Contents

List of Figures

List of Tables

1 Introduction

Computer memory is a necessary part of every computer. It holds the instructions that the processor executes and the data that those instructions work with. There are several types of memories in the computer, but essentially they can be divided into primary, that the processor directly works with (mainly operation memory), and secondary, where the processor is storing programs and data, that are not needed all the time (mainly disks). We will take a closer look at the primary memories. The memory elements in the computer are used as:

- internal processor memory
- registers
- stacks
- queues
- tables for various purposes
- memory of the microprograms in processor controller
- main memory including fast cache memories

Computer memory is dynamically progressing computer component. New technologies are being developed constantly (miniaturization, capacity, speed). The price strategy of the memory module manufacturers is constantly changing when new models arrive. The well known memory manufacturers are companies like Hitachi, Samsung, GoldStar, IBM, Intel, Micron, Hyundai, Hynix but also many others [1].

Prior to processors achieving 12MHz clock cycle operation, there was little interest in improving the performance of primary memory. At this clock frequency, the cycle time of processor and memory were approximately the same, and the data from the memory were available every processor clock cycle. This is no longer the case as we enter the era of Gigahertz frequency processors. Since 1980, processor speeds have improved at a rate of 80% annually, while memory speeds have improved at a rate of 7% annually [2]. In order to reduce the performance impact of this rapidly increasing gap, multiple levels of caches have been added, processors have been designed to prefetch and tolerate latency and the memory interface and architecture has undergone many revisions. In most cases, the innovative portion is the interface or access mechanism, while the memory core remains essentially unchanged.

The table 1 shows the differences in access speed and size of memories used in computer. As you can see the registers and cache memories are still much faster components compared to ordinary memories, that one can buy in computer stores. The values were acquired from an Intel Pentium 4, 3.2 GHz Server.

The reminder of this thesis is organized as follows: Chapter 2 talks about the history of computer memories, describes their classification and lists types of technologies used. Chapter 3 is dedicated to dynamic memories, first presenting the various aspects of the conventional DRAM, for example, operating modes and refreshing, and then it discusses the improved dynamic memories, like synchronous DRAMs, and double data rate DRAMs. The last section of this chapter introduces packaging styles of dynamic memories. Chapter 4 deals with static memories, it explains how these memories work and shows an example of reading and writing data. Chapter 5 talks about memories with permanent

Component	Access Speed	Size of Component
Registers	1 cycle = 0.3 nanoseconds	8 registers
L1 Cache	3 cycles = 1 nanosecond	16 Kbytes for both data/instruction
L2 Cache	20 cycles = 7 nanoseconds	256 Kbytes, 8-way set associative
L3 Cache	40 cycles = 13 nanoseconds	4096 Kbytes, 8-way set associative
Memory	300 cycles = 100 nanoseconds	16 Gigabytes (max)

Table 1: Computer memory speeds and sizes [3, slide 7]

content, what they are used for and how they operate. Chapter 6 (Other memory types) consists of three sections, in first one the video memories are described, in second FIFO memories are presented and the third section explains how cache memories are organized. Chapter 7 gives a short description of memory errors, their detection and correction. Chapter 8 concludes this document.

2 Computer memories

This chapter is divided into three sections. The first section summarizes the history of computer memories, what types were used at the beginning of computer era and how they developed in time. The second section describes the classification of memories considering the differences between the way they are accessed, the possibility of reading from them or writing into them and the way of the elementary cell operation. The third section lists and shortly describes the technologies used to produce computer memories.

2.1 Basic historical summary

- In 1955 **ferrite storage** was based on a principle of magnetized ferrite cores. Their biggest advantage was that they were nonvolatile. The disadvantages were their size, high manufacturing costs and big power and space requirements. It was possible to create relatively big memory capacities [4].
- In **drum memory** magnetic material was applied onto nonmagnetic drum that spinned fast. There were several combined heads (write/read) installed on the drum, that carried out the write and read procedures.
- **Bubble memory** was a magnetic memory, based on the usage of high capacity magnetic shift registers. They were nonvolatile. The size of the bubbles was determined on material properties and the intensity of the magnetic field. The bubbles were generated based on the logic signal $(0, 1)$ in magnetic bubble generator. A plate of a size approximately 400 mm^2 had a capacity of 10 Mbits and transfer speed of 5 Mb/s [5].
- **Delay lines** were based on a principle of sound transmission (information) by wire. The advantages were low manufacturing costs and relatively high speed of the sequence access. They could be used, for example, in display sets and monitors.
- **Semiconductor memories** were one-bit and more-bit shift registers. In 1966 there were first integrated memories with a capacity of 16 bits and from 1969 through 1971 semiconductor bipolar memories are being introduced with a capacity of several millions of bits. They were used as very fast cache memories.
- In 1960 **MOS semiconductor technology** was introduced. Four years later simple integrated MOS circuits appeared. In 1967 MOS circuits with a high level integration were introduced, and MOS shift registers had the size of 1024 bits. In 1974 a bit density on the chip reached 4096 bits. In 1970 **DRAMs** and in 1971 **SRAMs** were presented.
- In 1974 ferrite memories had capacity of 1.2 Mbits.
- In 1980 semiconductor dynamic memories reached capacity of 64 kbits on the chip.
- In 1999 memories with capacity of 256 MBytes on the chip were introduced.
- In 2001 memories with capacity of 512 MBytes on the chip were presented.

2.2 Classification

Based on the access type the memories can be divided into:

- RAM (Random Access Memory) - The access to this type of memory is called random because individual memory cells differ from each other just by their addresses which could be selected randomly and independently from addresses used in previous or next accesses. Memory cells are equivalent considering the access time and the way of its control.
- SAM (Serail Access Memory) - Serial access means that the addresses cannot be generated randomly but sequentially according to an order of the data stored in the memory cells. For example, when after accessing $n-th$ cell it's desired to access $m-th$ cell, where $m-n>1$, then it's necessary to first in sequence carry out accesses to all the cells between $n-th$ and $m-th$ cell. Classic example of the serial access is writing and reading on a punch card or a magnetic tape. A shift register is a representative of SAM in the field of semiconductor memory parts.
- memories with special access types - CAM associative memory, FIFO memories of queue type, FIFO memories of stack type, multigate memories, memories with combined control, FP, EDO and synchronous memories.

Based on a possibility of writing/reading the memories can be divided into:

- RWM (Read Write Memory) - They can be used for both reading and writing during the normal computer operation. Both reading and writing usually take the same time. Their disadvantage is that when the power supply is off, their content disappear, they are volatile.
- ROM (Read Only Memory) - The stored data can only be read from them and when the power supple is off, they will not disappear, they are nonvolatile.
- Combined memories
  - NVRAM (Non Volatile RAM) - They are a combination of RWM and E^2PROM or Flash EPROM.
  - WOM (Write Only Memory) - It's a memory where information can be only written. The question is, what good a memory would be, if it's unable to be read. The name cannot be taken exactly, it states that WOM memories can be read only in certain modes. For example, there are special writers in airplanes that store non-stop everything about controls and the state of the airplane and they can be read in a special devices outside the airplane while it is being checked or after the airplane crashes.
  - WORM (Write Once-Read many times Memory) - This refers to, for example, optical disks CDROM where the data can be written by laser just once and then it can only be read. It is basically mechanical-optical-electric high capacity PROM.

Based on the principle of the elementary cell the memories can be divided into:

- SRAM static memories - The bistable flip-flop or its functionally equal circuit is a memory element (cell) for holding one bit. They are also known as static RWM-RAM.

- DRAM dynamic memories - The value of a memorized bit is determined by an electrical charge in a capacitor. Since the values of the capacitor are far below 1 pF (Pico farad) then unlike ROM, PROM, EPROM and EEPROM (described below), it is required to periodically refresh the charge with special memory cycles, during which the memory is not accessible for the processor. They are also known as dynamic RWM-RAM.

- PROM, EPROM, EEPROM, FLASH (programmable read only memory, erasable programmable read only memory, electronically erasable programmable read only memory) - Programmable memories with cells' technologies FAMOS (Floating gate Avalanche MOS), FLOTOX (FLOating gate Tunnel OXide), ETOX (Extremely Thin-OXide), etc., where the information is stored in a form of a charge in a capacity but it is relatively big and well isolated so the charge is not required to be refreshed for many years.

2.3 Summary of memories based on a technology

Bipolar circuits

- TTL circuits - "Transistor Transistor Logic" - Those technologies are TTL, STTL, LSTTL, ALSTTL. TTL technology is the oldest one and it was used thanks to a single power supply, big logical gain and speed. Memories with MSI integration (16 to 64 bits) were realized with this technology.

- ECL circuits (Emitter Coupled Logic) - This technology uses current switches. These circuits has a signal delay only 1 to 3 ns because emitter bounded transistors work entirely in their active region. Access time is 5 to 10 ns [6]. However, two power supplies are required. In addition, these circuits are not compatible with TTL circuits.

- Bipolar PROM - it's programmed by burning fusible links.

Unipolar circuits The element of all unipolar memory circuits is a MOS (Metal Oxide Semiconductor) type field effected transistor and particular unipolar technologies differ in the type of the transistor channel, circuit control, manufacturing procedure, etc. This technology is slower compared to bipolar technology, but there is a possibility of the higher integration level. It is currently most often used in memories.

- Unipolar memory elements
  - PMOS (MOS transistors with P type drain).
  - NMOS (MOS transistors with N type drain).
  - CMOS (Complementary MOS) - PMOS and NMOS transistor pairs are used.
  - HMOS (High performance MOS) - developed from NMOS transistors by Intel.

- Unipolar memory technology
  - principle of SRAM bit cell - the cell is formed as a bistable flip-flop, several transistors are required but no refresh is required.
  - principle of DRAM bit cell - the cell consists of one transistor and one capacitor, refresh is required.

- principle of EPROM unipolar technology - the stored information can be erased with ultraviolet light.
- principle of E^2PROM unipolar technology - the stored information can be erased with electrical current.

PLA (Programmable Logic Array) In this technology bipolar and unipolar technologies are used. It is possible to realize sequence logic circuits. They were used only for special aplications (arithmetical units, etc.).

FPGA (Field Programmable Gate Array) It is a technology between PLAs (Programmable Logic Array) and custom gate arrays. The FPGA is a programmable logic device that consists of an array of logic blocks, surrounded by programmable I/O blocks, and connected with programmable interconnect. A typical FPGA contains from 64 to tens of thousands of logic blocks and an even greater number of flip-flops.

3 Dynamic memories

In dynamic memories the information is stored in the form of charges in a capacitor. The capacitor can be either charged (logic 1) or discharged (logic 0). In Figure 2 you can see a one transistor memory cell of the dynamic memory. The capacitors are placed in square matrix (or in a matrix that is close to square shape) and the number of address lines is reduced in half of normally required amount. At first the memory is supplied with a row address and on the same lines with column address of the selected cell. Thanks to that the number of lines is reduced and the capacity increased. Unfortunately, there is a certain limitation to be paid in this apparent genius solution. Unlike the flip-flop circuit, that can hold its value for theoretically unlimited time, a capacitor does not have this feature. Since the capacitors are extremely miniature, their capacity is very small - in the order of 10 fF (Femto Farads). This means, that even very small current flowing in or out this capacitor will invoke major changes of capacitor voltage in a short time. It is required to relatively often recharge the voltage on the capacitors - referred as refresh procedure. This procedure does not have to be carried out precisely, special circuits are implemented on the chip for that, there is a need to only read periodically a random cell from each row and as a consequence whole row will be refreshed. Dynamic memories forget everything, what they were told in a the order of 10 milliseconds [1]. More about refreshing can be found in section 3.3.

Figure 1 shows the simplified DRAM memory organization, where each circle represents one memory cell, the lines coming out of both of the decoders are used for selecting the memory cell, and the lines coming out of the memory cells are used for transferring the stored data into the I/O buffer. The memory cells can be placed in blocks one by one and each block has its own row and column decoders. All the elements will be discussed in the next section of this text.

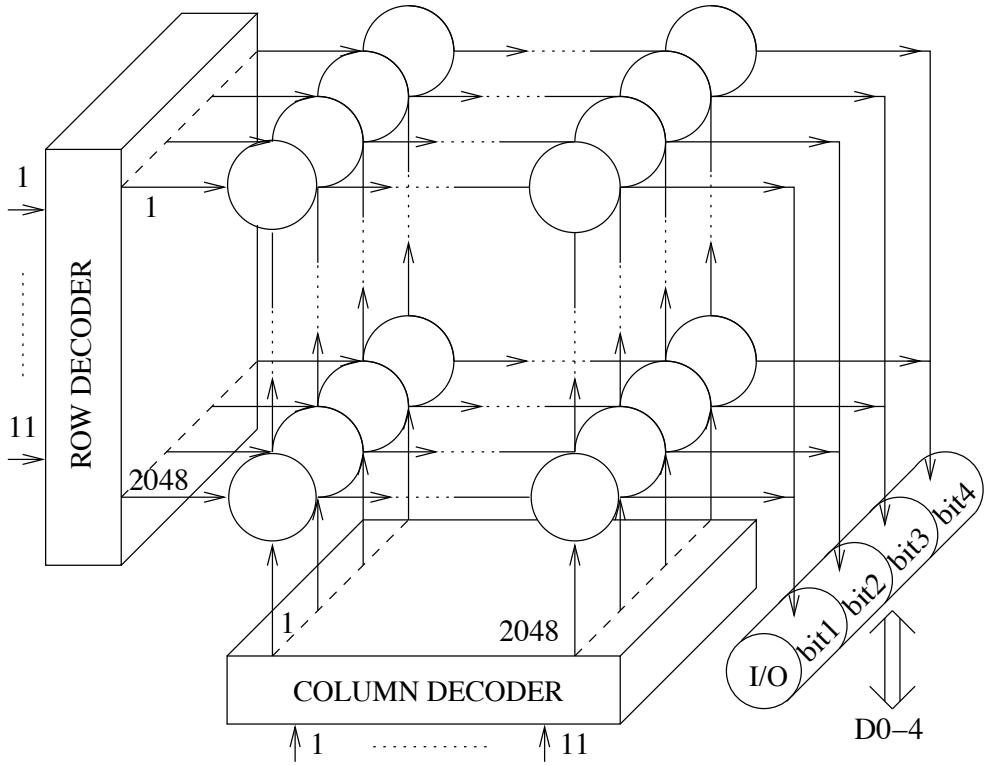

Figure 1: DRAM memory organization [7]

This chapter is divided into three sections. The first section presents the various aspects of the conventional DRAM, for example, operating modes and refreshing. The second section discusses the improved dynamic memories types, like synchronous DRAMs, and double data rate DRAMs. The last section of this chapter introduces packaging styles of dynamic memories.

3.1 Conventional DRAM

3.1.1 DRAM chip organization

With the continual development of memory chips with bigger capacities, various forms of organization have also been established. The 1 Mbit chip with its one data pin has a 1 Mword through 1 bit organization. This means that the memory chip comprises 1M words with a width of one bit per each pin, i.e. it has exactly one data pin. Another widely used organizational form for a 1 Mbit chip is the *256 Kword x 4* bit organization. These chips then have 256 Kwords with a width of four bits, thus they have four pins. The storage capacity is 1 Mbit here, too. Thus the first number always indicates the number of words and the second the number of bits per word. Unlike the *1Mbit x 1* chip, the *256K x 4* chip has four data pins because in a memory access one word is always output or read.

Nowadays, there is a multitude of memory chips in a wide range of organizational forms (x1, x4, x8, x16). DRAMs with an organization of more than 4 bit parallel are often called wide DRAMs. The main feature here is the number of data pins present, i.e. the width in which a data word can be input or output during a memory access. Therefore, a *1M x 8* type chip has eight data input and output buffers. Moreover, the memory array of these chips is divided into at least eight sub arrays, which

are usually assigned to one data pin each. The higher the number of data pins for a memory chip, the fewer chips you require for creating a memory module or for implementing a graphics memory for a graphics adapter.

3.1.2 The operation principle of the DRAM memory cell

The address buffer accepts the memory address output by the external memory controller according to the address received from the CPU. For this purpose, the address is divided into two parts, a row and a column address. These two addresses are read into the address buffer in succession. This process is called multiplexing. The reason for this division is obvious: to address one cell in a 4 Mb chip with 2048 rows and 2048 columns, 22 address bits are required in total (11 for the row and 11 for the column). If all address bits are to be transferred at once, 22 address pins would also be required. Thus the chip package would have to be very large. Moreover, a large address buffer would be necessary. For high integration, it is a disadvantage if all components that establish a connection to their surroundings (for example, the address or data buffer) have to be powerful. This means they have to occupy a relatively large area, because only then they can supply enough current for driving external components such as the memory controller or external data buffers.

Thus it is better to transfer the memory address in two portions. Generally, the address buffer first reads the row address and then the column address. This address multiplexing is controlled by the RAS (row address strobe) and CAS (column address strobe) control signals. If the memory controller passes a row address then it simultaneously activates the RAS signal, that is, it holds the RAS level low. RAS informs the DRAM chip that the supplied address is a row address. Now the DRAM control activates the address buffer to fetch the address and transfers it to the row decoder, which in turn decodes this address. If the memory controller later supplies the column address then it activates the CAS signal. Thus the DRAM control recognizes that the address now transferred is a column address, and activates the address buffer again. The address buffer accepts the supplied address and transfers it to the column decoder. The duration of the RAS and CAS signals as well as their interval (the so-called RAS - CAS delay) must fulfill the specification of the DRAM chip.

The memory cell addressed in this way outputs the stored data, which is amplified by a sense amplifier and transferred to a data output buffer via an I/O gate. The buffer finally supplies the information as read data *Dout* via the data pin of the memory chip.

If data is to be written the memory controller activates the WE (write enable) signal and transfers the write data *Din* to the data input buffer. Via the I/O gate and a sense amplifier, the information is amplified, transferred to the addressed memory cell, and stored in it. The precharge circuit (described later) serves to support the sense amplifier in this action.

The PC's memory control therefore carries out three different tasks: dividing the address received from the CPU into a row and a column address that are transferred to memory in succession; correctly activation the RAS, CAS, WE and READ signals; transferring the write data and accepting the read data. In addition, the memory control must flexibly request wait cycles, when advanced memory concepts such as interleaving and page mode are used, and prepare the addressed memory chips for theses modes (more about this can be found in section 3.1.4). The raw address and data signals from the CPU are not suitable for the memory, thus a memory controller is an essential element of the PC's memory subsystem.

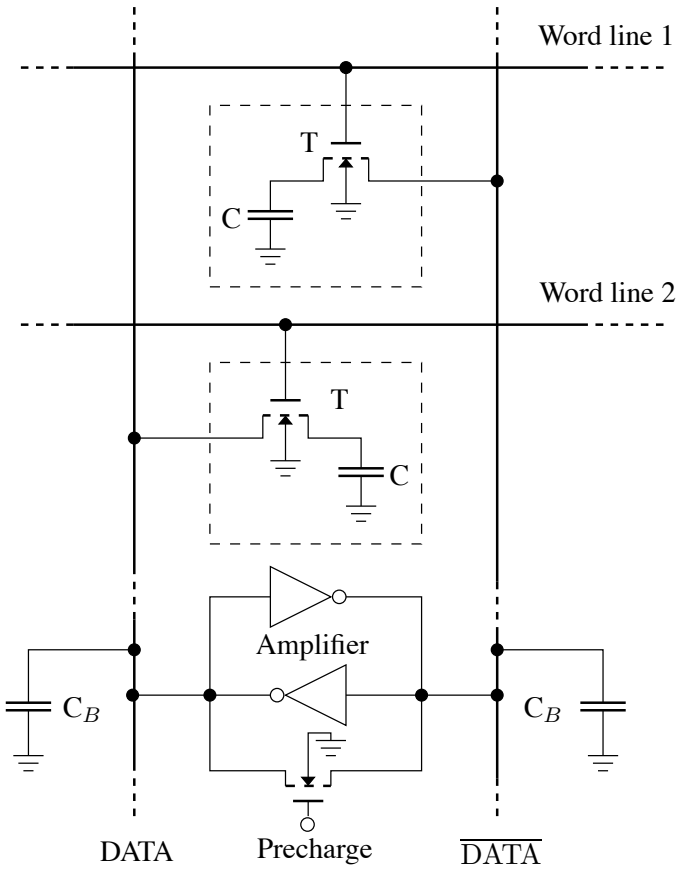

Figure 2: DRAM memory cell with 1 transistor [8]

3.1.3 Reading and writing data

The unit memory cell has a capacitor which holds the data in the form of electrical charges, and an access transistor which serves as a switch for selecting the capacitor. The transistor's gate is connected to the word line. The memory cell array contains one word line, numbered *1* to *n*, for every row formed.

Besides the word lines the memory cell array also comprises so-called bit line pairs DATA and $\overline{\text{DATA}}$. There is a bit line pair for every column in the memory cell array. The bit lines are alternately connected to the sources of the access transistors. Finally, the unit memory cell is the capacitor which constitutes the actual memory element of the cell. One of its electrodes is connected to the drain of the corresponding access transistor, and the other to ground.

The regular arrangement of access transistor, capacitors, word lines and bit line pairs is repeated until the chip's capacity is reached. Thus, for a 4 Mb memory chip, 4 194 304 access transistors, 4 194 304 storage capacitors, 2048 word lines and 2048 bit line pairs are present.

In advance of a memory controller access and the activation of a word line (which is directly connected to this access), the precharge circuit charges all bit lines pairs up to half of the supply potential, that is, Vcc/2. Additionally, the bit line pairs are short-circuited by a transistor so that they are at exactly the same potential. Once this equalizing and precharging process is completed, the precharge circuit is deactivated. The time required for precharging and equalizing is called the RAS

precharge time. The chip can only access one of its memory cells once this process is finished.

When a memory controller addresses a memory cell in the chip, the controller first supplies the row address signal, which is accepted by the address buffer and transferred to the row decoder. At this time a two bit lines of a pair have the same potential Vcc/2. The row decoder decodes the row address signal and activates the word line corresponding to the decoded row address. Now all the access transistors connected to this word line are switched on. The charges of all the storage capacitors of the addressed row flow onto the corresponding bit line and into the capacitors C_B. In the 4 Mb chip described here, 2048 access transistors are thus connected and the charges of 2048 storage capacitors flow onto the 2048 bit line pairs.

The problem, particularly with today's highly integrated memory chips, is that the capacity of the storage capacitors is much lower than the capacity of the bit lines connected to them by the access transistors. Thus the potential of the bit line changes only slightly, typically in order of $\pm$ 100 mV and in a very short time interval [9]. If the storage capacitor is empty, then the potential of the bit line decreases slightly; if charged then the potential increases. The read amplifier activated by the DRAM control amplifies the potential difference on the two bit lines of the pair. In the first case, it draws the potential of the bit line connected to the storage capacitor down to the ground and raises the potential of the other bit line up to Vcc. In the second case, the opposite happens: the bit line connected to the storage capacitor is raised to Vcc and the other bit line decreased to ground.

Without precharging and potential equalization by the precharge circuit, the read amplifier would need to amplify the absolute potential of the bit line. But because the potential change is only about 100 mV, this amplifying process would be less precise and therefore more likely to fail, compared with the difference forming of the two bit lines. Here the dynamic range is $\pm$ 100 mV, that is, 200 mV in total. Thus the precharge circuit enhances the memory reliability.

Each of the 2048 sense amplifiers supplies the amplified storage signal at its output and applies the signal to the I/O gate block. This block has gate circuits with two gate transistors, each controlled by the column decoder. The column decoder decodes the applied column address signal (which is applied after the row address signal) and activates exactly one gate. This means that the data of only one read amplifier is transmitted onto the I/O gate and transferred to the output data buffer. After supplying the row address and undergoing all these actions the column address becomes important. Multiplexing the row and column address therefore has no adverse effect as one might expect at the first glance.

The output data buffer amplifies the data signal again and outputs it as output data *Dout*. At the same time, the potential of the bit line pairs are on a low or a high level according to the data in the memory cell that is connected to the selected word line. Thus they correspond to the stored data. As the access transistors remain on due to the activated word line, the read-out data is written back into the memory cells of one row using capacitors C_B that hold the original data. The reading of a single memory cell therefore simultaneously leads to a refreshing of the whole line. The time period between creating the row address and outputting the data *Dout* via the data output buffer is called RAS access time t_{RAS}, or access time. The much shorter CAS access time t_{CAS} is significant for certain high-speed modes. This access time characterizes the time period between supplying the column address and outputting the data *Dout*.

After completing the data output, the row and column decoders are well as the read amplifiers are disabled again, and the gates in the I/O gate block are switched off. At the time the bit lines are still on the potentials according to the read data. The refreshed memory cells are disconnected from the bit lines by the disabled word line, and the access transistors thus switched off. Now the DRAM control activates the precharge circuit, which lowers and increases the potentials of the bit lines to Vcc/2 and equalizes them again. After stabilization of the whole DRAM circuitry, the chip is ready for another

memory cycle. The time required between the stabilization of the output data and supplying a new row address and activating RAS is called recovery time or RAS precharge time t_{RP}.

Adding the RAS precharge time to the access time gives the cycle time t_{cycle}. Generally, the RAS precharge time lasts about 80% of the access time, so that the cycle time is about 1.8 times longer than the access time. Therefore, a DRAM with an access time of 100 ns has a cycle time of 180 ns. Only when this 180 ns has elapsed can a new access to memory be carried out. Therefore, the time period between two successive memory accesses is not determined by the short access time but by the nearly double cycle time of 180 ns. If one adds the signal propagation delays between CPU and memory on the motherboard of about 20 ns, then an 80286 CPU with an access time of two processor clock cycles may not exceed a clock rate of 10 MHz, otherwise one or more wait states have to be inserted. However, advanced memory concepts such as interleaving can alter the RAS precharge time so that in most cases only the access time is decisive. In page mode or static column mode, even the shortest CAS access time determines the access rate.

Figure 3 shows the behavior of the most important memory signals if the chip carries out a read access.

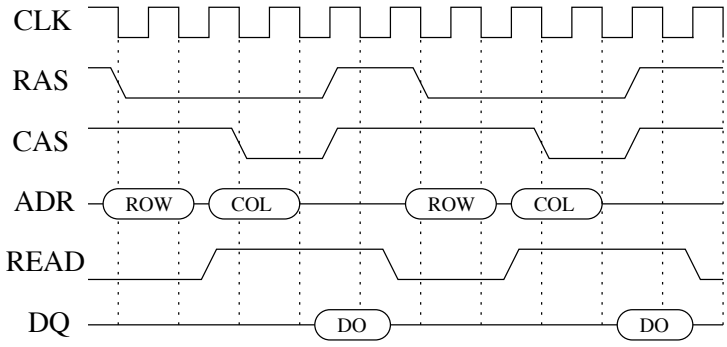

Figure 3: DRAM read time scheme [10, page 9]

Writing data is carried out in nearly the same way as reading data. At first the memory control supplies the row address signal upon an active RAS line address signal. At the same time, it activates the WE control signal to inform the DRAM that it should carry out a write procedure. The data to be written (*Din*) is sent to the data input buffer, amplified and transferred onto the I/O line. The data output buffer is not activated for the data write.

The row decoder decodes the row address signal and activates the corresponding word line. As is the case for data reading, here also the access transistors are turned on and they transfer the stored charges onto the bit line pairs DATA, $\overline{\text{DATA}}$. Afterwards, the memory controller activates the CAS signal and applies the column address via the address buffer to the column decoder. It decodes the address and switches on a single transfer gate through which the data from the I/O line pair is transmitted to the corresponding sense amplifier. This read amplifier amplifies the data signal and raises or lowers the potential of the bit lines in the pair concerned according to the value “1” or “0” of the write data. As the signal from the data input buffer is stronger than that from the memory cell concerned, the amplification of the write data gains the upper hand. The potential on the bit line pair of the selected memory cell reflects the value of the write data. All other read amplifiers amplify the data held in the memory cells so that after a short time potentials are present on all bit line pairs that correspond to the unchanged data and the new write data respectively. These potentials are transferred to the storage capacitors as the corresponding charges. Afterwards, the DRAM controller deactivates

the row decoder, the column decoder and the data input buffer. The capacitors of the memory cells are disconnected from the bit lines and the write process is completed. As was the case for the data read, the precharge circuit sets the bit line pairs to a potential lever Vcc/2 again, and the DRAM is ready for another memory cycle.

Besides the memory cell with one access transistor and one storage capacitor, there are other cell types with several transistors or capacitors. The structure of such cell is, of course, much more complicated and the integration of its elements gets more difficult because of their greater number. Such memory types are therefore mainly used for specific applications, for example, a dual-port RAM where the memory cells have one transistor for reading and another transistor for writing data so that data can be read and written simultaneously. This is advantageous, for example, for video memories because the CPU can write data into the video RAM to set up and image without having to wait for memory to be released. On the other hand, the video controller may continuously read out the memory to drive the monitor. For this purpose, VRAM chips also have a parallel random access port which is used by the CPU for writing data to the video memory. They also have a very fast serial output port that can clock a number of bits, for example a whole memory row. The monitor driver circuit can thus be supplied very quickly and continuously with image data.

Instead of the precharge circuit, other methods can be used. For example, you can install a dummy cell for every column in the memory cell array that holds only half of the charge which corresponds to a "1". Practically, this cell holds the value "1/2". The read amplifiers then compare the potential read from the addressed memory cell with the potential of the dummy cell. The effect is similar to that of the precharge circuit. Here too, a difference is amplified, and not an absolute value. This is done to filter the noise out.

It is not necessary to structure the memory cell array in a square form with an equal number of rows and columns and to use a symmetrical design with 2048 rows and 2048 columns. The designers have complete freedom on this respect. 4 Mb chips often have 1024 rows and 4096 columns simply because the chip is longer than it is wide. In this case, one of the supplied row address bits is used internally as an additional (that is, 12th) column address bit. The ten row address bits select one of 210 rows = 1024 rows, but the 12 column address bits select one of 212 = 4096 columns. In high-capacity memory chips, the memory cell array is also often divided into two or more sub-arrays. In a 4 Mb chip, for example, eight sub-arrays with 512 rows and 1024 columns may be present. One or more row address bits are then used as the sub-array address; the remaining row and column address bits then only select a row or column within the selected sub-array. The word and bit lines thus get shorter and the signals become stronger. But there is a disadvantage: the number of sense amplifiers and I/O gates increases. Such methods are usual, particularly in the new highly integrated DRAMs, because the cells are always decreasing in size. Therefore the capacitors reduce in capacity, so the long bit lines "eat" the signal before it can reach the sense amplifier. Which concept a manufacturer implements for the various chips cannot be recognized from the outside. Moreover, these concepts are often kept secret so that competitors don't get an insight into their rival's technologies.

Figure 4 shows the behavior of the most important memory signals if the chip carries out a write access.

3.1.4 Additional operating modes

The last section described the normal mode of DRAM. Memory chips can also execute one or more different column modes to reduce access time. The best known is page mode. What is actually behind this often-quoted catchword (and the less well-known static-column, nibble and serial modes) is discussed in the following sections. Figure 5 through 8 shows the behavior of the most important

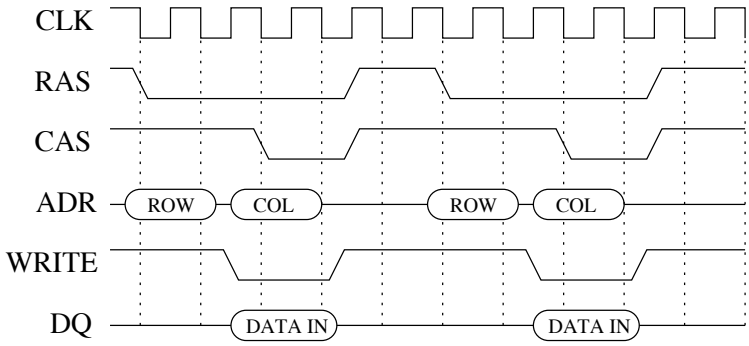

Figure 4: DRAM write time scheme [10, page 10]

memory signals if the chip carries out one of these high-speed modes in a read access.

Page mode Section 3.1.3, *Reading and Writing data*, mentioned that during the course of an access to a unit memory cell in the memory chip, the row address is input first with an active RAS signal, and then the column address with an active CAS signal. In addition, internally all memory cells of the addressed row are read onto the corresponding bit line pair. If the next memory access refers to a memory cell in the same row but another column (that is, the row address remains the same and only the column address has changed), it is not necessary to input and decode the row address again. In page mode, therefore, only the column address is changed, but the row address remains the same. Thus, one page corresponds exactly to one row in the memory cell array. You will find the signal's course in page mode in Figure 5.

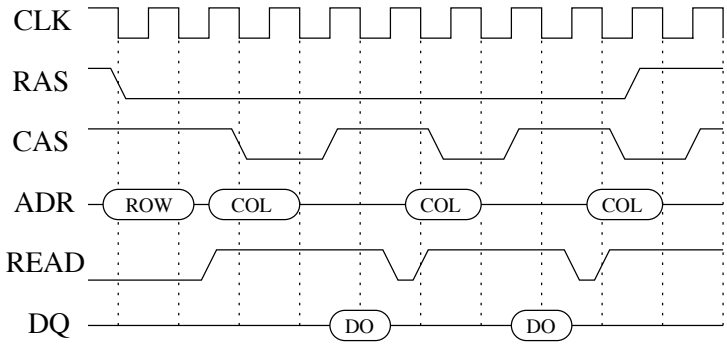

Figure 5: FP DRAM read time scheme [10, page 12]

To start the read access the memory controller first activates the RAS signal as usual, and transfers the row address. The address is transferred to the row decoder, decoded, and the corresponding word line is selected. Now the memory controller activates the CAS signal and passes the column address of the internal memory cell. The column decoder decodes this address and transfers the corresponding value from the addressed bit line pair to the data output buffer. In normal mode the DRAM control would now deactivate the RAS and CAS signals and the access would be completed.

If the memory controller, however, accesses in page mode a memory cell in the same row of the DRAM (that is, in same page) it doesn't deactivate the RAS signal but continues to hold the signal at

an active low level. Instead, only the CAS signal is disabled for a short time, and then reactivated to inform the DRAM control that the already decoded row address is still valid and only a column address is being supplied. All access transistors connected to the word line concerned thus also remain turned on, and all data read-out onto the bit line pairs is held stable by the read amplifiers. The new column address is decoded in the column decoder, which turns on a corresponding transfer gate. Thus, the RAS precharge time as well as the transfer and decoding of the row address is inapplicable for the second and all succeeding accesses to memory cells of the same row in page mode. Only the column address is transferred and decoded. In page mode, access time is about 50% (and the cycle time up to even 70% [9]) shorter than in normal mode. This, of course, applies only to the second and all subsequent accesses. However, for reasons of stability, the period during which the RAS signal remains active may not last for an unlimited time. Typically, 200 accesses within the same page can be carried out before the memory controller has to deactivate the RAS signal for one cycle.

However, operation in page mode is not limited to data reading only: data may be written in page mode, or read and write operations within one page can be alternated. The DRAM need not leave page mode for this purpose. In a 1 Mb chip with a memory cell array of 1024 rows and 1024 columns, one page comprises at least 1024 memory cells. If the RAM is implemented with a width of 32 bits, one main memory page holds 4Kb. As the instruction code and most data tend to form blocks, and the processor rarely accesses data that is more than 4 Kb away from value it has just accessed, the page mode can be used very efficiently to reduce the access and cycle times of the memory chips. However, if the CPU addresses a memory cell in another row (that is, another page), the DRAM must leave page mode and the RAS precharge time makes a significant difference. The same applies, of course, if the RAS signal is disabled by the memory controller.

Hyper page mode (EDO mode) In hyper page mode, also known as EDO mode, the distance (time) between two consecutive CAS activations is shorter than in normal page mode (see Figure 6). Thus column addresses are transferred quicker and access time is significantly shorter (usually by 30% compared with ordinary page mode), therefore the transfer rate is accordingly higher. Please also note that in this EDO mode the CAS signal must rise to a high level before every new column address.

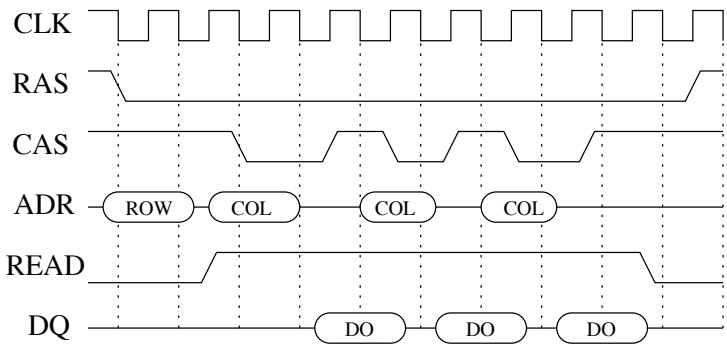

Figure 6: EDO DRAM read time scheme [11, page 14]

Static-column mode Closely related to the page mode is the static-column mode. Here the CAS signal is no longer switched to inform the chip that a new column address is applied. Instead, only the column address supplied changes, and CAS remains unaltered on a low level. The DRAM control

is complex enough to detect the column address change after a short reaction time without toggling CAS. This additionally saves part of the CAS switch and reaction time. Thus static-column mode is even faster than page mode. But here also the RAS and CAS signals may not remain at a low level for an unlimited time. Inside the chip only the corresponding gates are switched through to the output buffer. In static-column mode, therefore, all memory cells of one row are accessible randomly. But DRAM chips with static-column mode are still quite rare and are not widely used in PCs. Some IBM PS/2 modules, though, use static-column chips instead of DRAMs with page mode.

Nibble mode Nibble mode is a simple form of serial mode: by switching CAS four times, four data bits are clocked-out from an addressed row (one nibble is equal to four bits, or half a byte). The first data bit is designated by the applied column address, and the three others immediately follow this address. Internally, a DRAM chip with nibble mode has a 4-bit data buffer in most cases, which accommodates the 4 bits and shifts them, clocked by the CAS signal, successively to the output buffer. This is carried out very quickly because all four addressed data bits (one explicitly and three implicitly) are transferred into the intermediate buffer all at once. The three successive bits need only be shifted, not read again. DRAM chips with nibble mode are rarely used in PCs.

Serial mode Serial mode may be regarded as an extended nibble mode. Also in this case, the data bits within one row are clocked out by switching the CAS signal. Unlike in nibble mode, the number of CAS switches (and thus the number of data bits) is not limited to four. Instead, in principle, a whole row can be read serially. Thus, the internal organization of the chip plays an important role here, because one row may comprise, for example, 1024 or 2048 columns in a 1Mbit chip. The row and column address supplied characterize only the beginning of the access. With every switching of CAS the DRAM chip increments the column address internally and automatically. The serial mode is mainly an advantage for reading video memories or filling a cache line, as the read accesses by the CRT or the cache controller are of a serial nature over large address areas.

Interleaving Another way to avoid delays because of the RAS precharge time is memory interleaving. For this purpose, memory is divided into several banks interleaved at a particular ratio. This is explained in connection with a 2-way interleaved memory for an i386 CPU. For example, because of the 32-bit i386 address bus, the memory is also organized with a width of 32 bits. With 2-way interleaving, memory is divided into two banks that are each 32 bits wide. All data with even double-word addresses is located in bank 0 and all data with odd double word addresses in bank 1. For a sequential access to memory executed, for example, by the i386 prefetcher, the two banks are therefore accessed alternately. This means that the RAS precharge time of one bank overlaps the access time of the other bank. Stated differently: bank 0 is precharged while the CPU accesses bank 1, and vice versa. As only the access time and not the cycle time is significant for the CPU access rate, here the access rate can be doubled. Thus the effective access time for several successive memory accesses is halved.

3-way and 4-way interleaving is carried out according to the same principle, but memory is divided into three or four banks respectively, in those cases, and the temporal RAS and CAS shifts are only one third or one fourth of the time compared with half of the normal cycle time. Many NEAT boards allow custom setup of the interleaving factor. If your memory chips have four banks in total, you may choose either 2-way or 4-way interleaving. In the 4-way case, two banks are always combined into one group, and in the 2-way case, each bank is accessed individually.

So far the concepts of page mode and interleaving have been described in connection with a read access. Of course the same principles apply for writing data. Moreover, read and write accesses can

be mixed: there is no need to leave page mode, nor is interleaving without any value. To benefit from the advantages of both interleaving and page mode, many storage chips are now configured as paged/interleaved memory.

The CAS1 signal is phase-shifted by 180 degrees compared with CAS0. Thus, bank 0 accepts column addresses, decodes them and supplies data, while for bank 1 the CAS1 strobe signal is disabled to change the column address, and vice versa. The access rate is thus further increased compared with conventional interleaving or page mode. With conventional interleaving the DRAMs are interleaved according to the width of memory word by word or double word by double word. In page/interleaving this is done page by page. If a RAS precharge cycle is required during a page change, the access to the other bank is carried out with a probability of 50%. Thus, interleaving is effective in a similar way to conventional memory operation.

Unfortunately, page mode and interleaving are not always successful. As mentioned, it is necessary for the memory accesses to be carried out for the same page for page mode to be successful. A page change creates a RAS precharge time, and thus delays the memory access. In the same way, to gain an advantage from interleaving, it is necessary for the accesses with a 32-bit data bus to be carried out alternately for even and odd double-word addresses (or alternately for even and odd word addresses in the case of a 16-bit data bus). If the CPU twice accesses an odd or even double-word or word address, this also makes the disadvantageous - and noticeable - RAS precharge time necessary. Fortunately, program code and data tend to form blocks. Moreover, prefetching is executed sequentially so that page mode and interleaving significantly increase the memory access rate in most cases, but not always. The hit rate is typically about 80% with page mode/interleaving. A very complex memory controller is required for this: in page mode it must be able to detect whether the other bank needs to be accessed. If this condition is not fulfilled, the memory controller must flexibly insert wait states until the DRAMs have responded and output the required data or accepted the data supplied. Such memory controller is rather complicated but interesting (from a theoretical viewpoint).

3.2 Improved DRAM types

3.2.1 SDRAM

SDRAM stands for synchronous DRAM, which you should not mix up with SRAM, which stands for static RAM, described in chapter 4. SDRAM has a typical access time of merely 8 to 15 ns, and can operate synchronously with the system clock rate. This is typically 66 MHz and is currently a maximum of 133 MHz. EDO RAMs on the other hand usually have an access time of 50 to 60 ns [9]. In practice, this difference is not particularly significant. One reason is the L2 cache memory which boosts the performance of the slower EDO chips to some extent. You only really notice the faster operation of the SDRAMs, compared with EDO DRAMs, when the system clock rate is faster than 100 MHz, as on the majority of the nowadays systems.

For SDRAMs, 168-pin sockets (DIMMs, described in section 3.4) are used as memory modules. They have a data width of 64 bits. SDRAMs work in burst mode and with a synchronous clock rate (for the motherboard and system) and not with different CAS and RAS timing like other RAM chips. SDRAMs do use the corresponding RAS, CAS, WE, and CE signals, but they do so in order to transfer commands such write, read, or burst stop. The RAS and CAS signals are combined to form a command bus, as you can see in the timing diagram (Figure 7).

SDRAM modules are specified for 66 MHz up to typically 133 MHz. There are two types of SDRAM, one of which has an additional serial EEPROM, and the other does not. The chipset's system management bus controls the EEPROM, which contains data about the module type, the organization

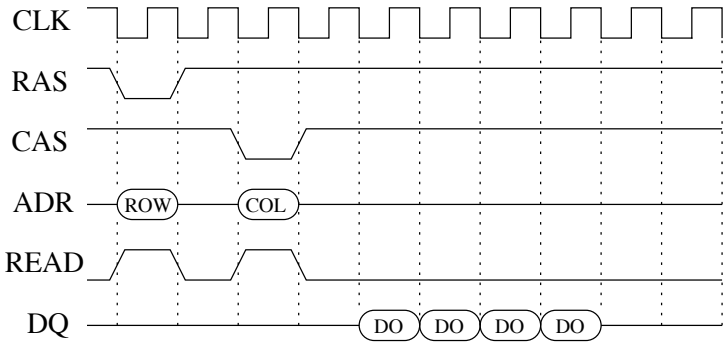

Figure 7: SDRAM read time scheme [12, page 27]

of the DRAMs in use, and the particular timing behavior. This consequently ensures that the best possible settings are used for SDRAMs, automatically. The presence detect signals (PD) are used to recognize the SIMMs, so the EEPROM is known as an SPD-EEPROM (serial presence detect).

If the SDRAM module does not incorporate an EEPROM, the optimum values need to be set manually in the BIOS setup. In PCs both types of SDRAM are used, which means that not every module works in every apparently suitable motherboard.

SDRAM uses a principle similar to interleaved memory fields in a way that while it works with one of them (it is being read), then the next one is getting ready to be accessed.

3.2.2 DDR SDRAM

The data transfer rate can be doubled if the data is transferred not only on the rising CLK clock pulse edge but also on the falling clock pulse edge. Exactly this principle is used by double data rate DRAMs (DDR-RAMs). This is consequently a new, but backwards-compatible, type of memory (unlike, for example, the RAMBus), which has led to the PC-266 modules. The Athlon can especially profit from it as it uses the DDR protocol as standard. A read cycle time scheme is shown on Figure 8.

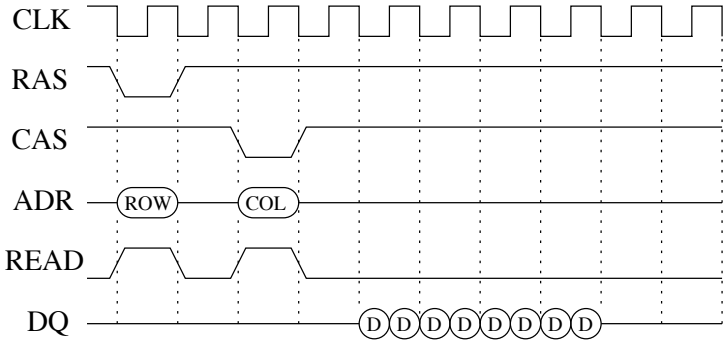

Figure 8: DDR SDRAM read time scheme

3.2.3 RAMBUS DRAM - RDRAM

Intel's preferred PC memory technology is RAMBus, which was used for the first time on motherboards with the Camino chipset (I820). Intel's competitors prefer DDR-RAM. The noteworthy thing about the RAMBus is that, as the name bus implies, the entire memory architecture is a bus system. On one side, the controller is located, the memory chips (RDRAM) are in the middle, and there is a terminator on the other end. Direct RAMBus can use a maximum of 32 RDRAM chips, i.e. any restriction to memory capacity is not caused by the number of sockets present but by the total number of memory chips on the modules present: there must not be more than 32. Furthermore, no RIMM sockets can be left empty, otherwise the bus would be incomplete and nothing would work. A solution is offered by CRIMM sockets (continuity RIMM) that contain no electronics and are simply used to bridge the signals.

On the 16-bit-wide data bus (DQ) the utilization data is transferred and a block-oriented protocol is executed. The necessary clock rate is generated by a square wave generator and transferred differentially to the CTM (clock to master) and CFM (clock from master) lines. During this, the clock signal practically runs from the generator (localized by the terminator) to the controller (MCH, memory controller hub) and back again. The consequence is that each of the memory chips, which basically use normal DRAM technology, can select the clock rate which is running in the right direction for it. If big data blocks are present, this leads to a "slide". The RQ signals are used for memory cell addressing and the Sxx signals are used for communication with the implemented control registers.

However, RAMBus has been unable to prove itself much faster than the PC-133 in practice, and test have shown that PC-266 memory (DDR-RAM) is ever faster than it. RAMBus memory was also very expensive. Equally it does not seem that RAMBus is very suitable for use as the standard memory technology in normal PCs. Instead it will be probably used in special workstations and servers. Even Intel has in the meantime declared its support for DDR-RAM.

3.2.4 CDRAM

Cache DRAM (CDRAM) is a development that has a localized, on chip cache with a wide internal bus composed of two sets of static data transfer buffers between cache and DRAM. This architecture achieves concurrent operation of DRAM and SRAM synchronized with an external clock. Separate control and address input terminals of the two portions enable independent control of the DRAM and SRAM, thus the system achieves continuous and concurrent operation of DRAM and SRAM. CDRAM can handle CPU, direct memory access (DMA) and video refresh at the same time, by utilizing half-time multiplexed interleaving through a high-speed video interface. The system transfers data from DRAM to SRAM during the CRT blanking period. Graphic memory, as well as main memory and cache memory, are unified in the CDRAM. As you can see, CDRAM can replace cache and main memory, and it is has already been proven that a CDRAM based system has a 10 to 50 percent performance advantage over a 256kbyte cache based system.

3.2.5 IDRAM

Intelligent DRAM, or IDRAM, merges processor and memory into a single chip in order to lower memory latency and increase bandwidth. It is a research model for the next generation of DRAM and has been tested in alpha 21164 processors. The reasoning behind placing a processor in DRAM rather than increasing the on-processor SRAM is that the DRAM is approximately 25 to 50 times denser than cache memory in a microprocessor. Merging a microprocessor and DRAM on the same chip provides some rather obvious opportunities in performance, energy efficiency, and cost. It affords a

reduction in latency by a factor of 5 to 10, an increase in bandwidth by a factor of 50 to 100, and has an advantage in energy efficiency at a factor of 2 to 4.

3.2.6 SLDRAM

Synchronous-Link DRAM. SLDRAM offers high sustainable bandwidth, low latency, low power consumption, is easily upgraded, and supports large hierarchical memory configurations. For video, graphics, and telecommunications applications, SLDRAM provides multiple independent banks, a fast read/write bus turn around, and the capability for small, fully pipelined burst. SLDRAM addresses the requirements of all major high volume DRAM applications. SLDRAM is an open standard to be formalized by IEEE and JEDEC specifications. Open standards permit manufacturers to develop varying products that address emerging applications and niche opportunities while inspiring competition that will ensure the continued rapid pace of development of DRAM technology, at the lowest possible cost.

SLDRAM technology is improved from SDRAM in a way of supporting higher bus speeds and using packets for supplying address requests, timing and commands for DRAM. As a result, there is a smaller dependency in improvements of DRAM chip designs and ideally cheaper solution for high performance memory.

A typical SLDRAM architecture uses a multi-drop bus that has one memory controller and up to eight loads. A load can be either a single SLDRAM device or a buffered module with many SLDRAM devices. Command, address and control information are on the unidirectional command link. The data link is a bi-directional bus for the transmission of write data from the controller to the SLDRAM, and read data from the SLDRAM back to the controller. Two sets of clocks allow control of the data link to pass from one device to the next with a minimum gap. Later versions of SLDRAM add a buffer on the command link and data link to provide higher memory bandwidth and larger memory depth.

3.3 Refreshing the DRAM

We already know that the data is stored in the form of electrical charges in a tiny capacitor. As is true for all technical equipment, this capacitor is not perfect, that is, it discharges over the course of time via the access transistor and its dielectric layer. Thus the stored charges and therefore also the data held get lost. The capacitor must be recharged periodically. Remember that during the course of a memory read, the memory cells in the addressed row are automatically refreshed because the read procedure is destructive. Normal DRAMs must be refreshed every 1 ms to 16 ms, depending upon the memory type. Currently, three refresh methods are used: RAS-only refresh, CAS before RAS refresh, and hidden refresh.

RAS-only refresh The simplest and most widely used method for refreshing a memory cell is to carry out a dummy read cycle. For this cycle the RAS signal is activated and a row address (the refresh address) is applied to the DRAM, but the CAS signal remains inactive. The DRAM thus internally reads one row onto the bit line pairs and amplifies the read data. However, because of the disabled CAS signal this data is not transferred to the I/O line pair and thus not to the data output buffer. To refresh the whole memory an external logic or the processor itself must supply all the DRAM row addresses in succession. This refresh type is called RAS-only refresh. The disadvantage of this outdated refresh method is that an external logic, or at least a program, is required to carry out the DRAM refresh. In the PC this is carried out by channel 0 of the 8237 DMA chip, which is periodically activated by counter 1 of the 8253/8254 timer chip and issues a dummy read cycle. In an RAS-only

refresh, several refresh cycles can be executed successively if the CPU or refresh control triggers the DRAM chip accordingly.

CAS-before-RAS refresh Most modern DRAM chips also have one or more internal refresh modes. The most important is the CAS-before-RAS refresh. For this purpose, the DRAM chip has its own refresh logic with an address counter. For a CAS-before-RAS refresh, CAS is held at low level for a certain time period before RAS also drops (thus CAS-before-RAS). The on-chip refresh (that is, the internal refresh logic) is thus activated, and the refresh logic carries out an automatic internal refresh. The refresh address is generated internally by the address counter and the refresh logic and need not be supplied externally. After every CAS-before-RAS refresh cycle, the internal address counter is incremented so that it indicates the new address to refresh. Thus it is sufficient if the memory controller "taps" the DRAM from time to time to issue a refresh cycle. With the CAS-before-RAS refresh, several refresh cycles can also be executed in succession.

Hidden refresh A more elegant option is the hidden refresh. Here the actual refresh cycle is "hidden" behind a normal read access. During a hidden refresh the CAS signal is further held on a low level, and only the RAS signal is switched. The data read during the read cycle remains valid even while the refresh cycle is in progress. Because the time required for a refresh cycle is usually shorter than a ready cycle, this refresh type saves time. For the hidden refresh, too, the address logic in the DRAM generates the refresh address. If the CAS signal remains low for a sufficiently long time, several refresh cycles can be carried out in succession. For this it is necessary only to switch the RAS signal frequently between low and high.

New motherboards implement the option of refreshing the DRAM memory with CAS-before-RAS or hidden refresh instead of the detour via the DMA chip and timer chip. This is usually faster and more effective. You should use this option, which comes directly from the field of mainframes and workstations, to free your PC from unnecessary and time-consuming DMA cycles.

3.4 Memory modules

Memory chips are called DIPs which stands for Dual Inline Packages. They are integrated circuits with pins on both sides. To make memory installation easier than it was in the past, these DIP chips were places on modules.

Today, more compact memory modules such SIMM, PS/2, and DIMM are often used instead of single chips. SIMM and SIP modules have a standard width of 9 bits. PS/2 modules have a data width of 36 bits (modules with parity), 32 bits (non-parity modules), or 40 bits (ECC modules). The appropriate number of memory chips is installed on the modules to reach memory capacity. The modules must be inserted into the sockets provided for them on the motherboard. A bank must always be completely filled with memory modules. From the Pentium CPUs onwards, a bank corresponds to two PS/2 sockets or one DIMM socket, because the DIMMs always have a data width of 64 bits.

SIMM and PS/2 modules have a contact strip similar to the adapter cards for the bus slots, and SIP modules are equipped with pins that must be inserted into the appropriate holes. Normally, only PS/2 and DIMM modules are used today. SIMM modules are old-fashioned, but there is no reason why you should not use them. You just need piggy-back boards on which you can usually install four SIMMs (with a data width of 8 bits plus one parity bit). The system then treats them like any PS/2 module with a data width of 32 or 36 bits.

The following text briefly discusses the terminals of the PS/2 module with and without parity. Similarly-named contacts on the SIMM and SIP modules have the same functions.

SIMM modules Single Inline Memory Module. They may have DIPs on one or both sides and will have 30 or 72 pins. They are normally available in the 72 pin size which supports a 32 bit data transfer between the processor and the memory.

DIMM modules Double Inline Memory Module. Pentium processors have sockets for double inline memory modules, or DIMMs for shorter. Although it is usually most possible to use both SIMMs and DIMMs on the same, suitable, motherboard, provided you don't mix up different kinds of memory module, on the same bank, and always completely fill each bank with the same kind of memory module, it is usually not possible to simultaneously use all the PS/2 SIMM and DIMM sockets present.

The 168-pin DIMMs always have a width of 64 bits, where each DIMM always corresponds to a bank each time so that, for basic Pentium PCs only one DIMM is necessary. Primarily SDRAMs are used on DIMMs. They use the voltage of 3.3 V [9]. There are also DIMMs that do not use SDRAMs but EDO RAMs, which require 5 V. If SRAMs are supplied a 5 V, even once, they are broken, by EDO DIMMs supplied with 3.3 V do not work at all, or work incorrectly. On some motherboards there is jumper that you can use to set the appropriate voltage.

RIMM modules RAMBus modules are known as RIMMs (Rambus Inline Memory Module). They have 184 contacts and are available in capacities of 64, 128, and 256 Mb. According to their definition they operate with a maximum clock rate of 400 MHz, which frequently leads to the value "800 MHz", although this fails to mention that, just as in the case of DDR-RAM, this requires data transfer on both clock pulse edges. Other standard clock rates are 350 and 300 MHz. Intel is using different implementation, Direct RAMBus, with a 16-bit-wide data bus. There have already been two other versions, Base and Concurrent, which had a maximum bus width of 9 bits and were used, for example, in the Nintendo 64. Apart from that RAMBus was available on some graphics cards years ago.

To install memory modules, you press them into the socket on the motherboard and lock them in with a plastic latch on both sides. Normally as the memory module is pressed into place the lock will automatically latch the module in place.

4 Static memories

In static memories the information is held as the state of a "flip-flop" circuit. Such a flip-flop has two stable states that can be alternated by a strong external signal. Figure 10 shows the structure of a memory cell in an SRAM.

You can see that the SRAM cell structure is far more complicated than of the DRAM memory cell, which is illustrated on Figure 2. The DRAM cell consists only of an access transistor T and a capacitor holding the charges according to the stored data. A typical SRAM cell is composed by two NMOS access transistors T1 and T2 and a flip-flop with two NMOS memory transistors T3 and T4 as well as two other elements, either PMOS transistors or resistors. Thus the integration of the SRAM memory cell is only possible with greater technical effort. SRAM chips are therefore more expensive and can generally store less data than DRAM chips because of the smaller memory cell density per unit area. The integration density of DRAM chips is about four times larger than that of SRAM chips using the same technology. For this reason, SRAM chips are primarily used for small, fast cache memories (described in section 6.3), while DRAM chips are used for the bigger and relatively slower main memory (RAM).

In an SRAM the unit memory cells are arranged in a matrix of rows and columns, see Figure 9, which are selected by a row and column decoder, respectively. The gates of the access transistors T1 and T2 are connected to the word line W and the sources are connected to the bit line pair DATA, $\overline{\text{DATA}}$ (Figure 10).

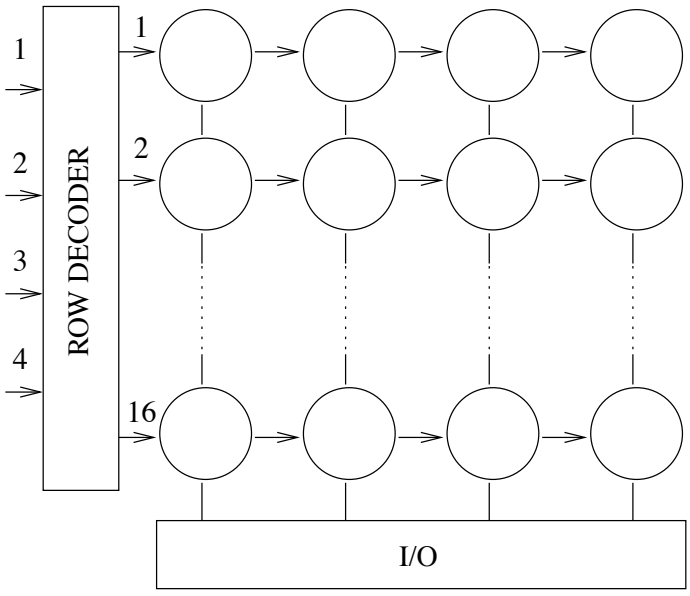

Figure 9: SRAM memory organization [7]

In the memory controller for SRAM chips, row and column addresses are supplied simultaneously. Because of the missing address multiplexing, more pins are required and the SRAM packages are larger than comparable DRAM chips. Furthermore, SRAM chips don't use any high-speed operating modes (for example, page mode or static-column mode). Internal addressing of the memory cells is thus easier. Because of the static design memory, a refresh is not required. The state of the memory flip-flops is kept as long as the SRAM chip is supplied with power. The SRAMs are faster than

DRAMs thanks to lack of address multiplexing and no need for refreshing the cells.

Figure 9 shows the simplified SRAM memory organization, where each circle represents one memory cell, horizontal lines are word lines and vertical lines are bit lines. All the elements will be discussed in the next section of this text.

This chapter is divided into two sections. The first section deals with static memories, it explains how the memories work and shows an example of reading and writing data. The second section shortly describes asynchronous, synchronous and pipeline burst SRAMs.

4.1 SRAM - Static Random Access Memory

4.1.1 The operation principle of the flip-flop

Now we turn to the flip-flop to get an understanding of how an SRAM memory cell functions. Figure 10 shows the structure of a flip-flop. The simple flip-flop shown consists of two feedback-coupled NMOS transistors, T3 and T4, and also two load elements R1 and R2. Feedback means that the source of T3 is connected to the gate of T4 and vice versa. At outputs DATA and $\overline{\text{DATA}}$ two stable levels will then occur. If T3 is turned on then in the left branch of the flip-flop the overall voltage drops at resistor R1 and the output DATA is grounded (low). The gate of transistor T4 is therefore also supplied with a low-level voltage. T4 is then turned off, and in the left branch the entire voltage drops at transistor T4. Thus output $\overline{\text{DATA}}$ is on high Vcc.

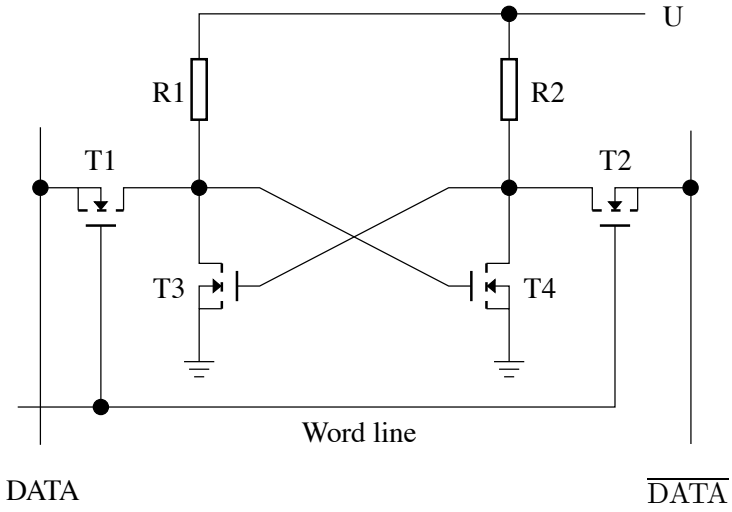

Figure 10: SRAM memory cell with 4 transistors [8]

If, on the other hand, T3 is turned off, the entire voltage in the left flip-flop branch drops at transistor T3 and output DATA is equal to Vcc (high). Therefore, a high voltage is applied to the gate of transistor T4, thus T4 is turned on and in the right branch the entire voltage drops at resistor R2. Output $\overline{\text{DATA}}$ is therefore grounded (low).

In addition to that, the outputs DATA and $\overline{\text{DATA}}$ can also be used as inputs to set up the flip-flop state, that is, switching the state of transistors T3 and T4 on and off. Setting the state is equal to the storing of a bit, because the flip-flop stable supplies, for example, a high or low signal at output $\overline{\text{DATA}}$.

In the text hereafter, an example is used to explain how the flip-flop state is programmed. If

transistor T3 is switched on then output DATA supplies a low-level signal, transistor T4 is turned off, and output $\overline{\text{DATA}}$ supplies a high-level signal. Every transistor has a certain resistance value even in the on state, that is, the so-called on-state resistance. The flip-flop's load elements R1 and R2 have a much higher resistance value than the on-state resistance of transistors T3 and T4. Thus, despite the on-state resistance of T3 and the accompanying voltage drop, the voltage at output DATA is small enough to represent a low level and, on the other hand, a voltage is applied to the gate of transistor T4 which turns off T4. If the value of R1 is, for example, nine times larger than the on-state resistance of T3, than 90% of the voltage Vcc drops at R1 and only 10% at T3 [9]. This is sufficient to hold output DATA stable at a low level and to keep T4 turned off.

To switch the state, the connection DATA (which is both an output and an input) must be supplied with a signal that is so strong that the transistor turned on is unable to lead this signal to ground completely because of its on-state resistance. Thus a signal is applied to the gate of transistor T4 which gives rise to a slight on-state of T4. Therefore the voltage at DATA slightly decreases because of the lower voltage drop at T4. This voltage, which is lower than previously, is simultaneously applied to the gate of T3 so that its conductivity is somewhat reduced and the voltage drop at T3 increases. By means of the feedback to the gate of T4, transistor T4 is further turned on and the process works itself up. During the course of this process, transistor T3 turns off and transistor T4 switches through increasingly so that the flip-flop finally "flips"(or flops); thus the name flip-flop. In other words, a bit has been loaded or programmed.

For the flip-flop's stability the ratio of the resistance values of the load elements R1 and R2 to the on-state resistances of the transistors T3 and T4 is decisive. The higher the load resistances compared with the on-state resistances, the more stable the stored states are. But it is also more difficult, then, to switch the flip-flop states. The flip-flop responds inertly to the programming signal supplied. If the resistance ratio is small then the flip-flop stability is lower. Yet, the switching can be carried out in easier and therefore quicker way. The designer of a flip-flop always treads a thin line between stability and speed of operation.

If connection DATA is supplied with a signal of the same level as it has just output, the new signal has no influence on the flip-flop state. If you write the same value that is already there into a memory cell, then there is, of course, no consequence for the stored value influence. You can also program a flip-flop by applying a signal to the complementary connection $\overline{\text{DATA}}$, which is complementary to the bit that is to be programmed. Thus flip-flops are well suited as memory elements, and they are widely used, for example, in latch circuits, shift registers and other digital technology components.

In the simple flip-flop described above one bit is always stored when a connection DATA or $\overline{\text{DATA}}$ is supplied with an external signal. For the clocked components in computer this is not very favorable, because at certain times an unpredictable and invalid signal may occur on the signal lines. Therefore, clocked flip-flops are mainly used in computers. They accept the applied bit signals only if the clock signal is also valid. Such flip-flops have one or more additional access transistors controlled by the clock signal, and which transmit the applied write signal only upon an active clock signal for a store operation by the flip-flop.

Unlike the storage capacitors in DRAM memory cells, the flip-flop cells supply a much stronger data signal, as transistors T3 and T4 are already present in the memory cell: they amplify the signal and are thus able to drive the bit lines. In a DRAM cell, however, only a tiny charge of a capacitor is transferred onto a bit line, without any amplification, so the signal is very weak. Accordingly, more time is needed for the sense amplifiers in a DRAM to amplify the signal, and the access time is longer. For addressing memory flip-flops in an SRAM, additional access transistors for the individual flip-flop cells, address decoders, etc. are required, as is the case in a DRAM.

There are also SRAM memories that consist of 6 transistors. Here the PMOS type transistors T5 and T6 function only as resistors. Figure 11 shows this type of SRAM memory.

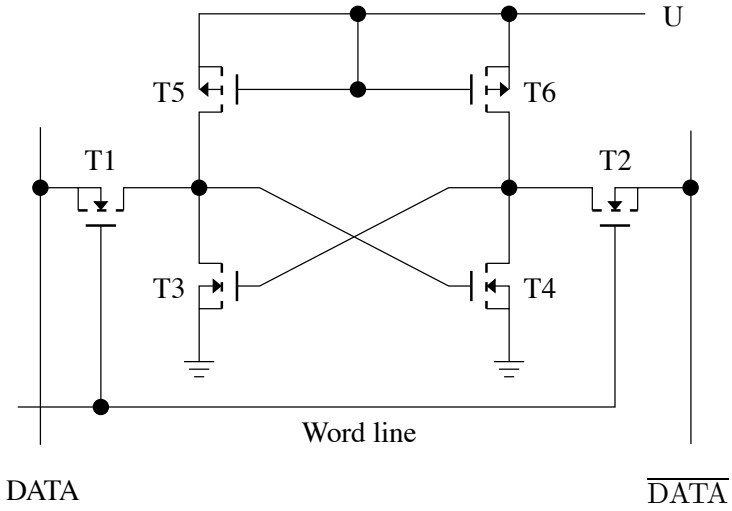

Figure 11: SRAM memory cell with 6 transistors [8]

SRAM memories could be made using the TTL technology. A cell of such a memory operates on the similar principle like a cell of memory made using the MOS technology, however, it only consists of 2 PNP transistors and 2 resistors [8].

4.1.2 Reading and writing data

If data is to be read from SRAM memory cell, the row decoder activates the corresponding word line W. The two access transistors turn on T1 and T2 and connect the memory flip-flop with the bit line pair DATA, $\overline{\text{DATA}}$. The signals are transmitted to the sense amplifier at the end of the bit line pair. Unlike in the DRAM, the T3 and T4 memory transistors in the flip-flop provide a very strong signal as they are themselves amplifying elements. The sense amplifier amplifies the potential difference on the bit line pair DATA, $\overline{\text{DATA}}$. Because of the large potential difference, this amplifying process is carried out much faster than in a DRAM so that SRAM chip needs the column address much earlier if the access time is not to be degraded. SRAM chips therefore don't carry out multiplexing of row and column addresses. Instead, the row and column address signals are provided simultaneously. The SRAM address decoder divides the address into a row and column part. After stabilization of the data, the column decoder selects the corresponding column (that is, the corresponding bit line pair DATA, $\overline{\text{DATA}}$) and outputs a data signal to the data output buffer, and thus to the external circuitry.

The data write proceeds in the opposite way. Via the data input buffer and the column decoder, the data to be written is applied to the corresponding sense amplifier. At the same time, the row decoder activates a word line and turns on the access transistor T1. As in the course of data reading, the flip-flop tries to output the stored data onto the bit line pair DATA, $\overline{\text{DATA}}$. However, the sense amplifier is stronger than the storage transistor T3, and supplies the bit line DATA, $\overline{\text{DATA}}$ with a signal that corresponds to the write data. Therefore, this flip-flop switches according to the new write data, or keeps the already-stored value depending upon whether the write data coincides with the stored data or not.

Unlike the DRAM, no RAS/CAS recovery times are necessary. The indicated access time is

usually equal to the SRAM's cycle time. Advanced DRAM memory concepts such as page mode, static-column mode or interleaving have no advantages for SRAMs because of the lack of address multiplexing and RAS recovery times. SRAM chips always run "normal mode", in which both row and column address are supplied.

4.2 Async, Sync and PB SRAM

Asynchronous SRAM This type of memory exists from the time of 386 processors and it could be found in L2 cache memories. It is called asynchronous because it is not synchronized with system clock, and thus CPU must wait for the required data from the L2 cache memory.

Synchronous Burst SRAM Similarly to SDRAM, this memory is synchronized with system clock that is why it is easier to access than asynchronous SRAM. Synchronous SRAM lacks synchronization with buses running with frequencies higher than 66 MHz [1].

Pipeline Burst SRAM Using the burst technology the requests can be pipelined or collected in such way that requests in one burst could be executed almost immediately. PB SRAM uses pipelining and though it falls behind the system synchronization frequencies, it represents improvement from synchronized SRAM because it is designed to work with buses with frequencies of 75 MHz and higher.

5 Memories with permanent content

The disadvantage of the memory modules described earlier is the volatility of the data stored in them. DRAM and SRAM chips are unsuitable for PC boot process startup routines because when the power is turned off their content is forgotten. ROM chips are used instead. The storage data is written once into the ROM in a non-volatile way so that it is held even if the power is turned off for a long time.

All basic passive and active electronic elements has varied as a function of ROM memory elements in the history. The resistors, inductors, ferrite cores, capacitors, diodes, unipolar and bipolar transistors were used [1].

The main function of these memories is to remember the data at the time, when the computer is turned off. For this reason, it is used, for example, for storing BIOS. The operating system controls hardware through BIOS, but since the ROM memories are quite slow, it slows down the computer's performance. For this reason, after starting the computer it is possible to read and store BIOS into faster RAM memory. The operating system then works with BIOS in RAM memory. This storing of BIOS into RAM from ROM is called *shadowing*. ROM memories can be easily recognized because they are mostly bigger chips than SRAMs and DRAMs.

This chapter is divided into five sections, ROM, PROM, EPROM, EEPROM, flash memories, and talks about what they are used for and how they operate.

5.1 ROM

In ROM chips the information is not held in a form of charges, as in DRAMs, or as an alterable circuit state in electronic elements, as in SRAMs, but generally as a fixed wiring state of the elements. Often, switching elements are connected between a word and a bit line, and their switching state (on, off) is fixed.

"Pure" ROM is very rare today. In these chips the data to be stored has already been taken into account in advance of manufacturing. The data is integrated into the circuit design as either present or absent connections between nodes within the chip's circuitry. Connected elements pass current, there is a minimum voltage, that means it carries logic 0. Disconnected elements do not pass current, there is a maximum voltage, it carries logic 1. A very compact circuit design can thus be achieved, as the circuit can be optimized according to the information to be stored. This can be carried out only at the expense of flexibility, because the change of only a single bit requires an alteration of the complete circuit.

The chip is called a mask ROM because programming in these ROM chips is carried out by means of a lithographic mask. Also, all other layers for creating gates, and the source and drain regions of transistors, conductive layers for word and bit lines, etc. are manufactured using such masks. The last process masks then contain the connection information according to the data to be stored. Thus the manufacturing steps up until the last masks are the same for all ROMs, and independent of the information to be stored. Different data only need to be handled by different masks.

The mask-programmable ROMs are advantageous if a large number of identical ROM chips with the same information are to be manufactured. The main effort is in the design and manufacture of the programming mask; the production of the last layer of the ROM chips is easy. The main disadvantage of mask-programmed ROMs is that the programming can be carried out only at the manufacturer's site. The time of the storage data is practically unlimited.

5.2 PROM

In programmable ROM (PROM for short) chips the information needs to be "burned in" by the user. The information is burned in by the user using a programmer. The data is written using electrical pulse, but, unlikely in DRAMs and SRAMs, this data is stored in a non-volatile fashion. One method for achieving non-volatility is to burn a fuse link between the word and bit line. During programming a stronger current pulse is supplied than in the normal operating mode. The generation and supply of such strong pulses can also be carried out by the user with a suitable programming device. The chip has already been mounted in a package and therefore no further manufacturing steps are required. The storage content cannot be altered afterwards, that is, such PROM's are one time programmable only and, thus, called OTPs (on-time programmable memories). This line is made of nickel and chrome or silicon.

Another possible realization of PROM memories is using of bipolar multi-emitor transistors. This type of PROM consists of one multi-emitter transistor for every word line. When the memory is to be read then logic 1 is applied on the certain word line, multi-emitter transistor opens, in the drain-source direction the current will pass. If the line was not burned then the current will open transistor that is connected as an invertor and a logic 0 is read on the output. If the line was burned then transistor will not open a logic 1 is read on the output.

5.3 EPROM

Another way to achieve non-volatile storage of data is to use a storage transistor with a so-called floating gate. This gate is located between the actual control gate and the substrate, and (unlike the control gate) is not connected to word, or other lines; it "floats" (that is, its potential has no defined value). In most cases, the control and floating gate consist of the same material, for example polycrystalline silicon, which is a reasonably good conductor.

In memory cells with a storage transistor, the gate is usually connected to a word line, the drain to a bit line, and the source to a reference current (such as Vcc). In the ROM chips described earlier, the connection between the word and bit line was realized by a simple conductor bridge, but now the storage transistor takes over this job. With the word line activated, a normal MOS transistor without a floating gate would always connect the bit line via the drain, channel and source to Vcc. Storage of any values is not yet carried out by this. Only the state of the floating gate leaves permanent information in the memory cell. If the floating gate is neutral (that is, no charges are stored), then it has no influence on the electrical field that the control gate generates in the channel region between the source and drain. The storage transistor operates as a normal MOS transistor, and applies the reference potential to the bit line as soon as the word line is activated.

The situation changes completely, though, if the floating gate holds electrons, i.e. it has a negative charge. The electric charges in the floating gate shield the field of the control gate and generate an electrical field in the channel region which is opposite to the field of an active control gate (the control gate is supplied with a positive voltage). Thus an activated word line cannot generate a sufficiently strong field with the control gate to turn on the transistor. The bit line is not supplied with the reference potential, and the storage transistor stores the value "0". Only with a much higher control gate potential can the storage transistor be turned on, that is, if the field of the control gate is strong enough to compensate the filed of the floating gate, and to make the channel between source and drain conductive. Thus the electric charge in the floating gate shifts the storage transistor's characteristics, and therefore also the threshold voltage, to higher values.

Loading the floating gate with electrons can be carried out by an electrical pulse. As the floating

gate is completely embedded in an insulating layer, and has no connection to other elements, the charges are held there for a long time (at least ten years). This long lifetime (and the well-insulated connection to it) is, of course, an obstacle for charging the floating gate. To program a "0" a pulse lasting 50 ms with a voltage of 5 V (some types require 20 V) is applied between the word and bit line, that is, between gate and drain [9]. Thus, in the channel region fast (hot) charges carriers (electrons) are generated that have enough energy to pass the insulation region between the substrate and floating gate. They accumulate in the floating gate and are held there after the programming pulse has been switched off, as the floating gate is insulated and the electrons do not have enough energy after slowing down to bridge the insulation layer again. Thus, the insulation layer is impermeable only to low-energy cold electrons. High-energy electrons can penetrate the insulation layer, but without destroying it. This type of storage transistor is also called FAMOST (floating gate avalanche MOS transistor).

The programming time of 50 ms is very long compared with the 7 ns of modern DRAMs or even to the 20 ns of SRAM chips. In addition, it can only be performed off circuit using special equipment. However, with a shorter programming time not enough electrons would be collected in the floating gate to achieve the intended effect. Remedial action might be taken by means of a higher voltage, but then there would be the danger of damaging the insulation layer and destroying the chip. For the timing and strength of the write pulses a PROM programmer is available. This is an equipment into which you can insert the PROM chips and which carries out the PROM chip programming according to the write data.

One may expect that the floating gate can be discharged and thus the data can be erased by reversing the polarity of the programming pulse. But this is not true, as in this case hot electrons are also generated in the substrate and not in the floating gate. To erase the data the ROM chip must be exposed to UV radiation. The electrons in the floating gate absorb the energy from the UV rays. They get "hot" and can leave the floating gate in the same way as they previously got in. If the chip package is equipped with a UV-permeable quartz window then we get an EPROM (erasable PROM). You may have already seen such chips.

Through the quartz window you have a clear view of the actual chip and the bonding wires. Otherwise, all these mysterious shining silicon chips are usually hidden in unexciting black or brown packages. After being irradiated for about 20 minutes, all charges carriers are removed from the floating gate and the EPROM chip can be programmed again.

5.4 EEPROM

A quartz glass window and a UV lamp for clearing data are complicated (and also expensive) equipment for erasing EPROM chips. It would be better and much easier if the chip could be cleared in the same way as it was programmed, by means of an electrical pulse. This is the case for an EEPROM (electrical erasable PROM)

Loading the floating gate with electrons (that is, the programming of the memory cell) is carried out in the same way as in an EPROM: by applying a relatively long 5 V (or 20 V) current impulse (duration 50 ms) between the gate and drain, high-energy charges carriers are generated in the substrate. These penetrate the gate oxide and collect in the floating gate. The positive potential of 5 V during programming "draws up" the negative electrons from the substrate into the floating gate. To clear the EEPROM, the thin tunnel oxide film between a region of the floating gate downwards in direction towards the substrate and the drain is essential. Due to the characteristics of the transistor, insulation layers never insulate perfectly. Instead, charge carriers can penetrate the insulation layer (but very infrequently). They are more likely to do so, the less the thickness of the insulation layer

and the higher the voltage between the two electrodes on the two sides of the layer.

For discharging the floating gate an inverse voltage is applied between gate and drain, that is, the drain is on a potential of 5 V against the gate. Thus the negative electrons in the floating gate are drawn to the drain through the thin tunnel oxide, and the stored data is thus erased. It is important to ensure that this charge drawing does not last too long, as otherwise too many electrons are drawn out of the floating gate and the gate is then charged positively. The transistor's characteristic is thus shifted too far to the left, and the threshold voltage is dropped too far, so that normal operation of the storage transistor becomes impossible.

5.5 Flash memory

In the past few years a new type of non-volatile memory has come onto the market as a substitute for floppy and hard disk drives, frequently used in small laptops, digital cameras, and mobile phones: so-called flash memory. This is used, for example, in memory cards. Compared with battery-buffered SRAM solutions, the inherent non-volatility of flash memory is a big advantage: there is no battery to fail and cause a data loss. The main advantage of flash memories is that they are in-circuit programmable. The storage time of information in a flash memory is at least ten years, and typically about 100 years.

Flash memory is mainly used in fields where a power failure would lead to disastrous consequences, or where the operating conditions are very rough. In comparison, hard disk drives are sensitive to shock and floppies to mechanical damage and humidity. Flash memories, sealed in a stable package with no mechanical movable parts, are immune to such external influences.

The structure of their memory cells is fundamentally the same as that of EEPROMs; only the tunnel oxide is thinner than in an EEPROM memory cell. Therefore, they need an erase and programming voltage of only 12 V so that 10 000 program and erase cycles can be carried out without any problems. Despite the memory cell array, several additional control circuits are registers are formed in a flash memory. The program and erase operations are carried out flexibly as in DRAMs or SRAMs, but flash memory does not lose the data it holds.

The central part of the flash memory is the memory cell array. The cells are addressed by an address buffer, which accepts the address signals and transfers them to the row and column decoders, respectively. Flash memories, like SRAM chips, don't carry out address multiplexing. The row and column decoders select one world line and one or more bit line pairs as in a conventional chip. The read data is output via a data input/output buffer or written into the addressed memory cell by this buffer via an I/O gate.

The read procedure of flash memory is similar to the read procedure of SRAM, where row and column addresses are selected and the data is then read. The write procedure is more difficult. It is possible to write "0", but it is not possible to write "1" in normal way. To be able to perform a write procedure of "1" into the flash memory, the whole sector has to be copied into the RAM and erased from the flash memory. In RAM the "1" is written to the row and then the whole row is written back to the flash to its original position. The external processor writes the instruction of the read, program and erase processes to the instruction register of the flash control. For a typical flash memory the following instructions are available:

- Read Memory: the flash memory supplies data via the data pins.
- Read Identifier Code: the flash memory supplies a code at the data pins, which indicates the type and version of the chip.

- Set-up Erase/Erase: prepares the flash memory for an erase process and carries out the erasure.
- Erase-Verify: erases all memory cells and verifies this process.
- Set-up Program/Program: prepares the programming of individual memory cells and carries out this process.
- Program-Verify: executes programming and verifies this process.
- Reset: resets the controller of the flash memory to a defined initial state.

The read process is carried out with the usual MOS voltage of 5 V. The programming and erasing is done by charge pumps. Newer types can also use 5 V as their programming voltage. To program a memory cell the flash control applies a short voltage pulse of typically 10 μs and 12 V [9]. This triggers an avalanche breakdown in the memory transistor which charges the floating gate. In this way a 1-Mbit flash memory chips can be programmed in around two seconds. In contrast to normal EEPROMs, however, deletion is carried out a chip at a time: during a deletion, the flash control uses the deletion current switch to send a delete impulse to the entire memory cell field, so all memory cells are deleted. The erase time for the whole flash memory is about one second.

There are also variants of this memory available that can be erased page by page, that is, one complete row of memory cells is always erased. Moreover, current flash memories generate that programming and erasing voltage internally. Every modern motherboard uses these types of flash memory: they contain the system BIOS.

Flash memories have more extensive functionality than normal EEPROMs, and are virtually autonomous memory subsystem. For example, unlike conventional EEPROMs they can be programmed and erased while installed in the computer. Most of the models even generate the high voltage Vpp from the supply voltage of 5 V internally. The so-called flash BIOS makes use of that. Ordinary ROM chips must be removed from the BIOS update, and replaced by new ones (PROM and EPROM), or they must be reprogrammed with the new BIOS data in an external programmer (EEPROM). For a flash BIOS is erased in system by an erase or erase-verify command. Afterwards, the setup software transfers the new BIOS bytes (data and code) to the flash chip and permanently writes them with a program or program-verify command into the flash memory.

It does not matter whether an EEPROM or a flash memory module is used for the BIOS on the motherboard, as the supplementary logic for EEPROMs is implemented on the motherboard and flash memory does not require supplementary logic. However, what is important, besides support by a suitable writer program, is the programming voltage which is either 12V or 5 V and is dependent on the particular type of module used. Many motherboards support only one particular rewritable memory type for the BIOS and therefore either only 5 V or only 12 V. Still, there are also exceptions: on some motherboards there is a jumper with a label such as Flash ROM voltage selector or similar. You can set this jumper in different ways to select one of the two programming voltages, as required.

6 Other memory types

This chapter consists of three sections, in first one the video memories are described, in second FIFO memories are presented and the third section explains how cache memories are organized.

6.1 Video memories

Early graphics cards simply worked with the system memory, today's cards can have 64 or even more MB of its own memory that is used to store the content of the screen but also textures required for 3D graphics and algorithms. These memories should be fast enough and mostly distinctively contribute to the prices of the cards.

6.1.1 Video RAM (VRAM)

The traditional, standard DRAM used for video cards typically does not have enough bandwidth to handle the demands of running a card at high resolution and color depths, with acceptable refresh rates. The main reason why is the two competing access factors for the video memory: the processor writing new information to the memory, and the RAMDAC (Random Access Memory Digital to Analog Converter) reading it many times per second in order to send video signals to the monitor.

To address this fundamental limitation, a new type of memory was created called video RAM or VRAM. As the name implies, this memory is specifically tailored for use in video systems. The fundamental difference between VRAM and standard DRAM is that VRAM is dual-ported [1]. This means that it has two access paths, and can be written to and read from simultaneously. The advantages of this are of course enormous given what the video card does: many times per second a new screen image is calculated and written to the memory, and many times per second this memory is read and sent to the monitor. Dual-porting allows these operations to occur without interference.

VRAM provides substantially more bandwidth than either standard DRAM or EDO DRAM; double in many cases. It is more suited for use in systems requiring high resolution and color depth displays. VRAM is more complex and requires more silicon per bit than standard DRAM, which makes it cost more.

6.1.2 WRAM

Window RAM or WRAM is a modification of regular VRAM that both improves performance and reduces cost. Designed specifically for use in graphics cards, WRAM is also dual-ported but has about 25% more bandwidth than VRAM (because it supports addressing of large blocks (windows) of video memory). It also incorporates additional features to allow for higher performance memory transfers for commonly used graphical operations such as text drawing and block fills. Furthermore, WRAM is less expensive than VRAM to manufacture (although still more expensive than DRAM).

WRAM is suitable for use in high-end graphics cards and was first made popular by Matrox's Millennium series. A card with WRAM and a sufficiently fast RAMDAC can handle even very high resolutions (1600x1200) at true color.

6.1.3 SGRAM

Synchronous Graphics RAM basically works like SDRAMs, even if the individual signals have different names in the two types of memory. The most important differences are that SDRAMs are

optimized for the highest possible memory capacity and SGRAMs are optimized for the fastest possible data transfer. SGRAM's instructions, such as block write and write per bit, are specific to graphic and video applications.

For some time now, SGRAMs and other types of memory for graphics cards have been manufactures as 72-pin SODIMMs (small outline DIMMs) and many graphics cards provide a suitable socket for them. As a result, something approaching a "standard socket" for graphics memory has been created - until now almost every manufacturer used their own sockets for their own modules. SODIMMs, for example in the form of EDOs, have already been used for a long time in laptops and some workstations.

6.1.4 Multibank DRAM (MDRAM)

A new type of memory that attempts to address two problems with conventional video memory, Multibank DRAM or MDRAM was invented by MoSys specifically for use in graphics cards. MDRAM differs substantially in design from other types of video memory. Conventional memory designs use a single monolithic "block" of memory for the frame buffer. MDRAM breaks its memory up into multiple 32 KB banks that can be accessed independently. This provides the following advantages:

- Interleaving: Memory accesses can be interleaved between banks, allowing accesses to overlap to provide greater performance. This has the effect of increasing performance without the use of dual porting; the concept is similar to the use of interleaving for the system memory on high-end chipsets.
- Flexibility in Memory Sizing: With conventional memory, it is only practical to make video cards with whole megabytes of memory: you see cards with 1 MB, 2 MB, 4 MB etc. of video RAM. This can cause a great deal of memory waste. For example, to run 1024x768 resolution in true color (24 bits) requires 2.25 MB of video memory for the frame buffer, but a conventional video card would have to be outfitted with 4 MB of memory to support this mode. With MDRAM this restriction is removed and a card can be created with exactly 2.25 MB if desired [13].
- No Size-Related Performance Penalties: In some conventional video card designs the access speed to the memory is related to the amount of memory used. This means that a 1 MB DRAM card will run slower than a 2 MB one. This limitation does not apply to MDRAM. MDRAM is also cost-effective to manufacture compared to VRAM and is efficiently organized to reduce waste. MDRAM is suitable for use in high-end applications and is becoming popular due to its performance-enhancing and cost-reducing features.

6.1.5 Comparison of video memory technologies

The table below summarizes the most common video memory technologies and contrasts some of their basic characteristics. Note that the bandwidth factor is intentionally approximate, as it should be considered a rough guideline. Among other things, the actual bandwidth of the memory depends on the speed of the memory, which can vary widely (some DRAMs have access time half that of others).

6.2 FIFO

FIFO memories are realized either right in the processor or like structural elements with various organization. Basically we can divide them into these types:

Technology	Access Ports	Approximate Bandwidth	Relative Cost
Standard DRAM	Single	Low	Low
EDO DRAM	Single	Low	Low
VRAM	Dual	High	High
WRAM	Dual	High	High
SGRAM	Single	Very High	Moderate
MDRAM	Single	Very High	High

Table 2: Comparison of video memory technologies [13]

- without moving content - reading and writing to a queue is controlled by two registers - it is read based on the content of the queue beginning register, it is written based on the content of the queue end register. The control circuits also have to create two important signals - empty queue and full queue (to prevent underflow and overflow [1]).
- with moving content - circuit realization of the queue with moving content when reading and writing is complicated; they have one additional register. A queue with bubble through shifting of every element after asynchronous writing procedure up to the last free position, by reading one element from the beginning of the queue one position is freed, after which they are occupied with the same mechanism by other elements. This principle requires to have an indicator of occupancy (flip-flop) by every memory position.

All the mentioned principles allow more or less simple lining of FIFO modules into cascades and extending the queue as needed. Increasing the length of the memory position in a queue is done by parallel module lining that does not cause fundamental problems.

6.3 Cache

Cache memory can be found in every computer architecture. It is some sort of a buffer storage between differently fast computer components. Its purpose is mutual speed adjustment - faster component reads data from cache and does not have to wait for the slower component (where cache already read the data).

6.3.1 Characteristics of cache

The basic parameters about cache activity is the probability of success (hit ratio), or the probability of failure (miss rate), or the probability of block failure. These parameters can be defined separately for reading and writing, for data and instructions. The time required for finding a block is the access time of cache in successful case, when failure occurs then fault time is added (miss penalty), which is time required to bringing up a block. It is determined by the time needed for freeing the position in cache, the access time of the first word of the required block in the further memory, plus the whole block transfer time.

6.3.2 Organization of cache

The problem of mutual driving out of the elements with the same pointer is solved by increasing the degree of association. In two way memory there can be two elements stored with the same pointer.

The degree of association can be raised all the way to the fully associative memory.

In two way cache there is a problem of choosing the element that is to be driven out. If all the elements for a certain pointer are occupied then it is necessary to decide which element will be canceled and thus the position will be freed for new address. This problem is solved by few replacement strategies:

- Least Recently Used (LRU) - the elements that has been used just recently are left and the longest not used element is canceled
- Most Frequently Used (MFU) - frequently used elements are left and least frequently used elements are canceled
- RAND - a randomly chosen element
- FIFO - the longest lasting element is chosen

LRU, MFU, FIFO strategies require additional circuitry, like registers to keep the time of usage.

6.3.3 L1 cache (first level cache)

This cache is integrated into the processor. It is used for supplying the processor with data from the bus. Cache reads more data from the bus that wait in this buffer storage. Whenever processor needs them it can read them from cache. Since cache operates faster than bus then the processor does not have to wait, as it would be the case when the data would be read directly from the bus.

6.3.4 Adjusting the speed of the processor and system memory

The developement of the new range of processors also brings the increase of their period. Other PC components can not keep up with still more faster processors. Generally, the speed of the PC is not determined by its fastest component but by its slowest part.

The processor most often cooperates with the system memory. The memory must supply the required data in two periods of internal processor frequency so that it does not have to wait. Slower Pentium works with frequency of 120 MHz, one period of the processor lasts 8.3 ns, the memory should then respond in 16.6 ns which highly exceeds the possibilities of DRAM circuits.

Basically, 3 possibilities exist of mutual speed adjusting of DRAM and processor:

- inserting of wait states - this principle is quite simple - the processor waits until the memory supplies the needed information. It does not work for few periods but waits. This type of adjustment stopped to be used at 386 processors for simple reason. Even though, we can not avoid this to happen in the most modern computers.
- using SRAM circuits as system memory would surely significantly increase the speed of memory work but the computer would be very expensive.
- using L2 cache (is PC's standard solution).

6.3.5 L2 cache (second level cache)

L2 cache is situated between the processor and the system memory so all the data that pass through these two units will stay in the cache and if the processor needs them again, it can read them from faster cache. In addition, cache is controlled with special driver that tries to predict what data will processor need in a next time. The driver copies this data from the system memory into cache where the processor finds them and does not have to reach for them to the system memory that would require wait states. The ratio between the successful requests from cache and all the requests repeated by the processor is determined by *Hit Rate* which is between 80 - 90% [1]. It is clear that the wait states do not occur often - cache then really speeds up the processor - system memory communication. It also works as a multi-port memory where data can be written and read in the same time.

Cache uses 3 modes:

- Write-Through - it is the oldest and the slowest way, typical for 486 processor. The data stored into cache are stored into the system memory as well. When reading takes place the cache driver compares the requested addresses of the system memory with the already stored addresses. If the requested data is found in the cache, they will be read from it.
- Write-Back - it is newer and faster method used by Pentium and some of the faster 486 processors. The data is only stored into cache and once they are removed from it then they are stored into the system memory. Before the data gets to the system memory, they can be changed several times in the cache. The time is saved with this mode, required for repeated storing into the slower system memory.
- Pipeline Burst - it is the newest and the fastest system of operating currently used. It will perform more pipelined operations - if it is reading the information from a certain address, it will read the information from the next address as the same time (it would be probably doing that shortly). The access time is between 9 to 15 ns.

The mode, that cache will be operating in, must be supported by the motherboard and memory modules and cache driver. The motherboard often supports more modes, the desired one can be chosen in SETUP mode.

7 Memory errors, detection and correction

This chapter is divided into four sections. The first one gives a short description of memory errors. The second one talks about the parity and non-parity memories and parity checking. The third section discusses the usage of ECC memories. In the last section of this chapter a list of advantages and disadvantages of the parity, non-parity and ECC memories is presented.

7.1 Memory errors

Memory is an electronic storage device, and all electronic storage devices have the potential to incorrectly return information different than what was originally stored. Some technologies are more likely to do this than others. DRAM memory, because of its nature, is likely to return occasional memory errors. DRAM memory stores ones and zeros as charges on small capacitors that must be continually refreshed to ensure that the data is not lost. This is less reliable than the static storage used by SRAMs.

Every bit of memory is either a zero or a one. This in itself helps to eliminate many errors, because slightly distorted values are usually recoverable. For example, in a 5 volt system, a “1” is +5 V and a “0” is 0 V. If the sensor that is reading the memory value sees +4.2 V, it knows that this is really a “1”, even though the value is not +5 V. That’s because the only other choice would be a “0” and 4.2 V is much closer to 5 V than to 0 V. However, on rare occasions +5 V might be read as +1.9 V and be considered a “0” instead of a “1”. When this happens, a memory error has occurred.

There are two kinds of errors that can typically occur in a memory system. The first is called a repeatable or hard error. In this situation, a piece of hardware is broken and will consistently return incorrect results. A bit may be stuck so that it always returns “0” for example, no matter what is written to it. Hard errors usually indicate loose memory modules, blown chips, motherboard defects or other physical problems. They are relatively easy to diagnose and correct because they are consistent and repeatable.

The second kind of errors are called a transient or soft errors. They occur when a bit reads back the wrong value once, but subsequently functions correctly. Such problems are, understandably, much more difficult to diagnose. They are also, unfortunately, more common. Eventually, a soft error will usually repeat itself, but it can take anywhere from minutes to several years for this to happen. Soft errors are sometimes caused by memory that is physically bad, but at least as often they are the result of poor quality motherboards, memory system timings that are set too fast, static shocks, or other similar problems that are not related to the memory directly. In addition, stray radioactivity that is naturally present in materials used in PC systems can cause the occasional soft error. On a system that is not using error detection, transient errors often are written off as operating system bugs or random glitches.

The exact rate of errors returned by modern memory is a matter of some debate. It is agreed that the DRAMs used today are far more reliable than those of five to ten years ago. This has been the chief excuse used by system vendors who have dropped error detection support from their PCs. However, there are factors that make the problem worse in modern systems as well. First, more memory is being used; 10 years ago the typical system had 1 MB to 4 MB of memory; today’s systems usually have 256 MB to 1024 MB, since RAM prices have fallen dramatically in the last decade. Second, systems today are running much faster than they used to; the typical memory bus is running from 3 to 10 times the speed of those older machines. Finally, the quality level of the average PC is way down from the levels of 10 years ago.

The only true protection from memory errors is to use some sort of memory detection or correction techniques. Some techniques can only detect errors in one bit of an eight-bit data byte; others can

detect errors in more than one bit automatically. Others can both detect and correct memory problems, seamlessly.

7.2 Parity and non-parity

If modules have parity at all, this can be one of two kinds: physical and logical. Physical parity means that the parity bits are supplied by the memory controller during data write, and are stored in the module's memory chips. In a read access the module outputs the stored parity information. When logical parity is used, the memory controller generates the parity bits and supplies them to the module during a write access, but the module ignores this parity information, that is, the parity bits are not stored. In a read access the simple circuitry of the module now generates the parity information from the stored data bits. Therefore, a parity error can never occur; the parity information provided by the module is irrelevant. Such modules save more than 10% (4 to 36 bits) of memory capacity and are accordingly cheaper. They serve mainly to implement main memory (RAM) which is controlled by a memory controller that requires parity information.

Memory modules have traditionally been available in two basic flavors: non-parity and parity. (Actually, some sizes and styles are only available in non-parity, but most are available either way). Non-parity is "regular" memory - it contains exactly one bit of memory for every bit of data to be stored. 8 bits are used to store each byte of data. Parity memory adds an extra single bit for every eight bits of data, used only for error detection. 9 bits of data are used to store each byte.

Parity memory can be used for parity checking, a basic form of error detection, on PCs that support it. Non-parity memory provides no error detection capabilities at all unless these are provided through external circuitry (which is basically never done on regular PCs).

7.2.1 Parity checking

Parity checking is a elementary method of detecting simple, single-bit errors in a memory system. It in fact has been present in PCs since the original IBM PC in 1981, and until the early 1990s was used in every PC sold on the market. It requires the use of parity memory, which provides an extra bit for every byte stored. This extra bit is used to store information to allow error detection. Parity checking on newer systems normally requires the appropriate BIOS setting to be enabled. ECC-only modules cannot be used in straight parity-checking mode.

Every byte of data that is stored in the system memory contains 8 bits of real data, each one a zero or a one. It is possible to count up the number of zeros or ones in a byte. For example, the byte 10110011 has 3 zeros and 5 ones. The byte 00100100 has 6 zeros and 2 ones. As you can see, some bytes will have an even number of ones, and some will have an odd number.

When parity checking is enabled, each time a byte is written to memory, a logic circuit called a parity generator/checker examines the byte and determines whether the data byte had an even or an odd number of ones. If it had an even number of ones, the ninth (parity) bit is set to a one, otherwise it is set to a zero. The result is that no matter how many ones there were in the original eight data bits, there is an odd number of ones when you look at all nine bits together. This is called odd parity. (It is also possible to have even parity, where the generator makes the sum always come out even, but the standard in PC memory is odd parity).

When all nine bits are taken together, there are always an odd number of ones. When the data is read back from memory, the parity circuit this time acts as a checker. It reads back all nine bits and determines again if there is an odd or an even number of ones. If there is an even number of ones, there must have been an error in one of the bits, because when it stored the byte the circuit set

the parity bit so that there would always be an odd number of ones. This is how parity memory is used to detect errors - the system knows one bit is wrong, although it doesn't know which one it is. When a parity error is detected, the parity circuit generates what is called a "non-maskable interrupt" or "NMI" [14], which is usually used to instruct the processor to immediately halt. This is done to ensure that the incorrect memory does not end up corrupting anything.

There are limitations of the parity checking. Let's say we have a data byte of 00100100 and this is stored as "00100100 1" including the parity bit. Now let's say this is read back as "01100000 1". Here we have two bits that have flipped, one of them from 1 to 0 and the other from 0 to 1. However, the amount of "ones" remains odd! As you can see, parity does not protect against double-bit errors.

Incidentally, contrary to popular myth, parity checking does not slow down the operation of the memory system. The parity bit generation and detection is done in parallel with the writing or reading of the system memory, in transistor-to-transistor logic that is much faster than the DRAM memory circuits being used. Nothing in the system ever waits on a "go ahead" signal from the parity checking circuit. It only does anything if it finds an error and when it does, it uses a system interrupt.

7.3 ECC memory

Parity checking provides single-bit error detection for the system memory, but does not handle multi-bit errors, and provides no way to correct memory errors. An advanced error detection and correction protocol was invented to go a step beyond simple parity checking. Called ECC, which stands for error correcting circuits, error correcting code, or error correction code, this protocol not only detects both single-bit and multi-bit errors, it will actually correct single-bit errors on the fly, transparently. Like parity checking, ECC requires a setting in the BIOS program to be enabled. Often there are two; one turns on parity checking and the other tells the system to use ECC mode.

ECC uses a special algorithm to encode information in a block of bits that contains sufficient detail to permit the recovery of a single bit error in the protected data. Unlike parity, which uses a single bit to provide protection to eight bits, ECC uses bit groups: 7 bits to protect 32 bits, or 8 bits to protect 64 bits. There are special ECC memory modules designed specifically for use in ECC mode, but most modern motherboards that support ECC will in fact work in that mode using standard parity memory modules as well. Since parity memory includes one extra bit for every eight bits of data, this means 64 bits worth of parity memory is 72 bits wide, which means there is enough to do ECC. In fact, parity SIMMs are 36 bits wide (two are used in a fifth or sixth generation system) and parity DIMMs are 72 bits.

ECC has the ability to correct a detected single-bit error in a 64-bit block of memory. When this happens, the computer will continue without a hiccup; it will have no idea that anything even happened. However, if you have a corrected error, it is useful to know this; a pattern of errors can indicate a hardware problem that needs to be addressed. Chipsets allowing ECC normally include a way to report corrected errors to the operating system, but it is up to the operating system to support this. Windows NT and Linux do account these messages, but Windows 95 does not. In the latter case, you will not know when ECC has corrected a single-bit error. The user must decide if this is a concern or not; setting the system for simple parity checking will cause notification when an error occurs, but on-the-fly correction will be lost.

ECC will detect (but not correct) errors of 2, 3 or even 4 bits, in addition to detecting (and correcting) single-bit errors. ECC memory handles these multi-bit errors similarly to how parity handles single-bit errors: a non-maskable interrupt (NMI) that instructs the system to shut down to avoid data corruption. Multi-bit errors are extremely rare in memory.

An ECC module contains one extra bit per byte the way parity ones do, the extra bits cannot

be individually accessed, which is required for parity operation. Some systems cannot use the ECC modules at all, because they are wired differently than parity modules.

Parity memory was once the only kind sold; it is now sold much less frequently than non-parity memory, especially the faster and newer types of memories. Parity memory uses 12.5% more DRAM memory than non-parity (1 bit extra for every 8) which makes it more expensive, but the major reason that it costs more is simply that it is produced in much smaller quantities today. With processors increasing in speed and more high-end applications coming into prominence on the PC platform, error-checking memory is now again on the increase.

Unlike parity checking, ECC will cause a slight slowdown in system operation. The reason is that the ECC algorithm is more complicated, and a bit of time must be allowed for ECC to correct any detected errors. The penalty is usually one extra wait state per memory read. This translates in most cases to a real world decrease in performance of approximately 2-3%.

7.4 Parity vs. non-parity vs. error correcting code

Few advantages of using parity checking or ECC:

- Data Integrity: You run far less risk of a memory problem causing corruption and possible data loss. With ECC errors can be corrected on the fly. Even with regular parity checking you avoid writing bad data to your disk and accumulating errors that can cause long-term data loss. See here for details on the types and causes of memory errors.
- Easier Troubleshooting: If you don't use parity checking, any time you have a strange glitch in your system you cannot rule out a memory problem being the cause. When you run with parity checking in place you know most of the time when a problem is related to memory.
- Advance Warning of Hardware Failure: Many hardware failures over time can manifest themselves through parity errors in a parity checking system. Without parity, these can remain hidden and hard to diagnose.

Few disadvantages of using parity checking or ECC:

- Greater Expense: Parity memory costs more than non-parity in most cases.
- Harder to Find: Parity memory can be more difficult to buy, especially in the newer technologies.
- Occasional False Positives: Any time a parity error occurs, there is a 1 in 9 chances that the bit that was inverted to cause the error was actually the parity bit itself. This means that the 8 data bits themselves were fine, and if parity had not been in use there would have been no problem at all. This really does not apply to ECC, which would handle this situation with no difficulty.
- Poor Error Handling: Regular parity checking deals with a parity error by sending a non-maskable interrupt (NMI) to the processor, halting the system immediately. Depending on the timing of this, it can cause lost data. (But not the accumulation of data errors over time as with non-parity). This does not apply to ECC (except in the very rare case of a double-bit error).
- Performance Penalty For ECC: ECC does cause a minor slowdown in system performance.

8 Conclusions

The thesis work describes computer memories. This work should be used as a studying material for students of the Department of Computer Science, VŠB-TU Ostrava, but generally it is for anyone with an interest in this field of computer science.

A detailed overview about all memories used in today's computers was presented. In the main part, all important types of computer memories were described using schemes. The aim was to create a full and understandable description of particular technologies. The explanation of the functional principles was supported by well processed schemes and images. The comparison between different types of memory technologies appears often in the text, so that the differences emerge.

In addition to the main memory types, a summary of other types was presented, e.g. video memories and cache type memories. Memory errors, their detection and correction were also discussed.

This work can earn reader's deeper understanding of the described problems, that cannot be explained at the lectures in a such detailed way. Together with other simultaneously elaborated texts at the Department of Computer Science, covering the production technology of digital circuits, external memories and others, a compact description of these not very detailed topics can be given. This text can be extended with updated information as needed in the future.

References

[1] LIČEV, L.: *Architektura počítačů II*., skriptum FEI VŠB-TU Ostrava, 1999.

[2] Davis B. T., *MODERN DRAM ARCHITECTURES*. [17-05-2004] Available at: http://www.eecs.umich.edu/tnm/theses/briand.pdf

[3] Breecher J., *Memory design I*. [23-05-2004] Available at: http://babbage.clarku.edu/jbreecher/comp_org/lectures/Chapter7.1-Memory1.pdf

[4] Vyskočil J., *Paměti*. [16-02-2004] Available at: http://ulita.ms.mff.cuni.cz/mff/seznamy/old/prg022_00_do/Pameti_Vyskocil/pameti.ppt

[5] Drajsajtl T., *Paměti*. [16-02-2004] Available at: http://www.drajsajtl.cz/PCK/projekty2001.php3?p_nr=0103&p_file=#1

[6] VALAŠEK, P., LOSKOT, R.: *Polovodičové paměti* - druhé vydání. BEN technická literatura, 2001.

[7] *ORGANIZACIJA I STRUKTURA DRS*. [16-02-2004] Available at: http://www.ktf-split.hr/informatika/str-20.htm

[8] Pelikán J.: *Paměti*. [16-02-2004] Available at: http://zsprazska.oknet.cz/navody/arch_PC/TEXTY/INTPAM.HTML

[9] Messmer H.: *The Indispensable PC Hardware Book - Fourth Edition*. Addison-Wesley, 2002, ISBN: 0-201-59616-4.

[10] Micron Technology, Inc.: *micron FP DRAM*. [20-05-2004] Available at: http://download.micron.com/pdf/datasheets/dram/D49_5V.pdf, Micron Technology, Inc., 2000.

[11] Micron Technology, Inc.: *micron EDO DRAM*. [20-05-2004] Available at: http://download.micron.com/pdf/datasheets/dram/d47.pdf, Micron Technology, Inc., 1997.

[12] Jacob B., Wang D.: *DRAM Memory System: Lecture 2*. [16-05-2004] Available at: http://www.ece.umd.edu/courses/enee759h.S2003/lectures/Lecture2.pdf, University of Maryland, 2003.

[13] Kozierok C. M., *PC Guide - Reference Guide - Video Memory Technologies*. [02-06-2004] Available at: http://www.pcguide.com/ref/video/tech.htm

[14] Kozierok C. M., *PC Guide - Reference Guide - Parity Checking*. [02-06-2004] Available at: http://www.pcguide.com/ref/ram/errChecking-c.html