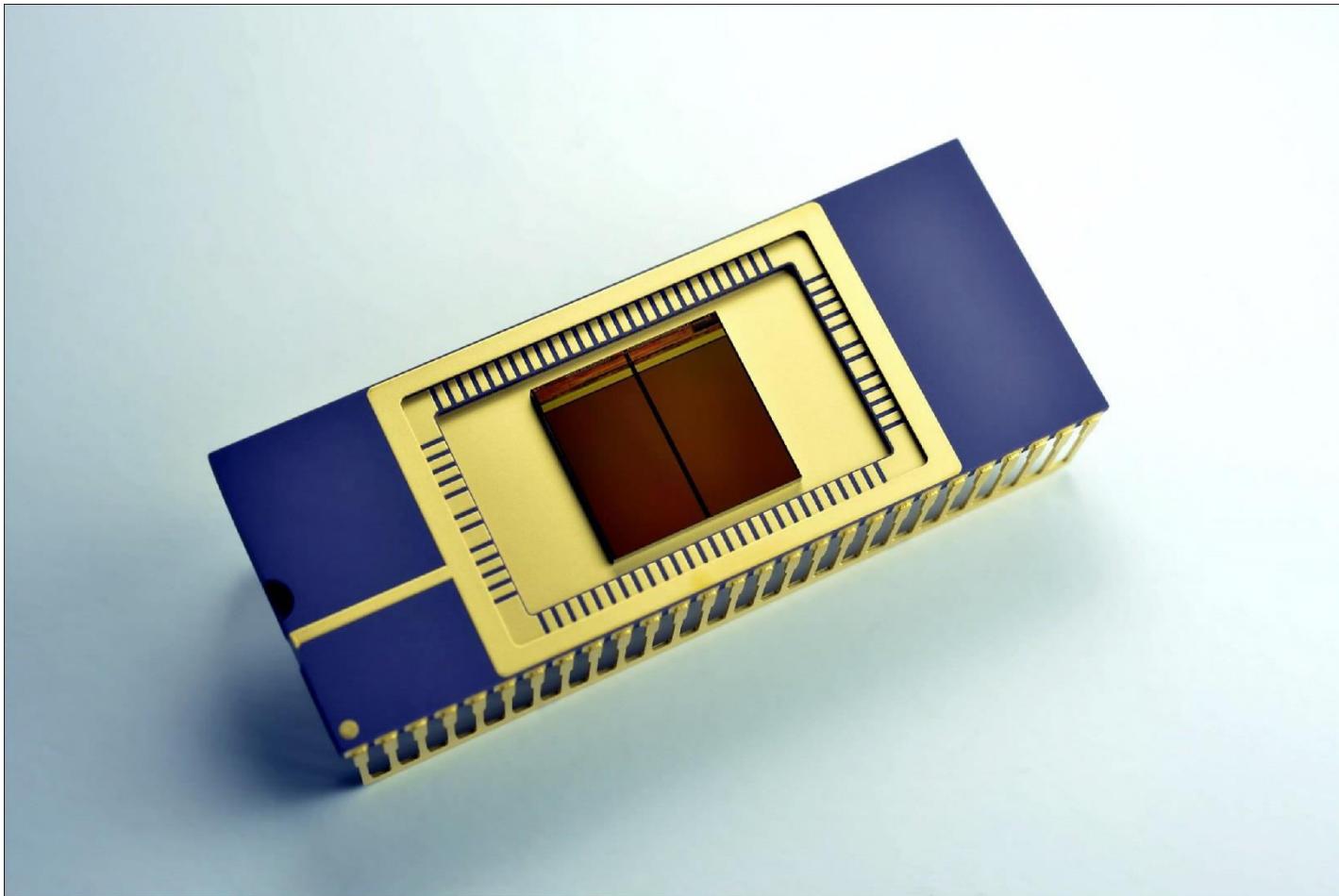
3D NAND Memory (the only way is up!)

- Higher density flash is being created by fabricating the *series* connections of NAND memory layout “**vertically**”, i.e. stacking cells on top of each other.
- Because of this, cell can be physically bigger than equivalent 2D cell leading to better reliability and faster access times
- This gives an expected “at least” **3x** density improvement over conventional 2D NAND technology.



NAND Flash E²PROM

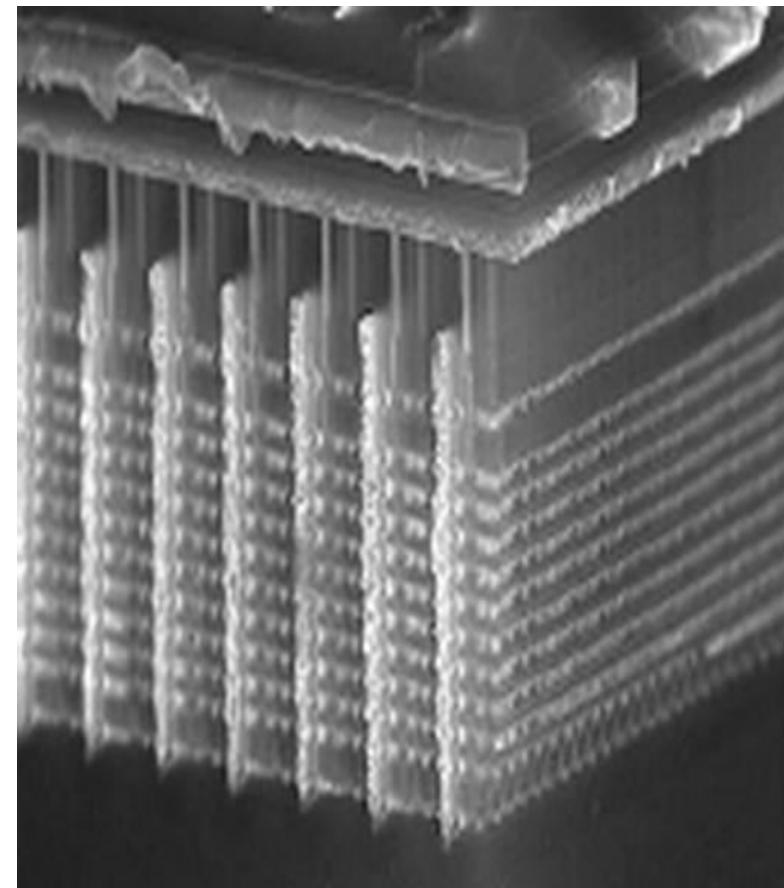
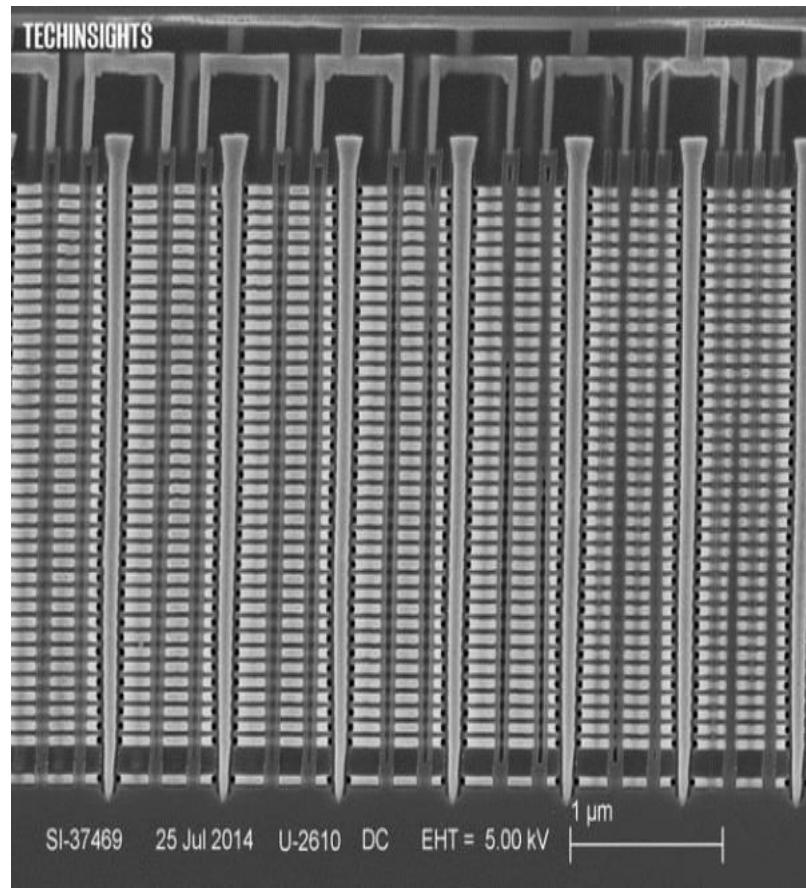
- **2013** - Samsung Electronics unveiled the industry's first **128Gbit** 3-D vertical NAND flash memory. Samsung V-NAND enables up to 100 layers of cells to be stacked with the *potential* to scale the density up to **1Tbit**.



Flash EEPROM

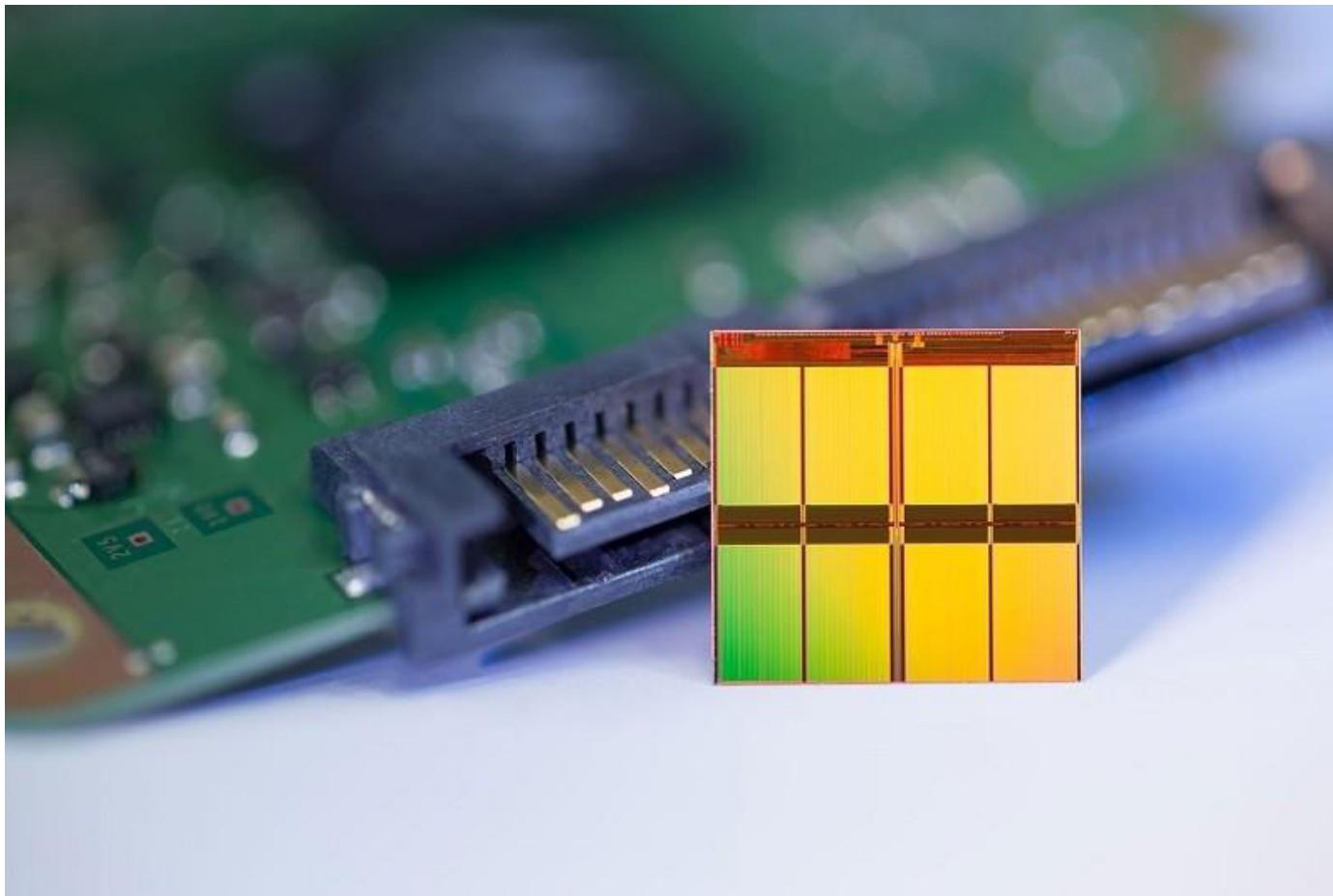
3D NAND Memory

- Electron microscope images of Samsung 3D Nand Flash Memory



NAND Flash EPROM

- Intel to use 3D Nand Triple Level Cell (TLC) Flash for High Density Solid State Drives.
- Prediction is for **10TByte** portable 2.5" SSD drives and **3.5TByte** thumb drives in **2016**
http://newsroom.intel.com/community/intel_newsroom/blog/2015/03/26/micron-and-intel-unveil-new-3d-nand-flash-memory



Flash Applications

- Flash technology has made rapid advances in the last few years. Cell density now rivals DRAM and is better than UV-EPROM and E²Prom.
- More recent NAND technology uses **8 different charge levels** stored on the floating gate, to encode **3 bits per cell**, http://en.wikipedia.org/wiki/Multi-level_cell. This requires sensitive on-chip electronics to detect 8 different conduction levels in the data transistor and leads to slower read times. This is not so much of a problem in Solid state drives where Nand tends to get used.
- ROMs, PROMS, UV-EPROMs and E²Prom rapidly becoming obsolete as they are superseded by **Flash** which is at least as cheap and as dense.
- There was a proposal from Information Storage Devices (ISD) for making a Flash cell with **256** different conduction levels, i.e. the equivalent of 8 bits of storage per cell, but it's too unreliable for data/program code applications. It does however have applications in the music industry and recording speech where bit inaccuracies can be tolerated.

http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=488619&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpis%2Fabs_all.jsp%3Farnumber%3D488619

- For example, 2 cells could record a single 16 bit sample from an analog to digital converter. At 44.1kHz (the standard used for CD quality music), 1 Sec of music could be stored on 88,200 cells. Put another way a typical 1 hour album could be stored on 40Mbytes rather than the ~650Mb of a CD.

Magnetic Ram (M-Ram)

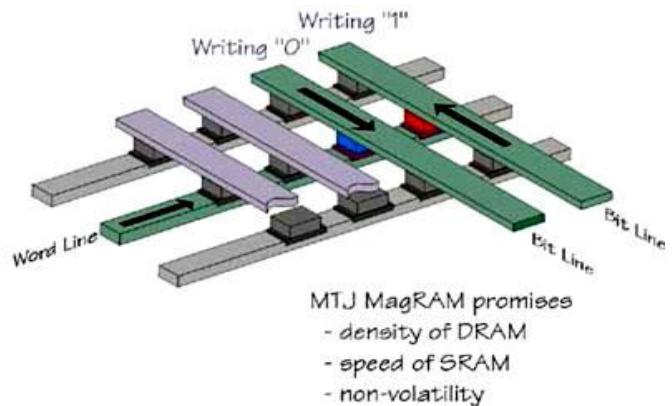
- Recently the subject of a great deal of research and patent activity within IBM, Motorola (*Freescale*) and Infineon Technology.
- Current capacities ~64MBit devices. (see <http://en.wikipedia.org/wiki/MRAM>). Still a long way off from Flash and Dram densities, but technology is young and density will improve.
- Has the *potential* to sweep away existing memory technologies - Sram, Dram and Flash.
- MRAM combines the best features of today's common memory technologies:

- The **storage capacity** and **low-cost** of DRAM.
- The **high speed** and **ease of use** of SRAM.
(No refreshing, symmetrical, fast read-write cycle times. 500x flash speed)
- The **non-volatility of flash**. *(No battery)*. Applications in low power, instant on applications such as the “**internet of things**”

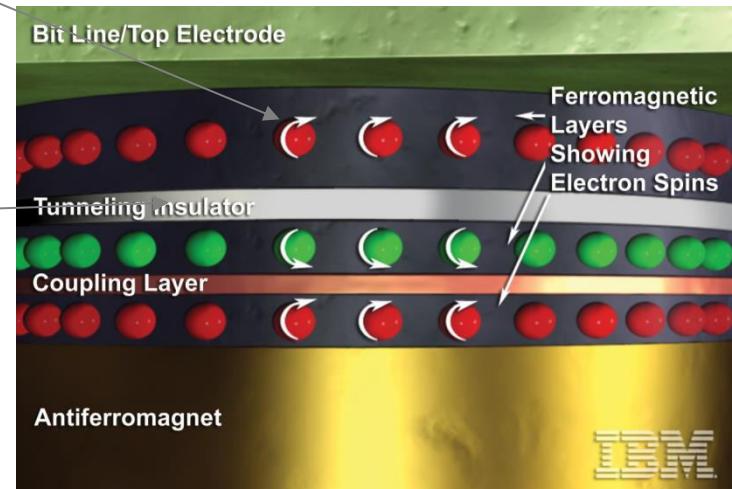
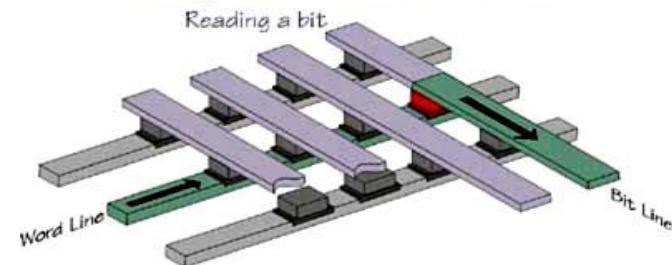
- Article from Extreme Tech outlining Toshiba's use of Mram in L2 Caches.
<http://www.extremetech.com/extreme/184183-toshibas-new-mram-cache-could-reduce-cpu-power-consumption-by-60>
- Everspin (a spin off from Freescale/Motorola) produce chips <http://www.everspin.com/>

Magnetic Ram (M-Ram)

- Each single bit cell comprises a small vertical ferromagnetic column sandwiched between a **Bit** (*i.e. data*) and **Word** (*i.e. cell select*) conductors.
- During a **write** operation, cell is selected and a current is pulsed in one of **two ways** along the **Bit (data) line**. The direction of current affects the direction of '**spin**' of electrons within the top-most layer of the selected columns, storing a **0** or **1**.
- The spin direction affects the polarity of a magnetic field 'stored' by the column and affects the ability of electrons flowing through the column to **tunnel** through the insulator thus affecting the **electrical resistance** of the column as a whole.
- **Reading** is achieved by passing a current down the word line (cell select line) and up the column onto the data line. Depending upon the *spin direction* of the top most layer, more or less current will flow through the tunnelling layer onto the bit line where it can be read. The difference in current represents the difference between a logic **0** or a logic **1**.

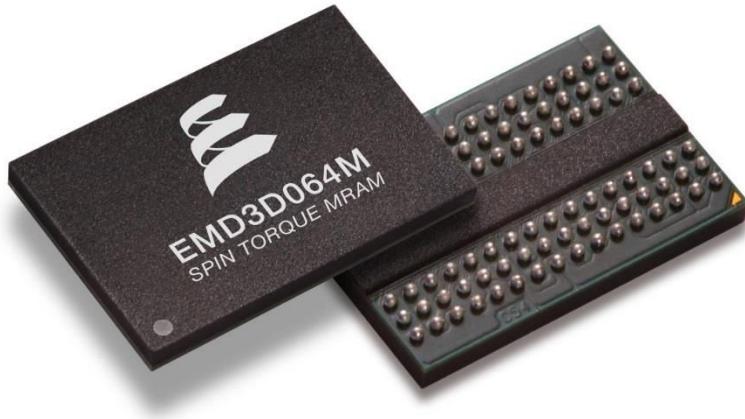


MagRAM Architecture

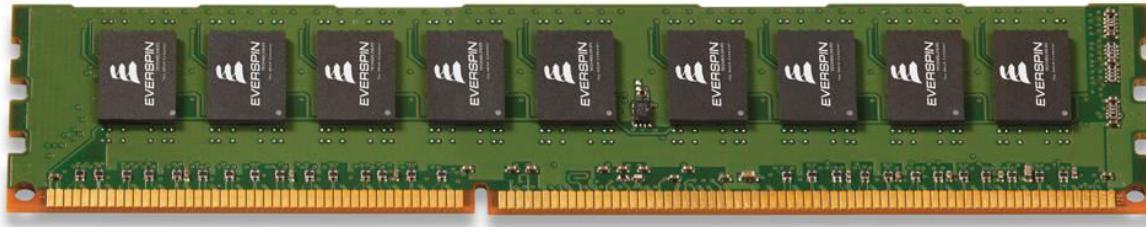


Magnetic Ram (M-Ram)

- Everspins **EMD3D064** – 64Mbit Mram chips



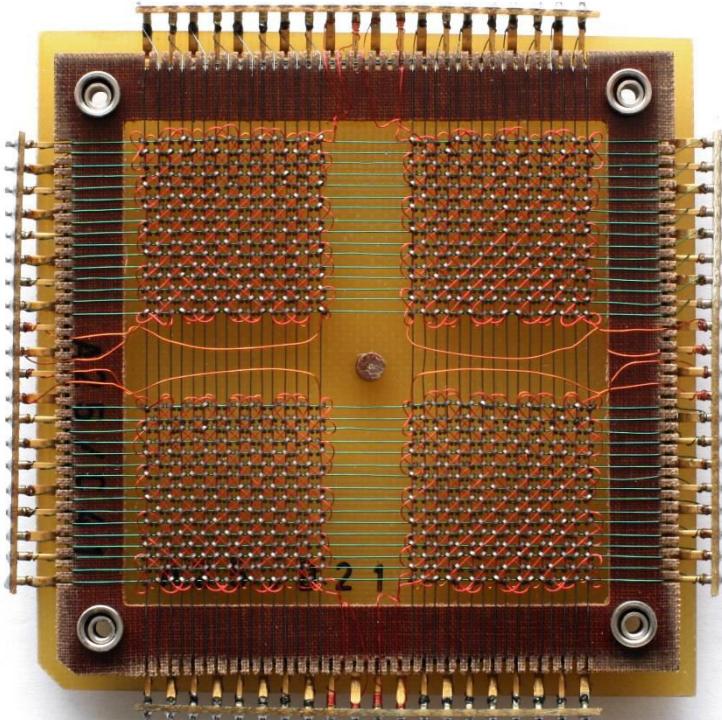
- PC Dram replacement module based on Mram technology (*non volatility is key featuring “instant on” power up without hibernation or sleep mode*)



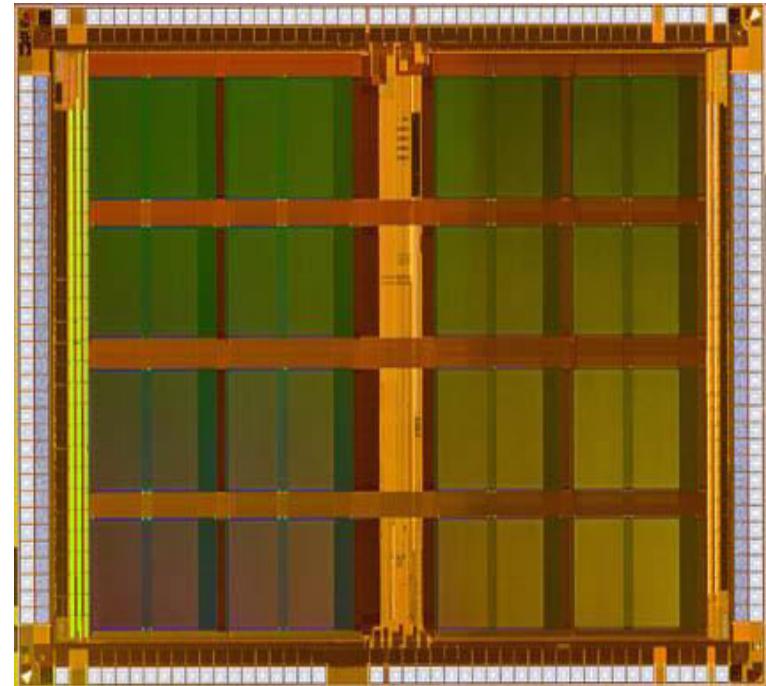
- Can buy **32k byte devices** from digi key for about \$7, check out this link
<http://www.digikey.com/product-detail/en/MR256A08BYS35/819-1024-ND/2665198>

Magnetic Ram (M-Ram)

- Full circle: From 1st generation Magnetic “Core” memory to latest generation MRam



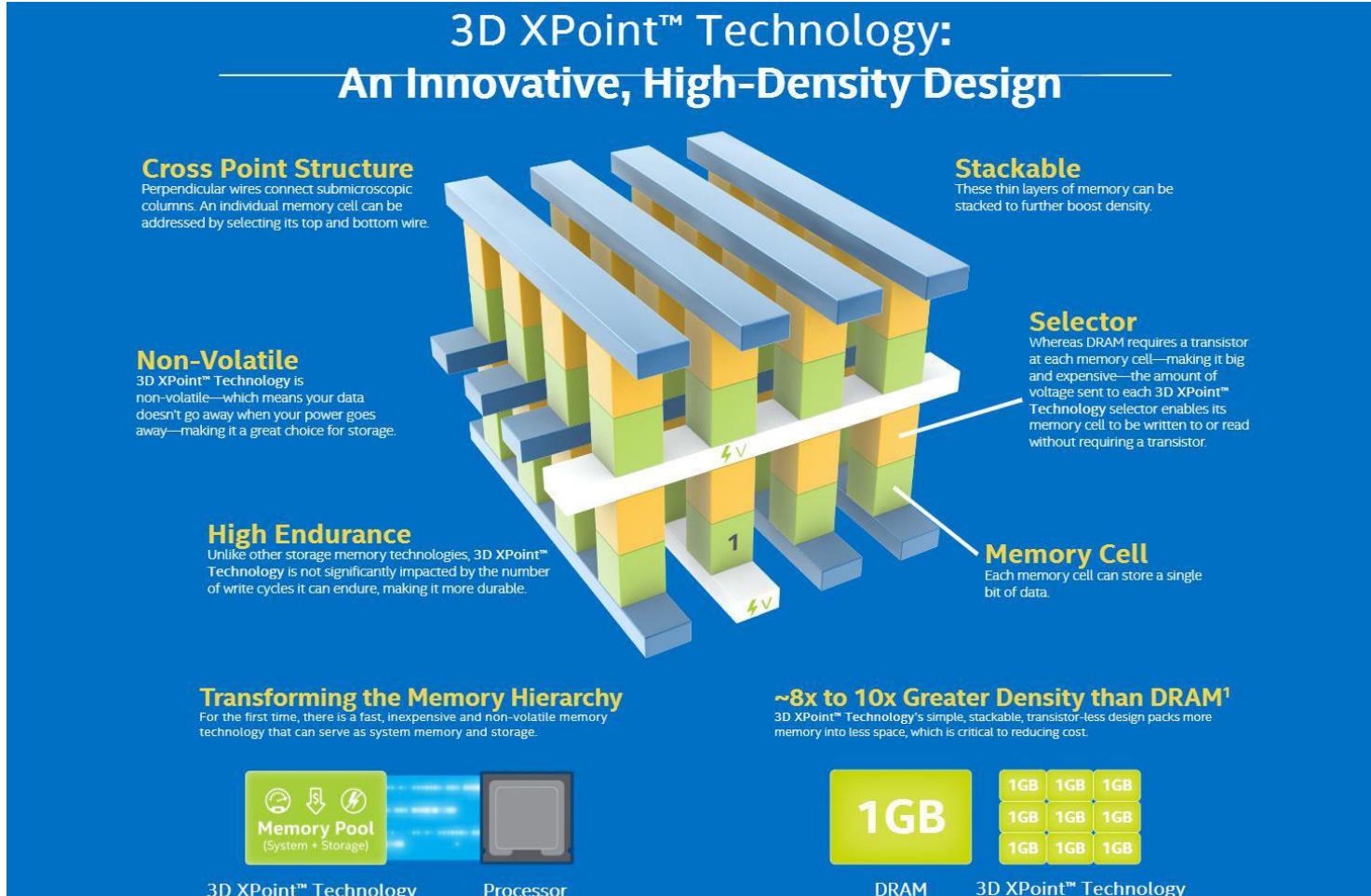
A **1k bit** magnetic “core” memory
(*circa 1960's*)



A **64Mbit** Mram chip from
Everspin/Freescale. (*circa 2015*)

Future Developments

- 2015 Intel/Micron announce 3D XPoint “**Transistor less**”, stackable memory cell technology. Details sketchy and no commercial products yet.



<https://www.youtube.com/watch?v=Wgk4U4qVpNY>