
CSE 220 Computer Organization

Lecture 24

Hussein Badr

Computer Science, Stony Brook University

<http://www.cs.sunysb.edu/~cse220>

Memory Chips (SRAM)

- A register file consists of a few locations (100 – 500 at most).
- Larger memories are built using SRAM or DRAM chips.
- SRAM: Static Random Access Memory.
- DRAM: Dynamic Random Access Memory.
- Compared to DRAM, SRAMs are fast, but consume more power and require more space for each bit.
- SRAMs do not need refreshing ('nonvolatile' memory).
- A good use of SRAM is Cache.
- As long as power is ON, an SRAM location will hold its value.
- Good for medium size memory.

SRAM Chip (1/3)

- This is a 2M x 16 chip.
- It has 2M locations.
Size of each location is 16 bits.
- In other words,
capacity or size of this
chip is 4 Mbytes.
- Address is 21 bits wide.
- Din [15-0] are 16 bits used while writing.
- Dout [15-0] are 16 bits used when we read.
- Chip select: we read a byte from one of several chips.
Before we read, we select the chip.

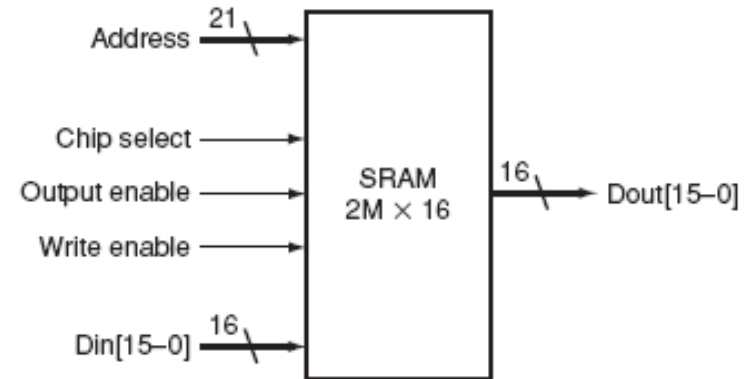
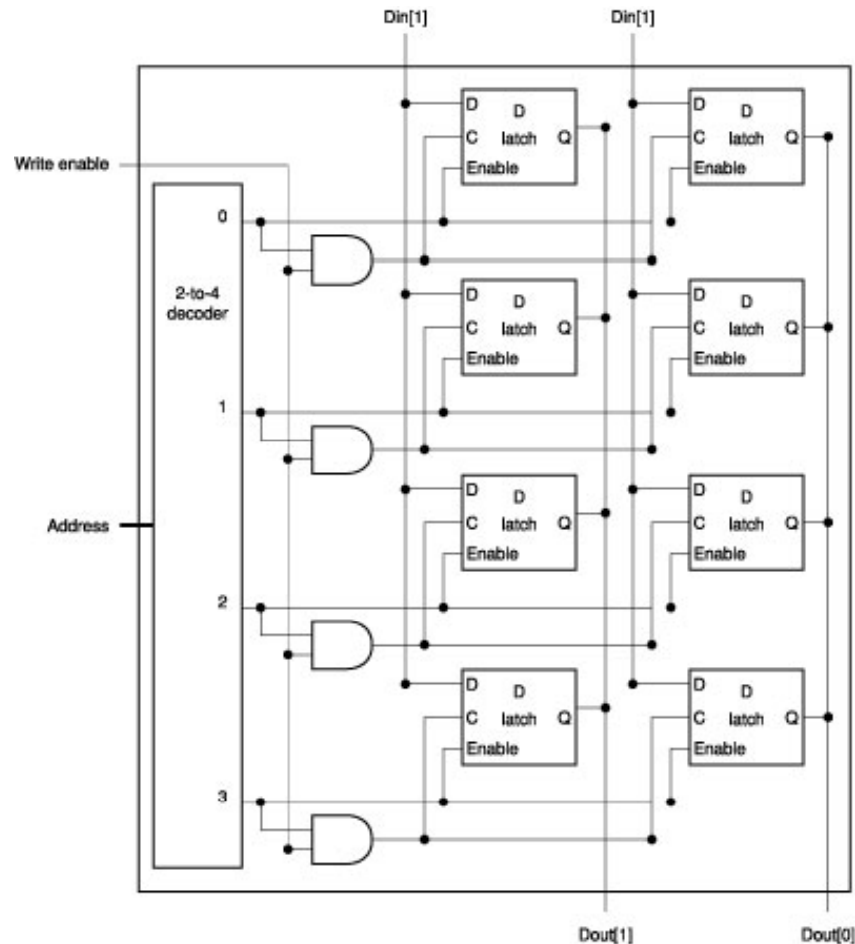


FIGURE C.9.1 A 2M x 16 SRAM showing the twenty-one address lines ($2M = 2^{21}$) and sixteen data inputs, the three control lines, and the sixteen data outputs.

SRAM Chip (2/3)

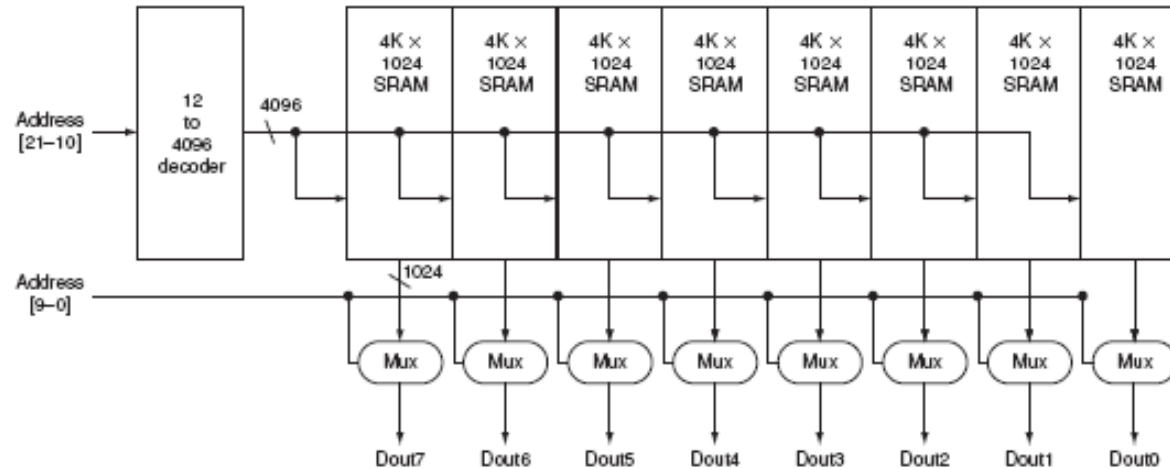


- 4 x 2 SRAM.
- Note that once a location is selected, it 'enables' all D latches in that location, allowing a read.
- Write enable will open an *AND* gate (with the location selected) and will clock all D latches, allowing us to write.

FIGURE C.9.3 The basic structure of a 4×2 SRAM consists of a decoder that selects which pair of cells to activate. The activated cells use a three-state output connected to the vertical bit lines that supply the requested data. The address that selects the cell is sent on one of a set of horizontal address lines, called word lines. For simplicity, the Output enable and Chip select signals have been omitted, but they could easily be added with a few AND gates.

SRAM Chip (3/3)

FIGURE C.9.4 Typical organization of a 4M×8 SRAM as an array of 4K×1024 arrays. The first decoder generates the addresses for eight 4K × 1024 arrays; then a set of multiplexors is used to select 1 bit from each 1024-bit-wide array. This is a much easier design than a single-level decode that would need either an enormous decoder or a gigantic multiplexor. In practice, a modern SRAM of this size would probably use an even larger number of blocks, each somewhat smaller.



- **As the number of locations increases, the size of decoder increases much faster (exponentially). So a two-step decoding is used.**
- **Consider a 4Mx8 SRAM: 4M locations of 8 bits (1 byte) each. It has a 22-bit address ($2^{22} = 4M$) and is built of an array of eight 4Kx1024 chips.**
- **The 22-bit address is split into two: a 12-bit and a 10-bit component.**
- **The 12-bit address selects one location from eight 4Kx1024 arrays.**
- **The 10 bits are used to select 1 bit from the 1024-bit location.**
- **There are eight such 1024-bit wide ‘columns’. So, finally, we get a byte out.**

DRAM (1/2)

- **Inexpensive, does not consume that much power.**
- **We can pack a lot of bits on a single chip. [64 Mbits]**
- **It is slow, requires regular refreshing ('volatile' memory).**
- **Each bit is stored as an electrical charge on a capacitor.**
- **A capacitor is like a very, very small battery.**
- **It loses its charge rather quickly.**
- **That is why we need to refresh it.**
- **Refreshing: all bits are read and written again.**
- **This is done several times every second.**

DRAM (2/2)

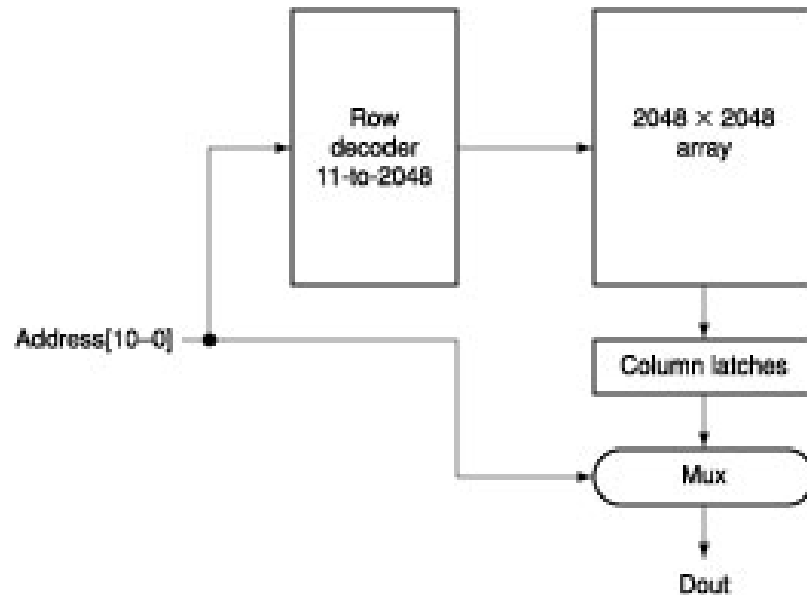


FIGURE C.9.6 A $4\text{M} \times 1$ DRAM is built with a 2048×2048 array. The row access uses 11 bits to select a row, which is then latched in 2048 1-bit latches. A multiplexor chooses the output bit from these 2048 latches. The RAS (Row Access Strobe) and CAS (Column Access Strobe) signals (not shown in the figure) control whether the address lines are sent to the row decoder or column multiplexor. Refresh is performed by reading the columns into the column latches and then writing the same values back. Thus, an entire row is refreshed in one cycle.

- This is 4-Mbit chip
- Organized as 2Kbits x 2Kbits
- 22-bit wide address is split into two 11bit addresses.
- Address is sent in two steps.
- Row address selects one row of 2048 bits.
- Column address then selects one bit out of these 2048 bits.

Error Detection (1/2)

- **As memory chips store more & more bits, the chance that some bits get read incorrectly increase.**
- **Also some external factors may cause errors while reading (*e.g.*, power supply spikes).**
- **How do we detect if a bit has been read incorrectly?**
- **One simple idea commonly used is parity.**
- **We count the number of 1's in a word.**
- **If it is odd, the parity bit is 1. Otherwise, it is 0.**
- **This parity bit is stored along with the memory word.**
- **When we read a 'word', we read a 32 bit-word and a parity bit.**

Error Detection (2/2)

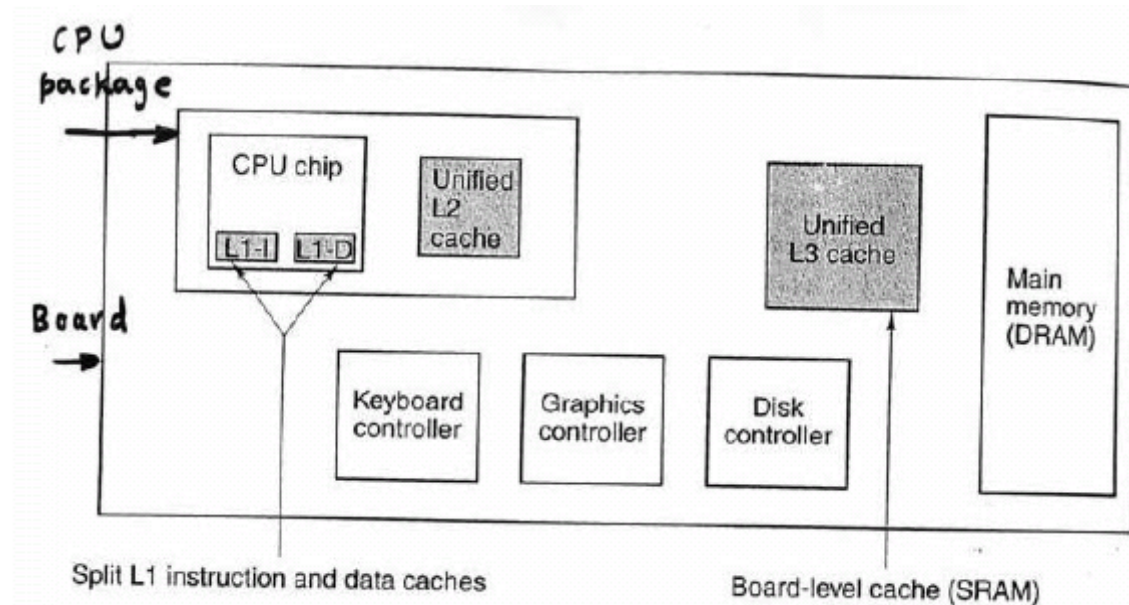
- After a word has been read, we calculate the parity by just looking at the 32 bits.
- If the calculated parity is the same as the parity bit read, then there were no errors while reading.
- 1 extra parity bit is capable of detecting multiple errors that affect an odd number of bits. If there are multiple errors in an even number of bits, they will not be detected.
- Error correction: This involves deciding which bit was read incorrectly.
- Error correction requires that several bits be stored along with each memory word.
- Typically, memory chips use 8 extra bits for every 128 bits of data.

Memory Timing

- **Memory access time**: for RAMs, this is the time it takes to perform a read/write operation.
- **Memory cycle time**: consists of access time plus any additional time required before a second access can begin (memory must be electrically stable).
- **Cycle time is slightly more than access time.**
- **Transfer rate / Bandwidth**: number of bytes that can be read/written in a second.
- **Example: Cycle time 20 nanoseconds; memory reads 4-byte words in a cycle.**

$$\text{Bandwidth} = (1000 / 20) * 10^6 * 4 = 200 \text{ Mbytes / sec.}$$

Cache



- **L1: on-chip split cache; 16 – 64 KB.**
- **L2: unified cache, in CPU package; 512KB – 1MB.**
- **L3: unified, board-level cache; 1 – 8MB.**

PLAs (1/2)

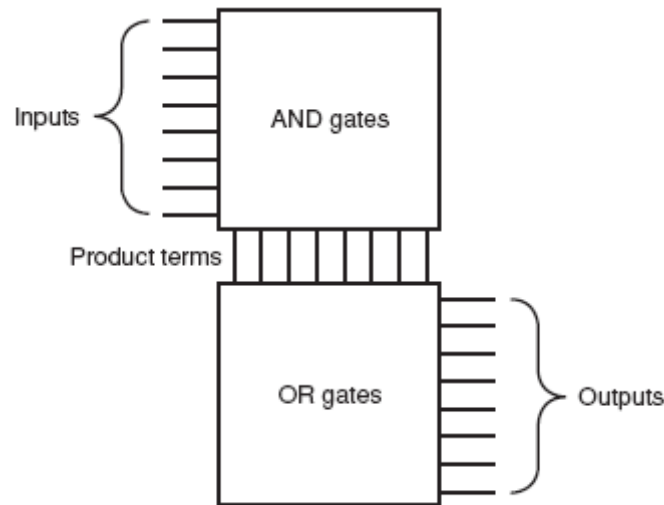
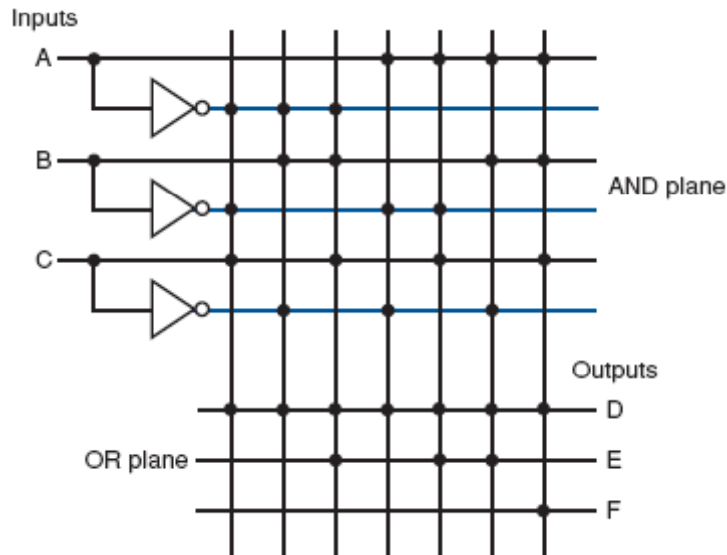


FIGURE C.3.3 The basic form of a PLA consists of an array of AND gates followed by an array of OR gates. Each entry in the AND gate array is a product term consisting of any number of inputs or inverted inputs. Each entry in the OR gate array is a sum term consisting of any number of these product terms.

- A programmable logic array (PLA) is composed of a set of inputs (and their complements) and two stages of logic.
- The first stage generates product terms (*AND*s) of the inputs (thus yielding 'minterms').
- The second generates sum terms (*OR*s) of the first-stage product terms.
- PLAs are structured-logic elements used to implement combinatorial logic functions that are expressed in Sum-of-Products form.

PLAs (2/2)



- Initially, a PLA has all its cross-points connected.
- Any connection can be removed by burning out the fuse at the corresponding cross-point.
- Example: the 3-input, 3-output PLA in the figure to the left implements the function given by the truth table below.

FIGURE C.3.5 A PLA drawn using dots to indicate the components of the product terms and sum terms in the array. Rather than use inverters on the gates, usually all the inputs are run the width of the AND plane in both true and complement forms. A dot in the AND plane indicates that the input, or its inverse, occurs in the product term. A dot in the OR plane indicates that the corresponding product term appears in the corresponding output.

Inputs			Outputs		
A	B	C	D	E	F
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

ROMs (1/2)

- **Read-Only Memory (ROM) is built using structured-logic elements very similar to PLAs.**
- **A ROM consists of a number of nonvolatile, fixed-sized memory locations. Each location is individually addressable.**
- **The contents of a ROM are written when the ROM is made. You cannot change them afterwards.**
- **PROM: Programmable ROM. The user (not the chip maker) writes the memory by blowing the fuses at the cross-points.**

ROMs (2/2)

- **EPROM: Erasable PROM.**
This uses a different technology than ROM/PROM. The user writes the memory, but it can be erased by exposure to ultra-violet light. The entire chip is erased, and not specific, selectable memory locations on it.
- **EEPROM: Electronically Erasable PROM.**
Does not require ultra-violet light for erasing, and specific memory locations can be selected for erasure.
- **Flash memory is a type of EEPROM.**