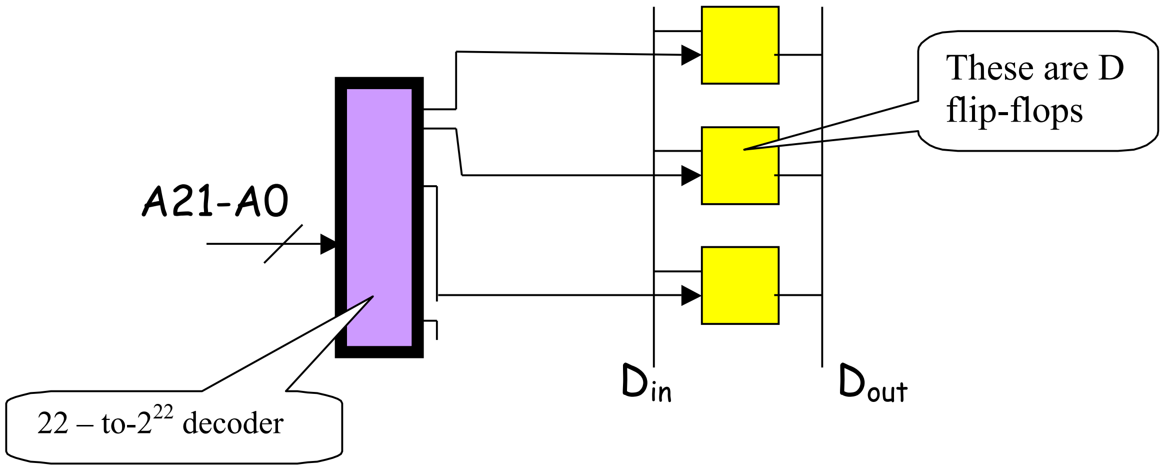
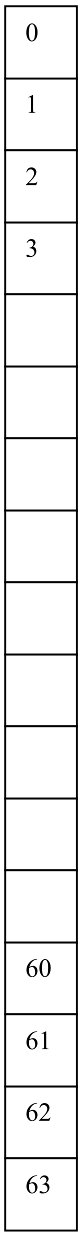
Inside RAM

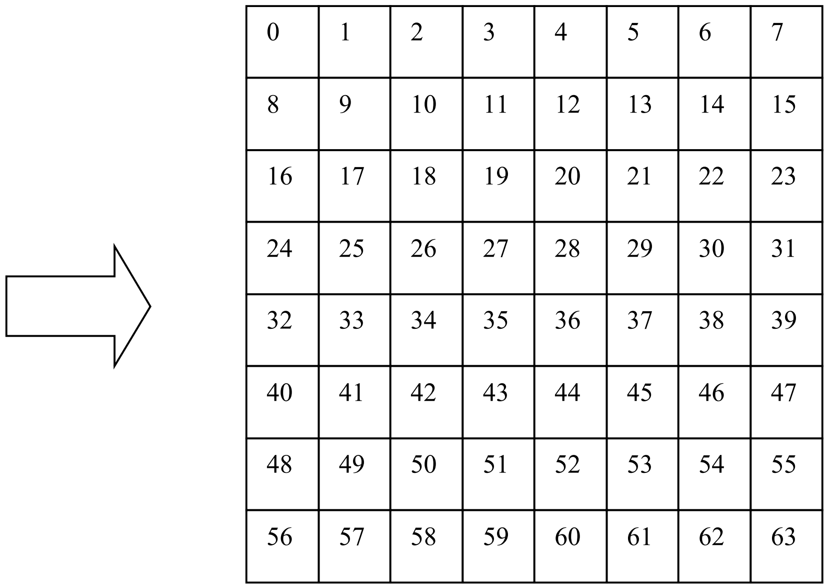
Two-level addressing

Consider a 4M x 1 bit RAM. It will apparently need a 22-to-4M decoder. 4 million output lines will make it complex! Complex! Tw0-level addressing simplifies it.



In two-level addressing, the memory cells for any bit position are logically arranged as a two-dimensional array. 11 address lines are used to select a row, and 11 address lines are used to select a column. This reduces the number of output lines of the decoders from 4M to 2048 + 2048 = 4096 only.





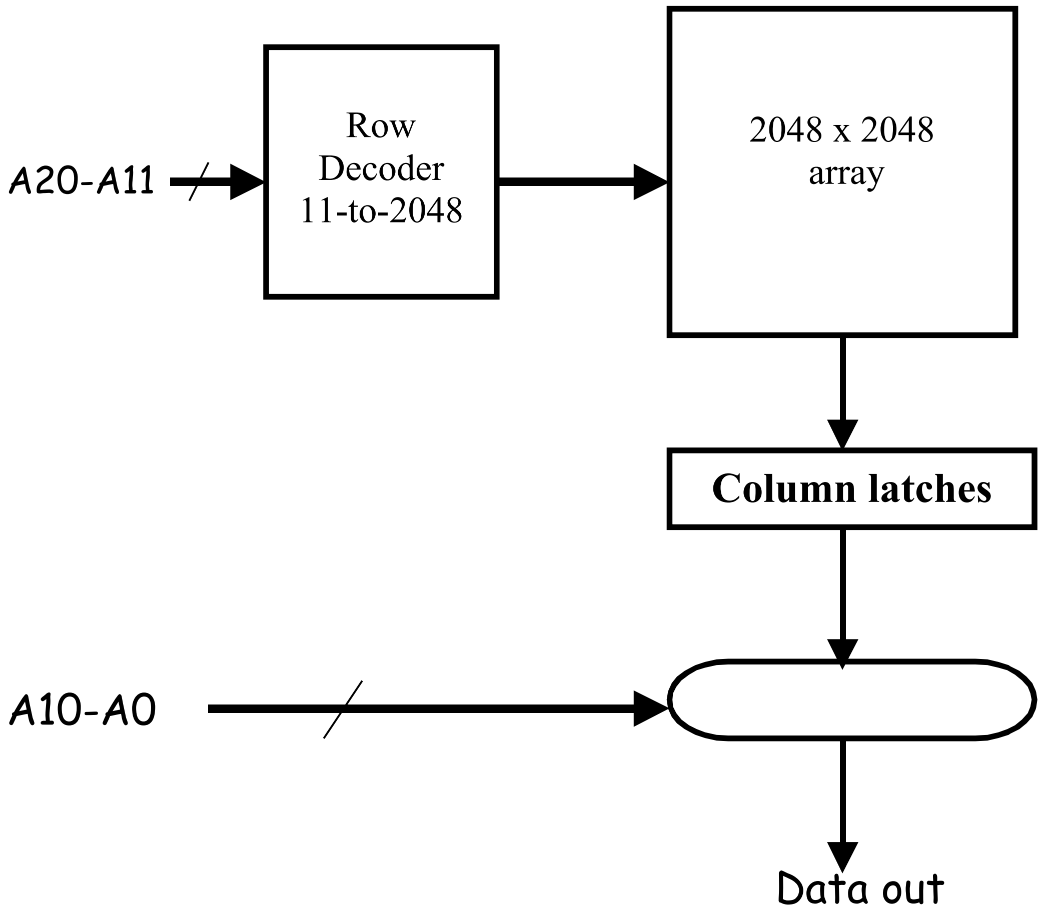
A 3-to-8 decoder for row
A 3-to-8 decoder for the column
(8+8=16 wires lead to the memory)

A 6-to-64 decoder will decode the address

(64 wires lead to the memory cells)

<u>Basic idea</u>

Scheme for two-level addressing

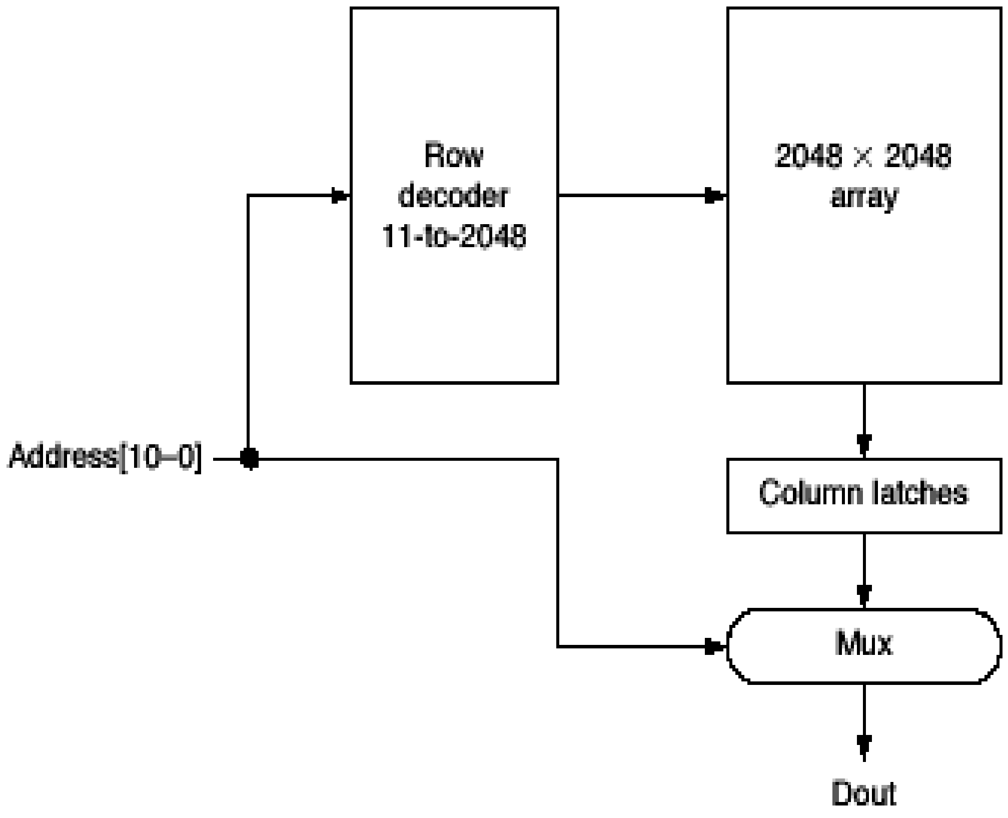


A 4M x 1 bit memory

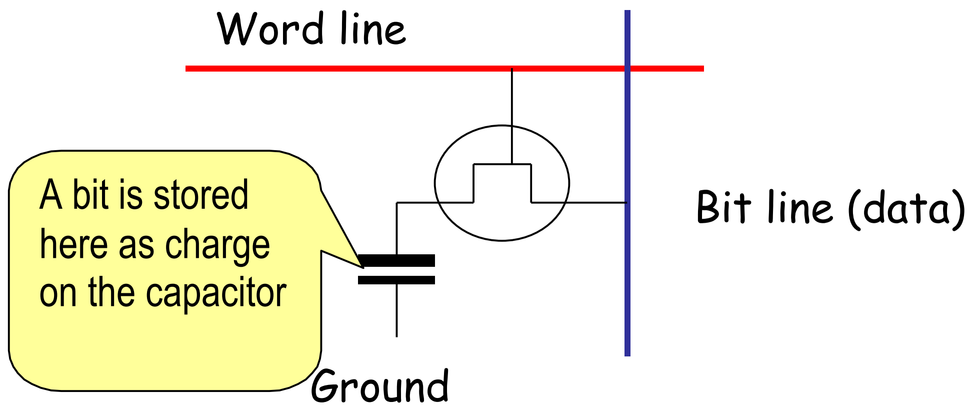
To save pins, in DRAMs, 11 common lines are used for both A20-A11 and A10-A0. A pair of signals (RAS and CAS) notify which part of the address is present at the input lines.

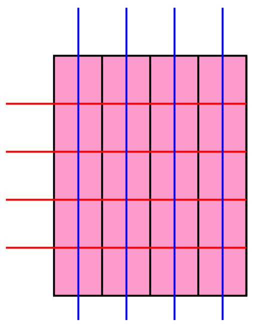
<u>**Modified scheme with shared address lines**</u>

(A21-A11) and (A10-A0) are fed to the address lines one after another



Dynamic RAM



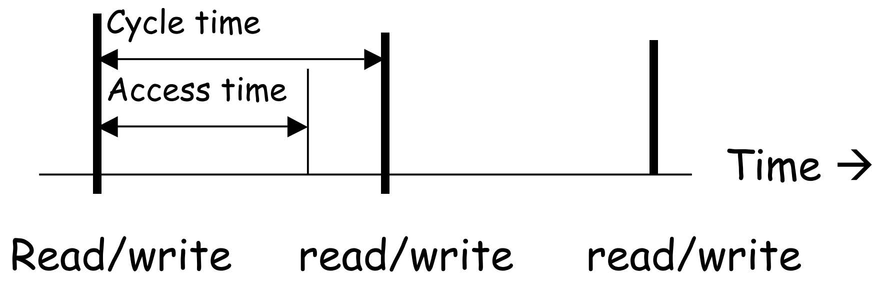


A "1" in the word line closes the switch, and the input data through the bit line charges the capacitor. During reading, the voltage on the capacitor is sensed through the bit line to determine if the content is a "1" or a "0".

To safeguard the leakage of the capacitor, it is periodically refreshed (An entire row in one cycle).

Very high packing density, but slow (45-65 ns access time) DRAM is a good choice for the main memory.

Cycle time



Smallest time between consecutive R/W operations.

Speeding up DRAMs

Page mode DRAM

Row address does not change – only the column address changes. Reduces the access time (25 ns) within a row.

SDRAM (Synchronous DRAM)

Series of bits from a row are automatically transferred in a burst – explicit column addresses are not specified. This saves times and leads to faster memory access.

ROM and PLA and FPGA

ROM = Read Only Memory

PLA = Programmable Logic Array

FPGA = Field Programmable Gate Array

Here a ROM that is a solid-state device (not an electro-mechanical device like a CD-ROM). Its contents can be randomly accessed, very similar to RAM, with the exception that it contains fixed codes/data that cannot be erased or overwritten by the programmer.

Needs special equipment to write data into it. Some can be programmed only once, and often done at the factory. In the programmable variety of ROMs (PROM), the contents can be erased (by ultraviolet radiation or electrical pulses as in EEPROM) and reprogrammed.

Examples

Flash Memory is a form of EEPROM (Electrically Erasable Programmable ROM). Extensively used in USB thumb drives, memory sticks etc. **Smart Media** and **Compact Flash** are used as electronic films in digital cameras.

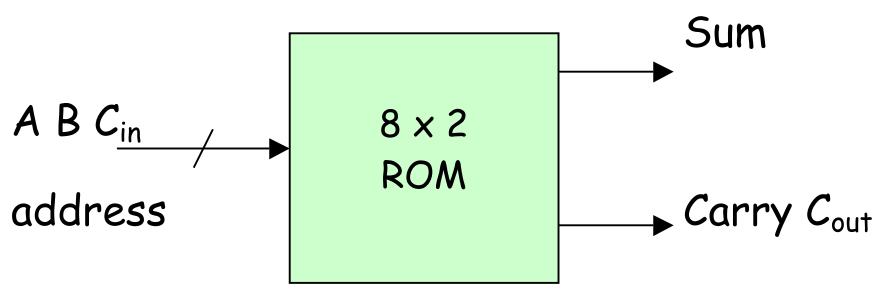
FPGA (Field Programmable Gate Array)

FPGA is a semiconductor device containing programmable logic components and programmable interconnects. Logic blocks can be programmed to perform the function of basic logic gates such as AND, and XOR, or more complex combinational functions such as decoders or mathematical functions. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. It is a useful building block for prototyping hardware design

Design using ROMs

ROMs can also be used in logic design, and it opens up interesting possibilities. The design of combinational circuits can be reduced to a simple table look-up

Example. Design a full adder using a ROM



The content of the ROM are given in the table below:

Address	Sum	Carry
000	0	0
001	1	0
010	1	0
011	0	1
100	1	0
101	0	1
110	0	1
111	1	1