

Memories

1. Memory classification

2 Important memory parameters

3 Parallel and serial memories

4. Internal structure of a memory

5 RAM memories

5.1 Static RAM

5.2 Dynamic RAM

5.3 Synchronous dynamic RAM

6 Permanent memories

6.1 Electrically programmable memories

6.2 NOR FLASH memory

6.3 NAND FLASH memory

Memories

The role of a memory is storing of large amounts of data. There are many physical principles on the basis of which the memory can be constructed. However, almost all semiconductor memories which have experienced a significant expansion are based on **CMOS** technology. In this text, main types of semiconductor memories will be discussed. Development of memories has always proceeded very intensively, so that their essential parameters, such as the access time and capacity are rapidly changing. Current data should be found in company materials.

1. Memory classification

Memories can be classified according to different criteria:

Method of addressing:

- **Addressable** memory - data is stored at stable addresses and address is required for both read and write operations.
- **Address-less** memory - addresses are generated automatically by internal circuitry in firmly specified sequence, and therefore external address does not exist. Very frequently used as buffer memories LIFO (Last In - First Out) or FIFO (First In - First Out).

Durability of the data after turning off the power:

- **Volatile** memory - data is lost after the power is turned off.
- **Non-volatile** memory - data is lost after the power is turned off. This aspect applies to the memory circuits, not to the whole memory system - it can be equipped with a backup battery and then even the energy-dependent memories do not lose data.

Speed of writing data:

- **Read-Write** memory, **RWM** - both read and write operations are easy and fast. RWM is also (even more often) referred to by the acronym **RAM** from Random Access Memory. In fact, this abbreviation is incorrect, but generally accepted.
- **Permanent** memory - writing new data is either not possible (contents is given during production) or writing is very slow compared to reading.

2. Important memory parameters

An important parameter characterizing the extent of memory is its **capacity**. It is given as the total number of bits. The unit is kilobit Kb ($1K = 2^{10} = 1024$), megabit Mb ($1M = 2^{20} = 1048576$), or gigabit Gb ($1G = 2^{30}$). The memory capacity can also be specified in the number of words, e.g. 256KB (here "B" means "byte", $1B = 8b$). The **organization** of the memory is also important, i.e. the number of addresses and the length of the word - usually 1 or 2 bytes. For example, $1M \times 8$ means a memory of 2^{20} addresses, at each address there is a word with a length of 1 byte. The total capacity would be 8Mb or 1MB.

An important parameter characterizing the memory speed is the **access time**, marked t_{AC} . It is expressed in nanoseconds ($1ns = 10^{-9}s$) and is measured from the beginning of the reading cycle to the release of valid data. The cycle usually begins by the change of address, but

depending on the type of memory also on the change of other signals. Depending on the type of memory there are also other dynamic parameters.

3. Parallel and serial memories

Typical signal systems are used in communication of the memory with other circuits in a digital system (e.g. a computer). Fig. 1 shows a **parallel access**, where the address and data are multi-bit. Data input (DI) and data output (DO) can be separated, or more often bi-directional (DI/O). Control signals determine the activity of memory (read - write, etc.) and the direction of data transfer. Depending on the type of memory, the control signal systems may differ from each other.

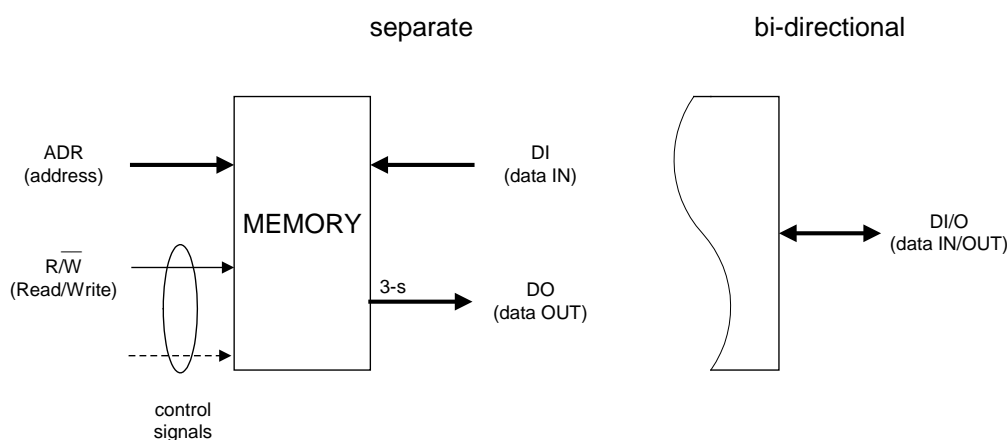


Fig. 1: Memory with parallel access

In addition to the parallel access, there is also a serial access. Serial memories are used where it is appropriate to reduce the number of connections and slower transmission is not a problem. A typical form of a serial memory is shown in Fig. 2.

It is a so-called **three-wire connection** (signals SI, SO, SCK), very frequent in computer peripherals **SPI** (Serial Peripheral Interface). A command is transferred from the SI input bit by bit to the internal shift register; this determines the next memory operation (read, write, etc.). Then the address is transferred. In case of a write cycle, the data to be written is further transferred via SI and finally the control circuits perform the required operation with the memory. In case of a read operation, serial data is sent via the SO output after entering the address. The three-state output is controlled by the control circuits.

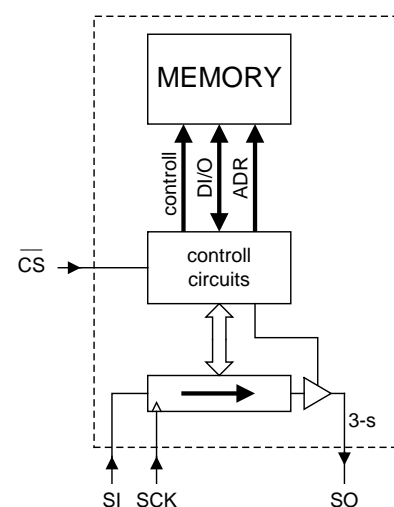


Fig. 2: Memory with serial access

The clock pulses SCK for the shift register must be generated by external circuits. The serial memory has an advantage of a package with a small number of pins. However, the disadvantage is the delay caused by bit-by-bit transmissions. In addition to the three-wire connection, there is also a **two-wire connection**, introduced for computer peripherals IIC. For circuits of this type, a selection signal is not used and the serial data is bidirectional.

4. Internal structure of a memory

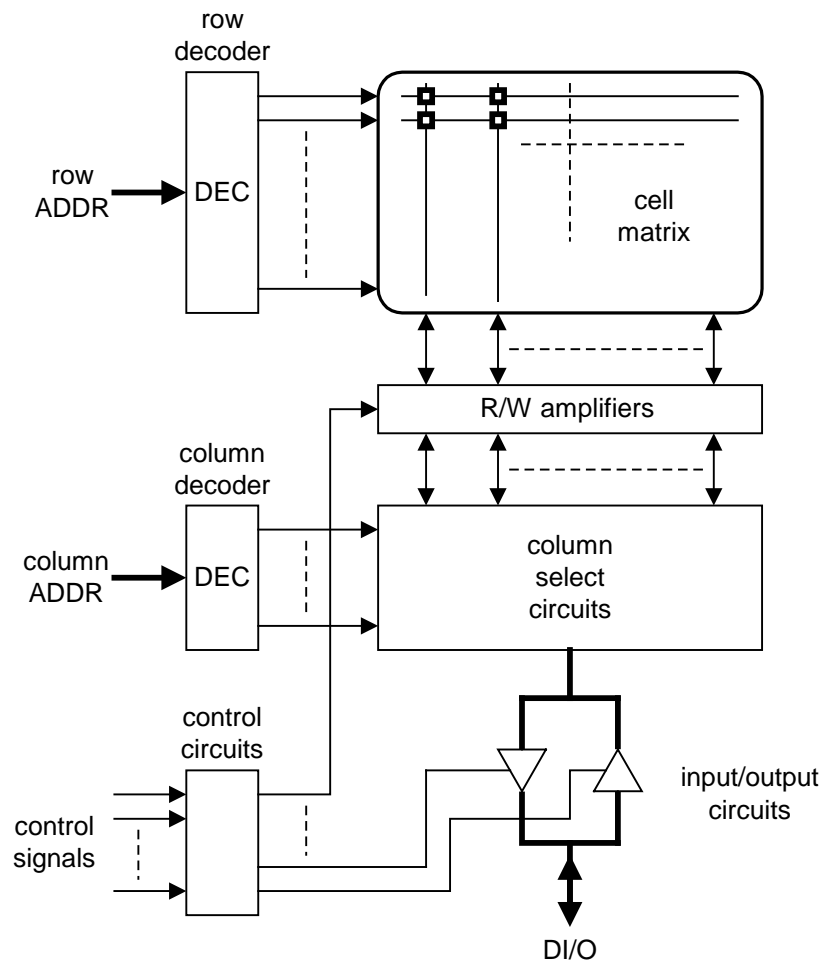


Fig. 3: Internal arrangement of a memory with address selection

Fig. 3 shows the internal circuits of the memory without details. The core is a **matrix** of memory cells, located at the intersection of row and column wires. One part of the address bits always selects one of the rows of the matrix via the **row decoder**. This selects all the cells in a given row and connects them to the column conductors. The second part of the address bits is loaded into the **column decoder**. It controls the **column selection** circuits, which is a group of CMOS switches connected as a large multiplexer/demultiplexer. The selection circuits are then followed by **input/output circuits**. Read and write **amplifiers** are necessary to amplify the signal from memory cells and in dynamic memories also to restore the contents of the cells. In the English literature, row wires are called "**word lines**" and column wires are called "**bit lines**".

Control signals determine the type of cycle of the memory (read - write, etc.), control three-state output circuits, start the power saving mode, etc. The set of control signals is always characteristic of the type of memory.

The switches in the selection circuits are grouped together so that the column decoder selects the groups of columns - this corresponds to the word length of the memory. Fig. 4 shows the principle of selection circuits for an 8-bit word. The column selection circuits serve to data both directions, i.e. for reading and writing.

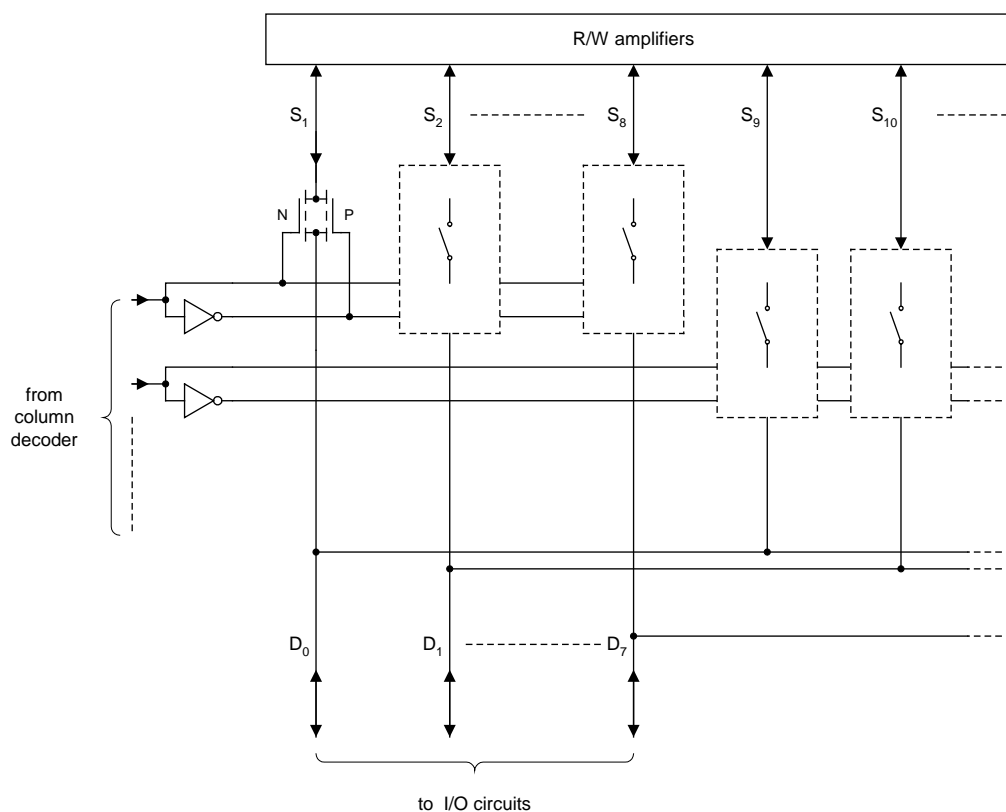


Fig. 4: Principle of column selection circuits

The loading of addresses into the row and column decoder is only outlined in Fig. 3 and differs for specific types of memory. There are three principal options - see Fig. 5. In the figure on the left is the simplest and most common option - a memory receives **simultaneously** all address bits and inside they are divided into two branches which are introduced into the respective decoders. This ensures the shortest delay, but requires large number of connections. Typically, this method of address loading is used in static **SRAMs** (see below).

As a second option, decoders are preceded by registers (indicated by dashed lines in Figure 5 on the left), into which the address is loaded by **Address Strobe** pulse. This can simplify the memory systems with multiplexed bus.

The third option is shown in the right half of Figure 5, where the address is loaded into the memory in **two steps**. The memory is equipped with two registers, controlled individually. The first part (row address) is loaded in the row register by impulse **Row Address Strobe**, and then the second part (column address) is loaded in the column register by **Column Address Strobe**. Both components must be switched at the appropriate time in circuits outside the memory - in the address multiplexer. The total delay caused by the successive writing to the address registers can be tolerated if the loading of the second part of the address

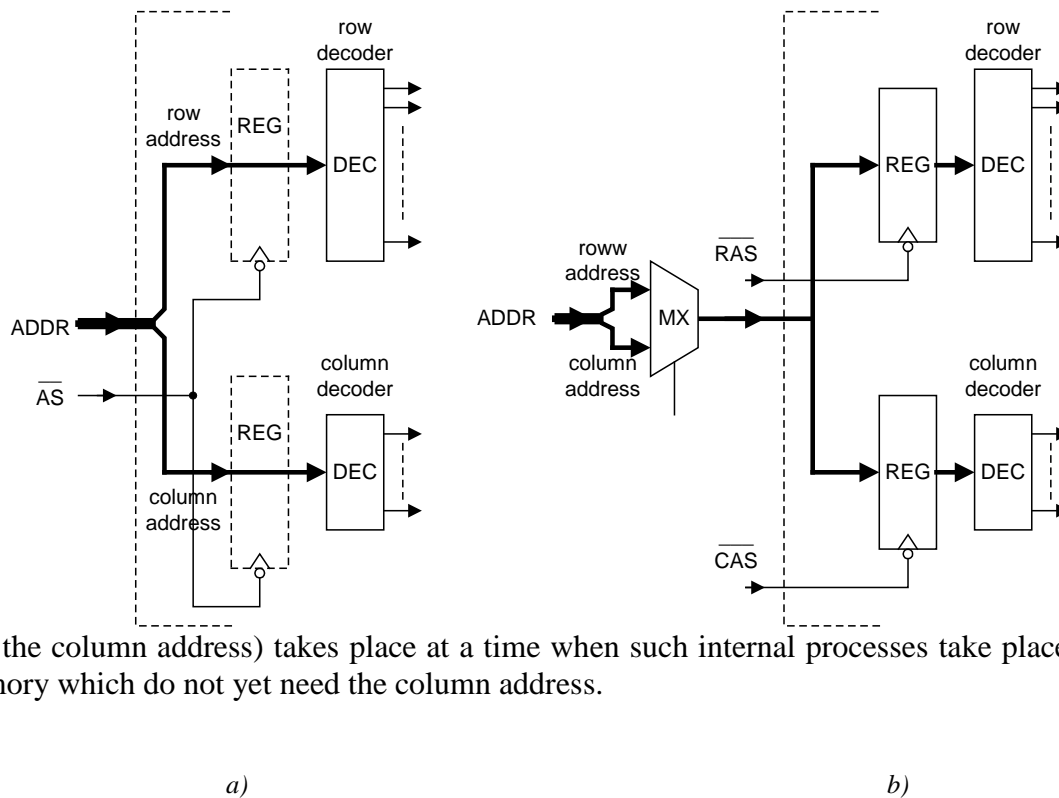


Fig. 5: Options of loading the address into the memory - a) as a whole, b) gradually

Typically, this method of address loading is used for high-capacity **dynamic DRAMs** (see below). Only half the number of package pins is required to load an address in two steps, which reduces the cost of the device.

5 RAM memories

5.1 Static RAM

In static random access memories – **SRAMs** – a bi-stable element of the simplest connection (a latch) is used for storing data in a memory cell. It can be toggled during writing data and its output signal can be processed during reading data. The bit remains stored in the cell until it is overwritten or until the power is turned off.

The overall arrangement corresponds to Fig. 3, 4 and 5 a), the address is delivered as a whole and, usually, directly into the decoders (no registers) - this minimizes delays. The memory matrix consists of six-transistor cells according to Fig. 6. The latch is implemented in the simplest way as two CMOS inverters (with NMOS and PMOS transistors). Four-transistor

cells are also used, where PMOS transistors are replaced by extremely high-value resistors. Latches are connected to two columns by two MOS switches. The voltage difference between the two columns is evaluated during reading by a differential amplifier - see Fig. 3. When writing, this amplifier, via a pair of column conductors and open switches, toggles the cell to the appropriate state. Although the arrangement with two conductors per column complicates the construction of the memory matrix, at the same time it substantially reduces the mutual interference between adjacent columns.

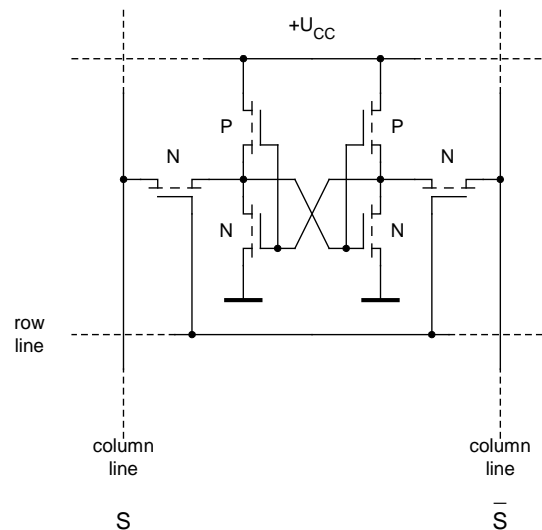


Fig.6: Static RAM memory cell

The system of SRAM control signals is very simple:

- \overline{CS} (Chip Select) in state 1 **blocks** writing to the memory, puts the outputs into the high-impedance state and limits the power consumption when the memory is unused. Power consumption will be limited by lowering the internal supply voltage of the matrix to a value at which the cells reliably hold the stored data. At state 0, the memory is put into operation - but this requires a relatively long time. This procedure is not suitable for quick connection and disconnection of outputs.
- \overline{OE} (Output Enable) has effect only on the output circuits - in state 1 puts them in a high impedance and in state 0 connects them, in both cases in a very short time. It can be roughly characterized as a "**reading pulse**".
- \overline{WR} (Write Enable) with state 0 enables the write operation. It can be roughly characterized as a "**writing pulse**". When writing, the output circuits are blocked.

Tab. 1 summarizes the function of the mentioned control signals ("—" indicates "any" state).

			operation	DI / O	supply current
H	—	—	none	high impedance	reduced
L	L	H	reading	TO	full
L	—	L	writing	DI	full
L	H	H	none	high impedance	full

Tab. 1: Meaning of SRAM control signals

Time diagram and main dynamic parameters are shown in Fig. 7. In the read cycle, especially significant is the **access time** (t_{AC}), the blocking time from \overline{CS} (t_{CZD}), and a substantially shorter blocking and unblocking time from \overline{OE} (t_{OZD}), (t_{ODZ}). In the write cycle, the time of the write pulse distance from the address change (t_{AW}), the minimum write pulse length (t_{WP}), and the setup and hold time of the written data around the end of the write pulse (t_{DS} and t_{DH}) are essential. Interval when the written data must be stable is framed by the thick line.

The signal \overline{CS} does not need to change in each cycle - it can be permanently in state 0. This speeds up the SRAM operation, as the delay t_{CZD} is eliminated. However, the memory then consumes more, as there are no idle times with reduced power consumption.

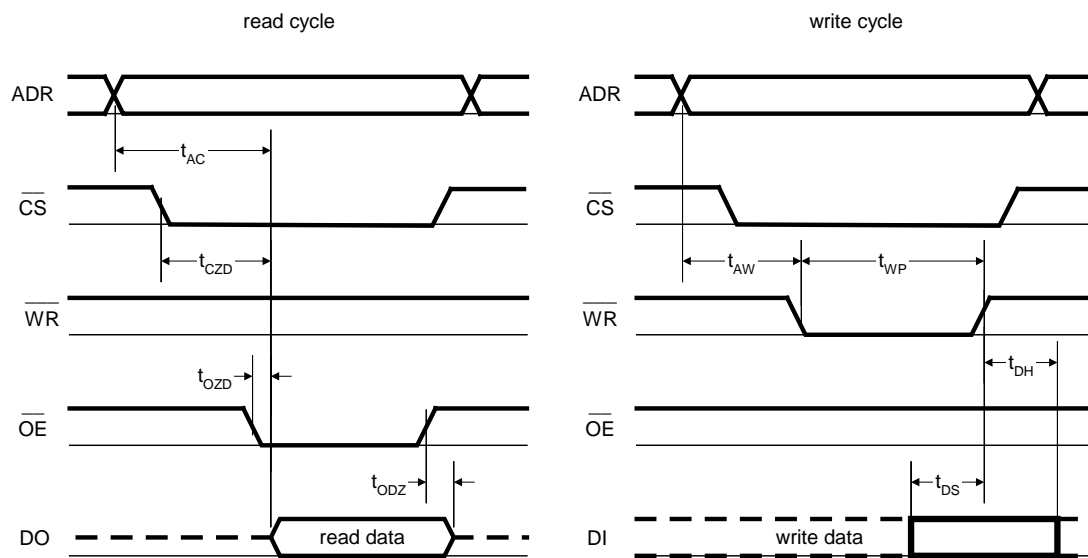


Fig. 7: SRAM read cycle (left) and write cycle (right)

SRAM memories are very fast, access times below 10 ns are not exceptional.

5.2 Dynamic RAM

In dynamic random access memories – **DRAMs** - a charge on a capacitor is used to store digital (two-state) signals. A capacitor, charged or discharged via a simple switch (MOSFET) during writing, retains the charge for a period of time. The voltage on the capacitor is compared with the reference voltage during reading to distinguish between states 0 and 1. This realizes the memory per bit of information. However, the small capacitors are subject to fast discharging. To prevent the loss of information, the charge must be renewed or **refreshed** often enough.

The overall arrangement of the DRAM is shown in Fig. 8. The address is loaded into the memory in two steps. First, the row address is written to the row register by pulse \overline{RAS} , then the column address is written to the column register by pulse \overline{CAS} . The memory array contains cells in their simplest form. Therefore, on the same area of the chip, DRAM has a

larger capacity than SRAM with its more complex six-transistor cells and a pair of columns. This corresponds to the **lower price per bit** for DRAMs.

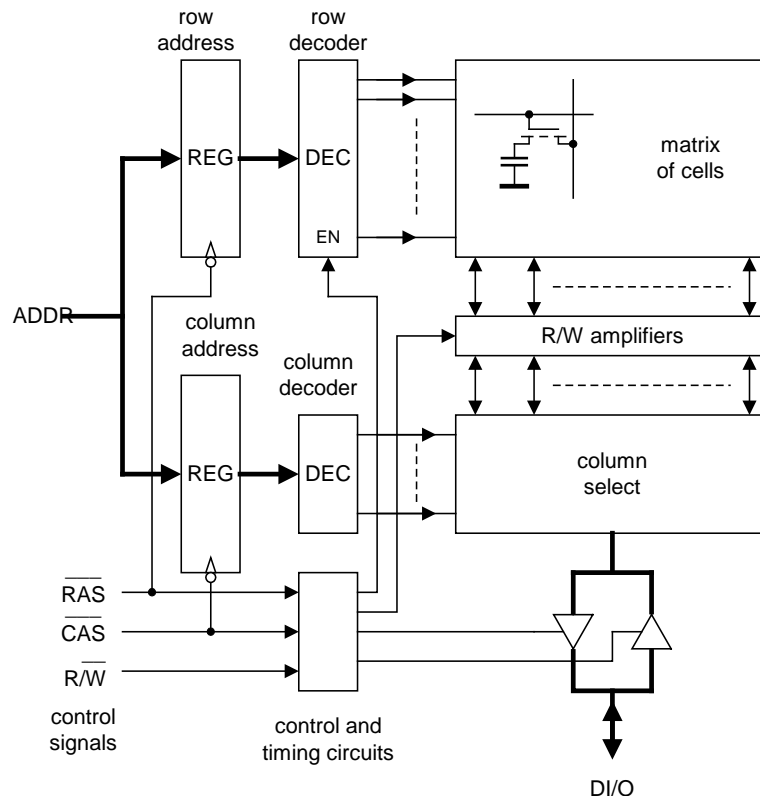


Fig.8: Internal arrangement of DRAM memory

However, the problem is reading the cell. The memory capacitor has a very small capacity, in the order of tens of fF (femtofarad, 10^{-15} F). In contrast, the capacitance of the column conductor to ground is much greater. By connecting a memory capacitor to the column conductor, a capacitive divider is created, which reduces the voltage step in proportion to the capacitance ratio – i.e. very significantly.

DRAM manufacturers have developed several clever read amplifier solutions. A very widespread variant is shown in Fig. 9. The read amplifier is designed as a trigger, brought into an equilibrium state at a suitable moment (i.e. with the same voltages at both outputs). Identical voltages are ensured by charging the column capacitor C_S to the reference voltage of approximately half the supply voltage U_{CC} via the switches – this is called **precharge** (preparatory charge). Then the supply voltage (U_{CC}) is applied to the read amplifiers. Due to the positive feedback, the toggling of the trigger depends on a slight asymmetry of both voltages. Unbalance is applied so that at one of the two column conductors is connected the memory cell with its memory capacitor, while at the other is not. It actually serves as a source of reference voltage at a given moment. After complete toggling, the trigger acts as a voltage source with the maximum amplitude ($U_L = 0$, $U_H = U_{CC}$) for the column conductor. Via the still open switch in the cell, its storage capacitor is charged (or discharged) to the extreme voltage value. The information in the cell is thus restored. The cell on the reference column is seemingly unused - but it will be used at a different row address when the column functions are reversed. The alternating use of columns as a memory and as a reference is ensured by a suitable connection of cells with rows and columns of the matrix - cells on odd rows are

column wires start charging, so the memory is ready for the next cycle. The sequence of suitably timed control signals is generated by the internal control and timing circuits. The entire waveforms of the reading cycle are shown in Fig. 10 on the left.

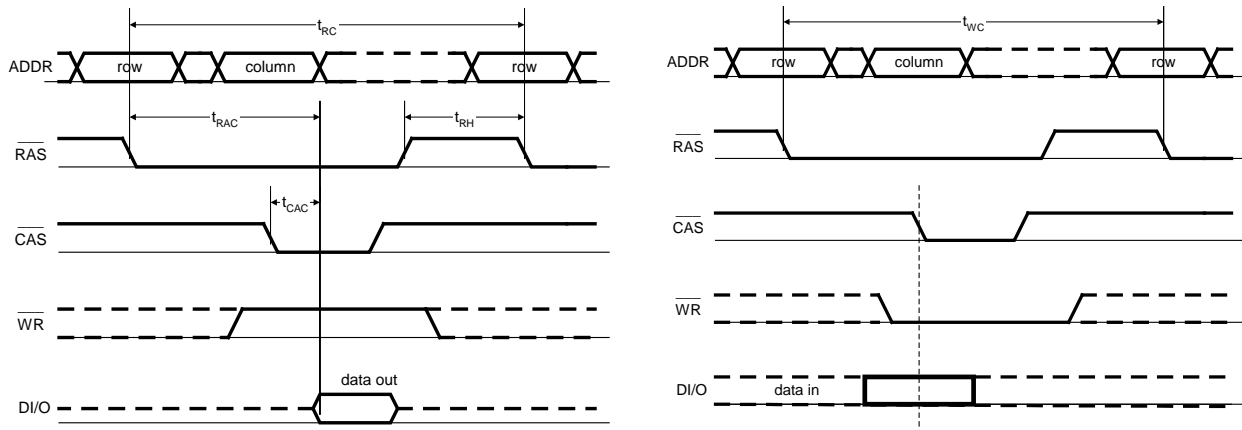


Fig. 10: Read (left) and write (right) DRAM cycle

The figure shows important parameters such as t_{RAC} (access time from \overline{RAS} edge), t_{CAC} (access time from \overline{CAS} edge), t_{RH} (column charge time), and t_{RC} (read cycle time). The write command \overline{WR} must be inactive, i.e. in state 1, around the falling edge of \overline{WR} . The read data is present on the output with a small delay for the time when $\overline{CAS} = 0$, i.e. only for a fraction of the cycle time. The entire access time t_{RAC} is only part of the cycle time. Normally t_{RAC} is around $50ns$, t_{RC} is then around $100ns$. The waveforms of the writing cycle are shown in Fig. 10 on the right. The write command must be active (at state 0) around the falling edge of \overline{CAS} . The interval when written data must be stable is framed by the thick line.

From the principle of dynamic memory cells implies the necessity of **refreshing** data. The refresh interval, i.e. the period of maintaining intact data, depends on the chip temperature, the type of memory and the production technology. At temperatures around $20^{\circ}C$, it can be units of seconds - but without warranty. Catalog data are much more reserved - order of tens of milliseconds. The information on the entire line is refreshed automatically when reading at the address of the line in question. However, during the refresh interval, all rows of the matrix must be refreshed. Therefore, refresh cycles are inserted into normal memory operation. These are abbreviated cycles, limited to inserting the row address and a \overline{RAS} pulse. This triggers the operation of the reading amplifiers and thus the refreshing the contents of all the cells in the row. Entering a column address is unnecessary because no data output from memory is expected in the refresh cycle.

To refresh the entire contents of the memory, all row addresses must be entered sequentially. These addresses are generated by a **refresh counter**, which is part of memory in modern DRAMs. From the external point of view, the refresh cycles are then hidden, automatically generated. Memories of this type are referred to as "**pseudostatic**". However, complete hiding of refresh cycles is not possible because during them the memory is **unavailable** for normal use in read and writes cycles.

Fig. 10 shows a significant difference between the access time and cycle time. The cycle time limits the frequency of repeated operations and the access time is underutilized. Dynamic memories allow significant acceleration of repetitive operations under certain conditions. It is a **page mode** reading or writing. After entering the row address and \overline{RAS} pulse, the content of the whole row is stored in the reading amplifiers. Then it is sufficient to repeatedly supply the increasing column addresses and accompany them only with \overline{CAS} pulses, without changing the row address - see Fig. 11. In this mode, the data output is always extended up to the falling edge of \overline{CAS} , and the system connected to the memory has data accessible over the whole period of \overline{CAS} . The output is placed in the high impedance state after the signals \overline{RAS} and \overline{CAS} return in state 1. The same applies to page mode writing.

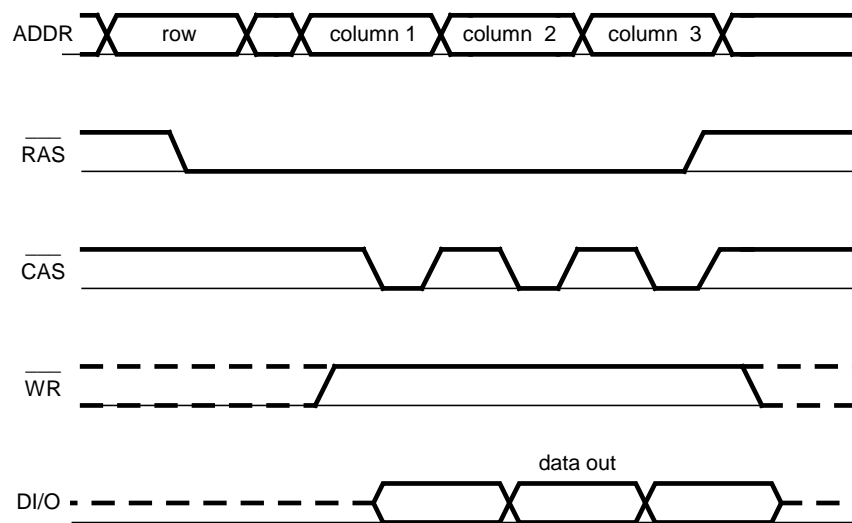


Fig. 11: Page mode reading

To simplify DRAM operating in page mode, modern memories are equipped with a **column address counter** that generates increasing addresses. It is possible to pre-program the starting address and page length. Of course, there is still the possibility of working by complete addresses (both rows and columns).

Further accelerated operation is achieved in **DDR** memories - **double data rate** - where the output data occurs at **both edges** \overline{CAS} , i.e. at double speed, with the period of the order of nanoseconds. However, the use of these parameters requires very precise timing of control signals (RAS, CAS, WR) – see the text below and Fig. 13.

5.3 Synchronous dynamic RAM

Significant acceleration of dynamic memory operation has been achieved by the introduction of synchronous dynamic memories. Synchronous dynamic RAMs, abbreviated **SDRAM**, use the same principle of memory cells, read amplifiers and address decoding as conventional DRAMs. However, they are equipped with other internal circuits that significantly speed up the reading and writing of data and simplify the timing of control signals for memory.

A significant change is the introduction of **synchronization** (clock) pulses. All input signals - addresses, control signals, data - are sampled by the edge of the clock pulse, i.e. loaded in the internal registers. The internal signals thus change at precisely defined moments and, conversely, the timing of the external signals is much less demanding than with a simple DRAM. Even the output data from the memory changes depending on the clock pulses. The entire operation of SDRAM can thus be synchronized with other circuits in the system.

Acceleration of SDRAM operation is achieved by a similar principle as with the DRAM memory described above. Memory mode is optimized for page mode operation, or **burst** read/write. The internal counter, incremented by clock pulses, addresses columns.

Figure 12 shows the internal arrangement of SDRAM. The main parts are memory blocks (four in the figure) consisting of the main circuits of DRAM memory - it is a row register and decoder, a memory matrix with amplifiers, and column selection. Other circuits are common to the entire SDRAM. Each block can be in different **mode**: precharge, refresh, read/write data. This helps to speed up memory operation, as while one block is performing a read or write operation, another may be performing a precharge and another may perform a refresh.

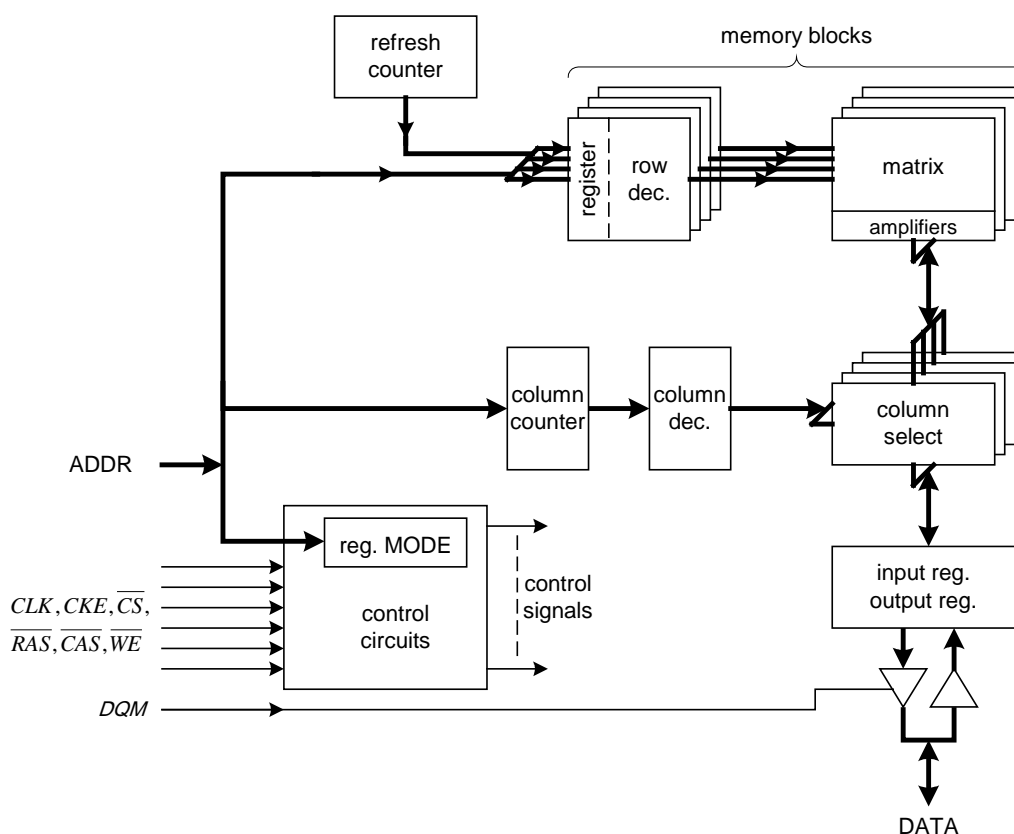


Fig. 12: Internal SDRAM layout

The system of control signals and especially their function differs from DRAM. Some are similar, but some are completely new - **CLK, CKE, CS**. The meaning of the synchronization pulses has already been explained. The basic cycle of reading or writing is shown in Fig. 13.

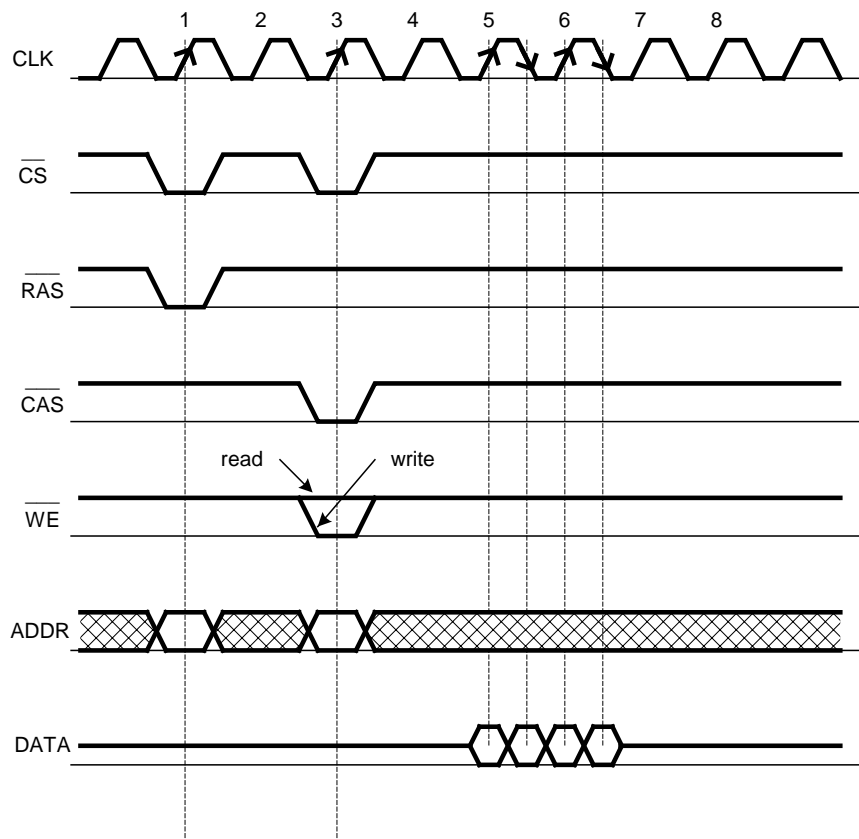


Fig. 13: SDRAM-DDR memory read and write cycle

The latency of the first issued data after \overline{CAS} is called the memory **latency time**. The number of words issued or the **length of the burst** is programmed in the memory control circuits. The column address is automatically generated by a column counter. When writing data, the data must be delivered to the memory - the same applies to the automatic increase of the column address and the length of the batch. All time parameters reported for DRAM in nanoseconds are defined very simply in the case of SDRAM in the number of **CLK cycles**:

- maximum CLK frequency
- distance \overline{CAS} after \overline{RAS}
- latency time after \overline{CAS}
- batch length - is programmable within certain limits
- precharge time

SDRAM dynamic parameters are specified as **three numbers**:

- latency
- distance \overline{CAS} after \overline{RAS}
- precharge time

Another parameter is the maximum CLK frequency. An example of a specification might be the memory:

256Mb , 32 M \times 8 , 133MHz, **3-2-2** .

6 Permanent memories

An essential feature of any non-volatile memory is that it is energy independent - stored data is not lost when the power is turned off. There are several types of non-volatile memories:

- **ROM** (Read-Only Memory) - content is given during production and can no longer be changed.
- **PROM** (Programmable ROM) - the content can be programmed by the user, but the programming is irreversible.
- **EPROM** (Erasable PROM) - the content is programmable and erasable. It is electrically programmed, erased by ultraviolet radiation.
- **EEPROM** (Electrically Erasable PROM) - the content is electrically programmable and electrically erasable. Programming is possible at any address without previous erasing.
- **FLASH** - the content is electrically programmable at any address, but is electrically erasable as a whole or in large parts - sectors or blocks. It must be erased before programming.

Only the last two types are now widely used.

The internal arrangement of the non-volatile memory is shown in Fig. 14. Only the data paths for reading are marked here - the internal signals required for programming the content are not normally shown. DO terminals are always used to enter data for programming; appropriate combinations of control signals are used to write them to the cell matrix.

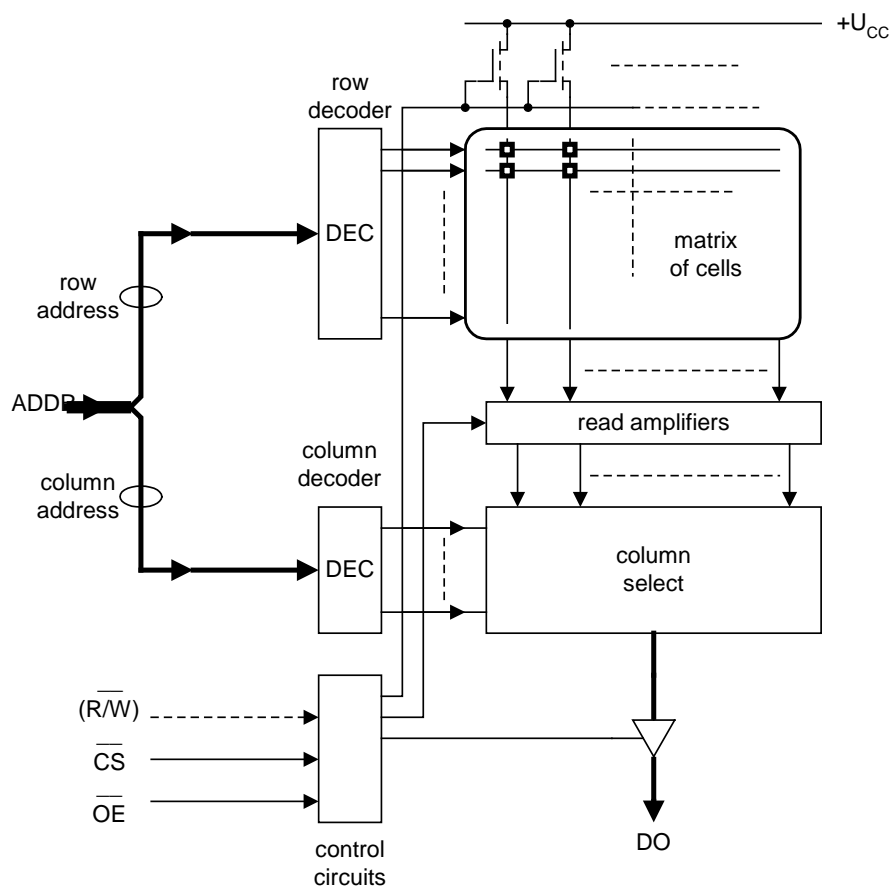


Fig. 14: Internal arrangement of non-volatile memory

6.1 Electrically programmable memories

CMOS technology uses cells in which another conductive layer is formed between the control gate and the semiconductor, totally insulated with silicon oxide - the so-called **floating gate**. Electrons can be transported to the floating gate which thus can be negatively charged. The gate charged in this way creates an electric field in the opposite direction to the control gate field of the NMOS transistor and thus prevents the formation of a conductive channel - the voltage U_H remains on the column conductor. Conversely, in the no-charge state, an induced channel is formed, the column is short-circuited to ground and there is an L level on it. In other words, the charge on the floating gate affects the threshold voltage of the MOS transistor. The principle is illustrated in Figure 15.

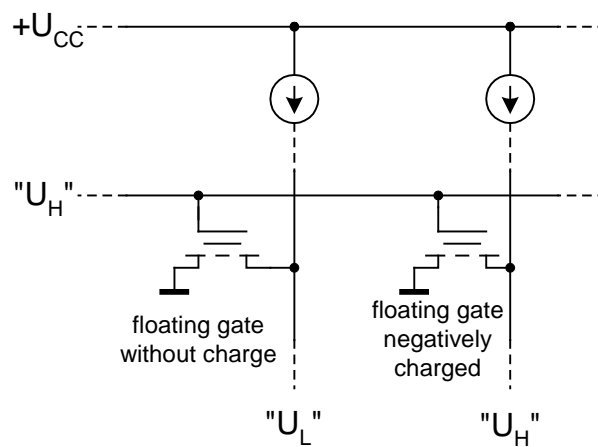


Fig. 15: Memory cell programmed by charge

There are two basic methods of bringing the charge to the floating gate and discharging it. It then determines the method of programming (or erasing) cells in the matrix. First, it is an **avalanche injection** of electrons. In this case, the electrons in the conductive channel are accelerated by an electric field between the ends of the channel and acquire energy sufficient to create a limited avalanche breakdown. At the same time, their path is deflected due to the positive voltage on the control gate. Accelerated electrons ("hot electrons") gain enough energy to pass through a thin layer of oxide under a floating gate. This mechanism of injecting "hot" electrons is very effective and allows the gate to be negatively charged in a short time, on the order of microseconds. Technology enabling this function is called FAMOS (Floating-gate Avalanche-injection MOS). Fig. 16 shows the principle of FAMOS transistor. The thick arrow indicates the path of the electrons.

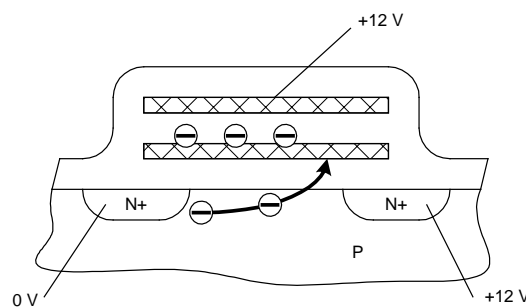


Fig. 16: Cross section of a FAMOS transistor

The second method utilizes quantum effect – transport of electrons through the layer of insulating material with very high electric field intensity, known as **Fowler-Nordheim tunneling**. In order to avoid using high voltage, the insulator layer must be very thin. With a layer of silicon oxide with a thickness under 10 nm, a voltage under 20 V is sufficient. Again, two gates are used - control and floating (totally insulated) - see Fig. 17. The programming voltage is marked as $+U_{PP}$. The figure on the left shows the arrangement for charging the floating gate with a negative voltage, the figure on the right shows the discharge of the charge of the floating gate. Thus, by manipulating the voltage on the floating gate, the state of the memory cell can be controlled. However, the whole mechanism is relatively slow - change of the state of the cells takes a millisecond.

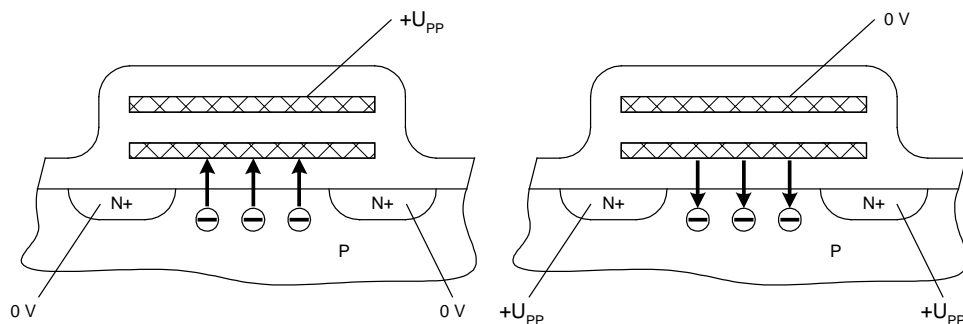


Fig. 17: Arrangement for charging and discharging a floating gate

It remains to explain how the appropriate voltage is applied to the totally insulated floating gate during programming. It is a capacitive coupling between both gates and between a floating gate and a semiconductor. This creates a capacitive divider, which transmits the impulse on the control gate (with the appropriate division ratio) to the floating gate.

The durability of the data in a correctly programmed cell depends mainly on the temperature. It is sufficient for ordinary purposes, in the order of decades.

Both of the above mechanisms are the basis of EEPROM and FLASH type memories.

The **EEPROM memory** uses electron tunneling in both directions. The matrix cell consists of two transistors - one for cell selection and the other for memory operations. The memory transistor has a floating gate and a control gate. The latter only applies when writing data and is not normally drawn in simplified schematic diagrams. The principle is shown in fig. 18 with inscribed voltage values for cell discharging, charging, and cell reading. The programming voltage is marked as $+U_{PP}$.

The EEPROM cell is relatively complex, but allows the floating gates to be charged or discharged at any address. Rather than "programming" or "erasing", it makes sense to talk here about **writing** a value of "0" or "1". This is similar to RAM, but writing is much slower. The number of writing operations per address is limited. Various manufacturers declare tens to hundreds of thousands.

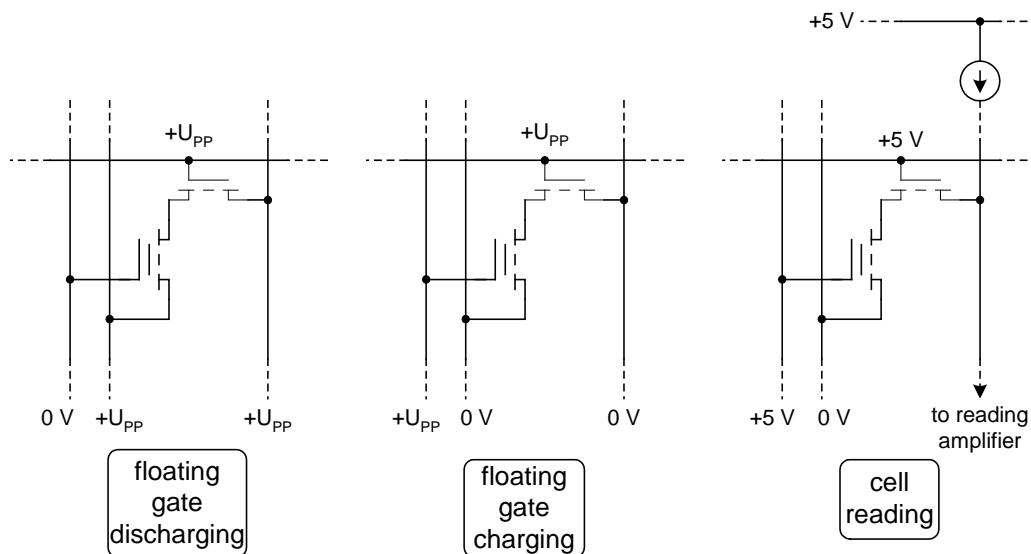


Fig. 18: Operating modes of the EEPROM cell

The **FLASH memory** is based on a cell employing the combination of the avalanche injection of electrons to the negative charge gate (programmed cell), and the principle of electron tunneling to discharge the gate (erased cell). The cells are single-transistor, simpler than EEPROM, and also have fewer terminals. The voltage applied to the cell in different modes is shown in Figure 19.

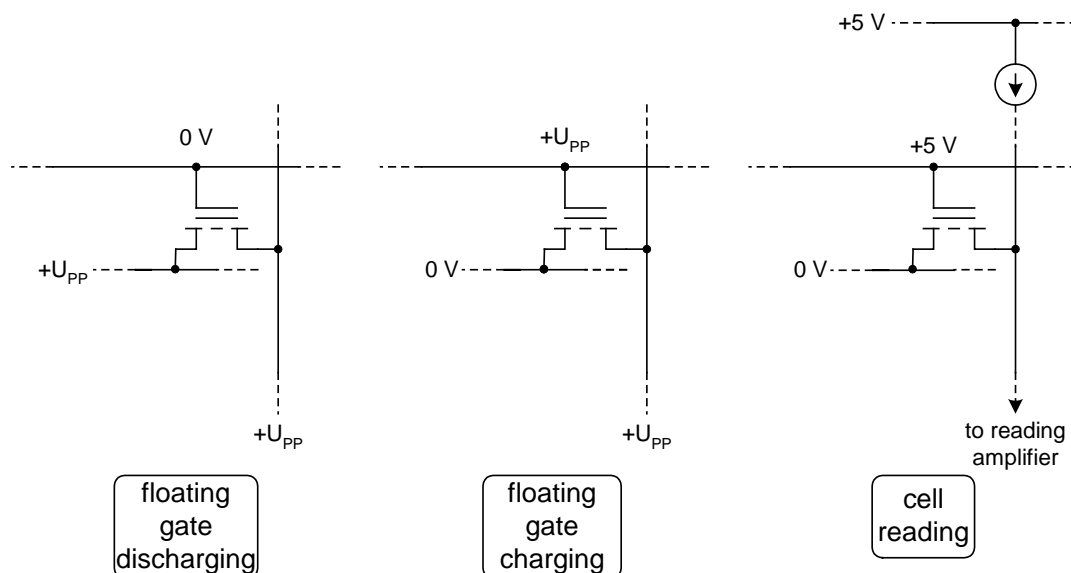


Fig. 19: Operating modes of the FLASH cell

For FLASH memory, the cell must be erased before it can be programmed. Discharge of the floating gate is achieved by applying a high positive voltage to both ends of the channel and zero to the gate - the negative charge is conducted by the tunnel current from the gate to the channel. To charge the gate, an avalanche breakdown is induced by applying a high voltage to

one end of the channel and deflecting fast electrons by a high positive voltage at the gate. If there is zero voltage on the control gate, the floating gate will not be charged. This corresponds to writing of zero or one.

Due to its simple structure, the FLASH memory does not allow selective erasing at each address, but only the **erasing of blocks** (sectors). The entire range of memory is divided into separate units - blocks or sectors - in order to speed up some operations (especially erasing). The block is selected by the highest bits of the address. Programming is quick, of the order of microseconds per bit. Erasing is slow, on the order of milliseconds. However, the entire large area (block) is deleted at once. The procedure for changing the content is such that first the relevant block is deleted and then it is reprogrammed with the new content. FLASH memory with its simple cell allows very **high scale** of integration.

The number of write operations at one address is limited and the guaranteed numbers are approximately the same as for EEPROM (tens to hundreds of thousands). It is important to note that these are the number of writes **at one address**, not in the entire memory - the number of writes for the entire memory is much larger.

Usually, EEPROM and FLASH memories are integrated in the system - therefore they are programmed and erased directly in the system. They are either part of the integrated circuit or they are soldered on the printed board. In latter case, some signals must be switched by additional circuits (multiplexers) or must be reserved for programming.

Programming of modern types of EEPROM and FLASH is greatly facilitated by the fact that, in addition to the memory itself, a sequential circuit - **a programming controller** - which controls the internal operations is also integrated directly on the chip. Also, the increased programming voltage (e.g. 20V) is generated on the chip by an integrated **charge pump**, so that from the outside look the memory is supplied only by the normal operating voltage (5V or less). Details of writing, erasing, and other operations depend on the type and manufacturer.

The slow write operation is significantly accelerated by adding an internal shift register at the input/output of the memory. Then the data is not programmed directly into the individual cells, but first is rapidly loaded into the register. When it is full, its individual flip-flops are connected to the columns of the cell matrix and the data is moved to the whole row of cells at once.

The access time of the EEPROM and FLASH memories is relatively long in comparison with fast SRAMs (10-times). This disadvantage has been mitigated in **synchronous** EEPROM and FLASH with a similar trick as in SDRAM - the memory is equipped with an internal column address counter and data is output in the rhythm of clock pulses.

6.2 NOR FLASH memory

Developments in the nonvolatile memories was concentrated on EEPROM and mainly on FLASH. A new arrangement of the cell matrix, called **NAND FLASH**, has appeared. The memory with the classical arrangement of cells is called **NOR FLASH** for a certain similarity with a logic NOR gate, in which a logic NOR function is created by parallel connections of transistors to the column conductor. However, the connection of memory cells has nothing to

do with the logical function. This arrangement requires the connection of each cell to both the column conductor and the ground - see Fig. 20. These connections occupy a large part of the cell area. The figure shows the signal paths for reading. The row wires are excited from the row decoder and the columns are connected with the selection circuits.

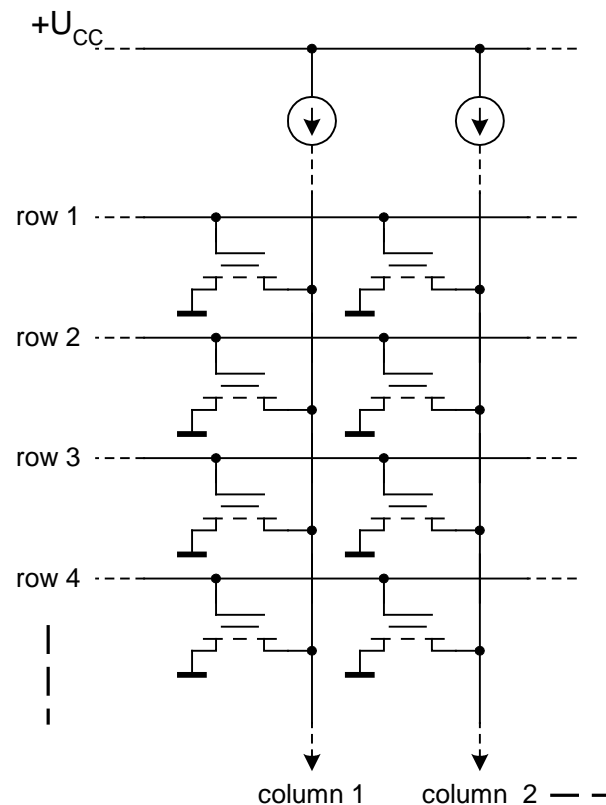


Fig. 20: Cell connection in NOR FLASH

NOR FLASH memory is reliable and has proven itself as a **program memory** for single-chip computers and other digital systems.

6.3 NAND FLASH memory

Significantly simpler cell interconnection is achieved with **serial** cell interconnection with NAND FLASH memory. Cell connection is completely different from NOR FLASH and cell function as well. The production technology is also different. The common principle is the transfer of charge by **tunneling electrons**, both during programming and during erasing - in contrast to NOR FLASH, which is programmed by an avalanche breakthrough. The arrangement of the cells is shown in Fig. 21.

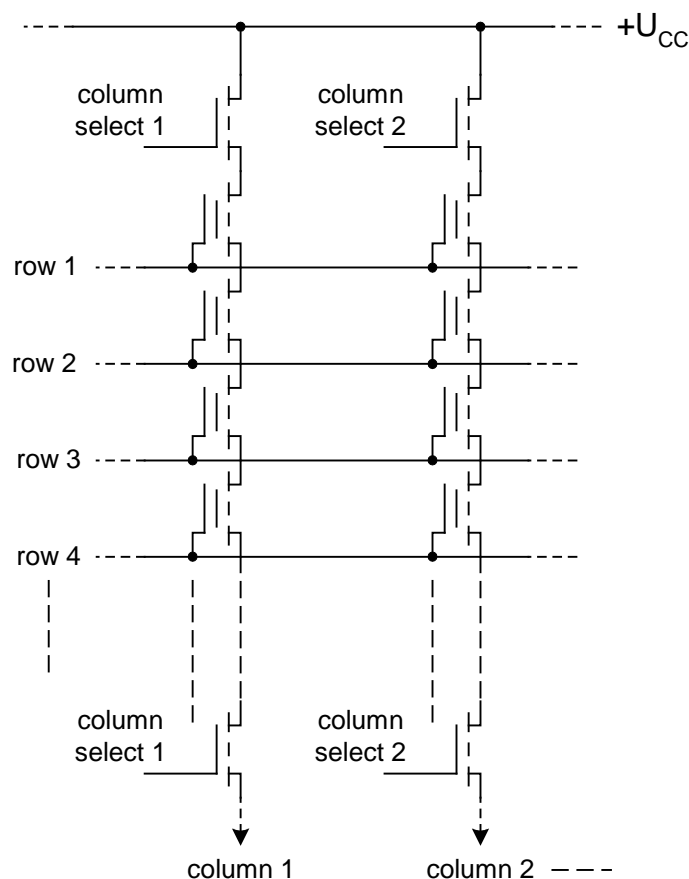


Fig. 21: Cell connection for NAND FLASH

The connection of cells in series resembles the connection of transistors in logic NAND gates. The channels of the neighboring transistors are interconnected and the only terminal of the cell is the control gate. Thus, a much higher **cell density** is achieved on the chip than with NOR memory. However, the cell column must be treated differently from NOR FLASH. In order to determine the state of the selected cell, i.e. the charge on its floating gate, all other cells must be **fully conductive**. Therefore, for unselected cells, the voltage at the row wires is so high that the channel is conductive even with a negative charge on the floating gate. Conversely, for a selected cell, only such a large voltage is applied to the control gate that the conductivity of the channel is fully affected by the charge on the floating gate. This is different from NOR FLASH, where zero voltage is applied to unselected rows.

During the development, the technological processes have been so improved that they made it possible to distinguish the charge on the floating gate very accurately. When reading from a cell, the reading amplifier can then distinguish **several voltage levels** at the output of the column. In this way, not one but 2 bits of information (with 4 levels) or 3 bits (with 8 levels) can be written to the cell, perhaps even more. This increases the total memory capacity proportionally.

However, the development went even further. In NOR FLASH and older NAND FLASH, both control and floating gates are made of conductive material (polycrystalline silicon) and it is necessary to separate and isolate them not only from each other, but also between adjacent

cells, which are close to each other in NAND FLASH. Silicon oxide (SiO_2) is used for insulation. For newer memories, a layer of **silicon nitride** (Si_3N_4) is used instead of a conductive layer, which is an excellent insulator with approximately twice the ϵ_r (rel. permittivity) of silicon oxide. The charge is trapped at the nitride-silicon oxide interface, and a higher ϵ_r reduces the voltage required to read the cell. Another advantage is that since the nitride is an insulator, its layer does not have to be interrupted between adjacent cells. The same, of course, applies to the oxide layer. This greatly simplifies the construction of the cell column. This technology is called **Charge-Trap NAND** - see Fig. 22. The row selection wires "R" are indicated only symbolically; in fact, they are made of polycrystalline silicon and are placed on the surface.

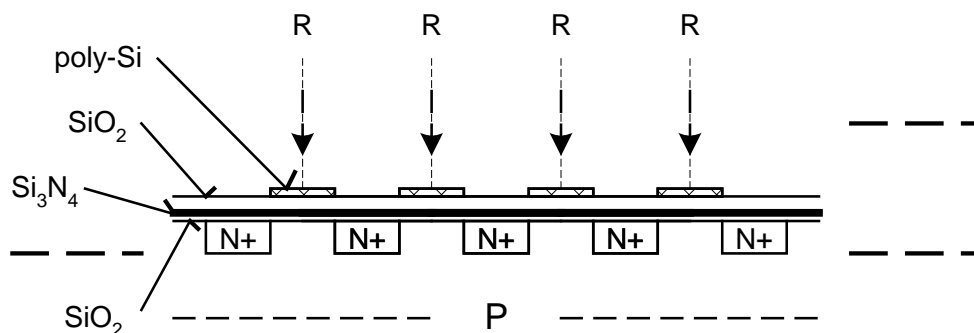


Fig. 22: Principle of Charge - Trap NAND

In recent years, there has been another leap in technology, especially mastering of precise deep etching. It is possible to create deep ditches to separate the layers or deep openings. This provided the means to implement a **vertical NAND** memory, V-NAND FLASH, also 3-D FLASH. The previous technology created cells horizontally, on the surface of the chip. Vertical technology builds cells **in layers**. This achieves a multiple increase in memory capacity - more than 100 layers, memory capacity of 1 Tb (terabit = 10^{12} b) - data from year 2020.

The disadvantage of NAND FLASH memory is **slower operation** than NOR FLASH memory and a larger **number of errors**. Erasing a block takes hundreds of *ns*, writing a little less. Acceleration is possible with a similar trick as with DRAM - **working on pages**. Read page is loaded into the internal shift register and its contents are then successively (and very fast) shifted out. Acceleration is partly proportional to the page size. The procedure for write operation is similar.

Errors occur quite often due to the crosstalk and charge spreading between cells, low tolerances of voltage levels in multi-bit cells, and other influences. Without **error correction**, these memories would be useless. Therefore, the NAND FLASH memories must work with self-correcting codes. The memory always has **spare blocks** in case of permanent failure or excessive error rate of the used blocks. All control functions are performed by a **memory controller**, which usually operates with several memory chips.

Another problem is the **limited number** of erase/write operations in individual cells (hundreds of thousands), which is unacceptable for some applications. It is true that NOR FLASH memories suffer from the same limitation, but due to their predominant use as

program memories of microcomputers, which are never reprogrammed so many times, this limitation is only theoretical. However, NAND FLASH memories are mostly used as large-capacity memories replacing hard disks, where a large number of writings is natural.

It is important to evenly **distribute the load** on the individual cells - "wear leveling". Since the cells cannot be manipulated separately, the data on the number of write/erase operations refers to the entire block. The memory controller optimizes utilization of individual blocks and monitors their load. If data needs to be overwritten, it is written to the reserve block, and the original block is then erased and placed in the reserve. This significantly reduces the time for writing data - there is no need to wait for erasing the block and then writing data. When allocating blocks, the controller takes into account their load and selects the less loaded ones. The controller also monitors the number of errors to be corrected and completely discards the block if the errors are too many. This allows the memory to run until the reserve is completely exhausted. Particularly demanding is the load balancing of a **flash disk storage**, where the directory area is heavily loaded - this is frequently being overwritten. The directories of NAND FLASH are therefore gradually moved to different blocks and the load is spread.