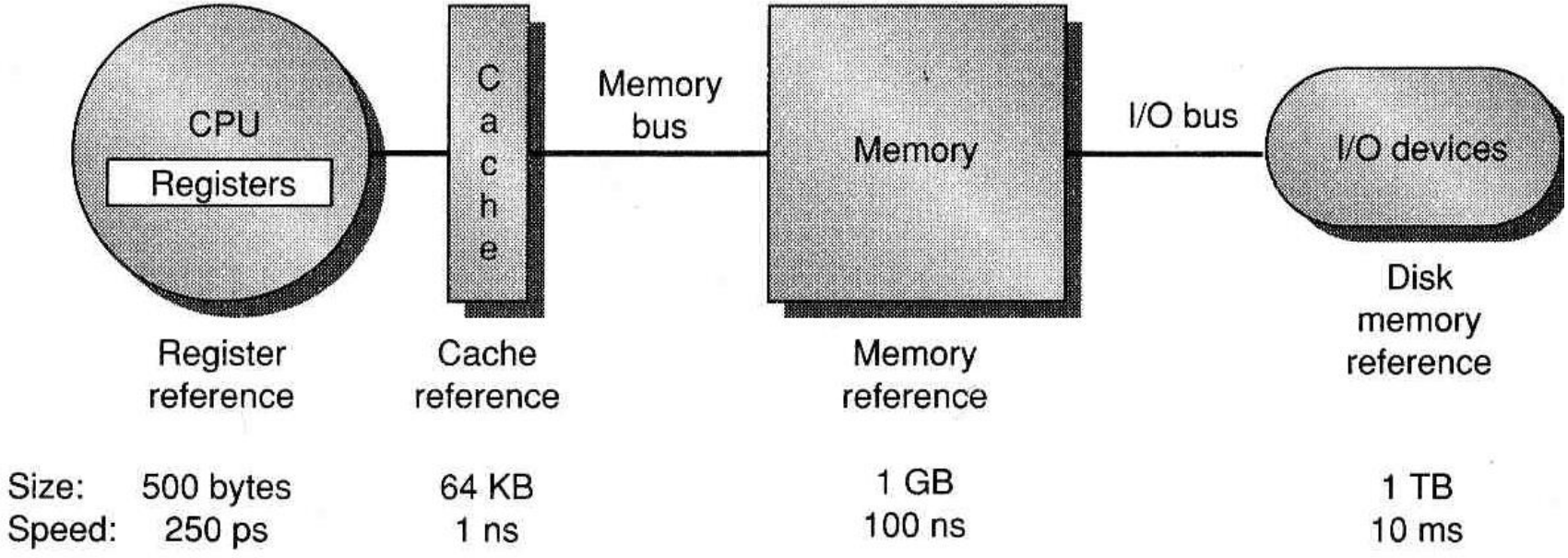
Random Access Memory

Lecture 3

SDRAM, DDR, CAS, RAS, memory channels, performance

Type of the memory



Type of the memory

Processor

- Cache (L1, L2, L3)
- Registers

Internal memory

- Random Acces Memory
- Video memory (used for storing information displayed on the monitor)
- HDD buffer
- BIOS (***b****asic* ***i****nput/****o****utput* ***s****ystem*)

External memory

- SSD, HDD
- CD-ROM, (DVD-ROM)
- FDD
- Magnetic tape

Memory characteristics

Capacity

- Length of the word
- Number of the words

Access type

- Sequential (tapes)
- Direct (HDD, CD, DVD)
- Random (RAM)
- Association (cache)

Transfered data

- Word
- Block

Memory characteristics

Performance

- Access time
- Cycle time
- Throughput
- Latency

Type

- Semiconductor
- Magnetic
- Optical
- Magneto-optical

Physical characteristics

- Saving information/not-saving information
- Erasable / not erasable

Registers

Processor register is a small amount of very fast computer memory used to speed the execution of computer programs by providing quick access to commonly used values—typically, the values being calculated at a given point in time.

Most, but not all, modern computer architectures operate on the principle of moving data from main memory into registers, operating on them, then moving the result back into main memory a so-called **load-store** architecture..

Registers

Categories of registers

- Registers are normally measured by the number of bits they can hold, for example, an "8-bit register" or a "32-bit register".

There are several classes of registers according to the content:

- **User-Visible Registers**
  - Data Registers
  - Address registers.

Data registers are used to store **integer** numbers (see also Floating Point Registers, below). In some older and simple current CPUs, a special data register is the **accumulator**, used implicitly for many operations.

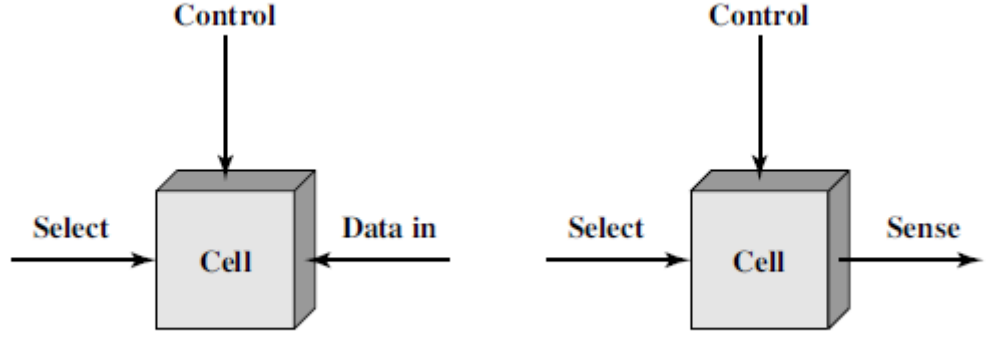
Registers

- **Address registers** hold memory addresses and are used to access memory. In some CPUs, a special address register is an index register, although often these hold numbers used to modify addresses rather than holding addresses.
- **Conditional registers** hold truth values often used to determine whether some instruction should or should not be executed.
- **General purpose registers** (**GPR**s) can store both data and addresses, i.e., they are combined Data/Address registers.
- **Floating point registers** (**FPR**s) are used to store floating point numbers in many architectures.
- **Constant registers** hold read-only values (e.g., zero, pi, ...).
- **Vector registers** hold data for vector processing done by SIMD instructions (Single Instruction, Multiple Data).
- **Special purpose registers** hold program state; they usually include the program counter (aka instruction pointer), stack pointer, and status register (aka processor status word).

Main memory

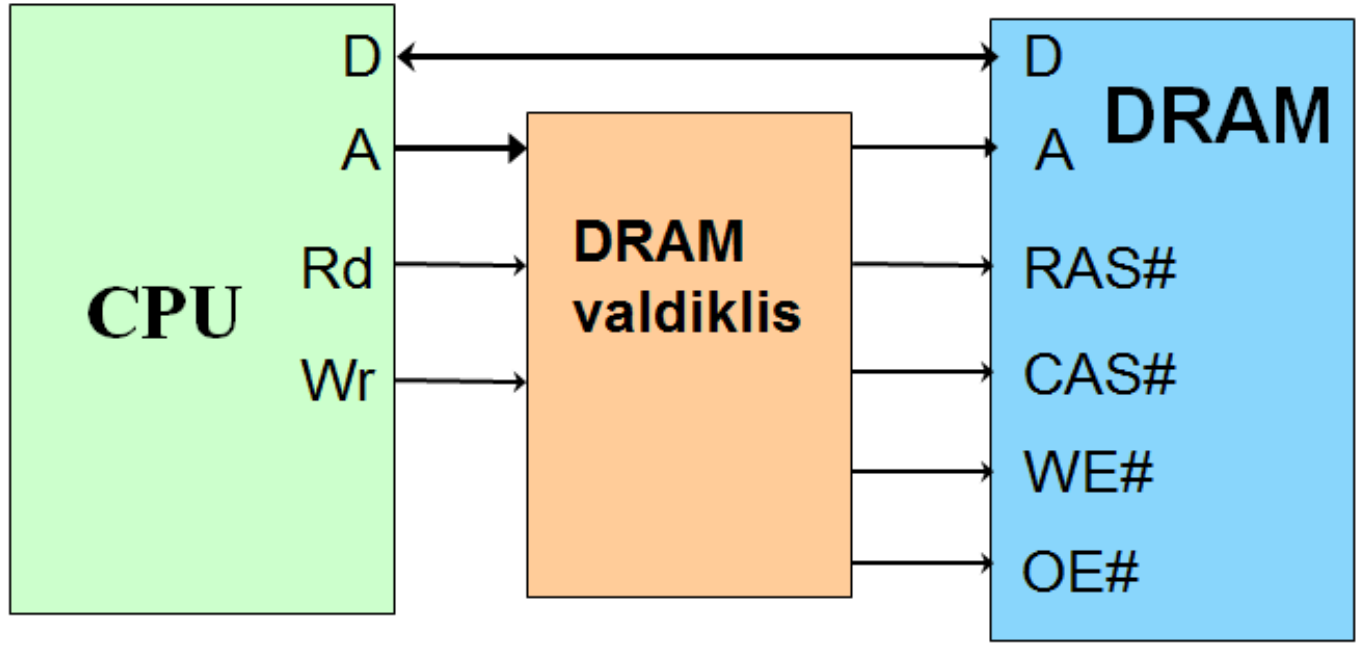
The basic element of a semiconductor memory is the **memory cell**. Although a variety of electronic technologies are used, all semiconductor memory cells share certain properties:

- They exhibit two stable (or semistable) states, which can be used to represent binary 1 and 0.
- They are capable of being written into (at least once), to set the state.
- They are capable of being read to sense the state.

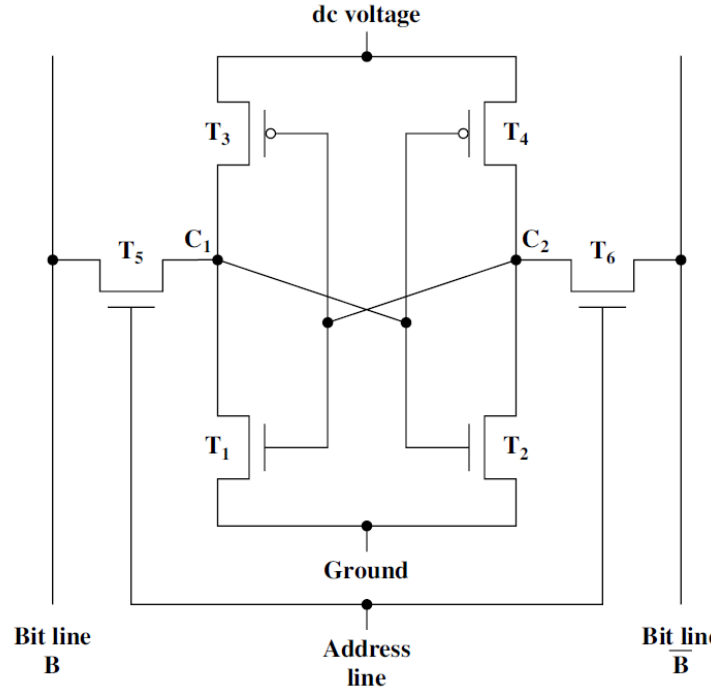


Random access memory

RAM – random access memory is semiconductor type memory, when any word in the memory is accessed directly using special addressing system.



SRAM

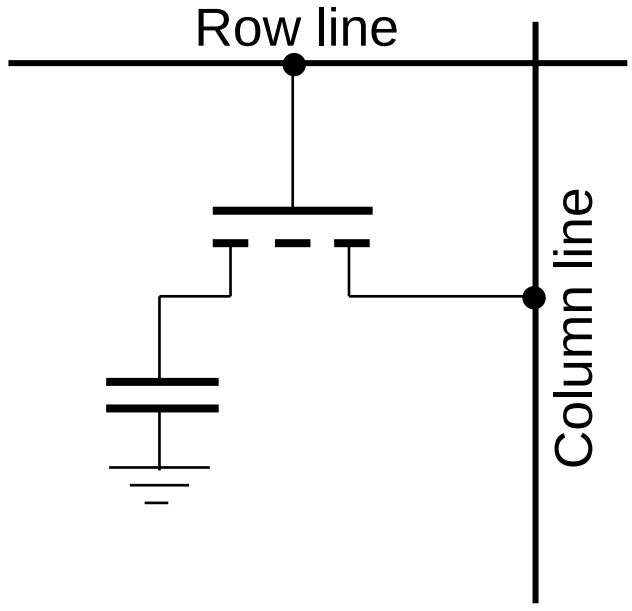


SRAM – static random access memory

SRAM is a digital device that uses the same logic elements used in the processor. In a SRAM, binary values are stored using traditional flip-flop logic-gate configurations. A static RAM will hold its data as long as power is supplied to it.

SRAMs are generally faster than DRAM. Because of these relative characteristics, SRAM is used for cache memory (both on and off chip), and DRAM is used for main memory.

DRAM



DRAM – dynamic random access memory.

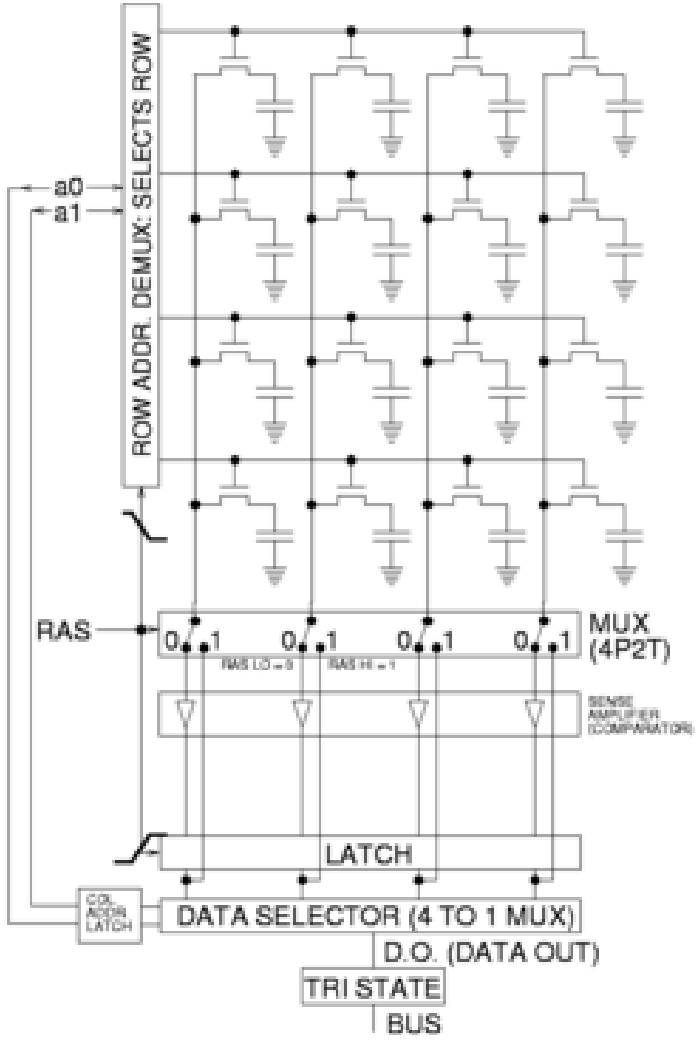
A dynamic RAM is made with cells that store data as charge on capacitors.The presence or absence of charge in a capacitor is interpreted as a binary 1 or 0.

Because capacitors have a natural tendency to discharge, dynamic RAMs require periodic charge refreshing to maintain data storage.

The term *dynamic* refers to this tendency of the stored charge to leak away, even with power continuously applied.

A typical DRAM structure for an individual cell that stores 1 bit. The address line is activated when the bit value from this cell is to be read or written. The transistor acts as a switch that is closed (allowing current to flow) if a voltage is applied to the address line and open (no current) if no voltage is present on the address line.

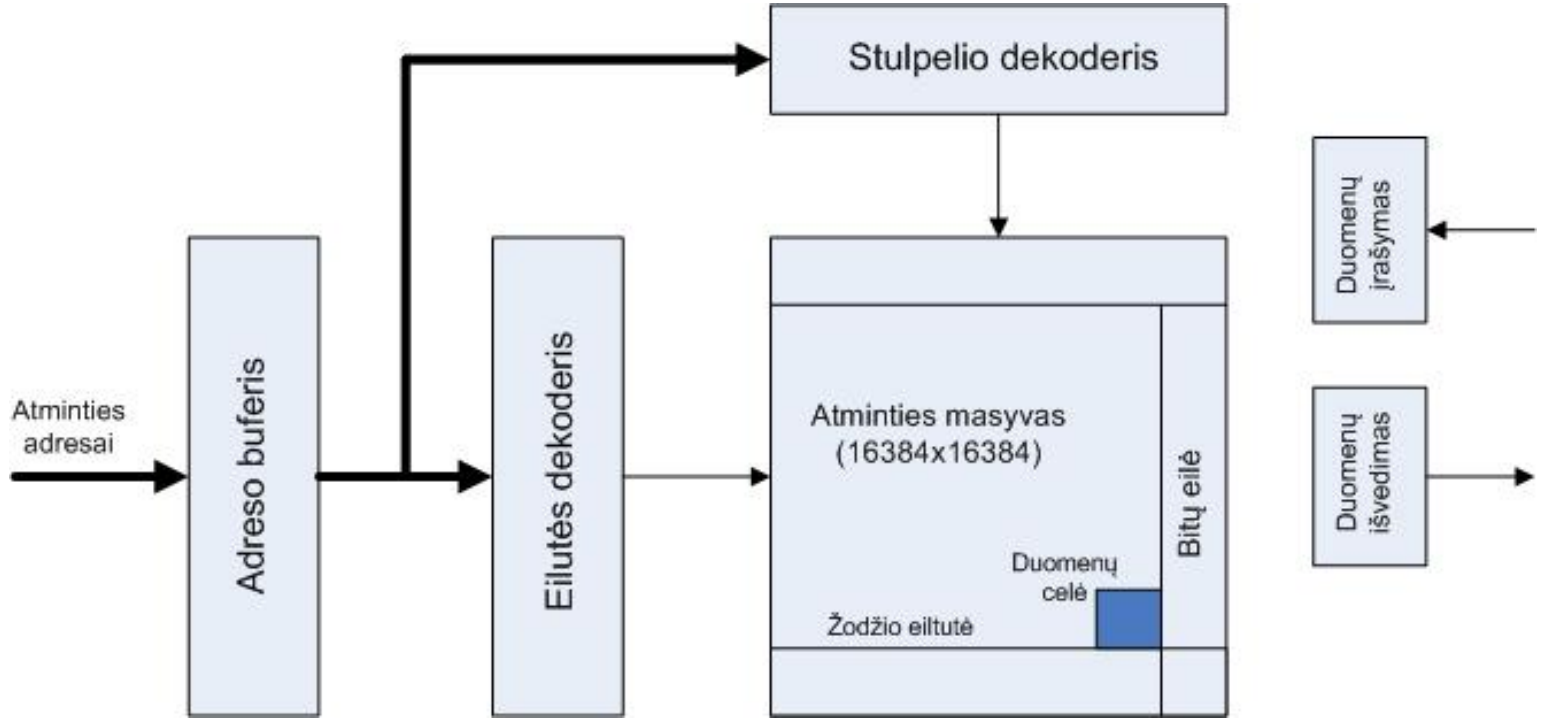
DRAM matrix



4x4 main memory matrix. Each memory cell has row and column number.

Main memory consists of several layers i.e. it is multilayer structure.

DRAM internal structure



Procedures used to read date form main memory.

From memory address the following data are obtained:

a) Row Access Strobe - RAS

b) Column Access Strobe -CAS

Memory latency depends on RAS. Memory bandwidth depends on CAS.

DRAM

Type of DRAM:

- Asynchronous
- Synchronous SDRAM

SDRAM exchanges data with the processor synchronized to an external clock signal and running at the full speed of the processor/memory bus without imposing wait states.

With synchronous access, the DRAM moves data in and out under control of the system clock. The processor or other master issues the instruction and address information, which is latched by the DRAM.

RAS, CAS

RAM is organized into rows and columns, and is accessed by electrical signals called **strobes**, which are sent along rows to the columns.

When data is needed, the CPU activates the **RAS** (Row Access Strobe) line to specify the row where data is to be found (high bits), then, after a short time, the **CAS**, or Column Access Strobe, to specify the column (low bits).

After that, the data goes to the output line and to its destination on the next clock tick. In other words, the Column Address Strobe dictates how many clocks the memory waits before sending data on.

RAS, CAS

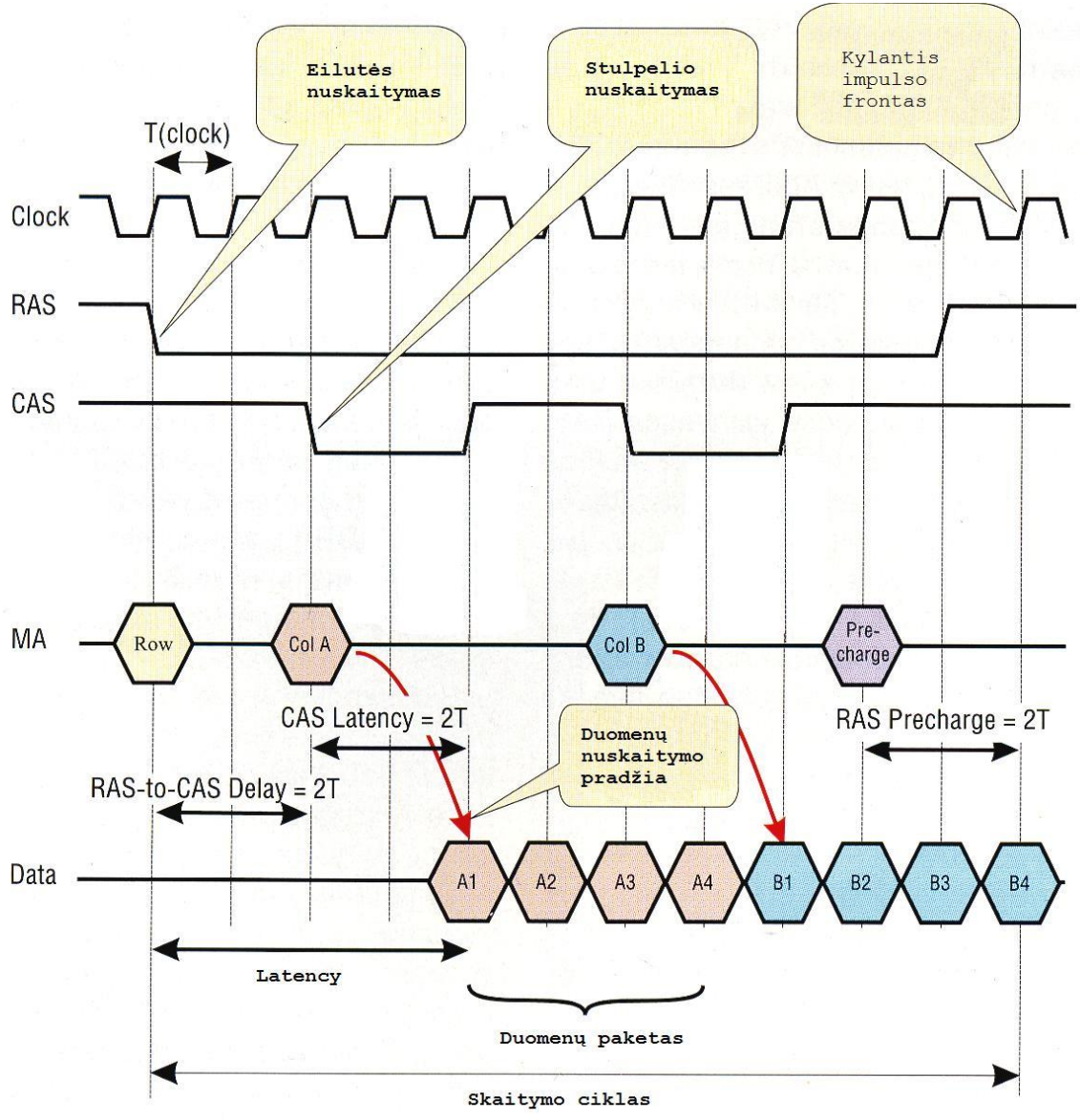
The shorter the cycle length, the faster the machine runs, at the expense of stability and data.

Linked with CAS are RAS and RAS-to-CAS, usually set to 2 or 3 with SDRAM Cycle Length, although you may be able to set them independently, and preferably in the reverse order to the above.

Numbers on the chip looking like 3-2-2 refer to CAS, RAS-to-CAS and RAS, respectively.

Running the chips at higher than rated speeds will mean dropping a CAS/RAS level.

SDRAM diagram

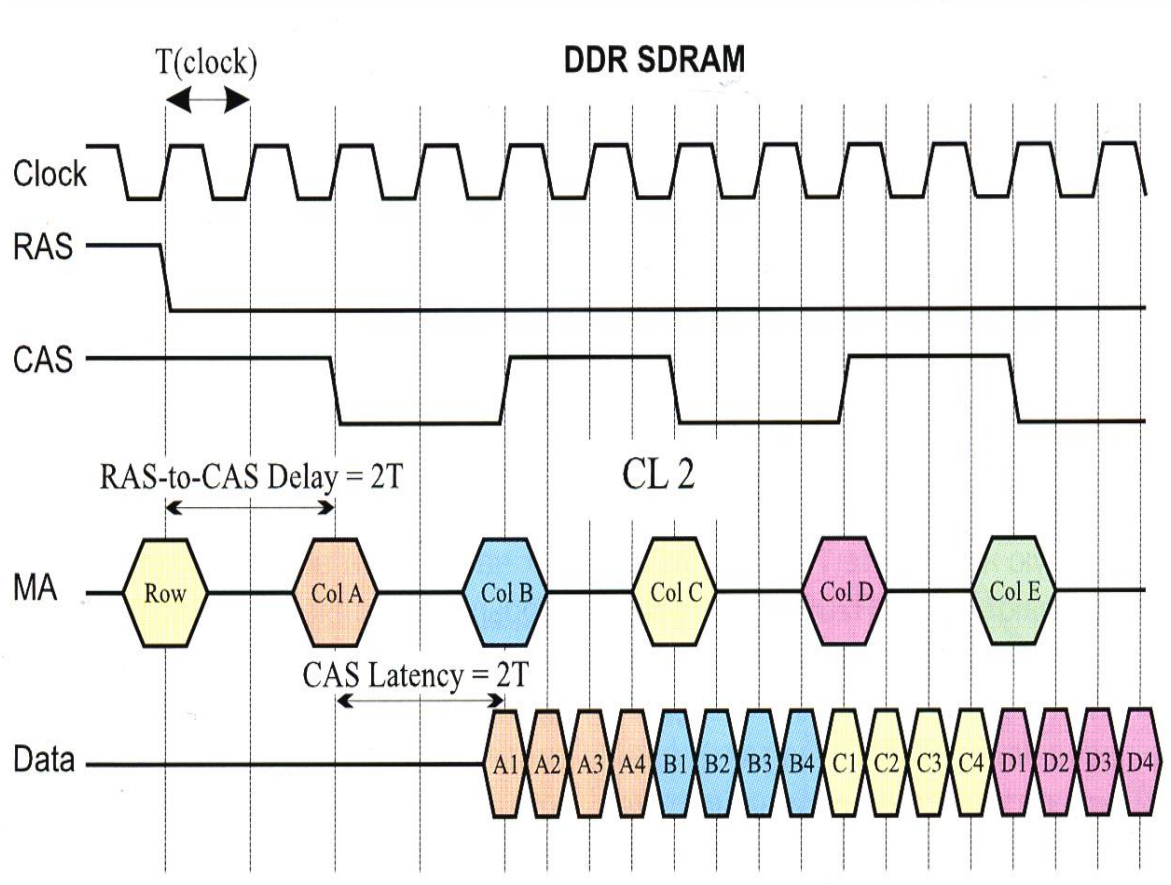


SDRAM

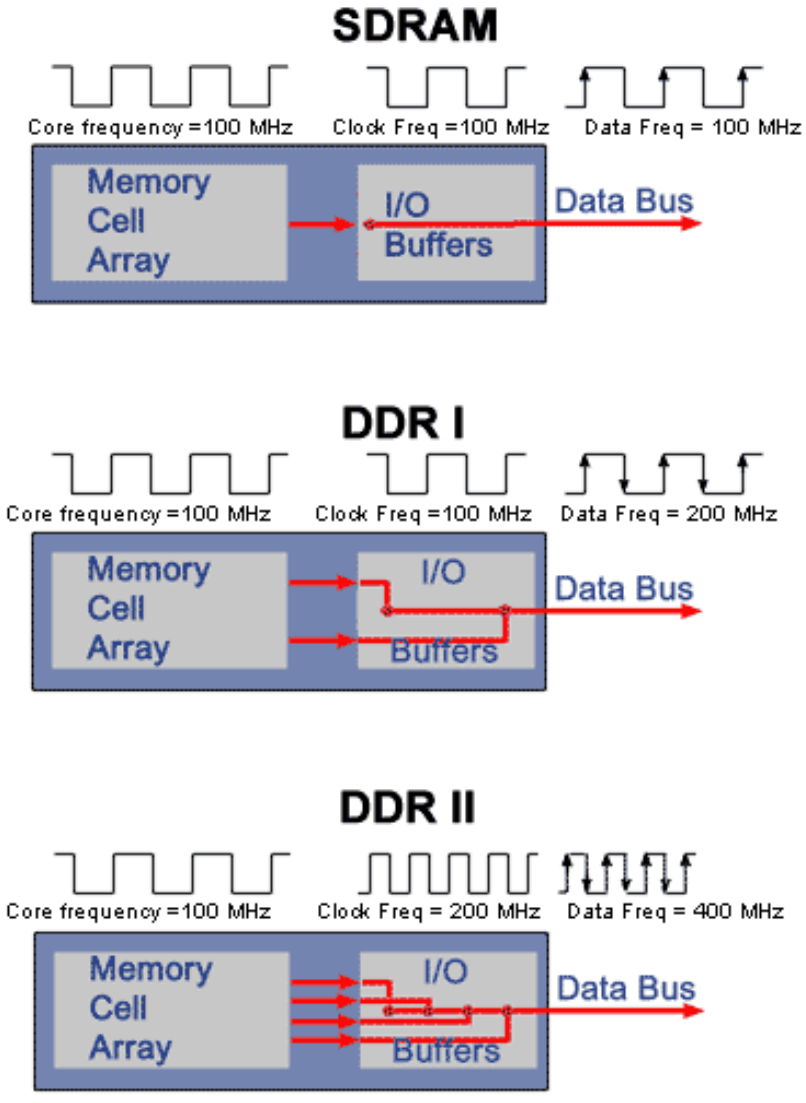
- **SDR** – single data rate
  - SDRAM PC100 (100MHz); SDRAM PC133 (133 MHz)
- **DDR** – double data rate
  - Frequency (100, 133, 166, 200 MHz) is muliplied by 2, therefore: DDR200 (PC1600), DDR266 (PC2100), DDR 333 (PC2700), DDR400 (PC3200)
- **DDR2** – double data rate 2 (100, 133, 166, 200, 266 MHz) multiply by 4.
- **DDR3** – double data rate 3 (100, 133, 166, 200, 233, 266 MHz) multiply by 8.
- **DDR4** – double data rate 4 (16 GB, 2xDDR3 MT/s, since 2014)

DDR SDRAM

DDR SDRAM, referred to as double-data-rate SDRAM can send data twice per clock cycle, once on the rising edge of the clock pulse and once on the falling edge.



SDRAM, DDR, DDR 2

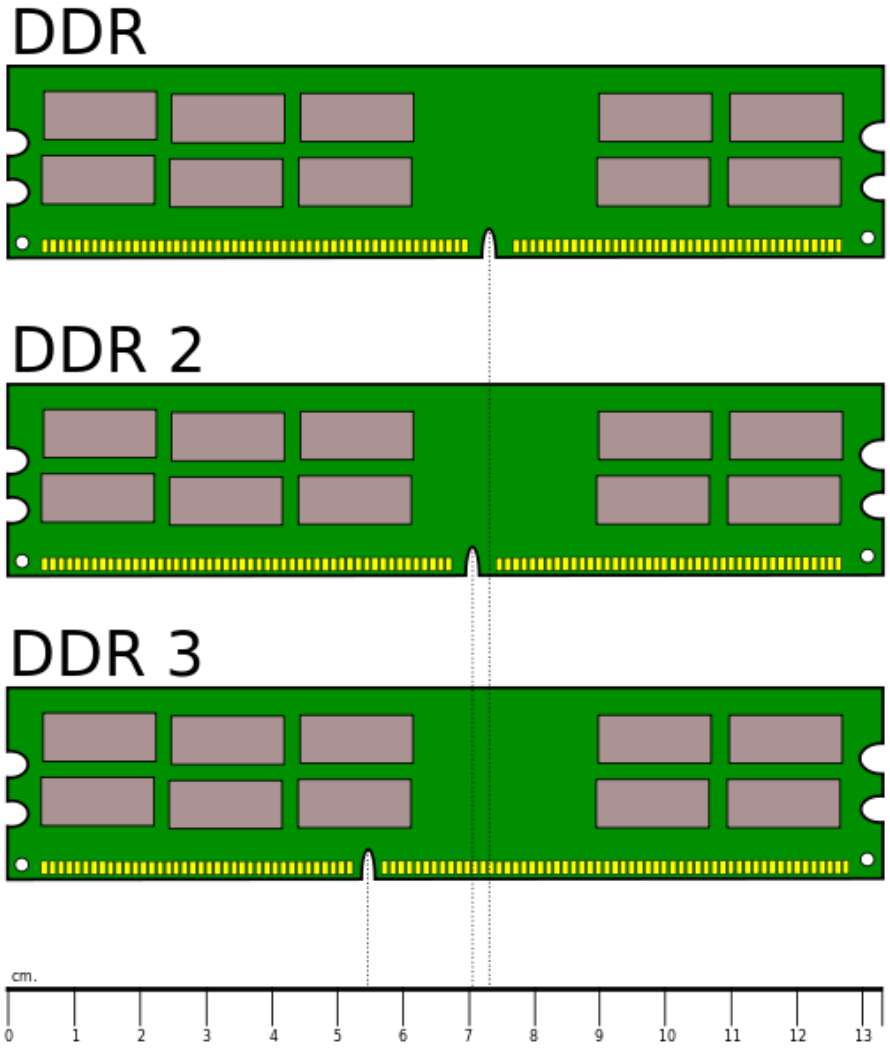


DDR2 increases the data transfer rate by increasing the operational frequency of the RAM chip and by increasing the prefetch buffer from 2 bits to 4 bits per chip.

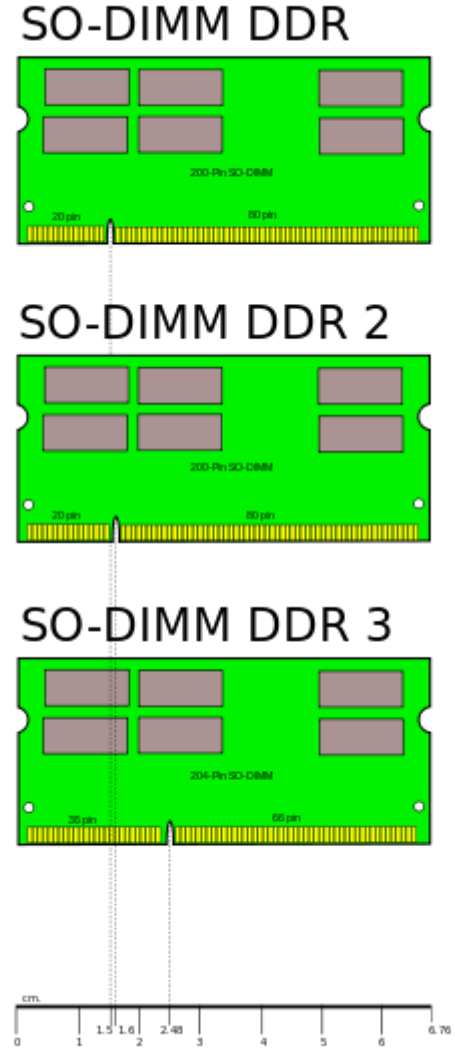
The prefetch buffer is a memory cache located on the RAM chip. The buffer enables the RAM chip to preposition bits to be placed on the data base as rapidly as possible.

DDR3, introduced in 2007, increases the prefetch buffer size to 8 bits. Theoretically, a DDR module can transfer data at a clock rate in the range of 200 to 600 MHz; a DDR2 module transfers at a clock rate of 400 to 1066 MHz; and a DDR3 module transfers at a clock rate of 800 to 1600 MHz. In practice, somewhat smaller rates are achieved.

DDR modules



13.3 cm (5.25 coliai)



6,7 cm

DDR3

DDR3 SDRAM - double-data-rate three synchronous dynamic random access memory.

Main characteristics:

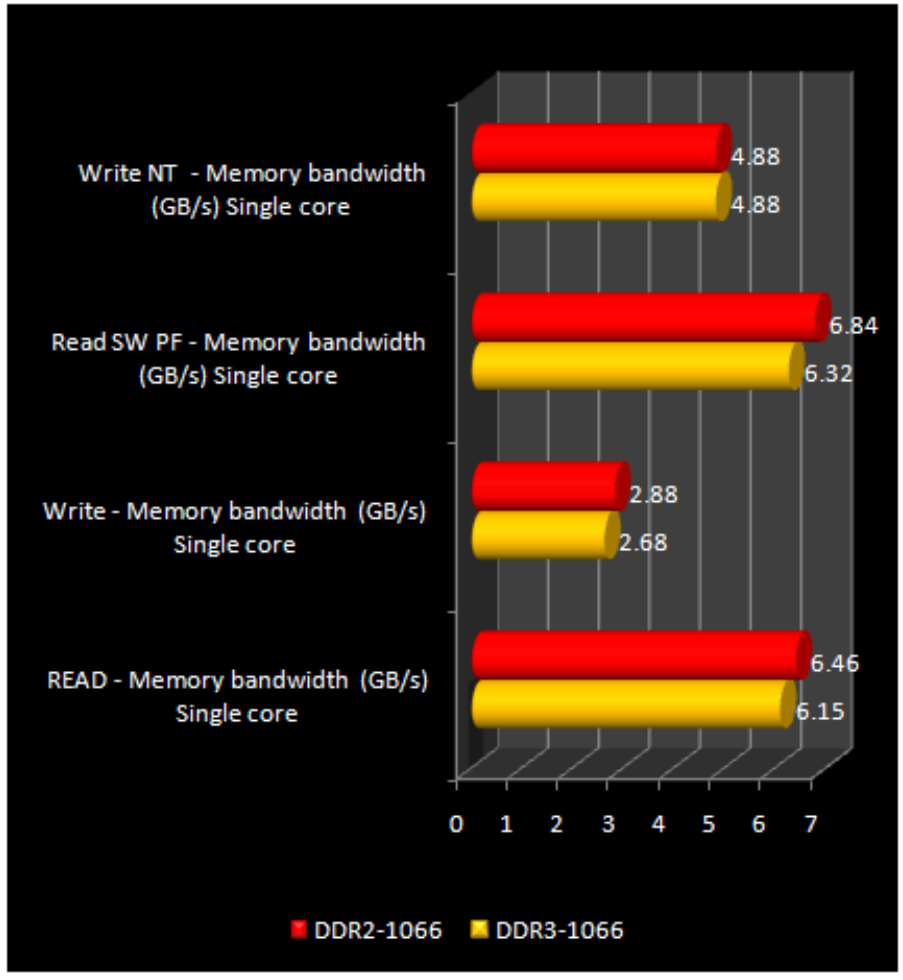
- Throughput is 2 times larger then DDR2 (~ 17000 MB/s)
- 16% less energy consumption:

  DDR4 – 1,2 V; DDR3 - 1.5 V, DDR2 - 1.8 V, DDR - 2.5V.
- Latency is smaller then DDR2

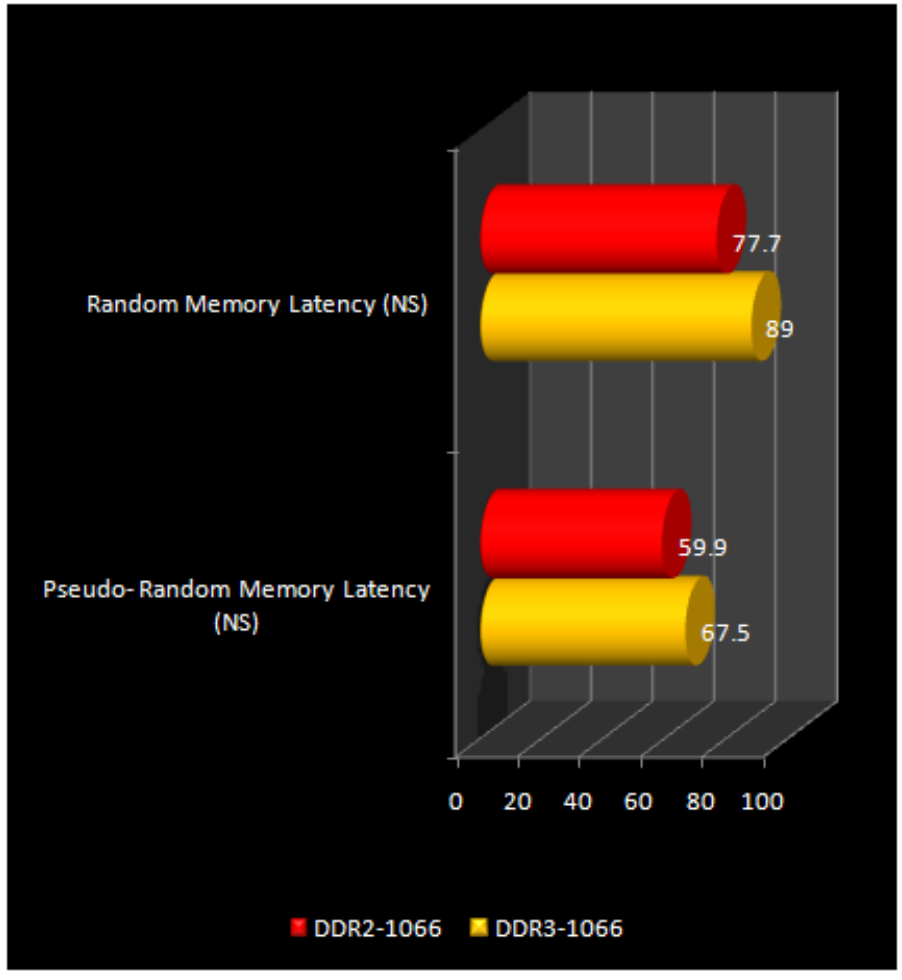
DDR3 characteristics

Module type	Notation	System bus, MHz	Frequency, MHz	Latency	Throughput, GB/s	
					1 channel	2 channels
DDR 3-800	PC 3-6400	400	100	6-6-6-18	6.40	12.80
DDR 3-1066	PC 3-8500	533	133	7-7-7-21	8.53	17.07
DDR 3-1333	PC 3-10667	667	166	8-8-8-24	10.67	21.33
DDR 3-1600	PC 3-12800	800	200	9-9-9-27	12.80	25.60
DDR 3-1866	PC 3-14900	933	233	10-10-10-30	14.93	29.87

DDR2 vs DDR3

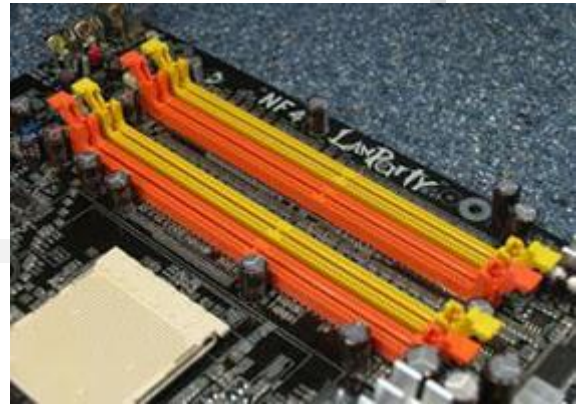


Throughput benchmark (1 channel)



Latency (DDR2 5-5-5, DDR3 7-7-7)

Memory channels



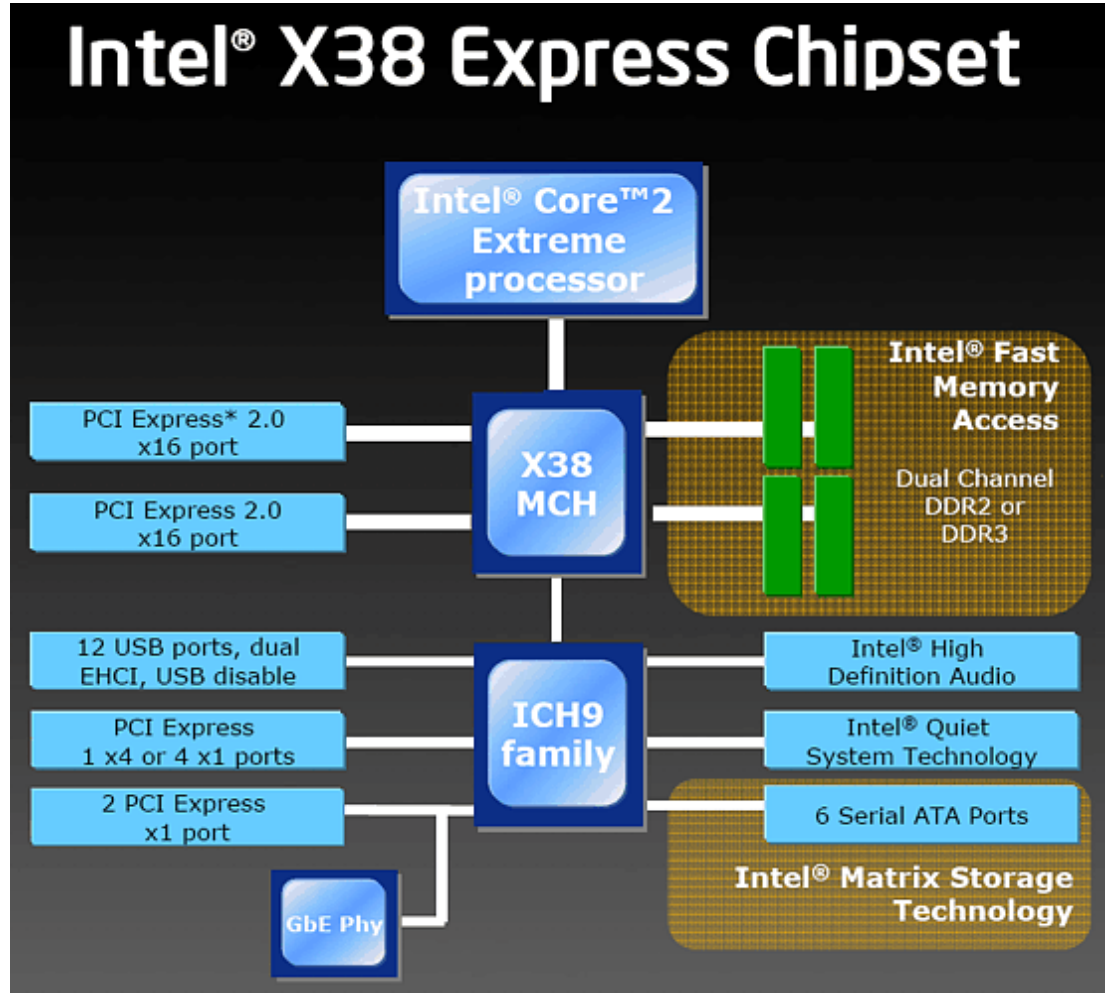
Multi-channel memory architecture is a technology that increases the data transfer rate between the DRAM memory and the memory controller by adding more channels of communication between them.

Theoretically this multiplies the data rate by exactly the number of channels present. Dual-channel memory employs two channels.

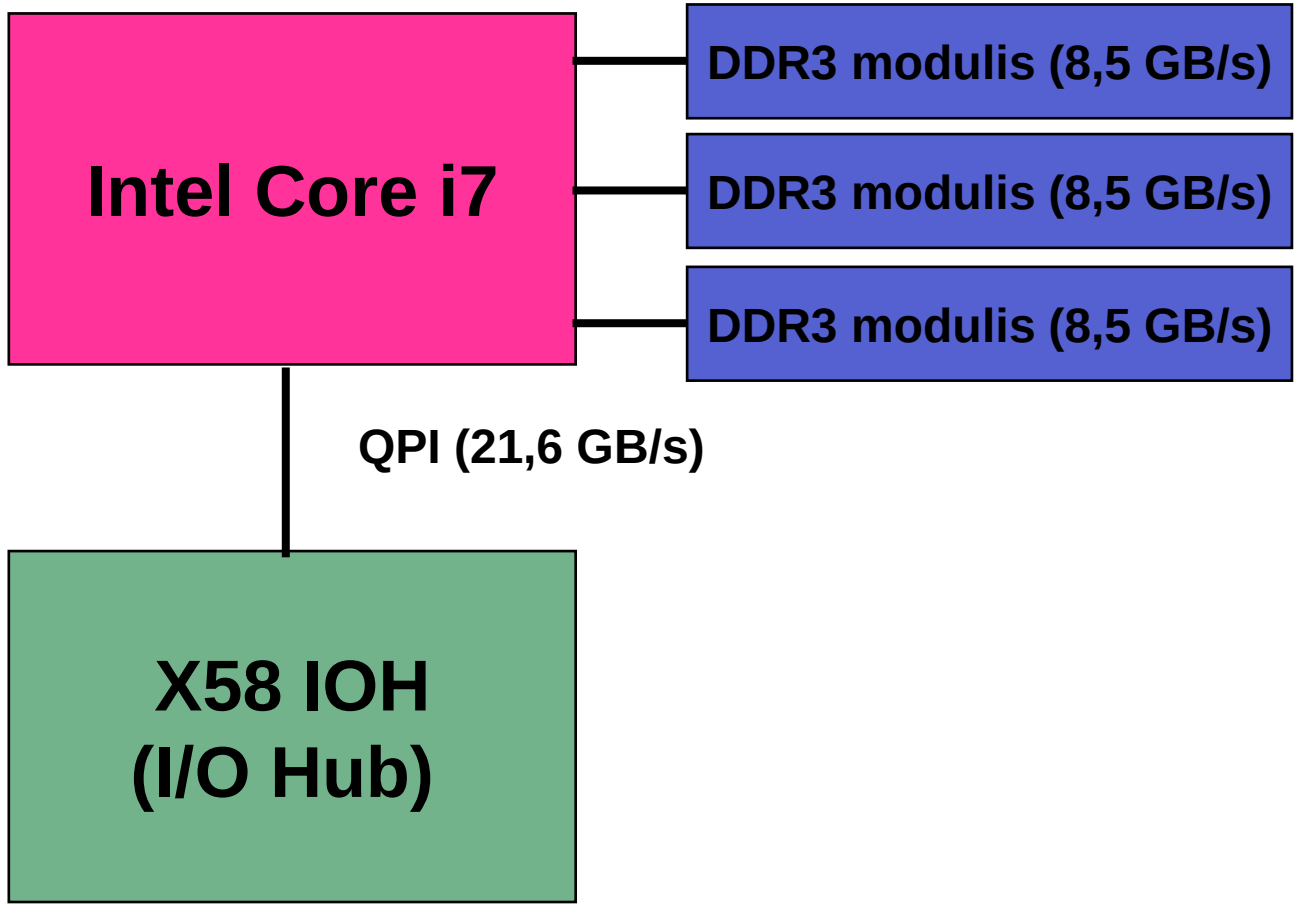
The technique goes back as far as the 1960s having been used in IBM System/360.

Modern high-end processors like the Intel i7 Extreme series and various Xeons support quad-channel memory.

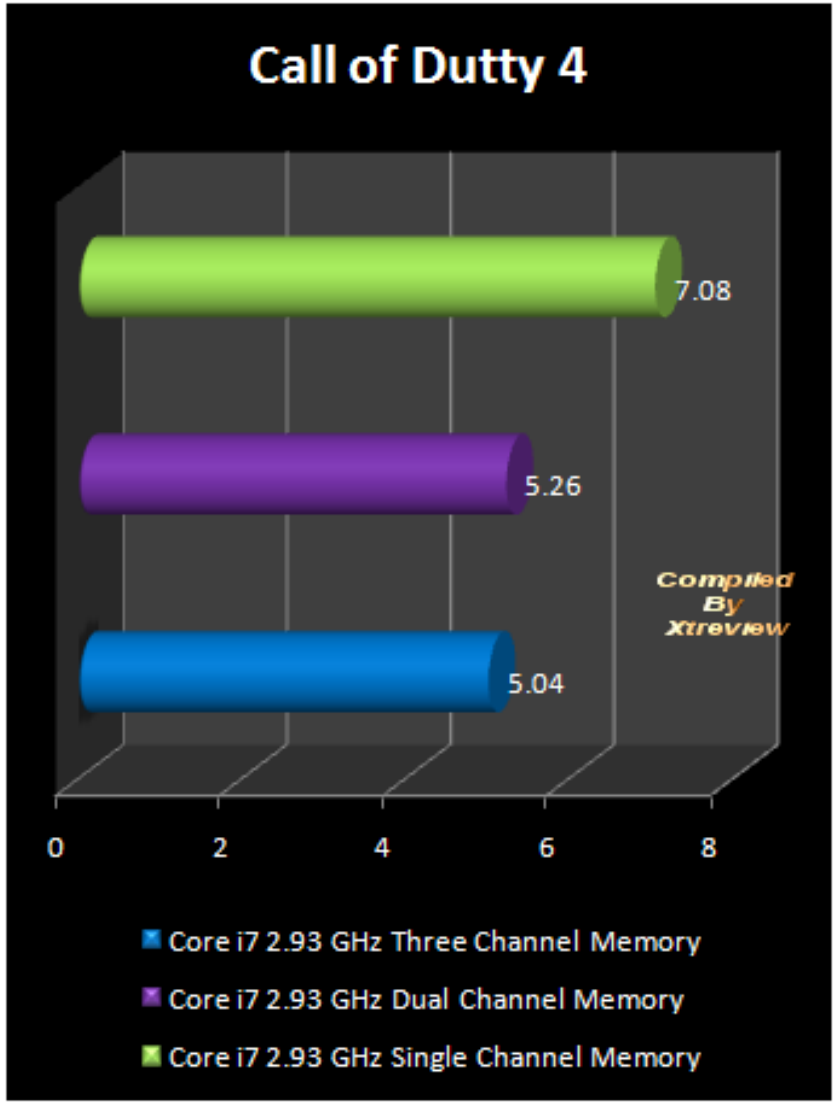
Two channels memory

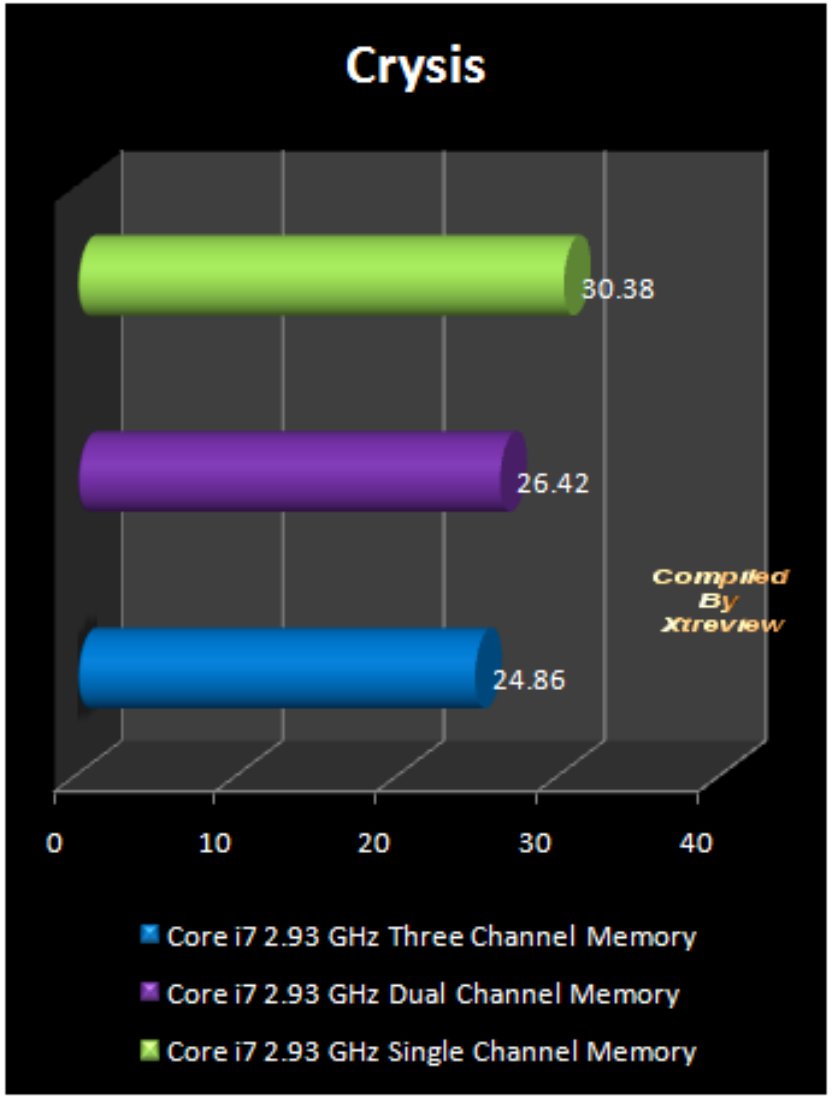


3 channels memory



Performance (3,2,1 channels)





Performance

Test	Dual Channel DDR3-1066 (9-9-9-20)	Triple Channel DDR3-1066 (9-9-9-20)
Memory Tests - Everest v1547		
Read Bandwidth	12859 MB/s	**13423 MB/s**
Write Bandwidth	**12410 MB/s**	12401 MB/s
Copy Bandwidth	16474 MB/s	**18074 MB/s**
Latency	**37.2 ns**	44.2 ns
Cinebench R10 (Multi-threaded test)	**18499**	18458
WinRAR 3.80 - 602MB Folder	118 seconds	**117 seconds**
PCMark Vantage	7438	**7490**
SuperPI - 32M (mins:seconds)	11:55	**11:52**
Far Cry 2 - Ranch Medium (1680 x 1050)	62.1 fps	**62.4 fps**
Company of Heroes - 1680 x 1050	**136.6 fps**	133.6 fps

Performance

Test	*DDR3-1066 (9-9-9-20)*	*DDR3-1333 (9-9-9-20)*	*DDR3-1600 (9-9-9-24)*
Memory Tests - Everest v1547			
Read Bandwidth	13423 MB/s	14127 MB/s	**17374 MB/s**
Write Bandwidth	12401 MB/s	12404 MB/s	**14169 MB/s**
Copy Bandwidth	18074 MB/s	16953 MB/s	**19447 MB/s**
Latency	44.2 ns	38.8 ns	**33.5 ns**
WinRAR 3.80 - 602MB Folder	117 seconds	111 seconds	**106 seconds**
PCMark Vantage	7490	7569	**8102**
SuperPI - 32M (mins:seconds)	11:52	11:36	**11:25**
Far Cry 2 - Ranch Medium (1680 x 1050)	62.4 fps	62.5 fps	**62.7 fps**
Company of Heroes - 1680 x 1050	133.6 fps	135.8 fps	**136.8 fps**

Error correction

A semiconductor memory system is subject to errors. These can be categorized as **hard failures and soft errors**.

A **hard failure** is a permanent physical defect so that the memory cell or cells affected cannot reliably store data but become stuck at 0 or 1 or switch erratically between 0 and 1. Hard errors can be caused by harsh environmental abuse, manufacturing defects, and wear.

A **soft error** is a random, nondestructive event that alters the contents of one or more memory cells without damaging the memory. Soft errors can be caused by power supply problems or alpha particles. These particles result from radioactive decay and are distressingly common because radioactive nuclei are found in small quantities in nearly all materials.

Error correction

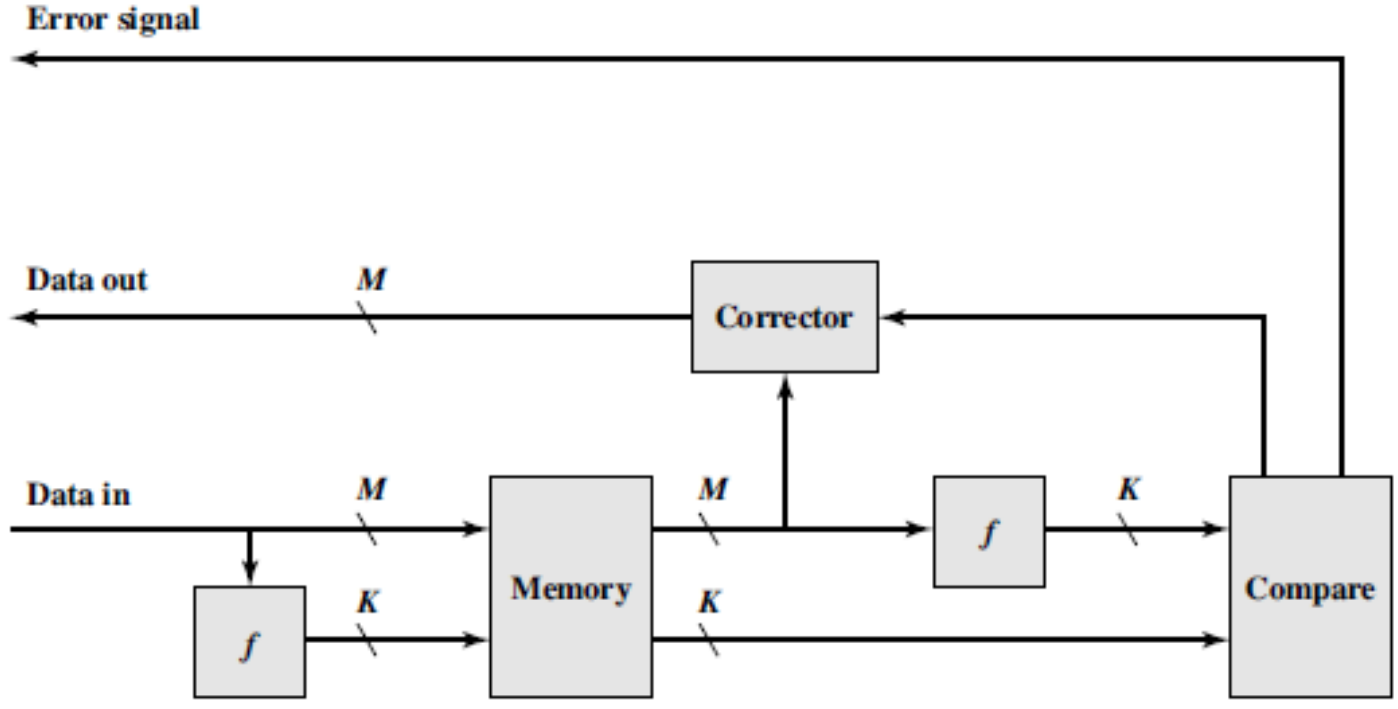
Both hard and soft errors are clearly undesirable, and most modern main memory systems include logic for both detecting and correcting errors.

Error correction algorithm

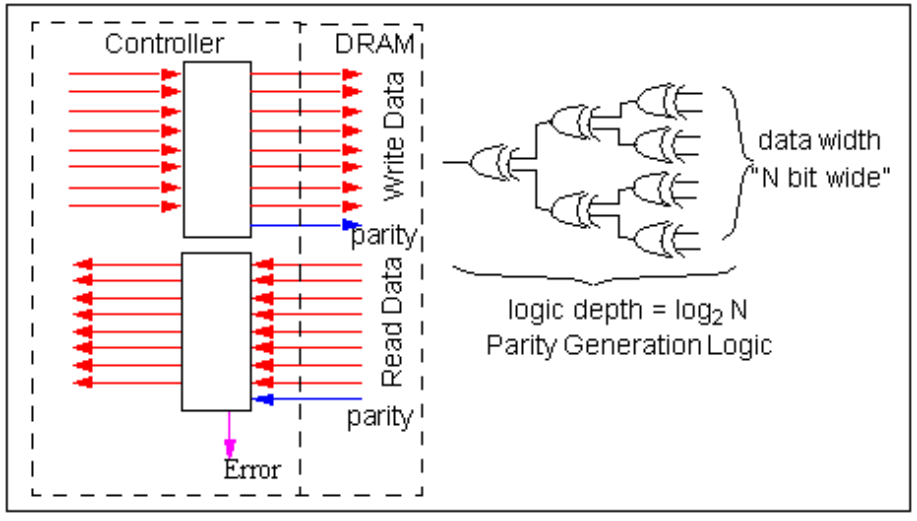
When data are to be read into memory, a calculation, depicted as a function **f**, is performed on the data to produce a code. Both the code and the data are stored. Thus, if an **M**-bit word of data is to be stored and the code is of length **K** bits, then the actual size of the stored word is **M** +**K** bits.

When the previously stored word is read out, the code is used to detect and possibly correct errors. A new set of **K** code bits is generated from the **M** data bits and compared with the fetched code bits.

Error correction



SDRAM parity



7 bits for data	Parity bit	
	Even	Odd
0000000	**00000000**	**10000000**
1010001	**11010001**	**01010001**
1101001	**01101001**	**11101001**
1111111	**11111111**	**01111111**

One attempt to provide minimal protection for data stored in DRAM is the use of parity memory. The intent of parity memory is to provide single bit error detection capability.

SDRAM parity

As seen in Figure, parity checking mechanism works by generation of a parity bit for the data to be written to DRAM. In a parity checking scheme, odd or even parity is generated from a recursive application of the **exclusive-or function** to all of the bits in the input vector. The parity bit is then stored into the DRAM memory system along with the input data vector. When the data vector along with the previously computed parity bit is read out from the DRAM system, the parity bit is re-computed from the data bit vector and compared to the retrieved parity bit. If the re-computed parity bit differs from the retrieved parity bit, an odd number of bits must have changed states, and a parity checking error is generated and reported to the system.

The disadvantage of the parity checking mechanism is that once an error has been detected, the mechanism cannot locate and correct the error. As a result of the disadvantage of the inability to correct errors, combined with the general trend toward wider data bus interfaces, modern memory systems has tended to migrate away from parity memory to some form of **ECC protection**.

Hamming code (ECC)

Codes that operate to detect and correct memory errors are referred to as *error-correcting codes.*

Consider a message having four data bits (D) which is to be transmitted as a 7-bit codeword by adding three error control bits. This would be called a (7,4) code. The three bits to be added are three EVEN Parity bits (P), where the parity of each is computed on different subsets of the message bits as shown below.

7	6	5	4	3	2	1	
D	D	D	P	D	P	P	4-Bits word
D	-	D	-	D	-	**P**	(Even parity bit)
D	D	-	-	D	**P**	-	(Even parity bit)
D	D	D	**P**	-	-	-	(Even parity bit)

Hamming code (ECC)

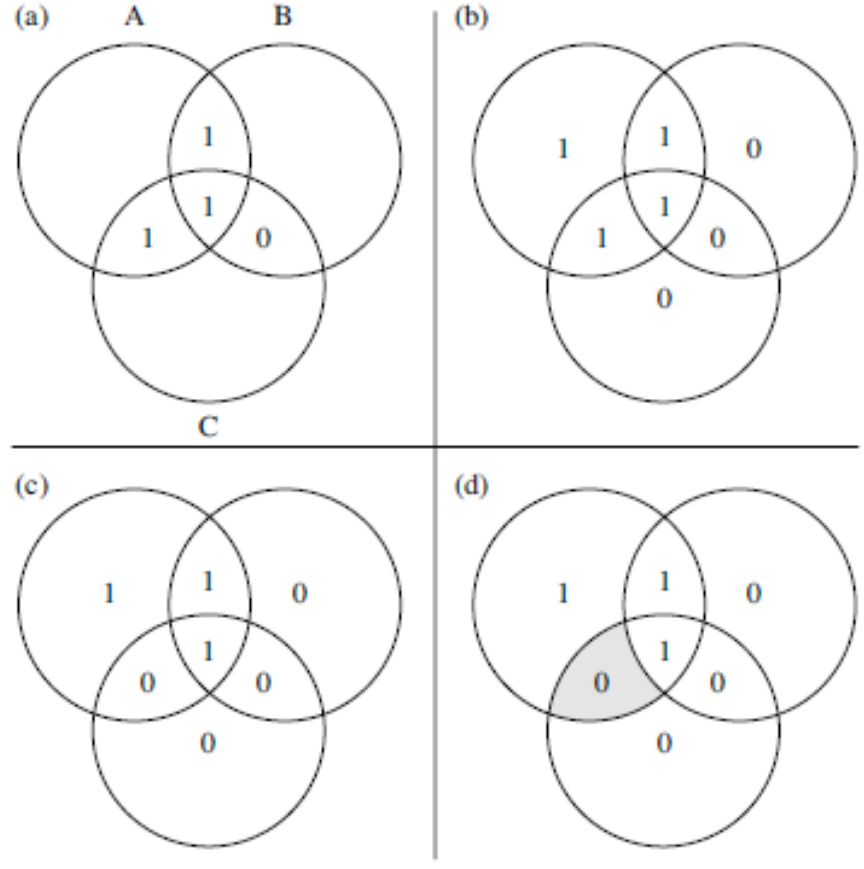
The simplest of the error-correcting codes is the Hamming code devised by Richard Hamming at Bell Laboratories.

Diagram (next slide) illustrates the use of this code on 4-bit words

(M= 4). With three intersecting circles, there are seven compartments. We assign the 4 data bits to the inner compartments The remaining compartments are filled with what are called parity bits. Each parity bit is chosen so that the total number of 1s in its circle is even. Thus, because circle A includes three data 1s, the parity bit in that circle is set to 1. Now, if an error changes one of the data bits, it is easily found.

By checking the parity bits, discrepancies are found in circle A and circle C but not in circle B. Only one of the seven compartments is in A and C but not B. The error can therefore be corrected by changing that bit.

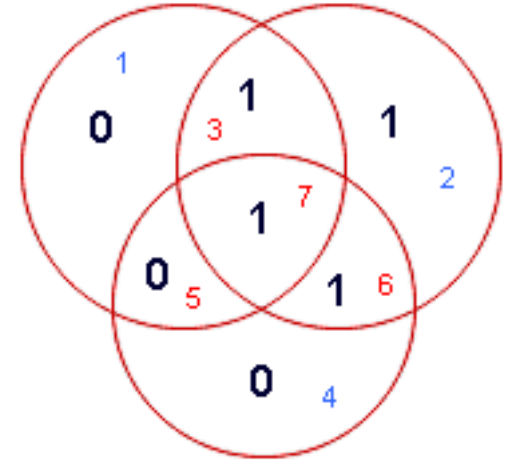
Hamming metodas (ECC)



Hamming code (ECC)

Example: data are as follows **1101**, packet is **1100110**

7	6	5	4	3	2	1	
1	1	0	0	1	1	0	4 data bits
1	-	0	-	1	-	0	Even parity bit
1	1	-	-	1	1	-	Even parity bit
1	1	0	0	-	-	-	Even parity bit



Consider packet **1100110** has error. It means that packet contains the following bits **1110110**. Using Hamming code error will be detected and corrected.

7	6	5	4	3	2	1			
1	1	1	0	1	1	0	4 data bits		
1	-	***1***	-	1	-	0	Even parity bit	**NOT!**	1
1	1	-	-	1	1	-	Even parity bit	**OK!**	0
1	1	***1***	0	-	-	-	Even parity bit	**NOT!**	1