How Memory Works

- Static RAM operation and timing
- Implementation of SRAM array
- Construction of larger RAMs from smaller ones
- Dynamic RAM implementation

Jon Turner
Random Access Memory

- Logically, a *random access memory* contains an array of numbered storage locations, called *words*
  - when read/write’ is high, data_out is equal to the value stored in word specified by address inputs
  - when read/write’ is low, the value on data_in replaces the value in word specified by address inputs
  - separate enable signal also usually provided
- Simplest RAMs are *asynchronous* - no clock input
  - circuits using RAMs must make sure that RAM timing requirements are satisfied to ensure correct operation
  - synchronous SRAMs handle timing issues internally
Static RAM Array (4x4)
Timing of Async RAM Operations

- **Read cycle**
  - *access time*: time from “last” address change until output data is valid

- **Write cycle**
  - $t_1$ is min time from address stable and enable asserted until r/w' is lowered
  - $t_2$ is min time that input data must remain stable before r/w' can be raised
  - $t_3$ is min time that address stays valid after r/w' is raised
  - *cycle time* is $t_1 + t_2 + t_3$

- Circuits using RAM must ensure timing conditions are met
Functional Simulation
Building Larger RAMs

- Systems often require larger RAMs than can be constructed using a single SRAM component.
- The use of an external decoder and the enable input allows larger RAMs to be constructed.

**64Kx16 RAM**

Alternative design uses 64Kx4 RAM chips
» no external decoder needed in this case
The SRAM Storage Cell

- Computers and other digital systems generally use large amounts of memory.
- Specialized memory cells have been developed to pack more memory in a given amount of space.
- Typical static RAM uses 6 transistor cell using pair of inverters and pair of pass transistors.
- Bit line asserted to read or write the cell.
- Complementary data lines used for input and output.
- Column drivers enabled when writing.
  - Powerful transistors allow them to force cell to desired state.

![Diagram of SRAM cell with data_in, data_out, bit line, column drivers, and sense amplifier.](image)
Dynamic RAMs

- Dynamic RAMs use simpler memory cell to enable more bits to be stored in a single chip (4-8x)
  - each storage cell consists of a selection transistor and a capacitor
  - reading contents, destroys value
    - need to write back after reading
  - stored charge leaks from capacitor after 10-100 ms
    - requires periodic refresh of memory contents
- DRAM cells are organized in 2D arrays, much like those for SRAM
  - single data line rather than pair
  - requires sensitive sense amplifiers to detect stored charge
  - takes more time (10x) to read values than with SRAM
Addressing in Dynamic RAMs

- Large memory chips require lots of address pins
- Many DRAM chips reduce number of address pins by dividing address into two parts
  - row address determines which row in 2D array is selected
  - column address selects one or more bits in the row
- Column address can be provided after row address without slowing down memory access
  - so, same address pins can be used to supply both row and column addresses
  - Row Address Strobe (RAS), Column Address Strobe (CAS) used to load row and column addresses into on-chip registers
- Refresh circuitry periodically reads each row in memory array and writes it back
  - refresh circuits now commonly built into memory chips
Exercises

1. Consider a consider a 256x8 memory. How many bits does this memory require? What are the dimensions of its storage array? How many address bits does it require? How many of these go to its row decoder and how many to its column decoder? How many words are stored in each row?

2. In the memory described in problem 1, assume that the most-significant address bits are connected to the row decoder and the least-significant bits are connected to the column decoder. Which row in the array contains word number 113 (assume that the rows are numbered from top-to-bottom, starting with 0). Which column within this row contains bit number 3 (assume that the columns are numbered from left-to-right, starting with 0).

3. Suppose, we wanted to implement a memory that could store 64 bits, and could be configured as either an 8x8 memory, of a 16x4 memory. Draw a diagram, showing how you could implement this.

4. Show how you can implement a 32Kx24 bit memory using two 32Kx8 components plus four 8Kx8 components. Show exactly which address bits go to which component and which data bits go to which component. Show any auxiliary decoders you may require.