Memory

Memory is the third key component of a microprocessor-based system (besides the CPU and I/O devices). More specifically, the primary storage directly addressed by the CPU is referred to as *main memory* to distinguish it from other “memory” structures such as CPU registers, caches, and disk drives. Main memory is typically built from several discrete semiconductor memory devices.

Most systems contain two or more types of main memory. All memory types can be categorized as ROM or RAM, and as volatile or non-volatile:

- Read-Only Memory (**ROM**) cannot be modified (written), as the name implies. A ROM chip’s contents are set before the chip is placed in the system.
- Read-Write Memory is referred to as **RAM** (for Random-Access Memory). This distinction is inaccurate, since ROMs are also random access, but we are stuck with it for historical reasons.
- **Volatile** memories lose their contents when their power is turned off.
- **Non-volatile** memories do not.

The memory types currently in common usage are:

	ROM	RAM
Volatile	(nothing)	Static RAM (SRAM) Dynamic RAM (DRAM)
Non-volatile	Mask ROM PROM EPROM	EEPROM Flash memory BBSRAM

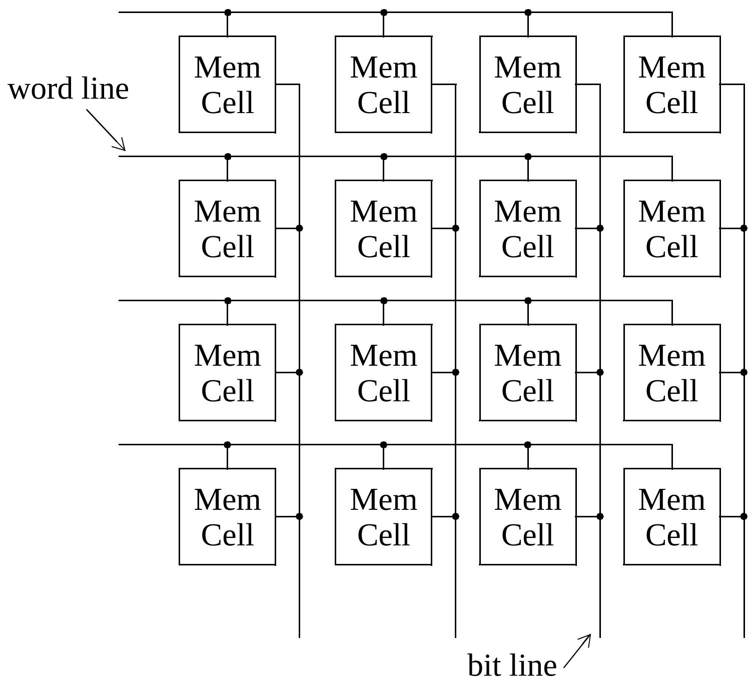
Every system requires some non-volatile memory to store the instructions that get executed when the system is powered up (the boot code) as well as some (typically volatile) RAM to store program state while the system is running.

For a very low-cost system, you can buy microprocessors that include both volatile and non-volatile memory integrated on the same chip (along with some I/O devices as on the MPC823). All you need to add are power and maybe a clock crystal. These “systems-on-a-chip” are called *microcontrollers*. The Xilinx download process on the lab expansion boards is run by an 8-bit microcontroller containing 256 bytes of SRAM and 2K bytes of Flash memory.

Inside a Memory Device: the Memory Array

The heart of a memory device is the *memory array*, which is where the bit values are actually stored. Each row of memory cells is controlled by a *word line*. When a specific row is activated via its word line, the data values of that row's cells can be read or written via the *bit lines*.

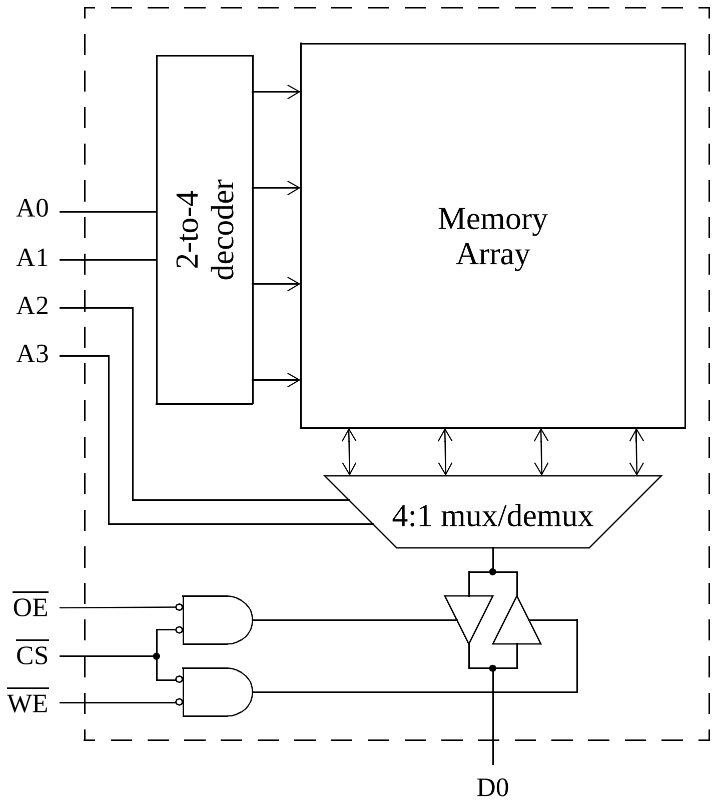
This diagram shows a 16-bit memory organized as a 4x4 array:



The different types of memory are basically distinguished by how each memory cell stores its bit value.

Inside a Memory Device: Support Circuitry

Support circuitry around the memory array provides the external device interface. The following (simplified) diagram shows how our 4x4 16-bit memory array might be packaged.



The support circuitry has two functions:

- Maps the internal physical memory array to the external configuration (4x4 to 16x1, in this example).
- Provides control signals for reading and writing the array.

Generic Memory Device Interface

Internal vs. external configuration:

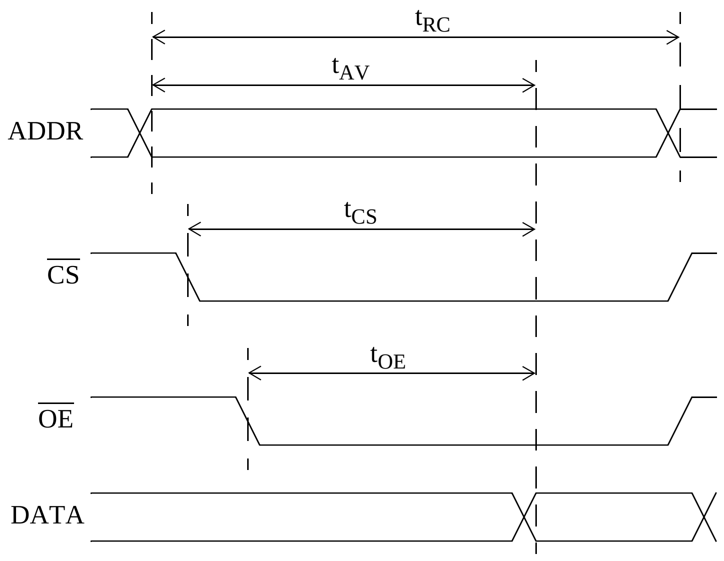
- Physical memory arrays are typically close to square. A square memory array equalizes the row and column wire delays, providing the shortest access time.
- External configurations are always tall and narrow. Adding one address line halves the number of data lines, so narrow external configurations (x1 or x4) are most pin-efficient.
- Several configurations are typically available at a given capacity. For example, devices with 64K bits of storage may be available as 64Kx1, 8Kx8, 4Kx16, etc.

Control signals:

- The chip select signal ($\overline{CS}$) enables the device; if $\overline{CS}$ is not asserted, the device will ignore all other inputs. Deasserting $\overline{CS}$ may put the device in a low-power mode. This signal is sometimes called chip enable ($\overline{CE}$).
- Output enable ($\overline{OE}$) causes the chip to drive the value stored at the specified address onto the data pins (i.e., do a read).
- Write enable ($\overline{WE}$) causes the chip to store the value from the data pins at the specified address. ROM devices do not have $\overline{WE}$ inputs (obviously).
- Most devices have bidirectional data pins (D) that are tristated unless $\overline{OE}$ is asserted. Some devices have separate data-in (DI) and data-out (DO) pins.

Memory Timing: Reads

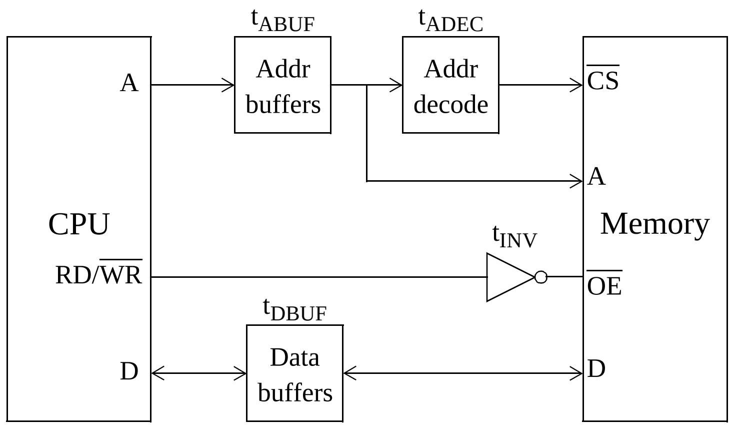
- The time required from the start of a read access to valid data output is a memory device's *access time*.
- An access time is specified for each of the three conditions required for valid data out (address valid, $\overline{CS}$ asserted, $\overline{OE}$ asserted). Let's call these t_{AV}, t_{CS}, and t_{OE}. The time to valid data out will depend on which of these is the critical path.



- Typically, $t_{AV} = t_{CS}$, but t_{OE} is much smaller. (Why?)
- The read cycle time (t_{RC}) is the minimum time required from the start of one access to the start of the next. For most memories, the cycle time equals the access time, but for DRAMs it may be as much as twice the access time.

System-level Read Timing

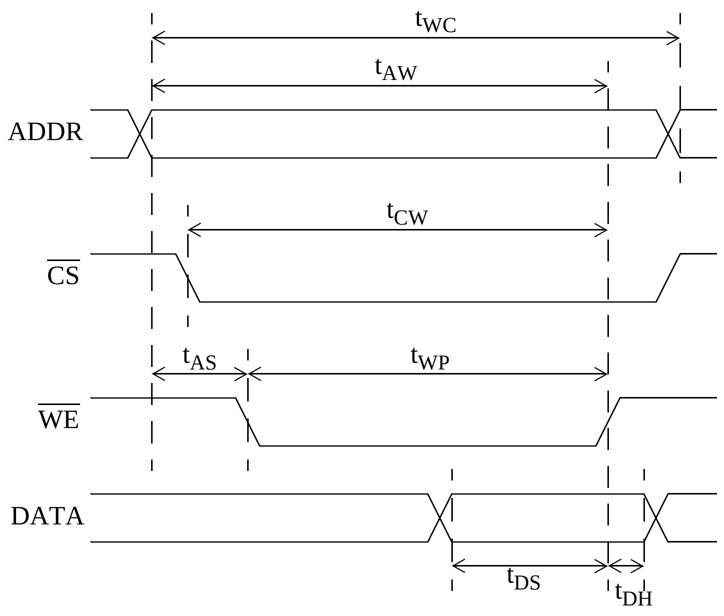
To determine the total access time from the CPU's perspective, delays through all of the glue logic must be considered:



If the CPU starts driving the address and RD/WR at the same time, what is the effective memory access time?

Memory Timing: Writes

Write timing is a little more complicated:

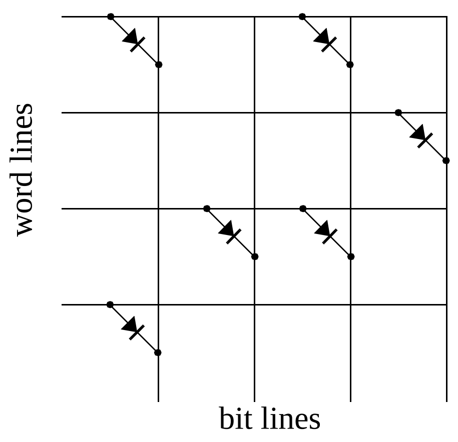


- The write happens on the *rising* edge of $\overline{WE}$. As with reads, a separate "access time" is specified for address valid, CS asserted, and $\overline{WE}$ asserted: t_{AW}, t_{CW}, and t_{WP}. (t_{WP} is the "write pulse" width.)
- Typically t_{AS} is 0, meaning only that WE may *not* be asserted *before* the address is valid.
- Note the required data setup and hold times.
- The write cycle time t_{WC} is typically not much larger than t_{AW}.

Memory Types

The key distinction between the various types of memory is the means by which each memory cell stores a bit value.

- By far the simplest technique uses the presence or absence of a diode at each cell to store a value, as shown here:



  - What value is stored by the presence of a diode?
  - Why is a diode required rather than a straight connection (i.e. a wire)?

- This type of memory is known as a **mask ROM**, since the pattern of diodes is defined by a mask used in the semiconductor fabrication process. Thus the memory contents are fixed when the chip is made and cannot be changed.

ROM Types

- Mask ROM characteristics:
  - Large setup charge to get initial mask designed
  - Small marginal cost per part
  - Ready to go off production line
  - Only used in high-volume applications (many thousands or millions of units) where upgrading the contents is not an issue (e.g., consumer electronics). (Used to be common for PC boot code, but obsoleted due to cost & density improvements in flash memories.)

Programmable ROM (PROM)

- Replace diode with diode + fuse, put one at every cell (a.k.a. “fusible-link” PROM)
- Initial contents all 1s; users program by blowing fuses to create 0s
- Plug chip into PROM programmer (“burner”) device, download data file
- One-time programmable (OTP), bug $\rightarrow$ throw it away

ROMs cont'd

UV Erasable PROM (UV EPROM, or just EPROM)

- Replace PROM fuse with pass transistor controlled by "floating" (electrically isolated) gate
- Program by charging gate; switches pass transistor
  - Careful application of high voltages to overcome gate insulation (again using special "burner")
- Erase by discharging all gates using ultraviolet light
  - UV photons carry electrons across insulation
  - Chip has window to let light in
- Insulation eventually breaks down $\rightarrow$ limited number of erase/reprogram cycles (100s/1000s)
- Biggest cost: special package with window
- UV EPROM w/o window is an OTP device, equivalent to (and often cheaper than) fusible-link PROM
- Again, largely displaced by flash memory

Non-volatile RAM Types

Three basic types: EEPROM, Flash, BBSRAM

Electrically Erasable PROM (EEPROM, E^{2}PROM)

- Similar to UV EPROM, but with on-chip circuitry to electrically charge/discharge floating gates (no UV needed)
- Writable by CPU $\rightarrow$ it's RAM, not ROM (despite name)
- Reads & writes much like generic RAM
  - on writes, internal circuitry transparently erases affected byte/word, then reprograms to new value
- Write cycle time on the order of a millisecond
  - typically poll status pin to know when write is done
- High-voltage input (e.g. 12V) often required for writing
- Limited number of write cycles (e.g. 10,000)
- selective erasing requires extra circuitry (additional transistor) *per memory cell* $\rightarrow$ lower density, higher cost than EPROM

Flash Memory

- Again, floating-gate technology like EPROM, EEPROM
- Electrically erasable like EEPROM, but only in large 8K-128K blocks (not a byte at a time)
    - Moves erase circuitry out of cells to periphery of memory array
    - Back to one transistor/cell $\rightarrow$ excellent density
- Reads just like memory
- Writes like memory *for locations in erased blocks*
    - typ. write cycle time is a few microseconds
    - slower than volatile RAM, but faster than EEPROM
- To rewrite a location, software must explicitly erase entire block
    - initiated via control registers on flash memory device
    - erase can take several *seconds*
    - erased blocks can be written (programmed) a byte at a time
- Still have erase/reprogram cycle limit (10K-100K cycles per block)

Flash Applications

- Flash technology has made rapid advances in last few years
    - cell density rivals DRAM; better than EPROM, much better than EEPROM
    - multiple gate voltage levels can encode 2 bits per cell
    - 64 Mbit devices available
- ROMs & EPROMs rapidly becoming obsolete
    - as cheap or cheaper, allows field upgrades
- Replacing hard disks in some applications
    - smaller, lighter, faster
    - more reliable (no moving parts)
    - cost-effective up to tens of megabytes
    - block erase good match for file-system type interface

Battery-backed Static RAM (BBSRAM)

- Take volatile static RAM device and add battery backup
- Key advantage: write performance
    - write cycle time same as read cycle time
- Need circuitry to switch to battery on power-off
- Have to worry about battery running out
- Effective for small amount of storage when you need battery anyway (e.g. PC built-in clock)

Volatile RAM Types

Two basic types: static and dynamic

Static RAM (SRAM)

- Each cell is basically a flip-flop
- Four or six transistors (4T/6T) $\rightarrow$ relatively poor density
- Very simple interfacing; writes & reads at same speed
- Very fast (access times under 10 ns available)

Dynamic RAM (DRAM)

- One transistor per cell (drain acts as capacitor)
- Highest density memory available
- Very small charges involved
    - bit lines must be *precharged* to detect bit values: cycle time > access time
    - reads are *destructive*; internally does writeback on read
    - values must be *refreshed* (rewritten) periodically by touching each row of array or charge will leak away
- External row/column addressing saves pins, $
- Row/column addressing + refresh $\rightarrow$ complex interfacing