

RAM

Random Access Memory is called random-access because it takes the same amount of time access any one memory location.

Magnetic-core Memory

This was the predominant form of random-access memory from about 1955 to 1975, consisting of magnetic cores or rings that could have one of two possible orientations. One main problem is that reading a value erases it.

Modern Memory

Modern memory has the same configuration as magnetic core memory, but the rings have been replaced with electronic memory cells such as the D-latch. These D-latches are fast but take up a lot of space.

Each row represents a memory location where a binary value is stored. The columns represent different bits of those memory locations.

Memory is like a matrix where the number of rows identifies the number of memory locations and the number of columns identifies the number of bits in each memory location.

To store or retrieve data from memory, the processor places a binary number called an address on special inputs to the memory device.

The address identifies which row of the memory matrix or array the processor is interested in communicating with, and enables it. The memory cells in that row are connected to bi-directional connections (called **data lines**) that allow data to be written to or read from the cells.

Three additional lines (**chip select**, **read enable** and **write enable**) are used to control the transaction:

- When read enable is zero, we are reading data from memory.
- When write enable is zero, we are writing data to memory.
- They should never both be zero.
- When reading the data lines connect to the Q inputs of the latches, and when writing they connect to the D inputs.
- If the chip select is zero, the memory activates all of its input and output lines.

The processor (the arbiter) uses digital logic to control the devices so that only one is talking or listening at any one time. The arbiter decides who gets the bus.

DRAM

Unlike SRAM, which functions as a D-flip-flop like we have described, DRAM is a transistor and capacitor paired to create a **memory cell**.

- The capacitor holds the bit of information by holding a tiny amount of charge.
- The transistor acts as a switch that lets the control circuitry on the memory chip read the capacitor or change its state.

Since the capacitors are so small, they discharge very rapidly (e.g. every couple of millisecond) once they're charged. To combat this, there's a piece of logic that detects whether there's a charge on the capacitor, and refreshes the charge if it's there. The memory controller reads the memory and then writes it back.

This needs to be done continuously, and is the reason for "dynamic" in the name. This happens **thousands of times per second per bit**. This is where the power in the machine goes.

Error-checking and Parity

Memory is very reliable, and most systems have the memory controller check for errors at start-up and rely on that.

Parity chips have an extra bit for every 8 bits of data.

Even and Odd Parity

The memory chip adds up the total number of 1s in each byte written to the memory. For **Even Parity**:

- If the total number of 1s is odd, the parity bit is set to 1
- If the total is even, it's set to 0.
- When the data is read from the memory, the process is repeated and checked against the parity bit.

Odd Parity works the same way, but sets the parity bit the other way around.

Parity assumes that it's unlikely more than 1 bit will go wrong. It cannot detect if two bits have gone wrong.