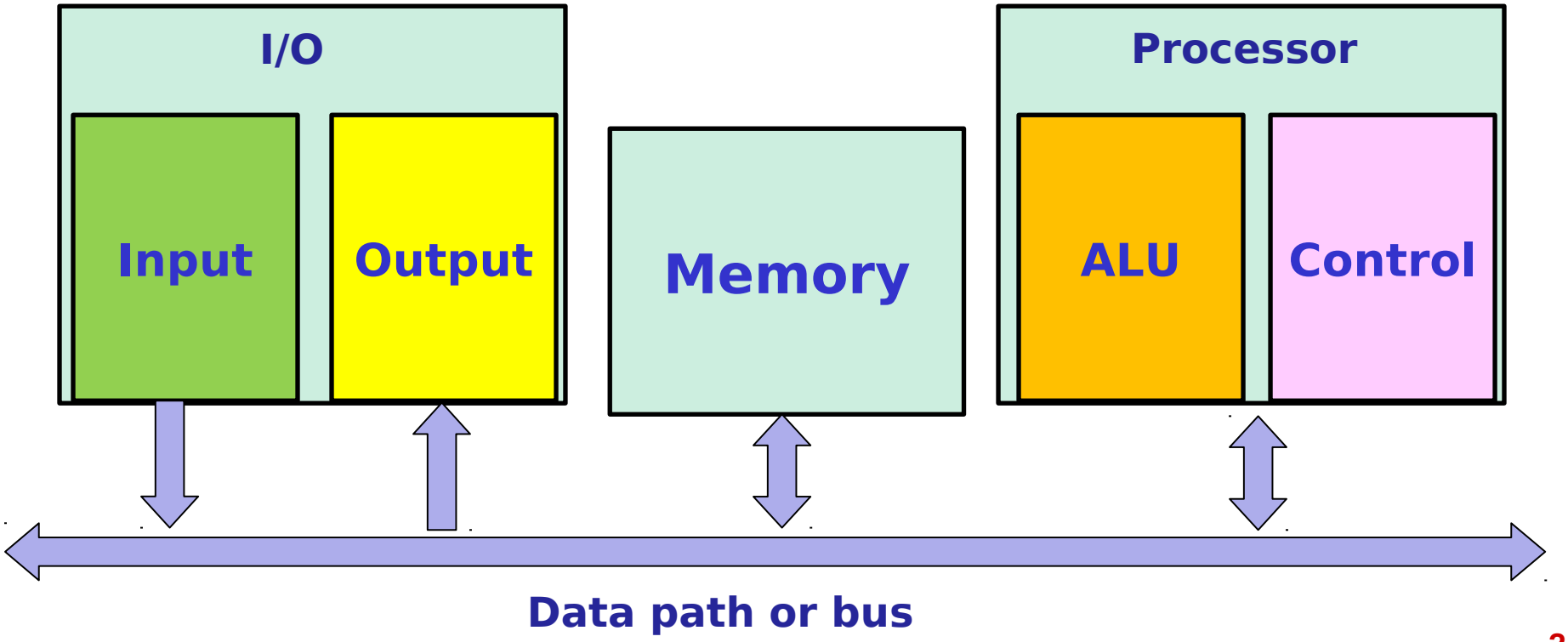
Memory Unit

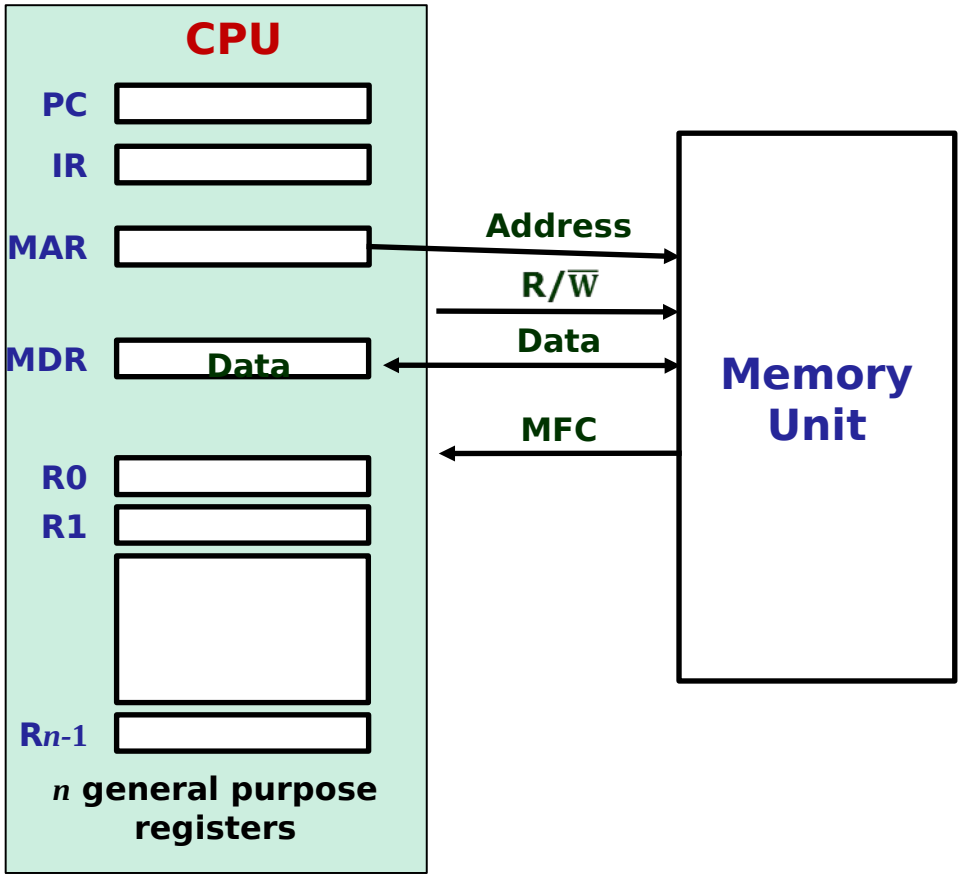
- **Functional Units**:
  - Input Unit
  - Memory Unit
  - Arithmetic and Logic Unit (ALU)
  - Output Unit
  - Control Unit



Computer Memory

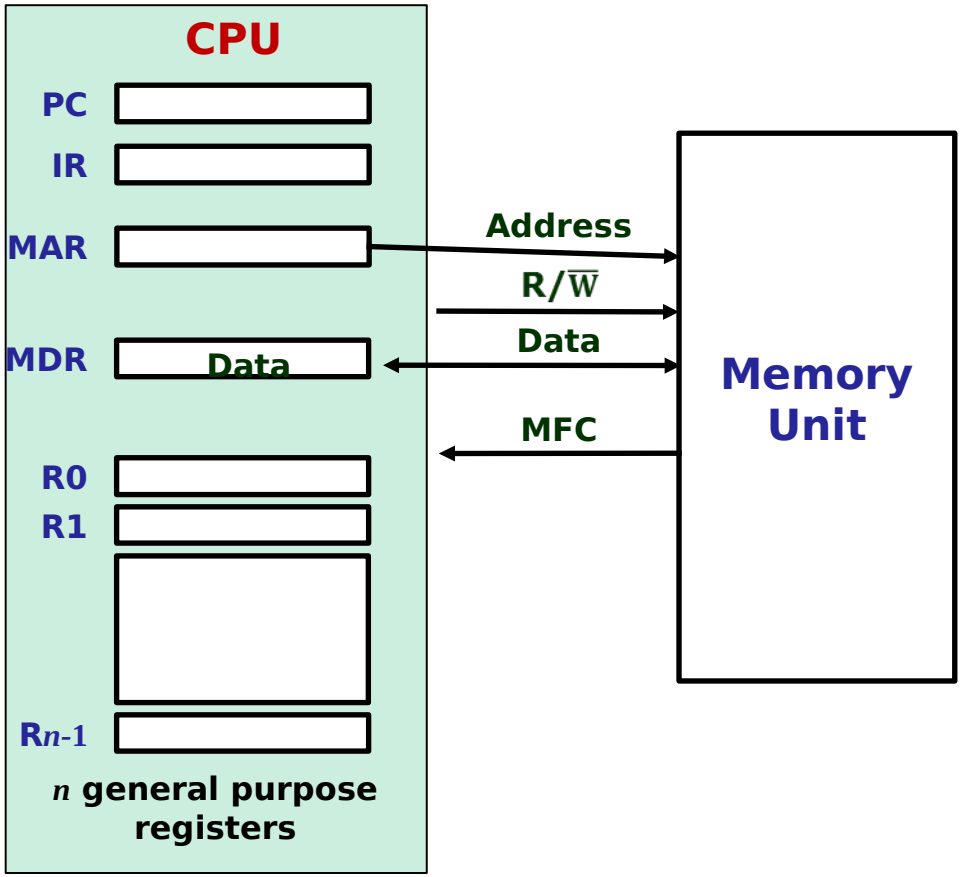
- Number & character operands, as well as instructions are stored in the memory of the computer
- Stored program concept
- CPU executes the instructions for which the instructions and operands have to come from memory unit
- Operations which involve memory:
  - Instruction fetch
    - Memory read
  - Memory operand fetch and store
    - Memory read
    - Memory write
- Instructions involving memory access:
  - LOAD and STORE instructions

Memory Read and Write Operation



- **Read**
  - Processor loads the address of memory location into MAR
  - Set the R/$\overline{W}$ line to 1
  - Memory responds by placing the data from address location onto data line
  - Confirm the action by asserting MFC (memory function complete) signal
  - Upon receiving MFC signal, processor loads the data on data line into MDR

Memory Read and Write Operation

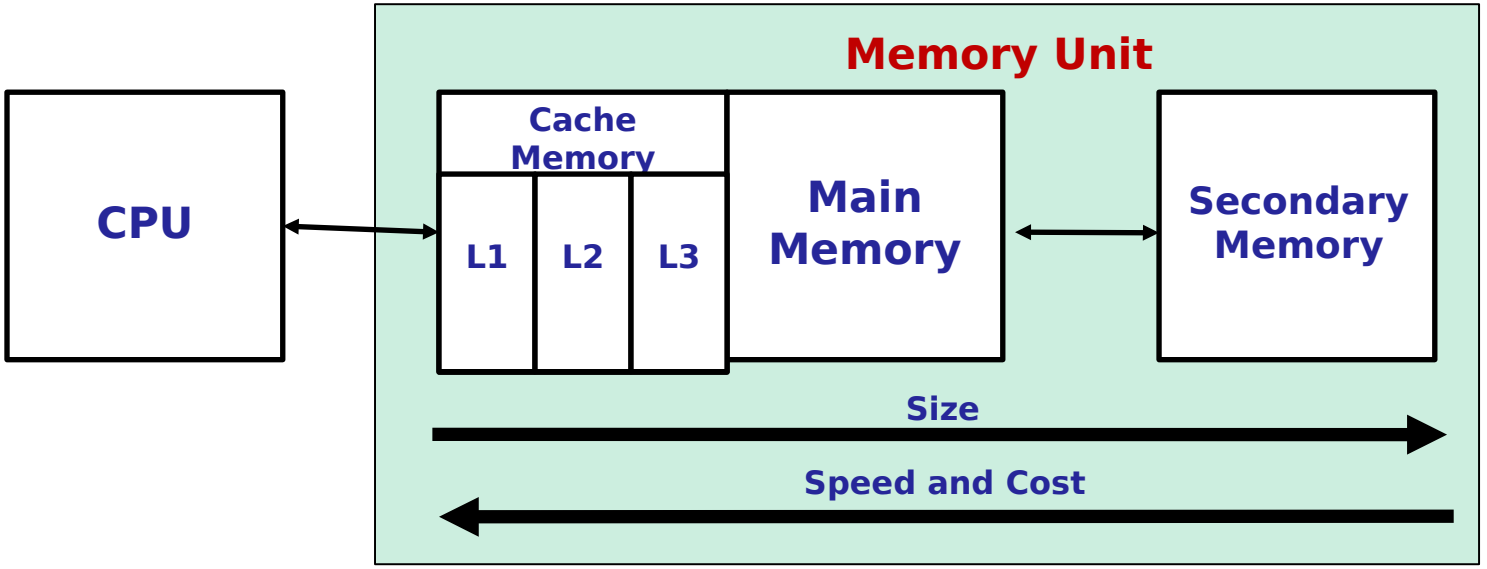


- **Write**
  - Processor loads the address of memory location into MAR
  - Processor loads data into MDR
  - Set the R/$\overline{W}$ line to 0 to indicate write operation
  - Processor places the data in MDR onto data line
  - Data on data line is written into memory location
  - Memory confirms the action by asserting MFC signal

Memory Latency and Memory Organization

- **Latency**: Time to access the first of the sequence of memory words
- What is involved in determining the latency of the memory operation?
  - Processor issues the logical address to memory unit
  - The logical address need to be converted into physical address
- Memory unit is called random access memory (RAM)
  - Any location can be accessed for read/write operation independent of the location's address
- Memory unit is organised in hierarchical manner

Memory Hierarchy



- Processor processes instructions and data faster than it can be fetched from memory unit
- Memory access time is the bottleneck
- One way to reduce memory access time is to use faster memory
  - A small and faster memory bridge the gap between processor and main memory

Memory Performance Parameters

- **Memory Access Time**:
  - Time interval between initiation of one operation and completion of that operation
  - Example: Time between assertion of Read signal and MFC signal
- **Memory Cycle Time**:
  - Minimum time delay between the initiation of two successive memory operations
  - Time delay between start of a read/write operation to start of next memory operation
- Memory cycle time is usually slightly larger than access time

Internal Organization of Memory

- The memory is organised such that a group of n-bits can be stored or retrieved in a single basic operation
- Each group of n-bits is referred as one **memory word**
- Accessing the memory to store or retrieve information require address for each location
- Possible number of address locations are decided by the number of address lines in the processor
- For k-address lines, there will be 2^k locations, each of n-bit memory word
- 2^k addresses constitute address space of computer
- Example: Let k=10 and n=32
  - Number of locations: 2^{10}
  - Size of the memory: $2^{10} \times 2^5$ bits $= 2^{15}$ bits
    $= 2^{12}$ Bytes
    $= 4$ KB

Memory Content Example

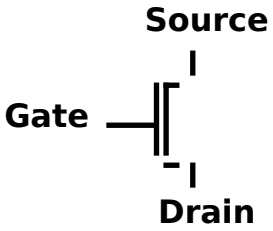
1024 memory locations: 1x0 bit address

16 bit data

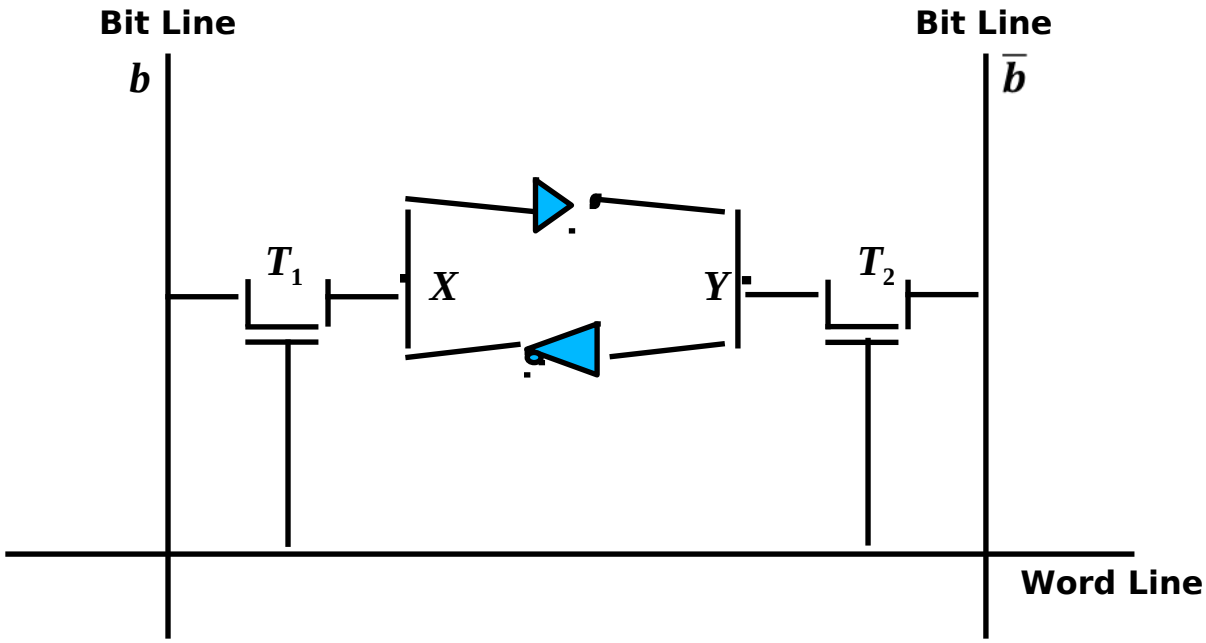
Memory address		Memory contest
Binary	decimal	
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
⋮	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

Semiconductor Memories

- Two basic ways of designing memory
  - Static RAM (SRAM)
  - Dynamic RAM (DRAM)
- **Static RAM**:
  - Built using metal-oxide semiconductor (MOS) transistors
  - MOS transistors acts as switch
    - +5 v (when Gate input is 1): Transistor conducts: ON state
    - 0 v (when Gate input is 0): Transistor does not conducts: OFF state

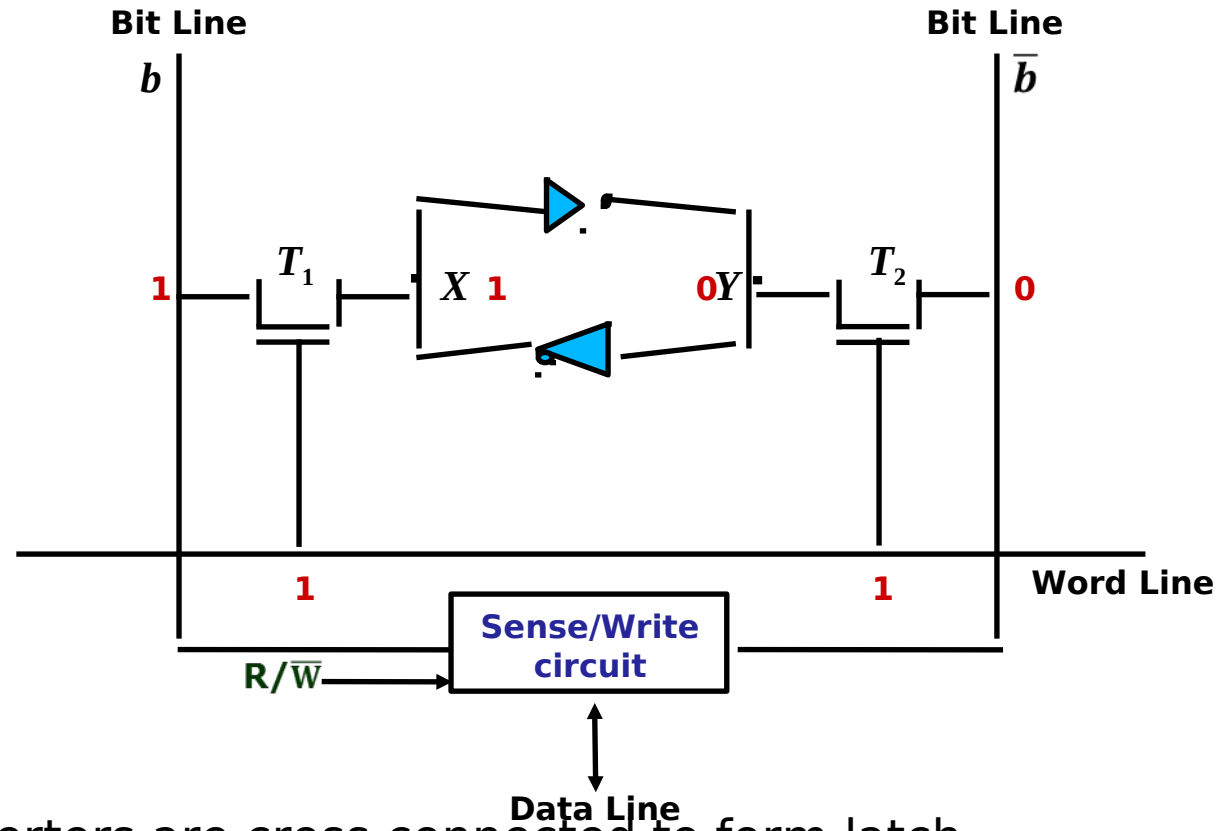


Static RAM Cell



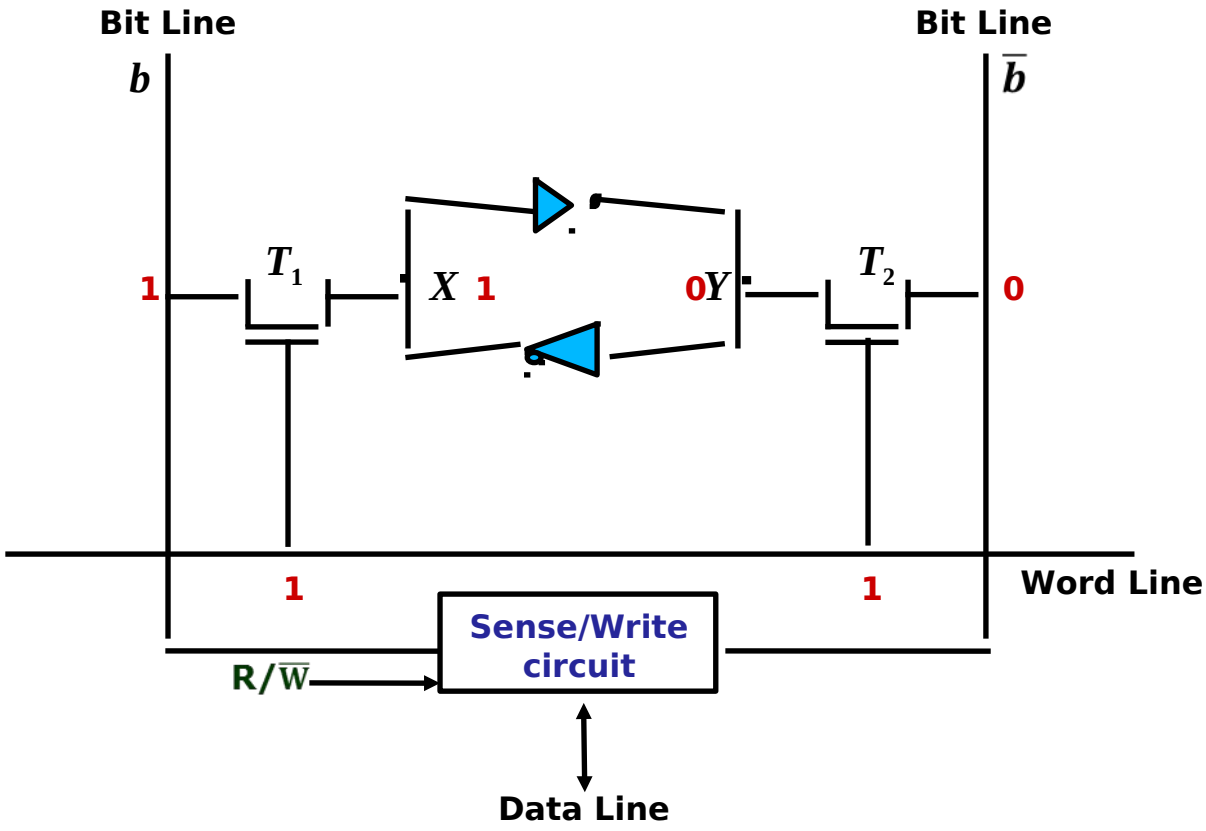
- Two inverters are cross connected to form latch
- Inverters are connected to 2 transistors which act as switches
- Switches are opened or closed under the control of word line
- This circuit retain the state (bit) as long as power is applied (Static Memory)

Static RAM Cell - Read



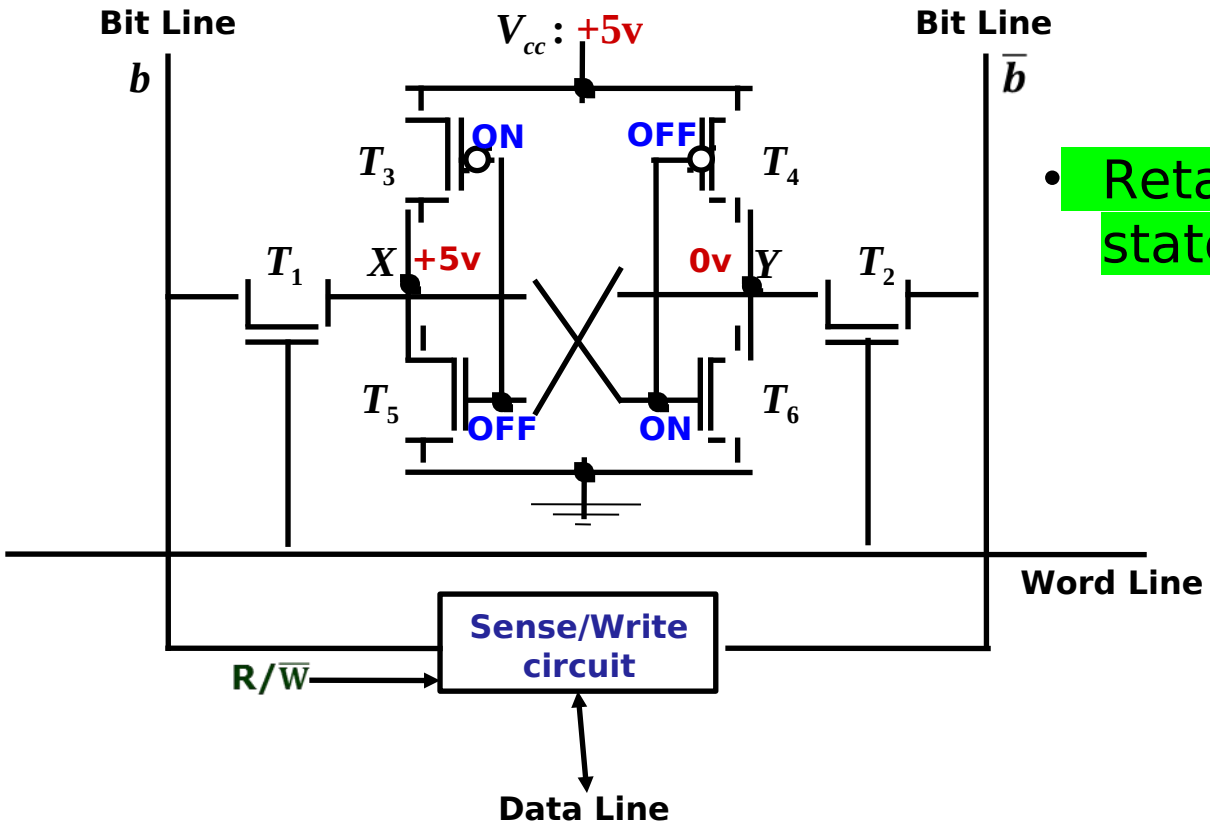
- Two inverters are cross connected to form latch
- Inverters are connected to 2 transistors which act as switches
- Switches are opened or closed under the control of word line
- This circuit retain the state (bit) as long as power is applied (Static Memory)

Static RAM Cell - Write



- Inverters are connected to 2 transistors which act as switches
- Two inverters are cross connected to form latch
- Switches are opened or closed under the control of word line
- This circuit retain the state (bit) as long as power is applied (Static Memory)

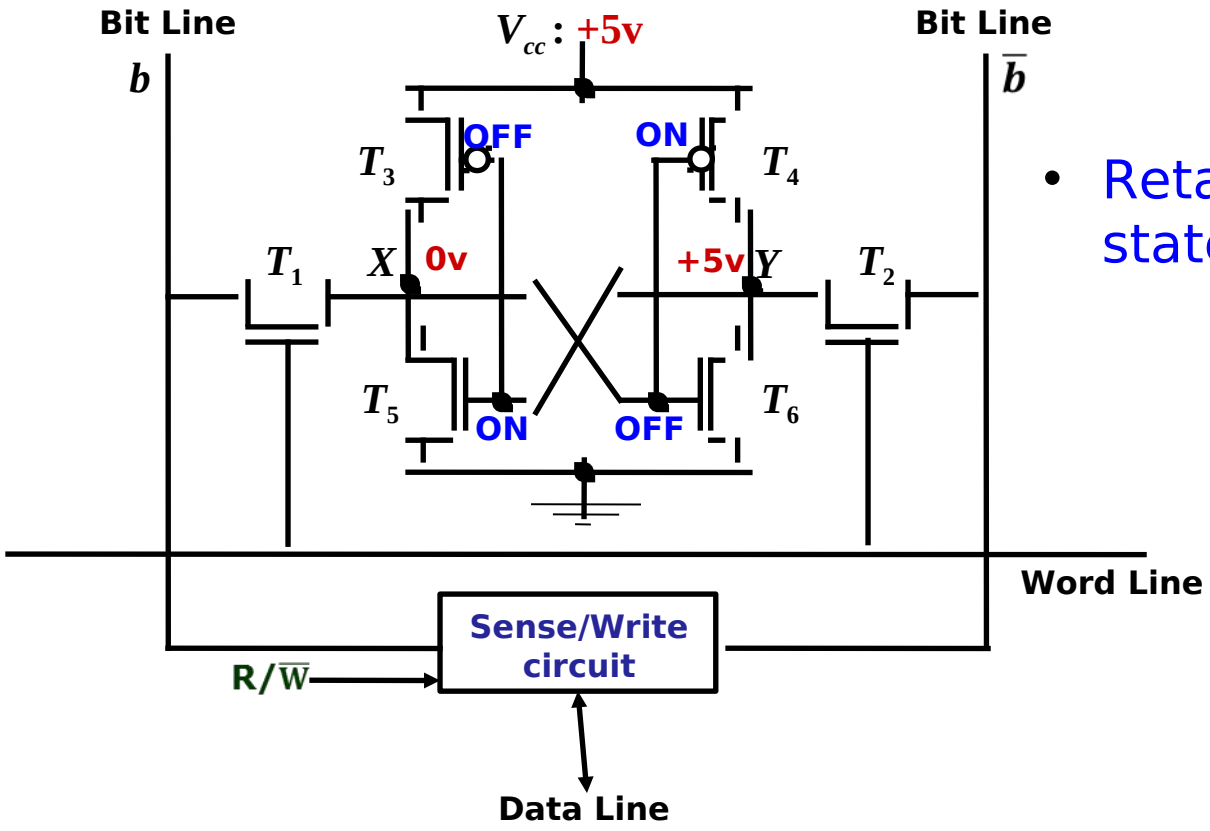
CMOS Static RAM Cell



- Retaining the state 1

- This circuit retain the state (bit) as long as power is applied (Static Memory)
- Continuous power is needed for a cell to retain the state

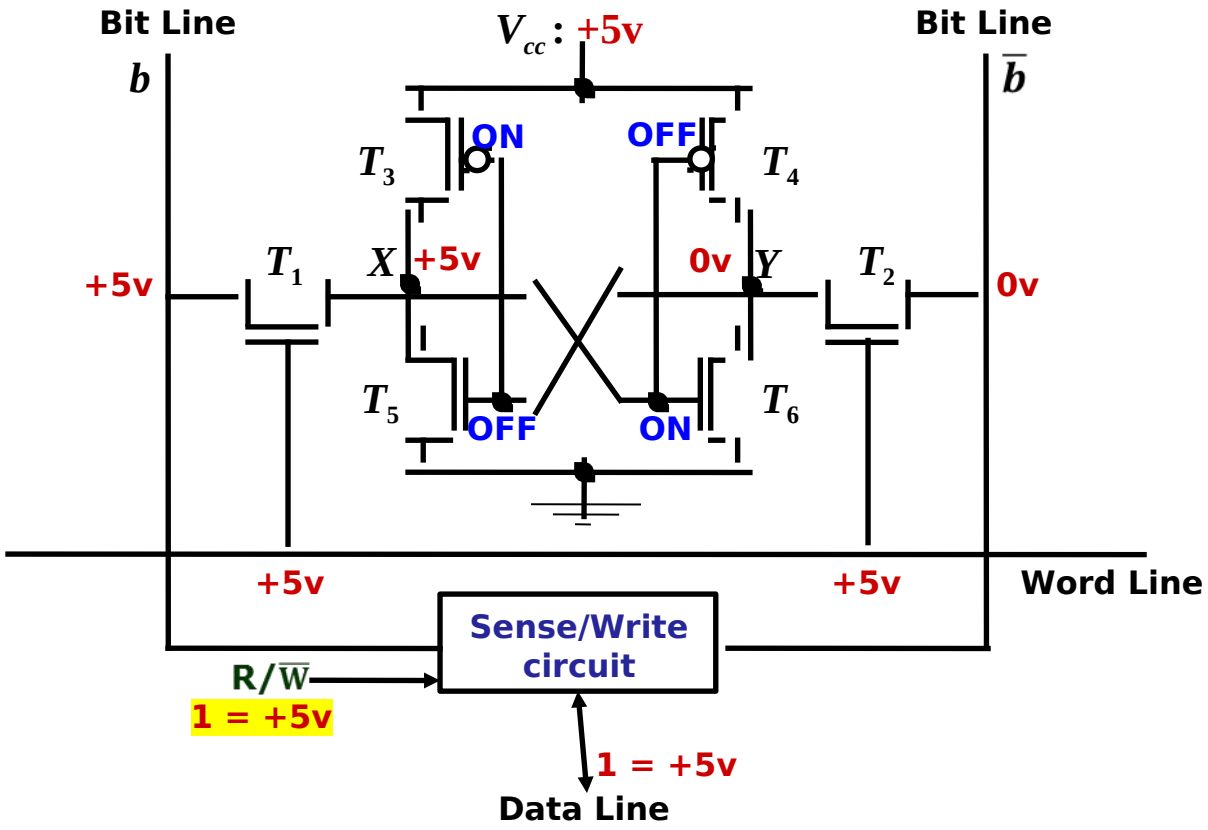
CMOS Static RAM Cell



- Retaining the state 0

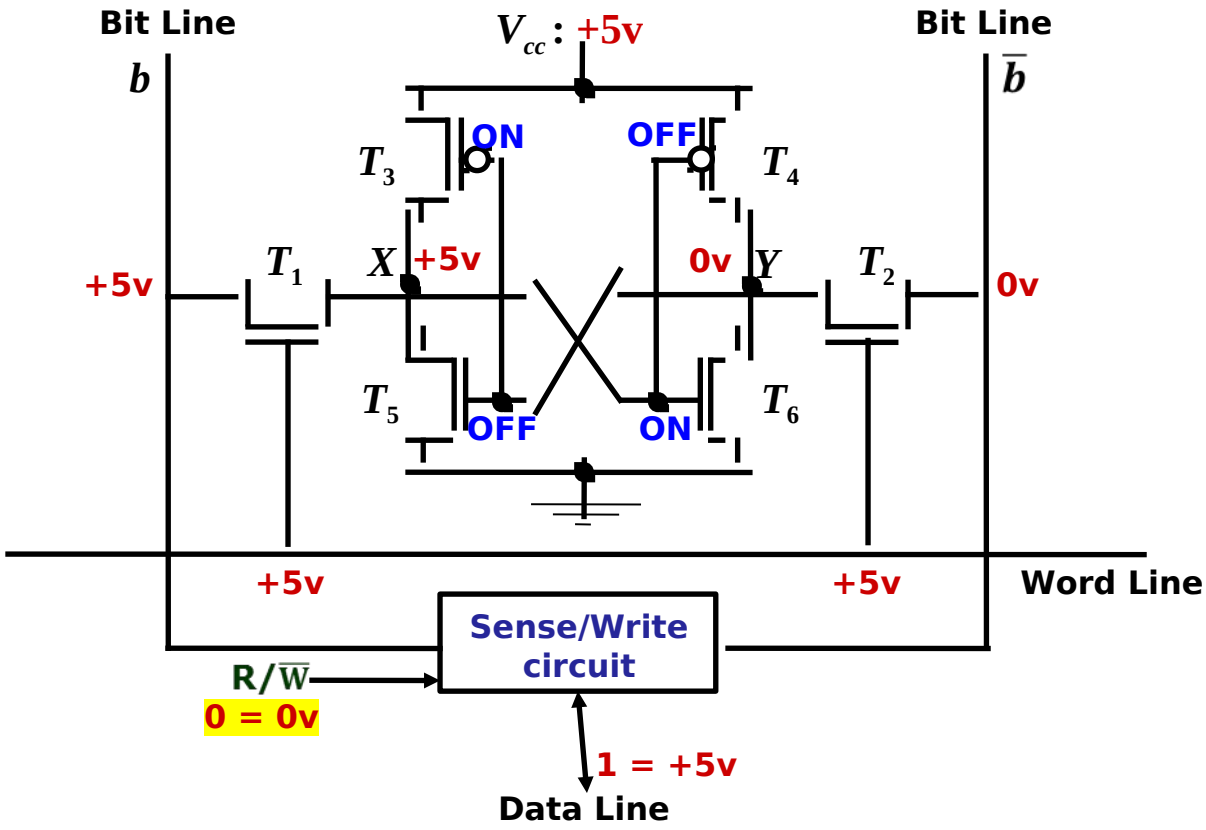
- This circuit retain the state (bit) as long as power is applied (Static Memory)
- Continuous power is needed for a cell to retain the state

CMOS Static RAM Cell - Read



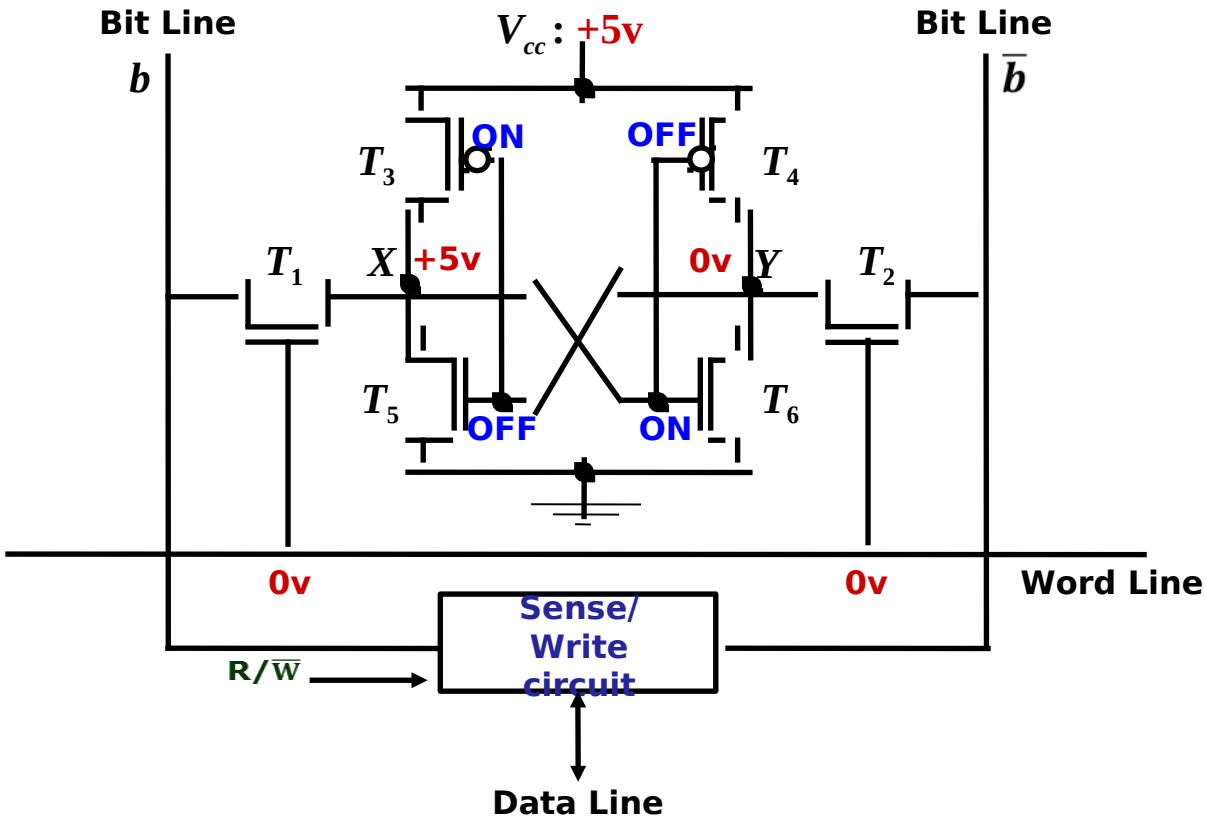
- This circuit retain the state (bit) as long as power is applied (Static Memory)
- Continuous power is needed for a cell to retain the state

CMOS Static RAM Cell - Write



- This circuit retain the state (bit) as long as power is applied (Static Memory)
- Continuous power is needed for a cell to retain the state

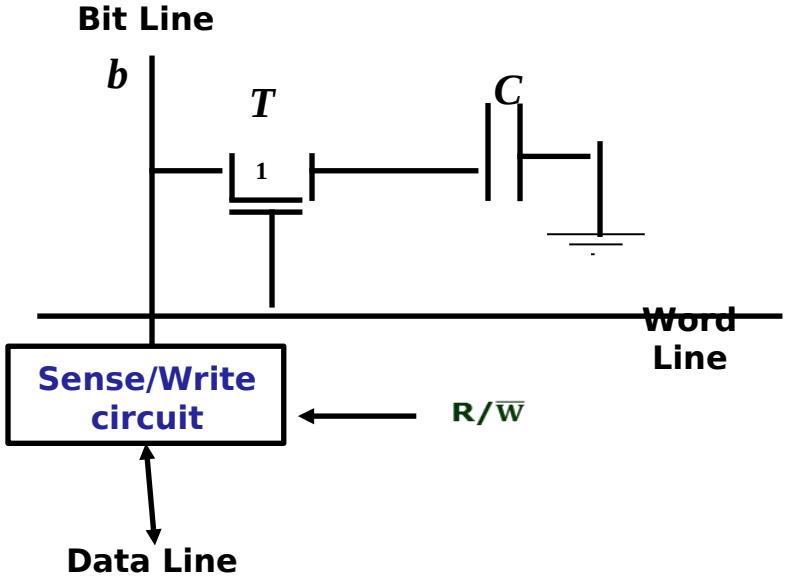
CMOS Static RAM Cell: Illustration



- Access time is less i.e. faster memory
- Low power consumption
- Uses 6 transistors: Costly
- Used in applications where speed is critical concern: Cache

Dynamic RAM (DRAM) Cell

- Less expensive and simpler cell
- Information is stored in the form of a charge on a capacitor (C)
  - Charge in capacitor is stored only for short time
  - However, a cell is required to store information for a much longer time
- To retain information for longer time, content of capacitor mush be periodically refreshed



- Low speed as refresh needed
- Only 1 transistor is used
- Used to build main memory

Memory Content Example

1024 memory locations: 10 bit address

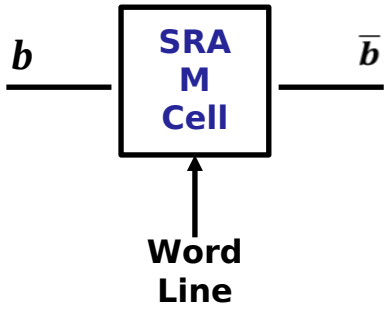
16 bit data

Memory address		Memory contest
Binary	decimal	
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
⋮	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

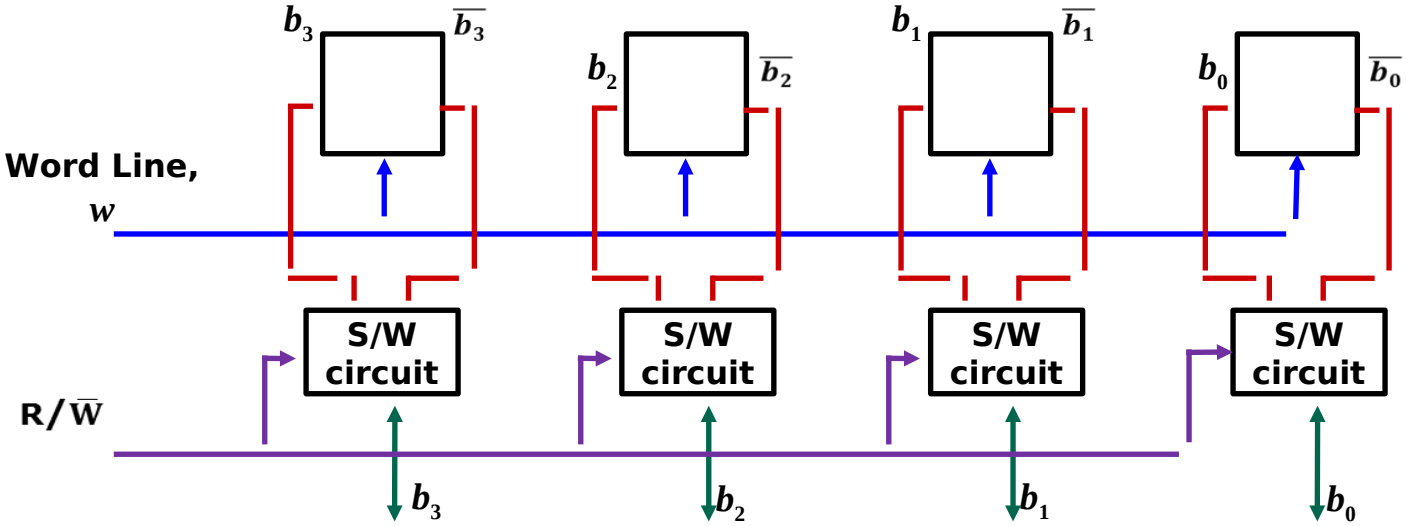
Fig. 7-3 Content of a 1024 × 16 Memory

Internal Organization of SRAM Chip

- SRAM cell (block diagram):



- Memory cells are usually organised in the form of an array
- At each memory location, n SRAM cells are placed next to each other to form n-bit memory word
- Example: 4-bit memory word



Internal Organization of 2^k x n SRAM Chip

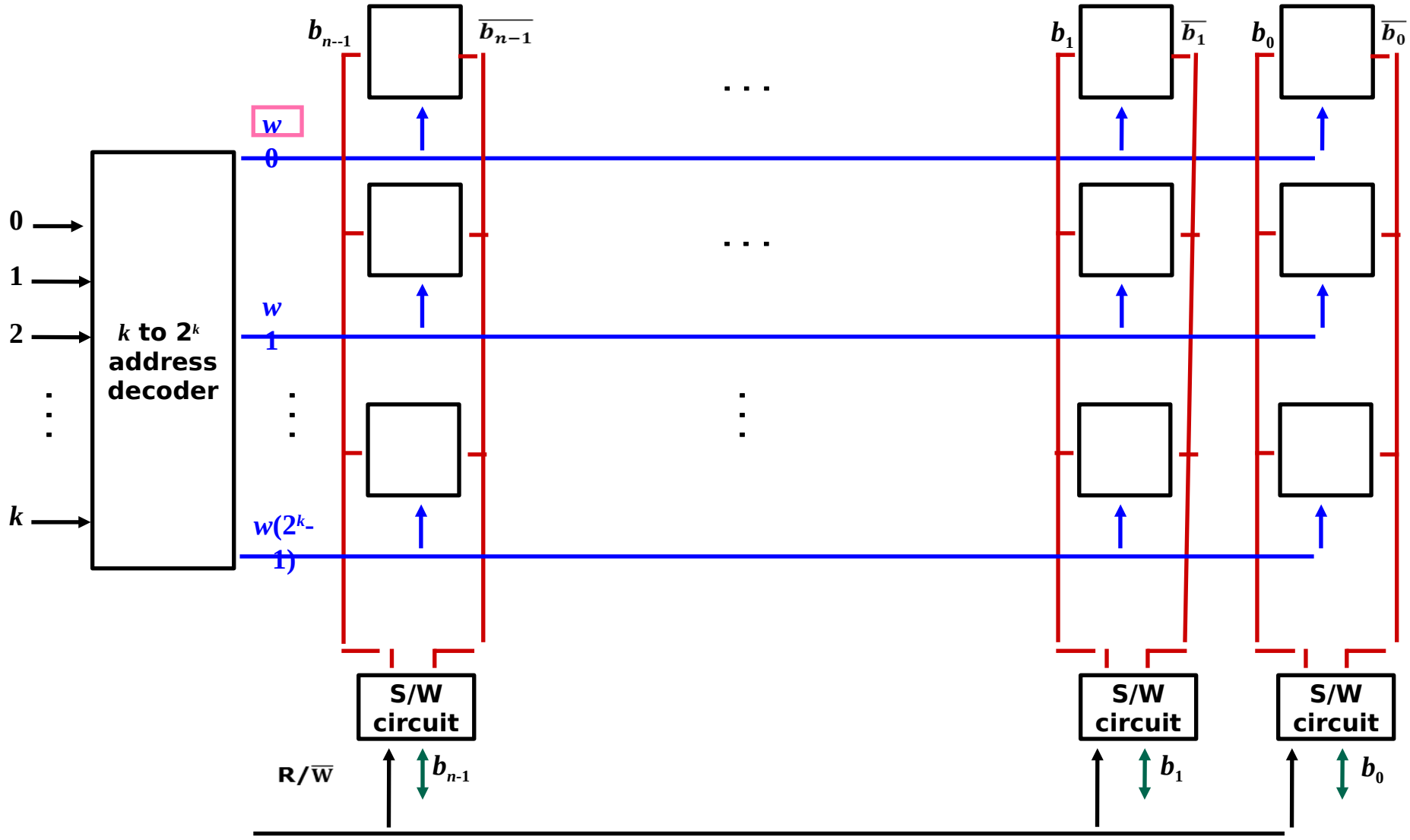


Illustration: 16 x 4 bit Memory Chip

- 1-dimensional address decoding:
  - *Example*: ***Design of 64b memory chip***
  - Number of address line be : 4
  - Address decoder configuration : 4-to-2^4

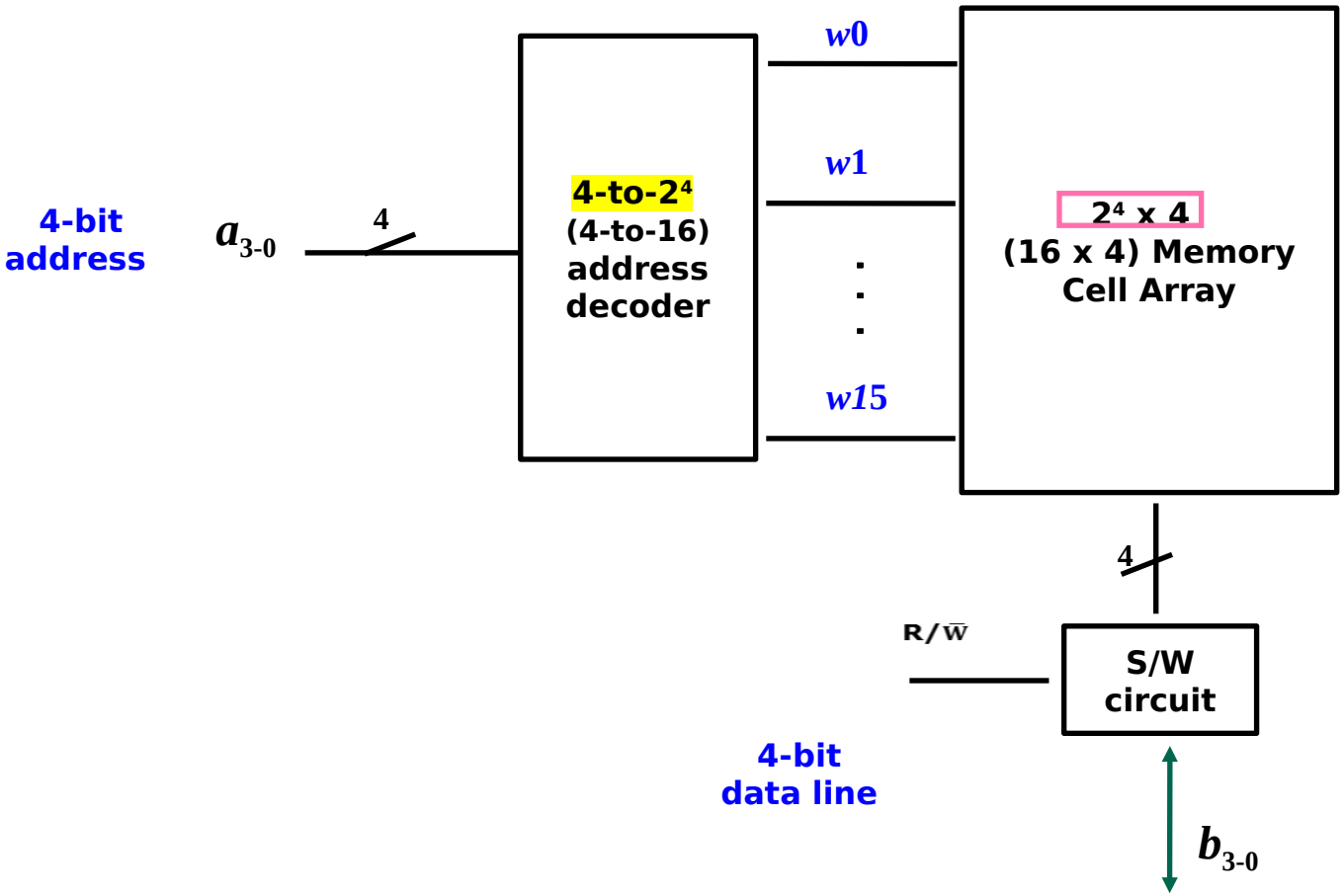


Illustration: 512 x 8 bit Memory Chip

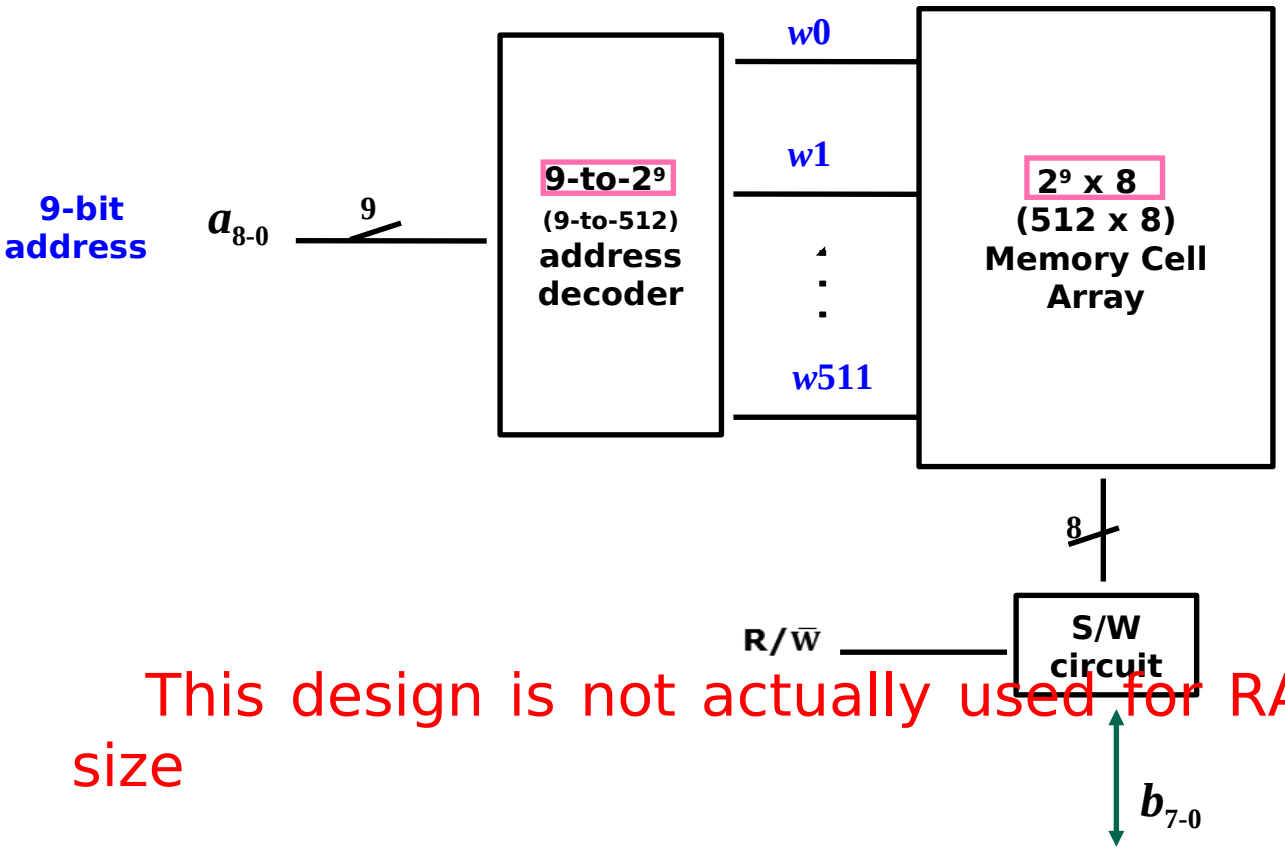
- 1-dimensional address decoding:
  - *Example*:

Memory chip organization : 512 x 8

Memory Size : 512B

Number of address line be: 9

Address decoder configuration : 9-to-2^9



This design is not actually used for RAM of any size

Coincident Decoding

Regular decoding is costly:

A decoder with k inputs and 2^K outputs requires 2^K AND gates with k inputs per gate.

Total number of gates can be reduced by using two-dimensional decoding:

Basic idea: arrange memory cells in a (as close as possible to) square configuration.

Use two k/2 input decoders instead of one k input decoder

Two-Dimensional Decoding

Instead of using a single 10 x 1024 decoder
we use two 5x32 decoders.

One decoder picks the row, one the column

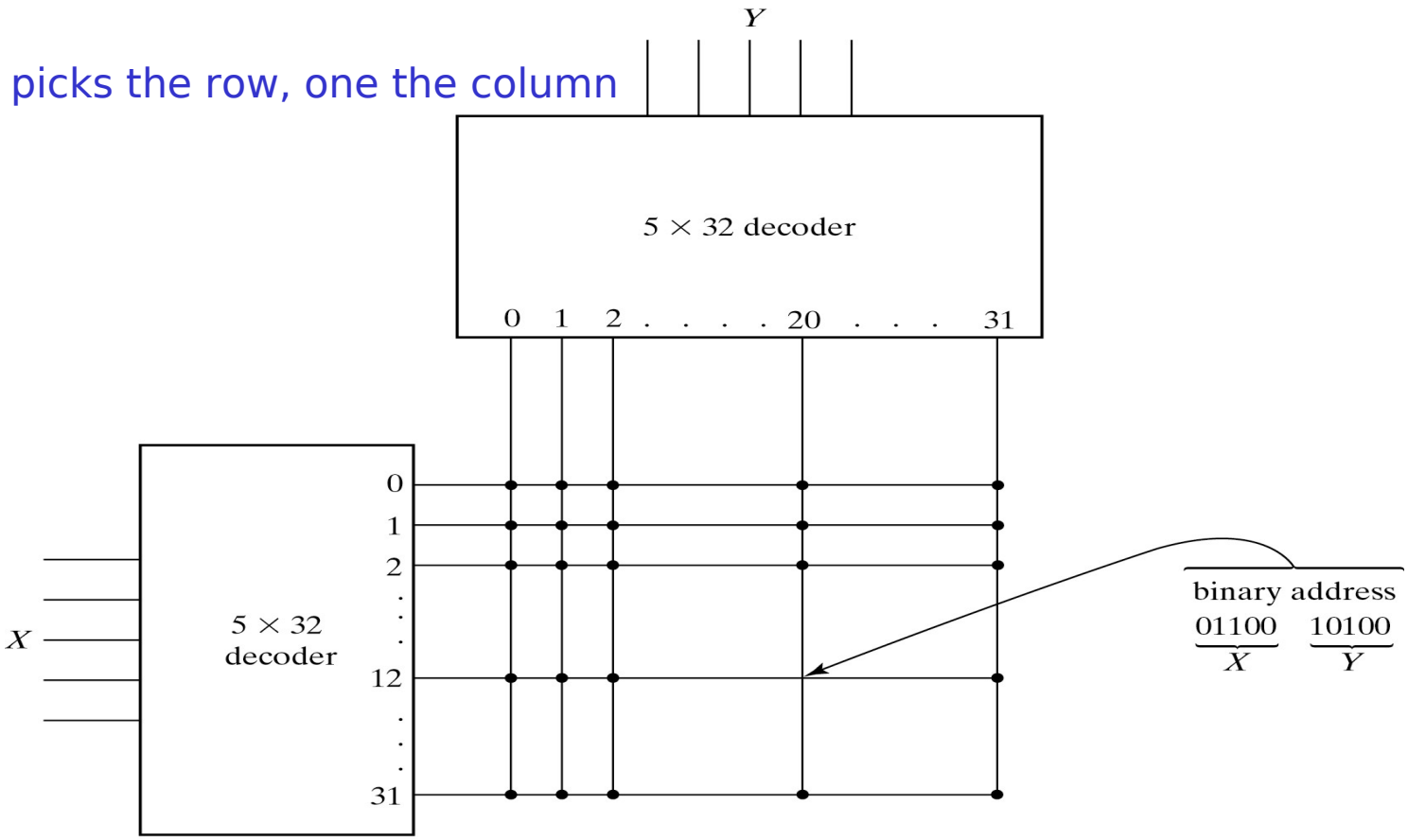


Fig. 7-7 Two-Dimensional Decoding Structure for a 1K-Word Memory

Two-Dimensional Decoding

Needs 64 5-input AND gates instead of 1024 10-input gates.

Address is divided to two equal parts

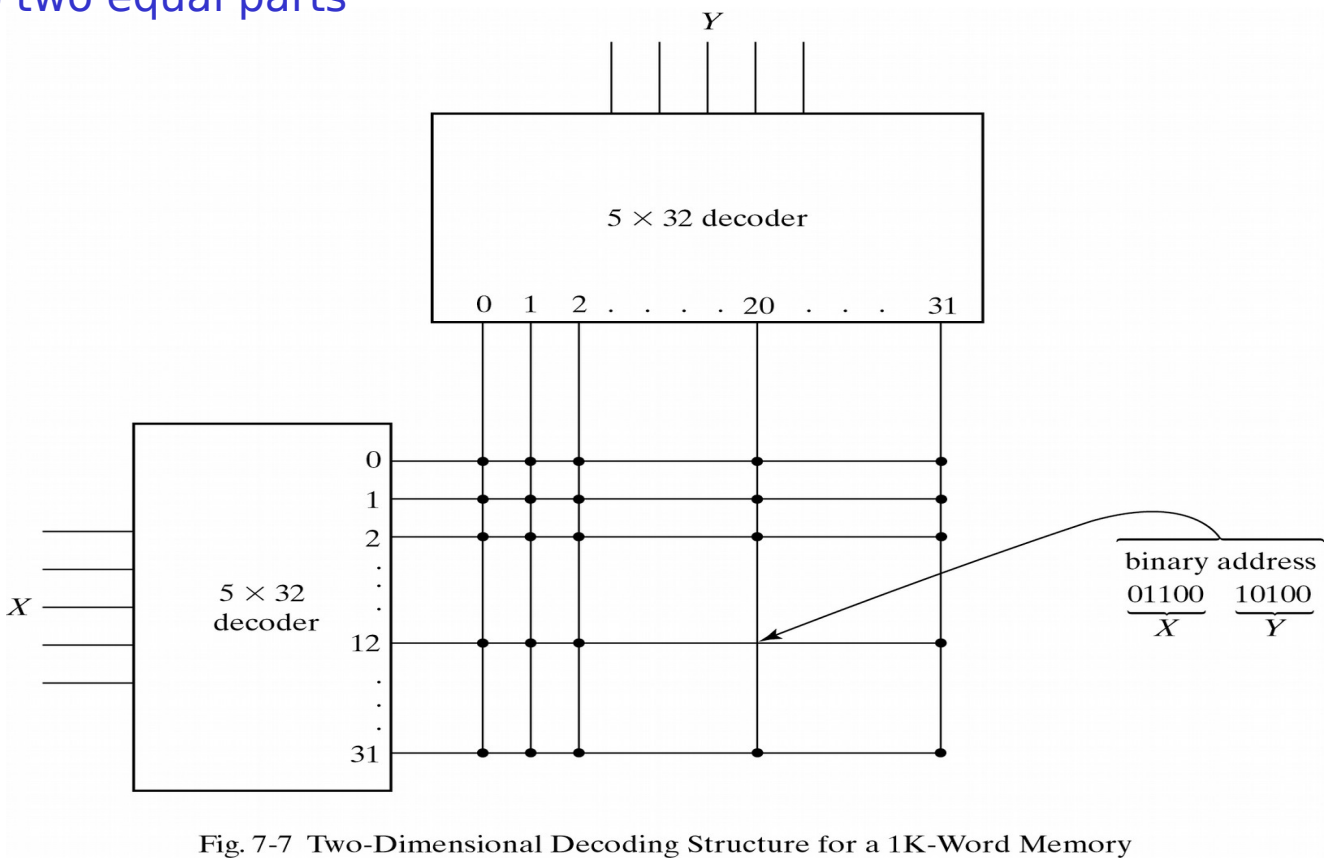
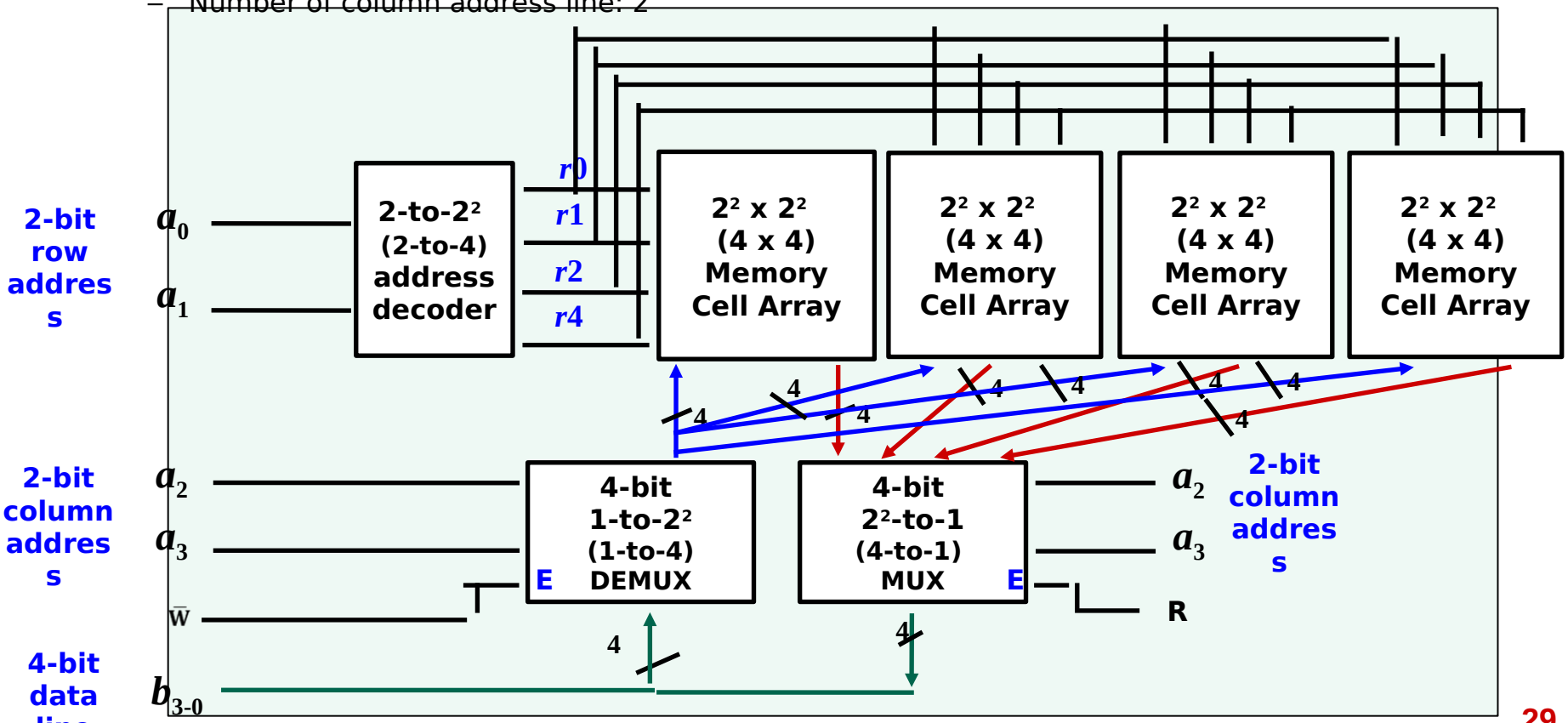


Fig. 7-7 Two-Dimensional Decoding Structure for a 1K-Word Memory

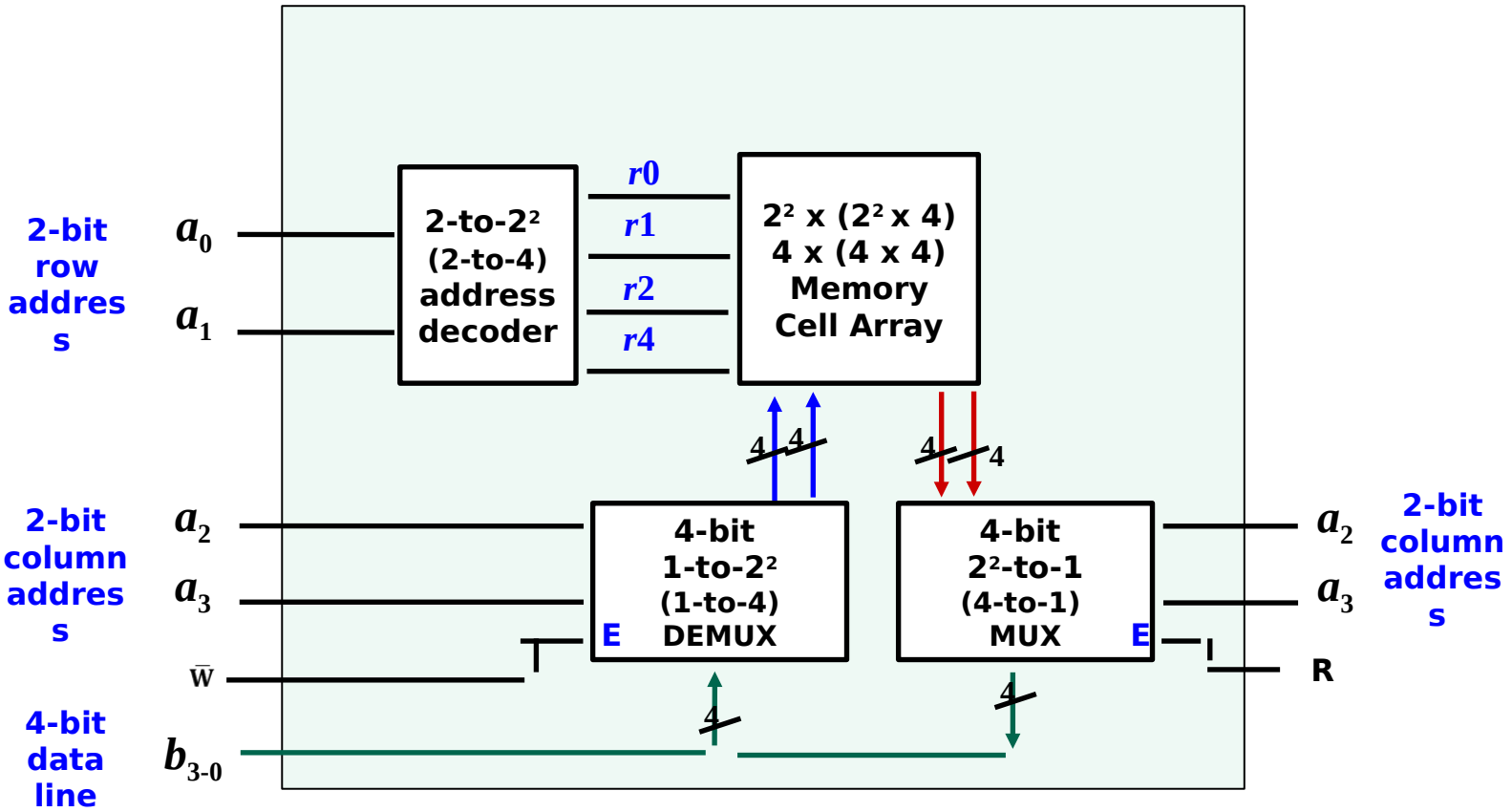
2-dimensional Address Decoding

- k-address line is divided into $k/2$ row address and $k/2$ column address
- Now, memory chip is considered as $2^{k/2}$ x $2^{k/2}$ memory cell array
- Example: Design of 16 x 4 bit memory chip
  - Number of address lines: 4
  - Number of row address lines: 2
  - Number of column address line: 2



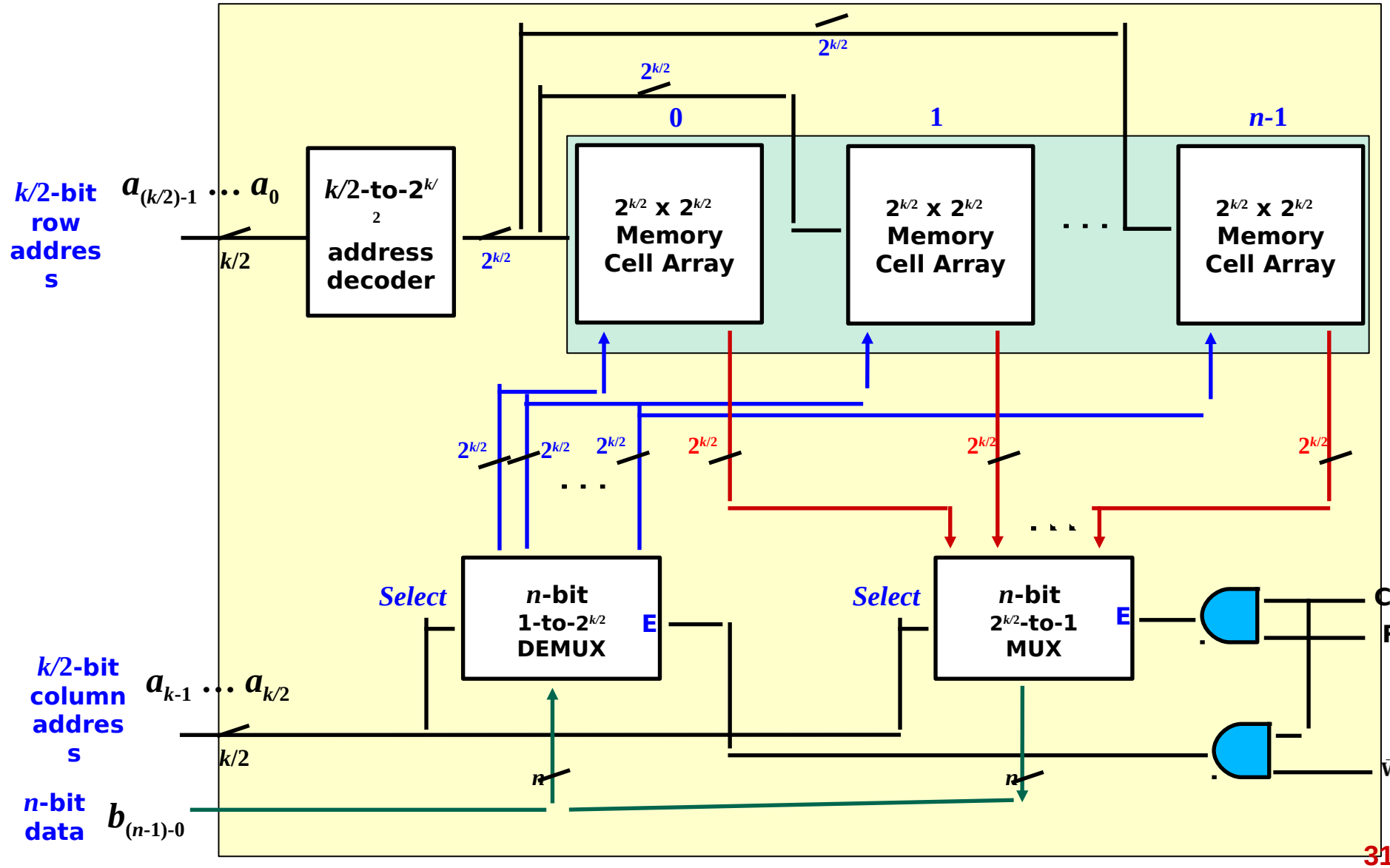
2-dimensional Address Decoding

- k-address line is divided into $k/2$ row address and $k/2$ column address
- Now, memory chip is considered as $2^{k/2}$ x $2^{k/2}$ memory cell array
- Example: Design of 16 x 4 bit memory chip
  - Number of address lines: 4
  - Number of row address lines: 2
  - Number of column address line: 2



2-dimensional Address Decoding

- **Organization of 2^k x n SRAM Chip**



2-dimensional Address Decoding

- **Organization of 2^k x n SRAM Chip**
  - The number of cell array depends on the width of the memory chip (word length) expected

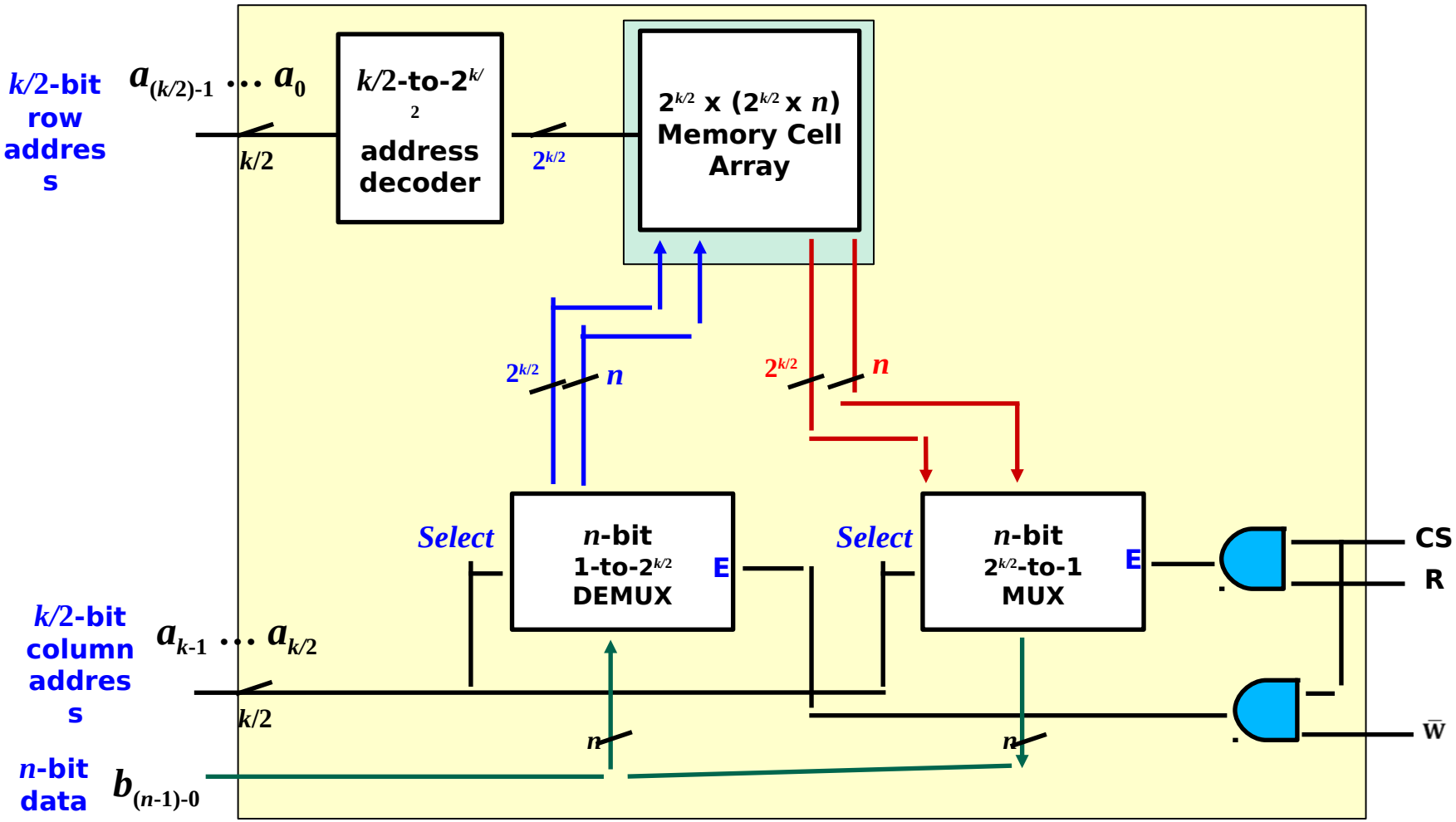


Illustration: Design of 1MB SRAM Chip

- Capacity of memory: 1 MB
- Word length: 8 bits
- Memory organization: 1M x 8

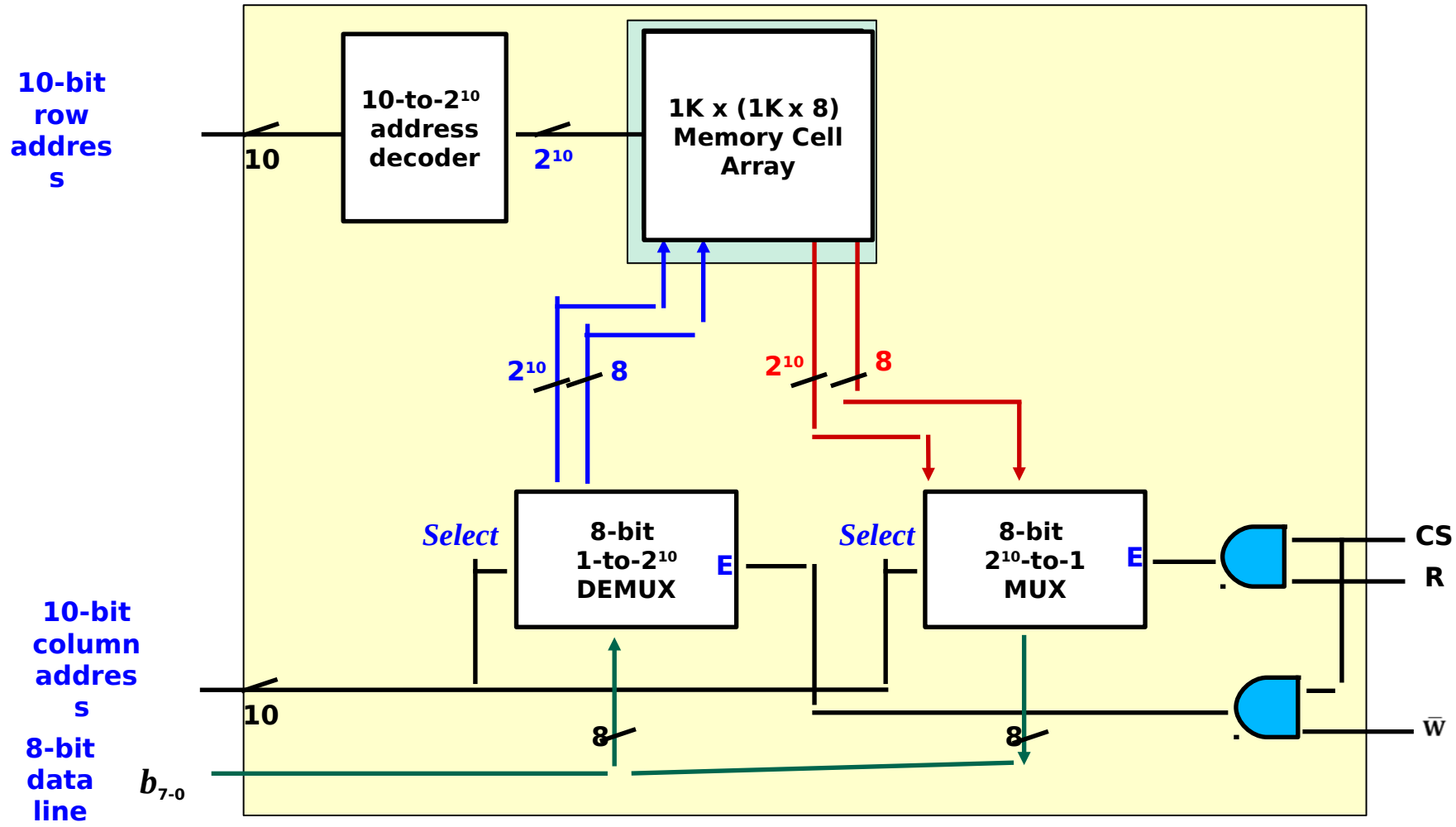
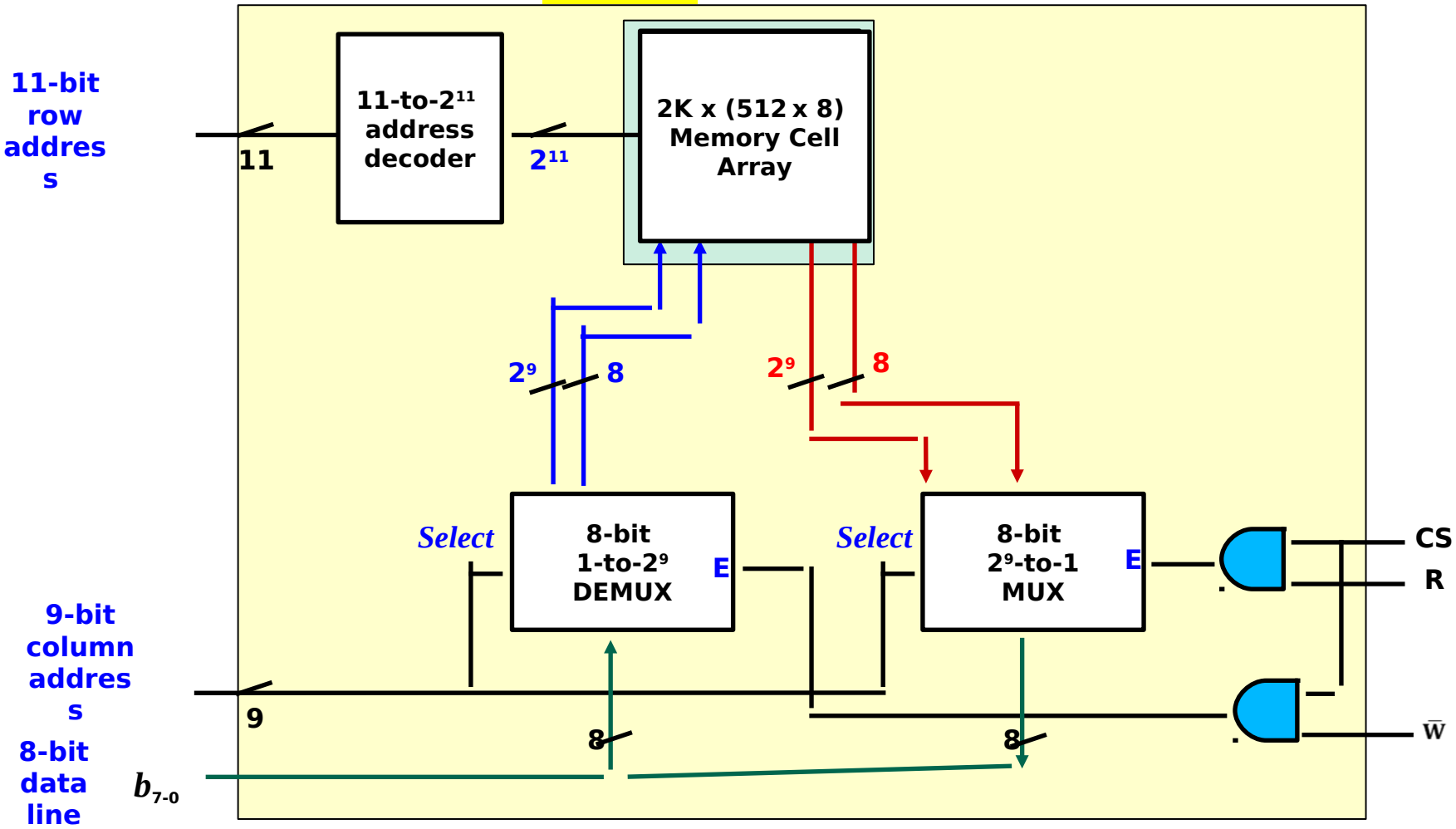


Illustration: Design of 1MB SRAM Chip

- Capacity of memory: 1 MB (2^{20} B = 2^{23} b)
- Word length: 8 bits
- Number of row address: 11
- Number of column address: 20-11 = 9



Static RAM Module

- **Byte Addressable Memory**
- Illustration:
  - Task: Design SRAM with capacity 8MB (2^{23} **B**)
  - Requirements:
    - Memory organization depends on the word size and word size decides the width of the data bus
      - Let word size be 32 bit (**4 B**)
      - Now, memory organization: 2M x 32 bit
    - Memory should be byte addressable
      - Let the organization of cell array to incorporate byte addressability be 1M x 8 bit
  - Organization include:
    - 2 rows of chips, each of size 1M x 8 bit
    - Each row contain 4 chips
  - Number of address lines: 23 (2^{23} **B**)

Illustration: 8MB Static RAM Module

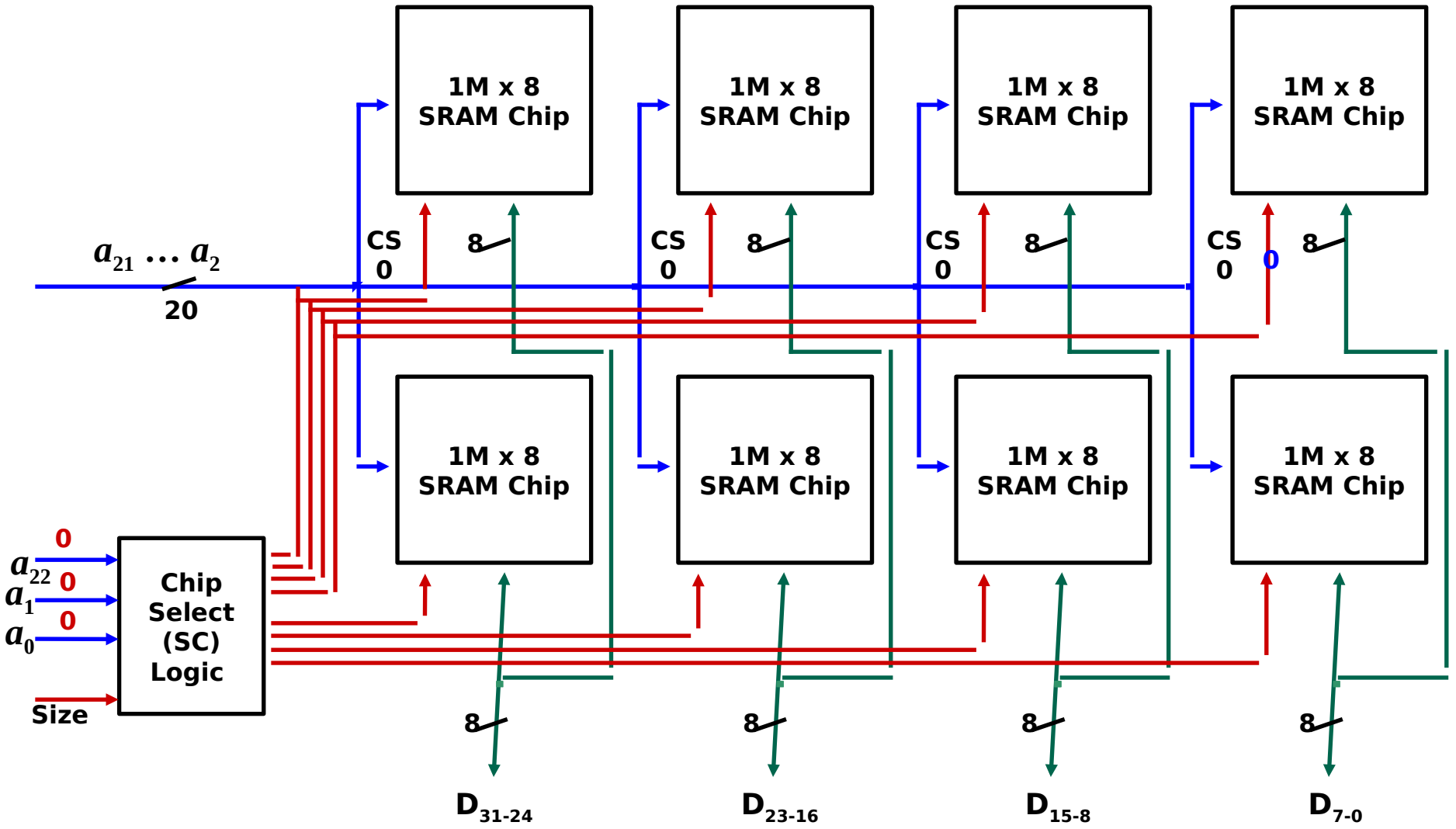


Illustration: 8MB Static RAM Module

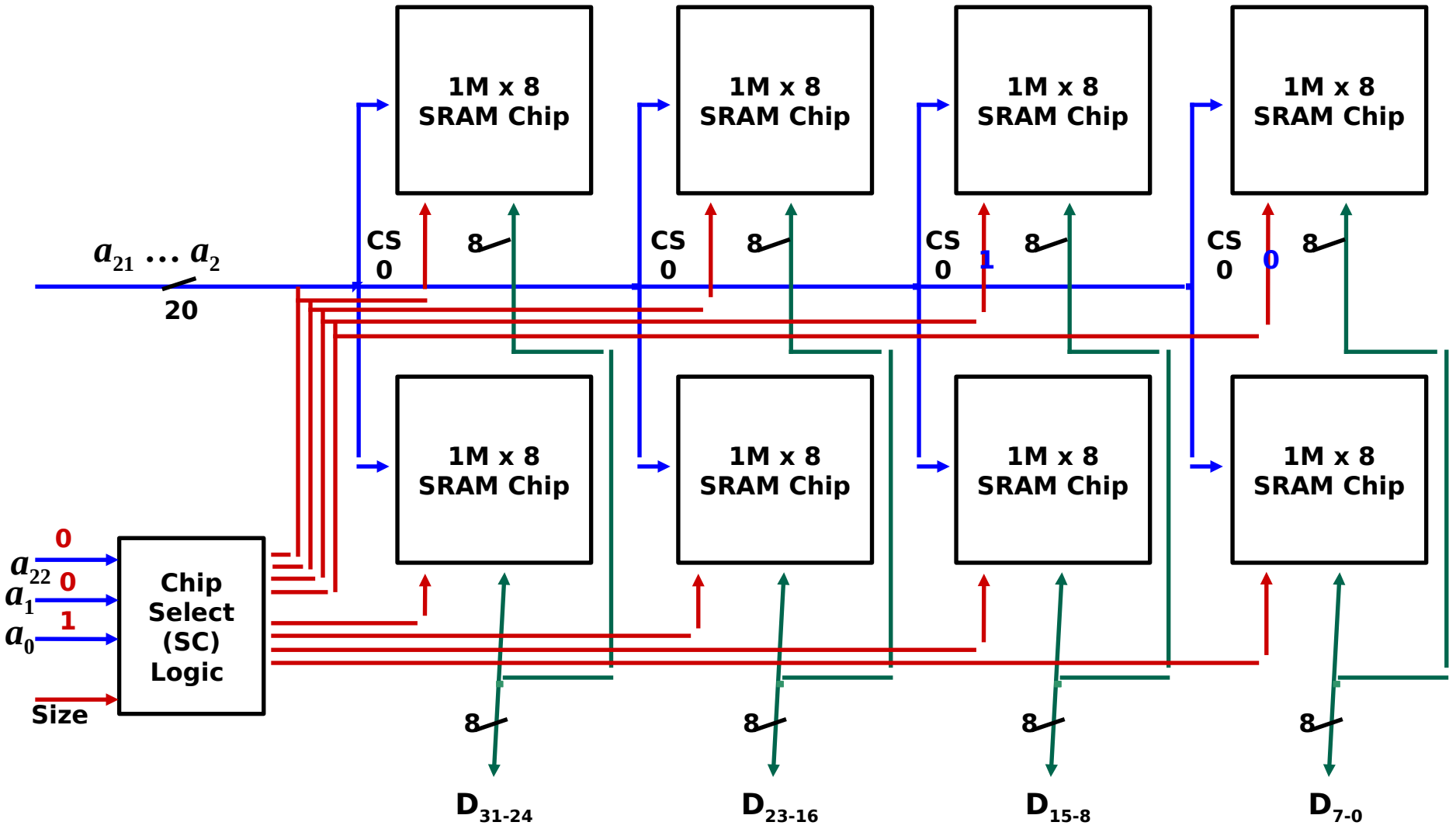


Illustration: 8MB Static RAM Module

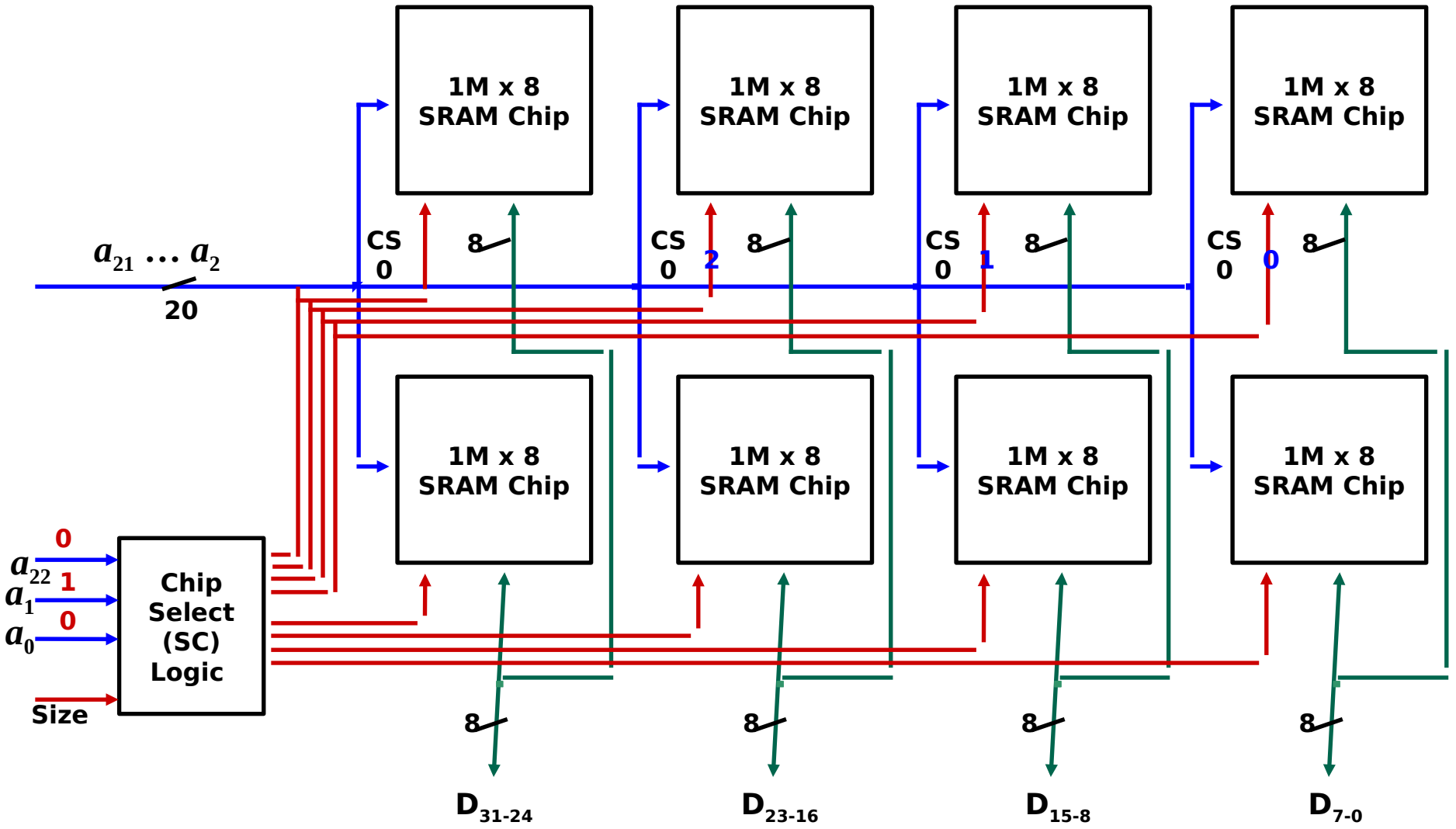


Illustration: 8MB Static RAM Module

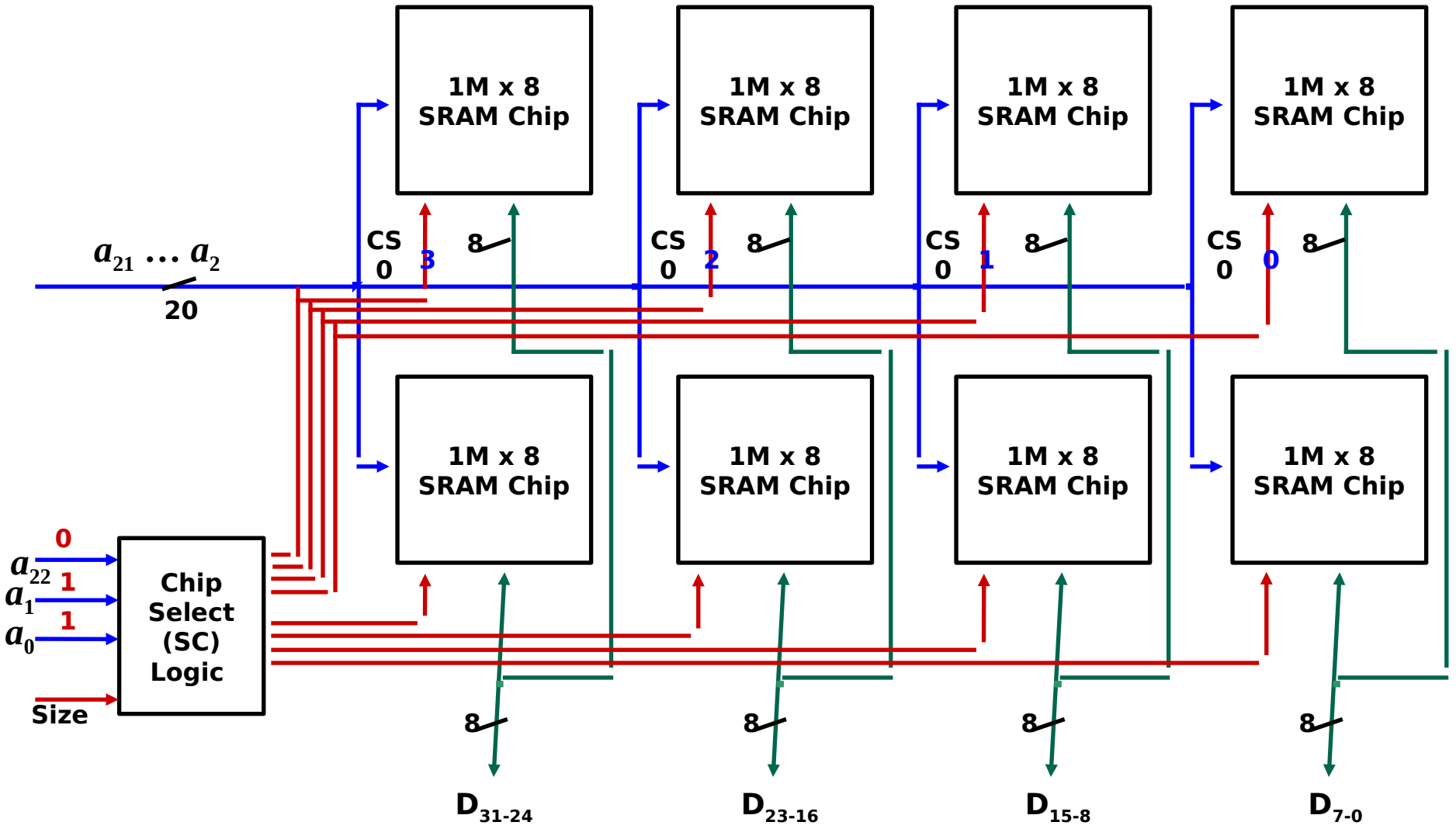


Illustration: 8MB Static RAM Module

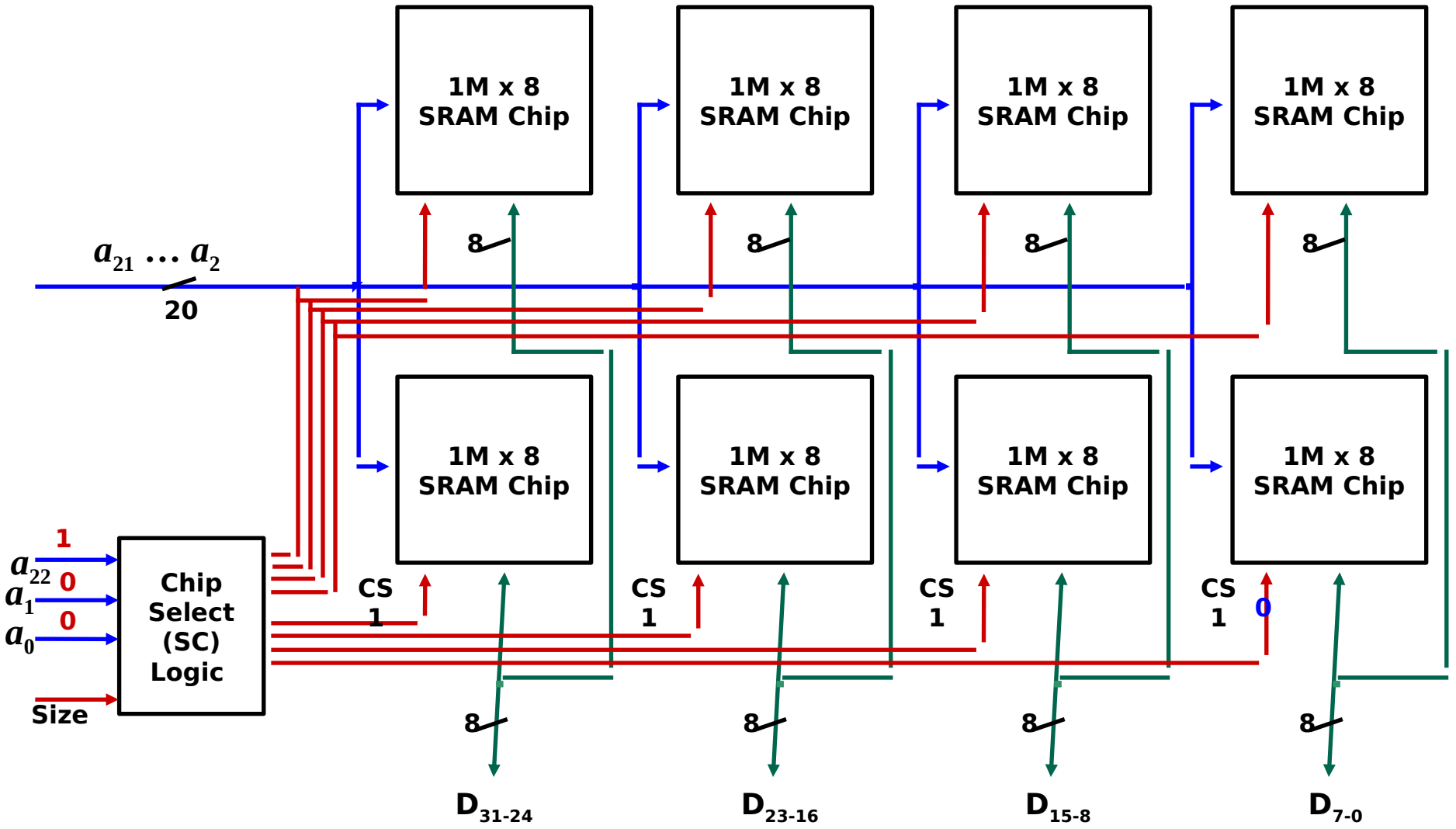


Illustration: 8MB Static RAM Module

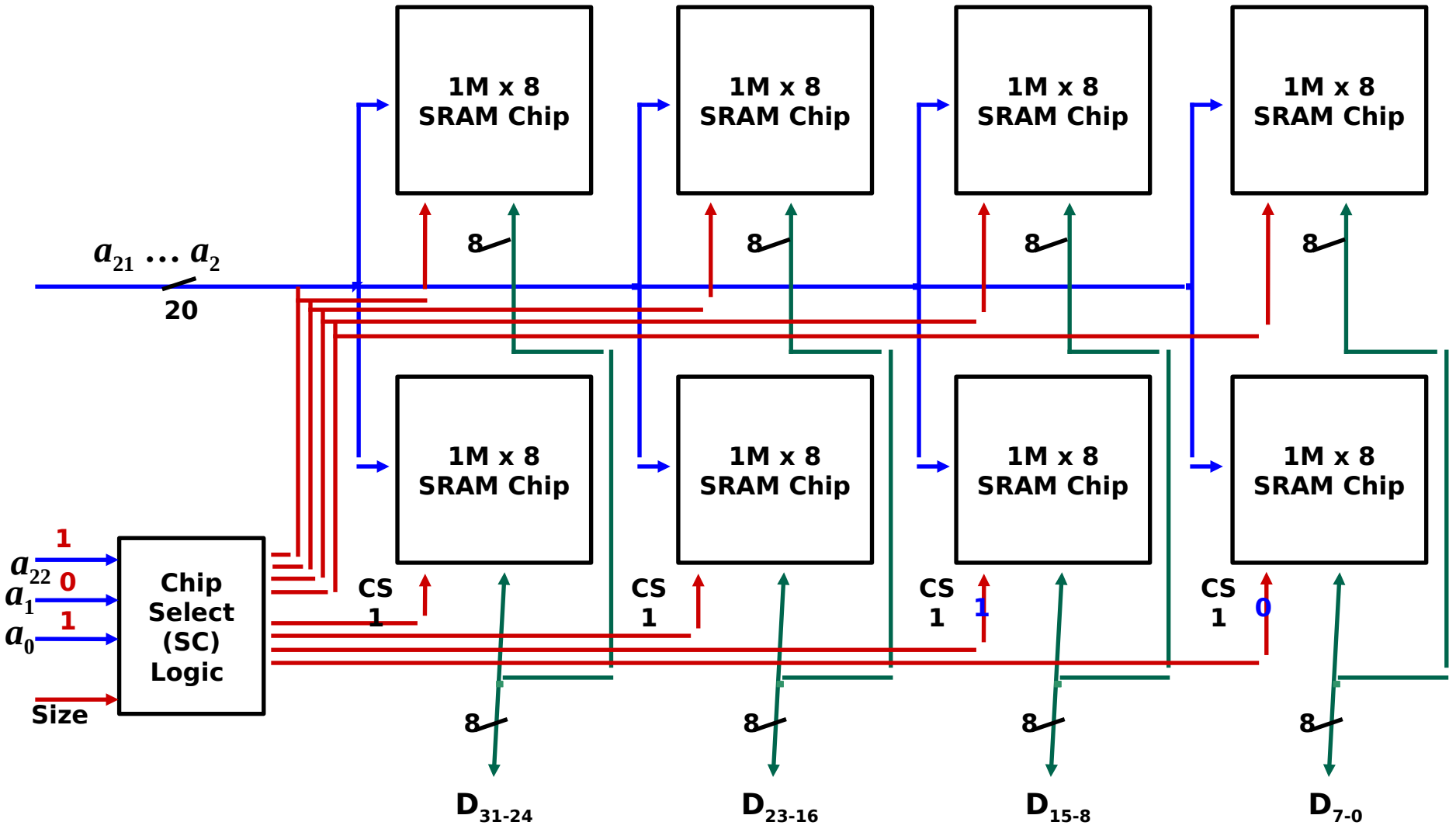


Illustration: 8MB Static RAM Module

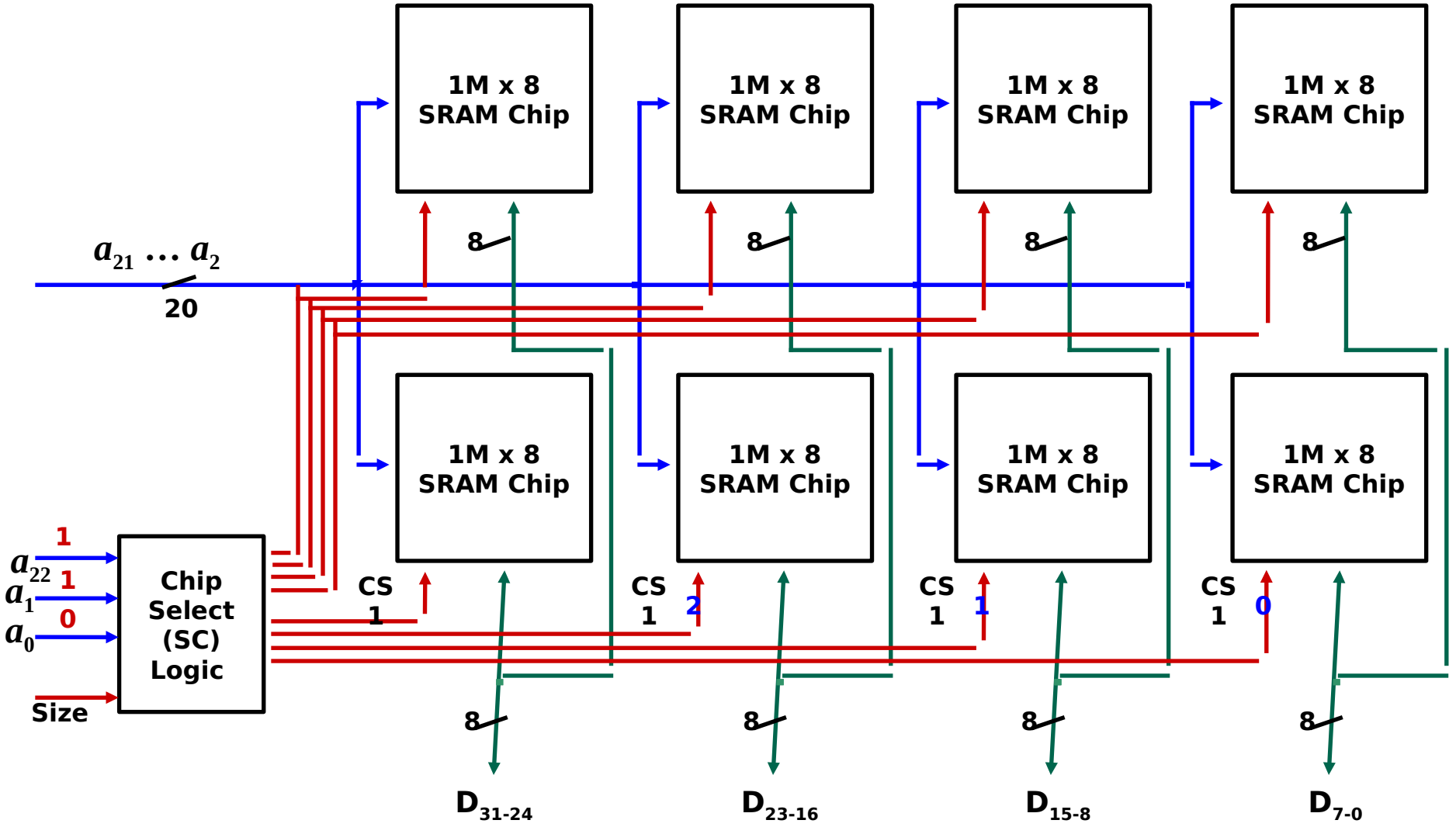
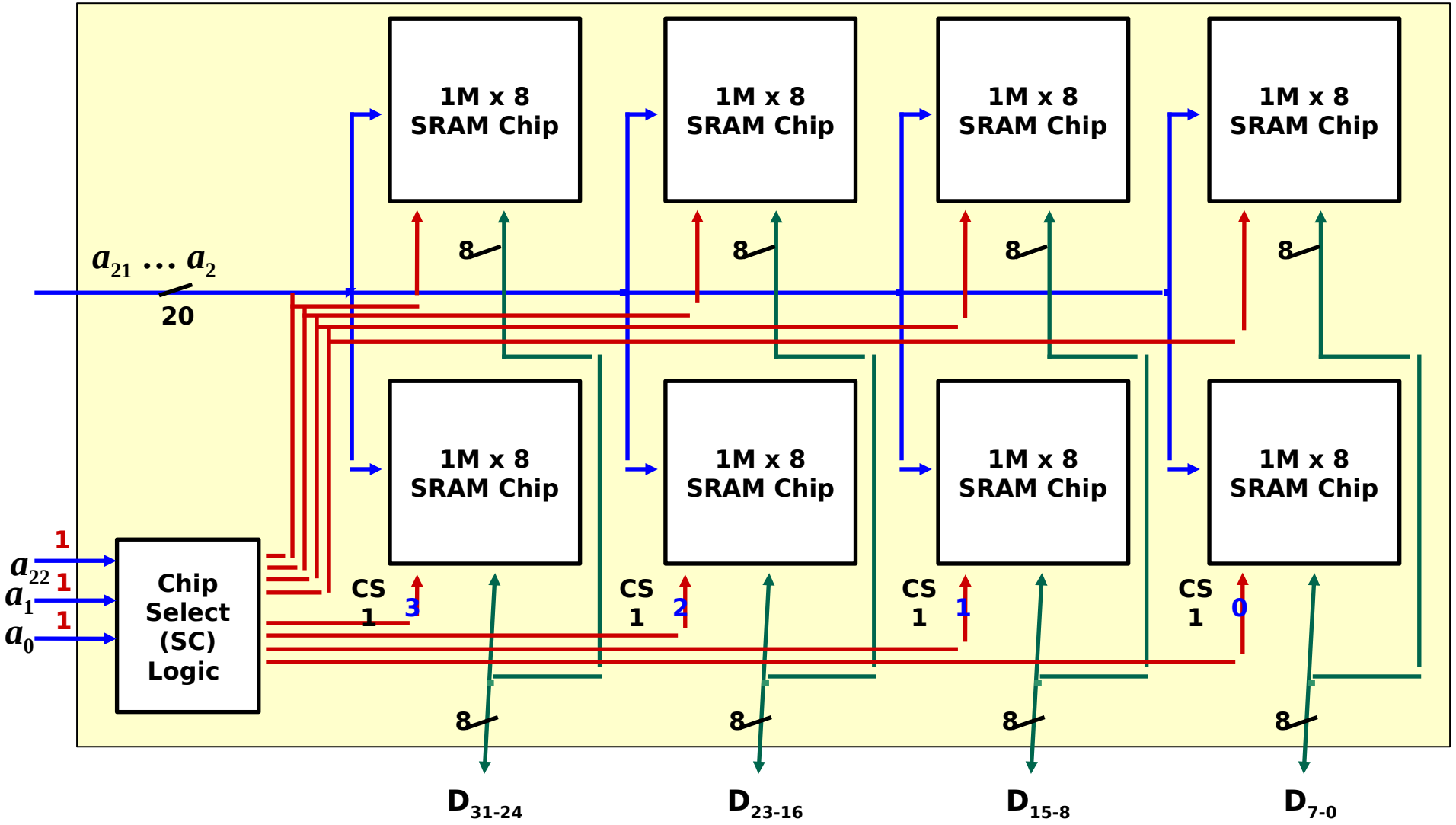
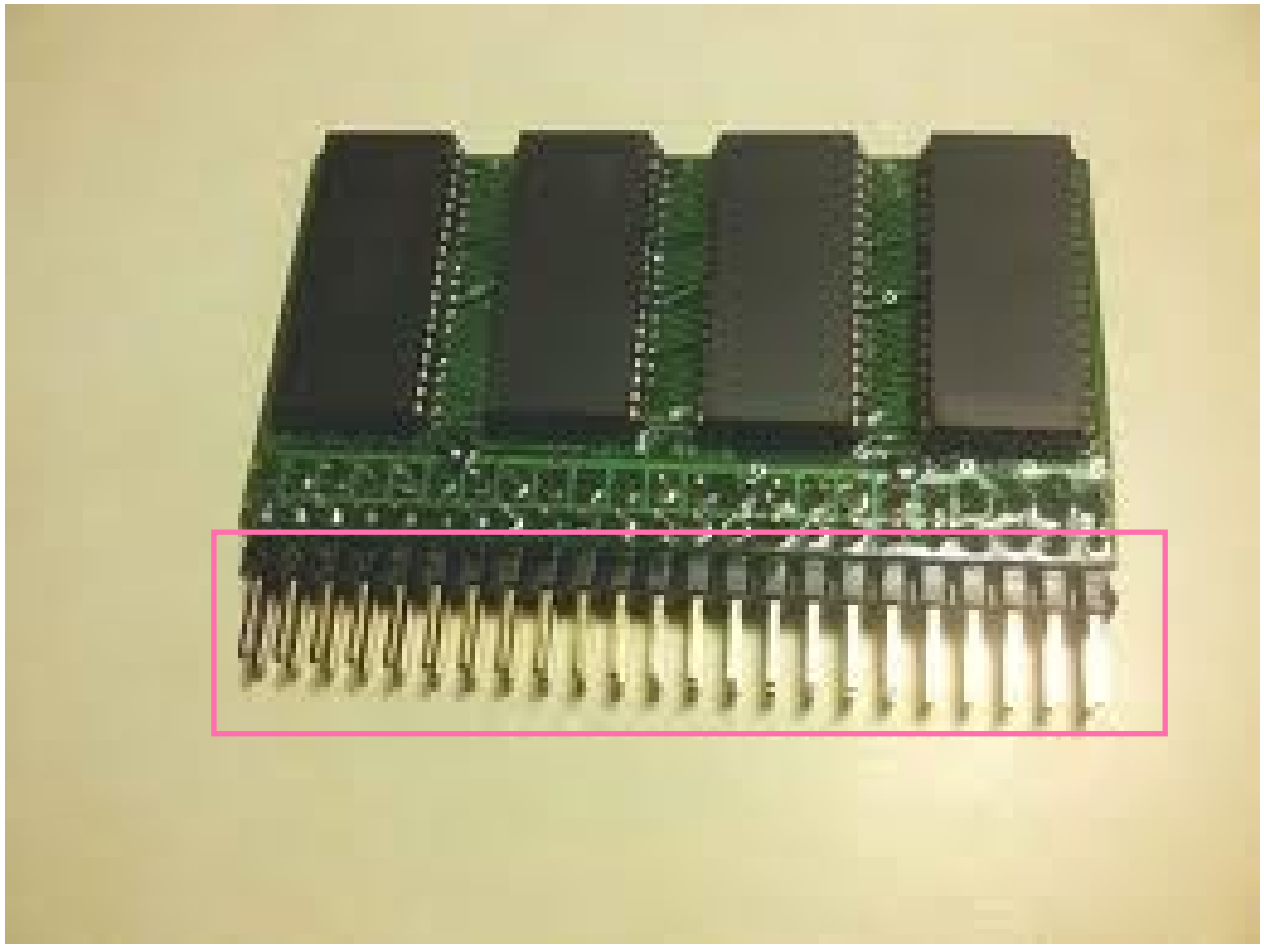


Illustration: 8MB Static RAM Module



Static RAM Module

- **4MB SRAM Module**

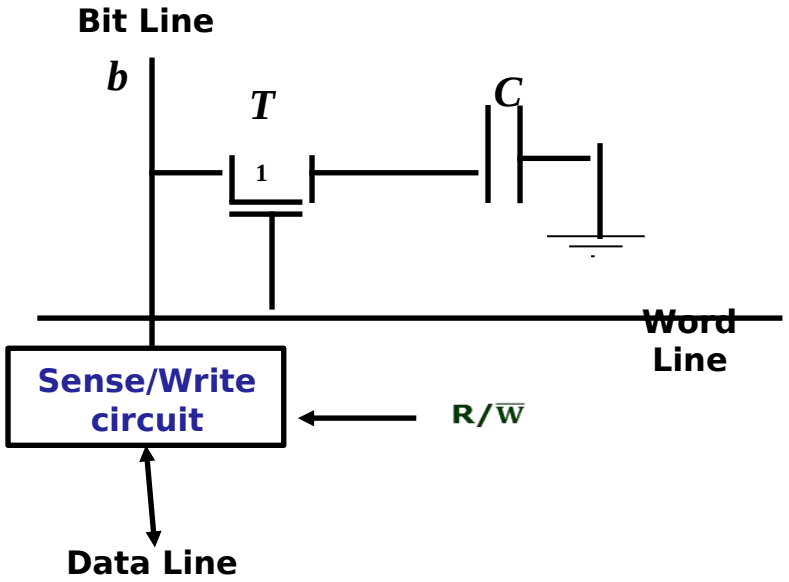


Static RAM Module

- N: Capacity of SRAM
- n bits: Width of SRAM
- M: Capacity of one SRAM chip
- m bits: Width of one SRAM chip
- Number of rows of memory chips: N/M
- Number of chips in a row: n/m
- According to the number of rows of memory chips and number of chips in a row, Chip Select (CS) logic is designed
- Higher order bits in address select a row of memory chips
- Lower order bits in address select a byte in a word
- Size line in CS logic indicates how many bytes in a word need to be selected

Dynamic RAM (DRAM) Cell

- Less expensive and simpler cell
- Information is stored in the form of a charge on a capacitor (C)
  - Charge in capacitor is stored only for short time
  - However, a cell is required to store information for a much longer time
- To retain information for longer time, content of capacitor mush be periodically refreshed



- Low speed as refresh needed
- Only 1 transistor is used
- Used to build main memory

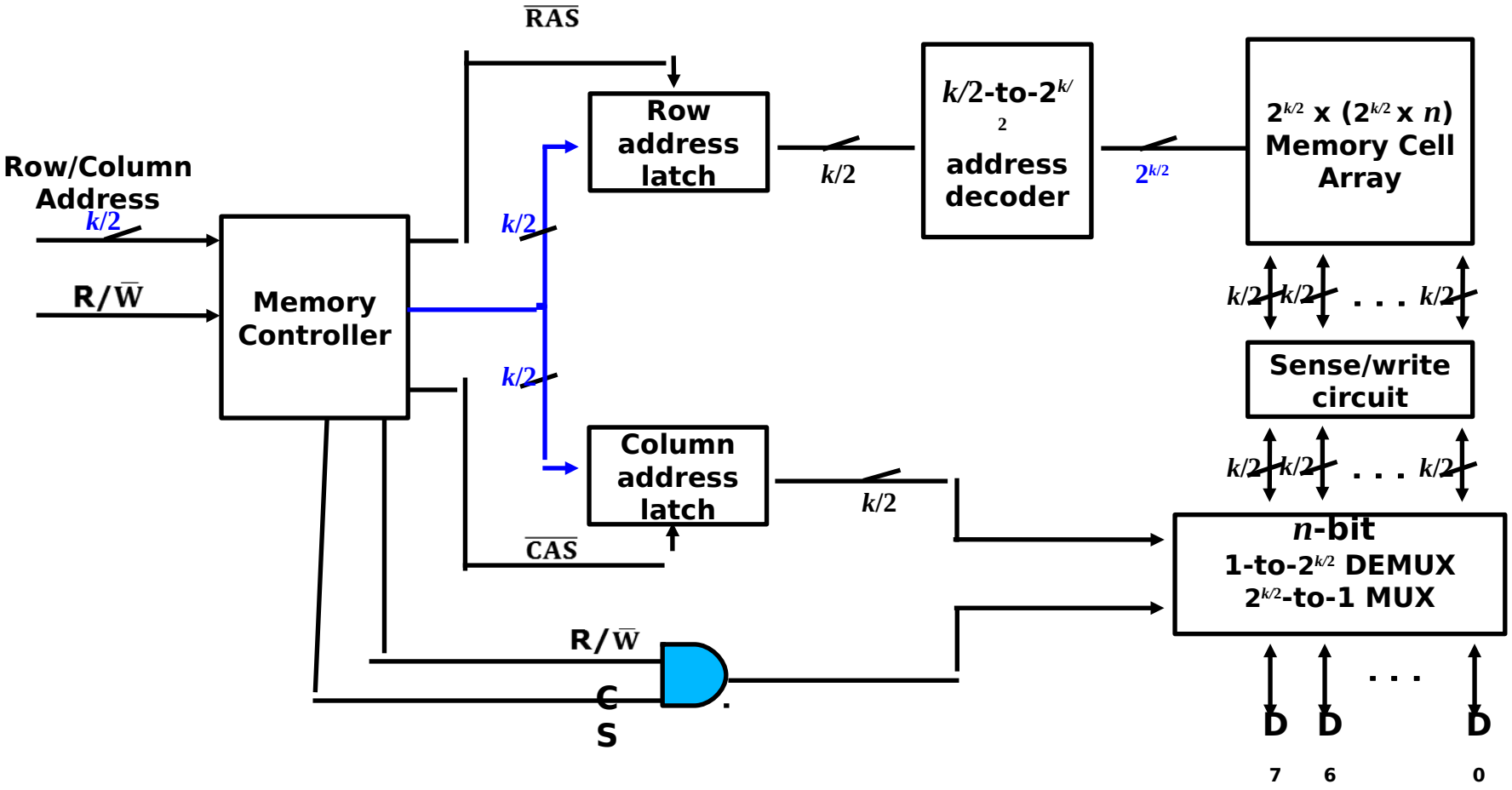
Dynamic RAM (DRAM) Chip Organization

- DRAM cells are also arranged in 2-diemsional array form
- Here also, row address lines are used to select row and column address lines are used to select column
- Two important factors influence the design of the DRAM chip are:
  - Number of input/output pins i.e. external pins
  - Need to refresh the cells
- Scheme for saving pins:
  - Row address and column address are transmitted over the same line one after the other
  - This is called time multiplexing
  - This is usually performed by a memory controller circuit
  - It generates the different control signals

Dynamic RAM (DRAM) Chip Organization

- Two additional control signals are needed to inform the chip when the row address and column address is valid on address line
  - Row address strobe ($\overline{RAS}$):
    - Inform the chip when the row address is valid on address lines
  - Column address strobe ($\overline{CAS}$):
    - Inform the chip when the row address is valid on address lines
  - Both $\overline{RAS}$ and $\overline{CAS}$ are usually active low

DRAM Chip Organization



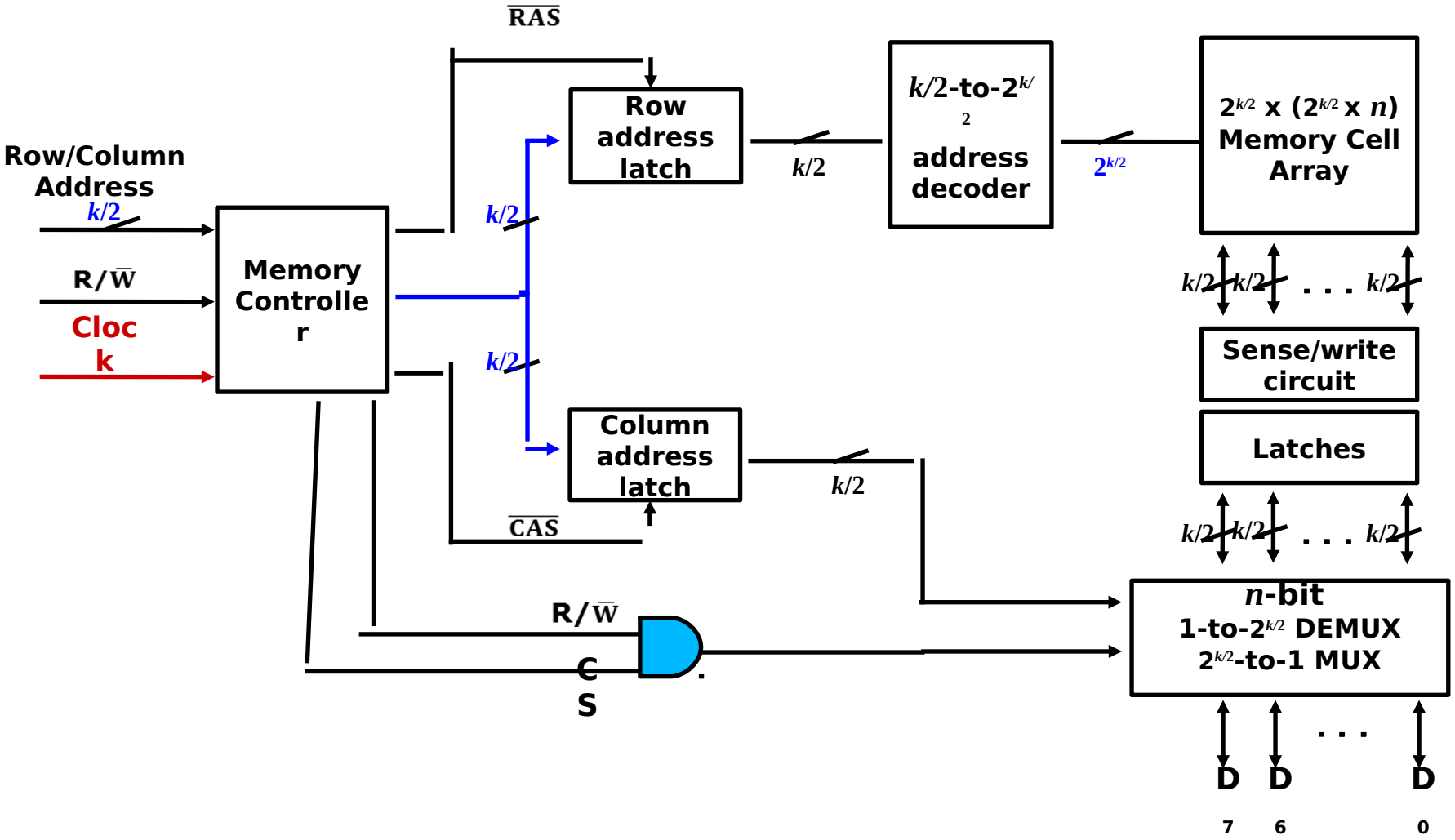
DRAM Chip Organization

- During read or write operation, the row address is applied first
  - It is loaded into row address latch in response to $\overline{RAS}$
  - Then read operation is initiated, in which all cells in a selected row are read and refreshed
- Shortly after row address is loaded, the column address is applied
  - It is loaded onto the column address latch under the control of $\overline{CAS}$
- The information in the column address latch is decoded and appropriate n sense/write circuits are selected
- To ensure that the contents of DRAM are maintained, each row of cells must be accessed and refreshed periodically

DRAM Chip Organization

- Memory controller circuit provide the necessary signals $\overline{CAS}$ and $\overline{RAS}$ that governs timing
- These operations are directly synchronised with clock signal
- Such a DRAM chip is called Synchronous DRAM (SDRAM)

SDRAM Chip Organization



Fast Page Mode Feature of SDRAM

- Transfer capability of SDRAM
- Contents of all $2^{k/2}$ cells in a selected row are sensed in each on the n cell arrays
- Only n bits (one from each of the cells arrays) are placed in the data lines, $D_{7\text{-}0}$
- To access other bytes in the same row, without having to reselect the row, a latch is used at the output of the sense/write circuits in each column
- The row address will load the latches corresponding to all bits in the selected row
- Then different column address are applied to place the different bytes on data lines
- Transfer bytes in sequential order
- This arrangement allows transferring a block of data at much faster rate

Refresh Overhead in SDRAM

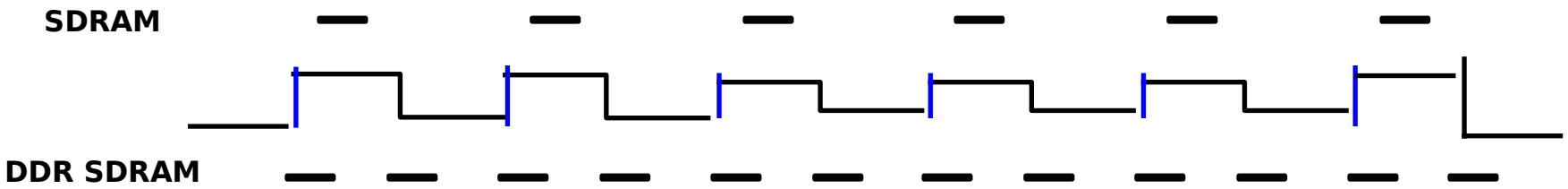
- All dynamic memories need to be refreshed
- In SDRAM typical period of refreshing all rows is ***64ms***
- Each row is refreshed at least in 64*ms*
- Example:
  - *Suppose a SDRAM chip has 8K (8192) rows*
  - *Number of clock cycles to access each row:* 4 clock cycles
  - Number of clock cycles to refresh all rows:

    8192 x 4 = 32768 clock cycles
  - *Suppose clock rate is* 133 MHz
  - Times needed to refresh all rows:

    $32768/133x10^{6} = 246x10^{-6}\ s = 0.246\ ms$
  - **Refreshing overhead is 0.246*ms* out of 64*ms***

Double-Data-Rate SDRAM (DDR SDRAM)

- Faster version of SDRAM
- The standard SDRAM performs all actions on the raising edge of the clock cycle
- DDR SDRAM access the cell array in the same way, but transfers the data on both edges of the clock
- Hence, their bandwidth is essentially doubled for long burst transfers
- Bandwidth: The number of bits/bytes that can be transferred in one second



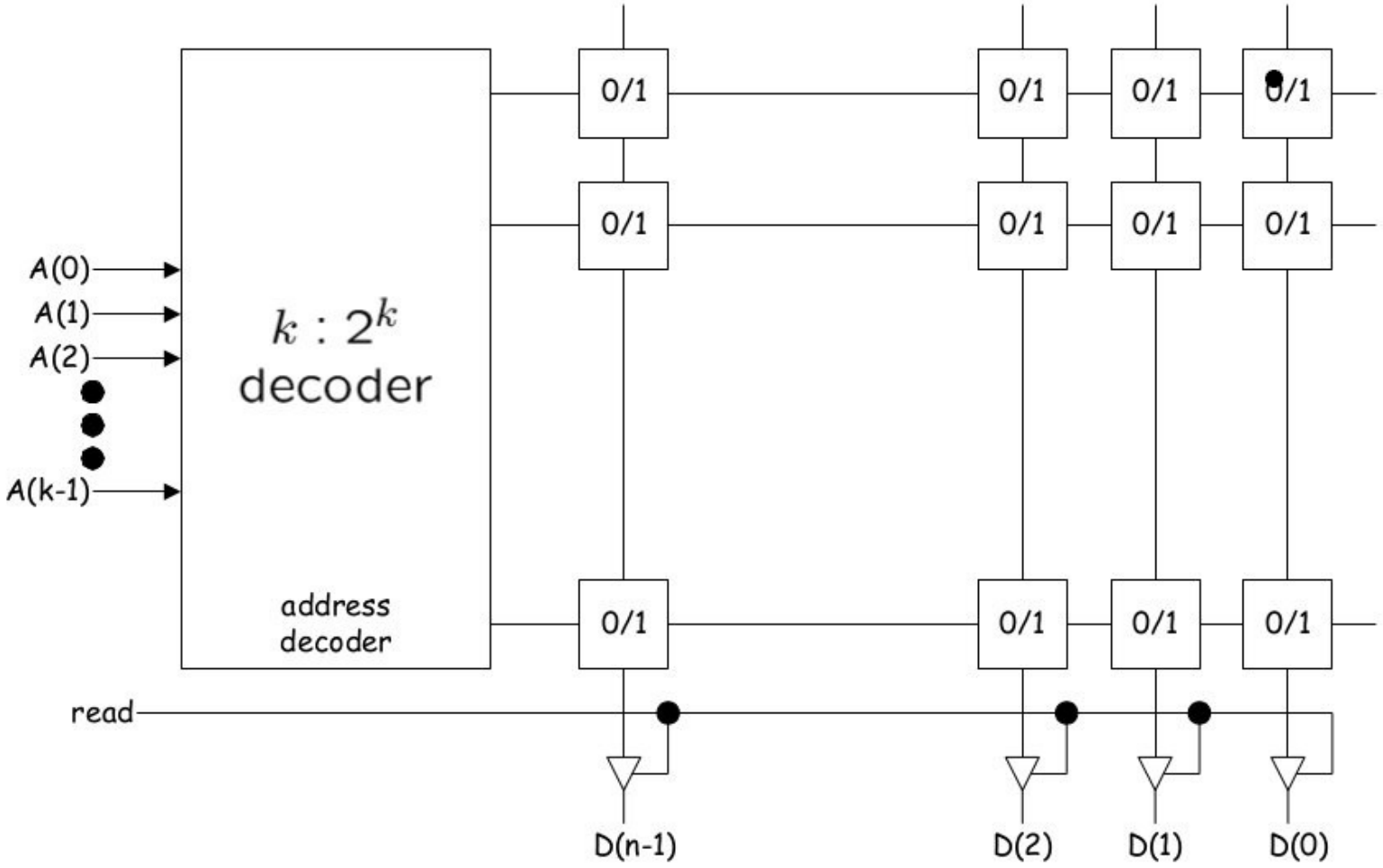
BM 33L5039 RAM Module

(1 GB, DDR2 RAM, 266 MHz, DIMM 184-pin)

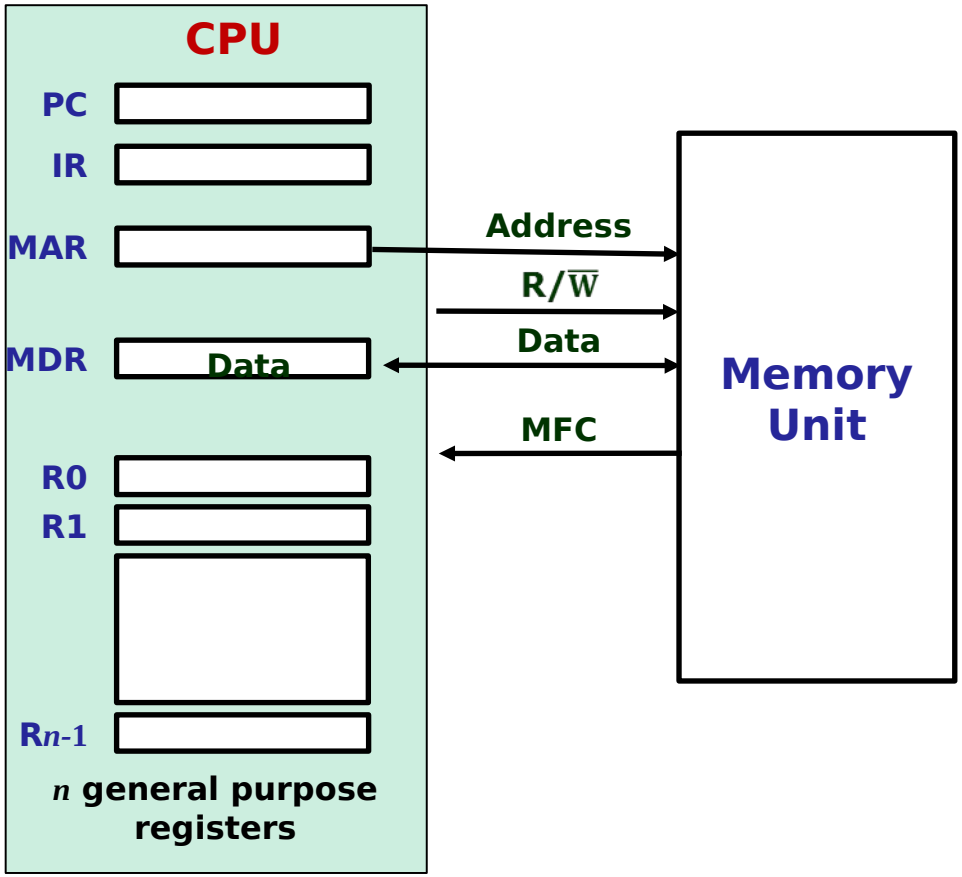
Read-Only Memories (ROMs)

- SRAM and DRAM are volatile i.e. they loose the stored information if power is turned off
- Read-only memories are semiconductor, non-volatile memories
- Their normal operation involve only reading the stored data
- They are extensively used in embedded systems
- Different types of ROMs
  - Read Only Memory (ROM)
  - Programmable ROM (PROM)
  - Erasable, reprogrammable ROM (EPROM)
  - Electicrally erasable reprogrammable ROM (EEPROM)
  - Flash memory

Read-Only Memories (ROMs)

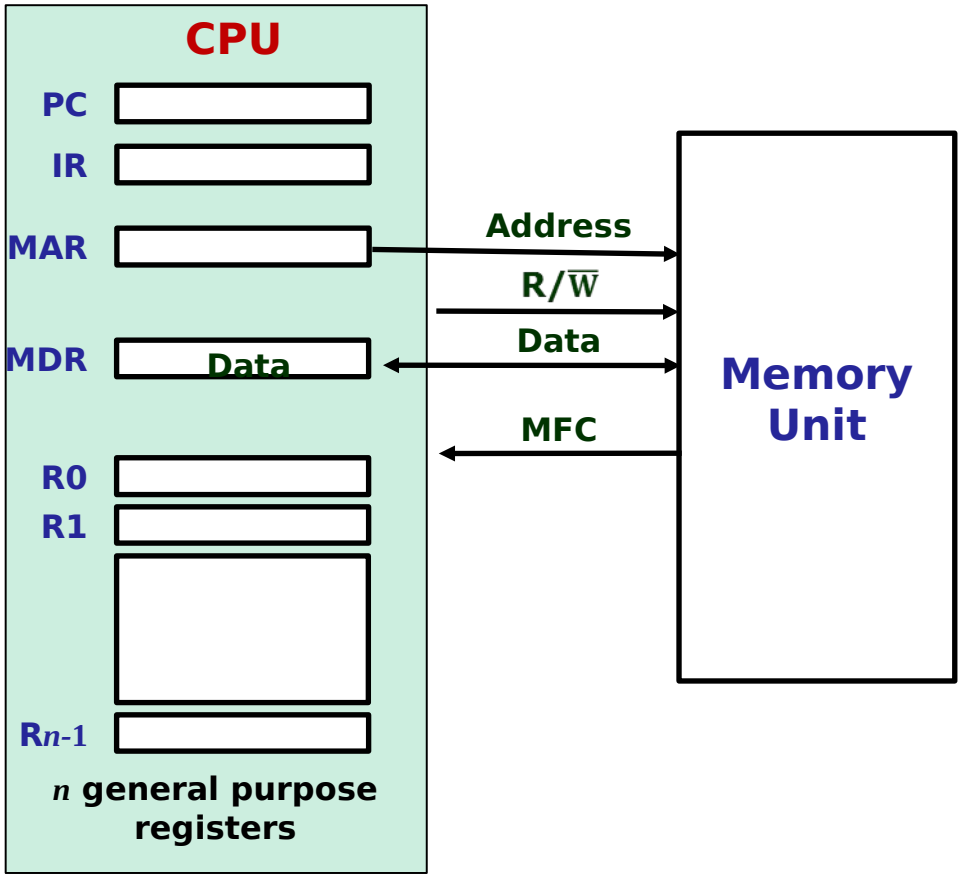


Memory Read and Write Operation



- **Read**
  - Processor loads the address of memory location into MAR
  - Set the R/$\overline{W}$ line to 1
  - Memory responds by placing the data from address location onto data line
  - Confirm the action by asserting MFC (memory function complete) signal
  - Upon receiving MFC signal, processor loads the data on data line into MDR

Memory Read and Write Operation

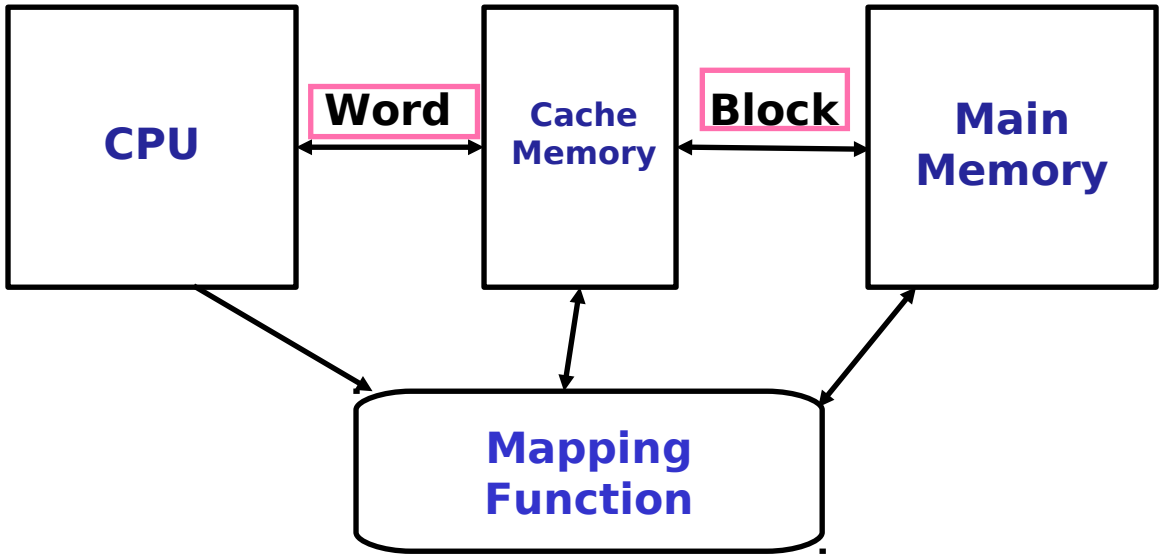


- **Read**
  - Processor loads the address of memory location into MAR
  - Set the R/$\overline{W}$ line to 1
  - Memory responds by placing the data from address location onto data line
  - Confirm the action by asserting MFC (memory function complete) signal
  - Upon receiving MFC signal, processor loads the data on data line into MDR

Cache Memory

- The cache memories are designed to exploit the locality of reference in the program
- **Locality of reference**:
  - Many instructions in localized areas of the program are executed repeatedly during some time period, and the remainder of the program is accessed relatively infrequently
- Different ways of locality of reference
  1. Temporal locality:
     - Recently executed instruction/data is likely to be used again very soon
  2. Spatial locality:
     - Instructions in close proximity to a recently executed instruction/data (with respect to instruction address) are likely to be executed very soon

Use of a Cache Memory



- Unit of transfer between main memory and cache is block
  - A block is a set of fixed number of words in contiguous address locations
  - Cache block is also called as Cache line
- Mapping function: Correspondence between main memory blocks and those in the cache

Replacement Algorithm

- Read hit/Write hit [Cache hit]:
  - When processor issues read or write request, cache control circuit determines whether the requested word exits in cache
  - If it exists in cache, the read or write operation is performed on the appropriate cache location
  - This means, read hit or write hit is said to have occurred
- Cache miss:
  - If the desired word is not there in cache during read/write operation, then cache miss is said to have occurred
  - During that time new block need to be brought into cache
- Cache control hardware decide which block to removed to create space for new block that contain referenced word when cache is full
- The collection of rules for making this decision constitutes the replacement algorithms

Read and Write Operations on Cache

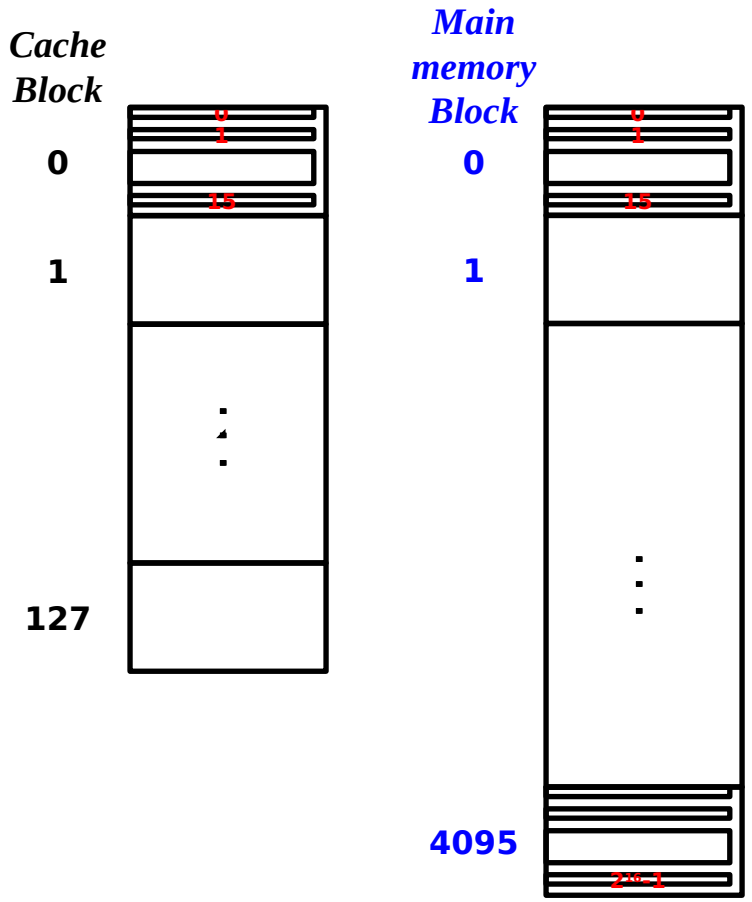
- Read operation:
  - Handling read miss:
    - Approach 1:
      - The block of words that contains the requested word is copied into the cache
      - After entire block is loaded into cache, particular requested word is forwarded to processor
    - Approach 2: Load through or early restart
      - A word is sent to processor as soon as it is read from the main memory
      - At the same time, the block of words that contains the requested word is also copied into the cache
      - This approach reduces the processor waiting period, but with the expense of complex circuitry

Read and Write Operations on Cache

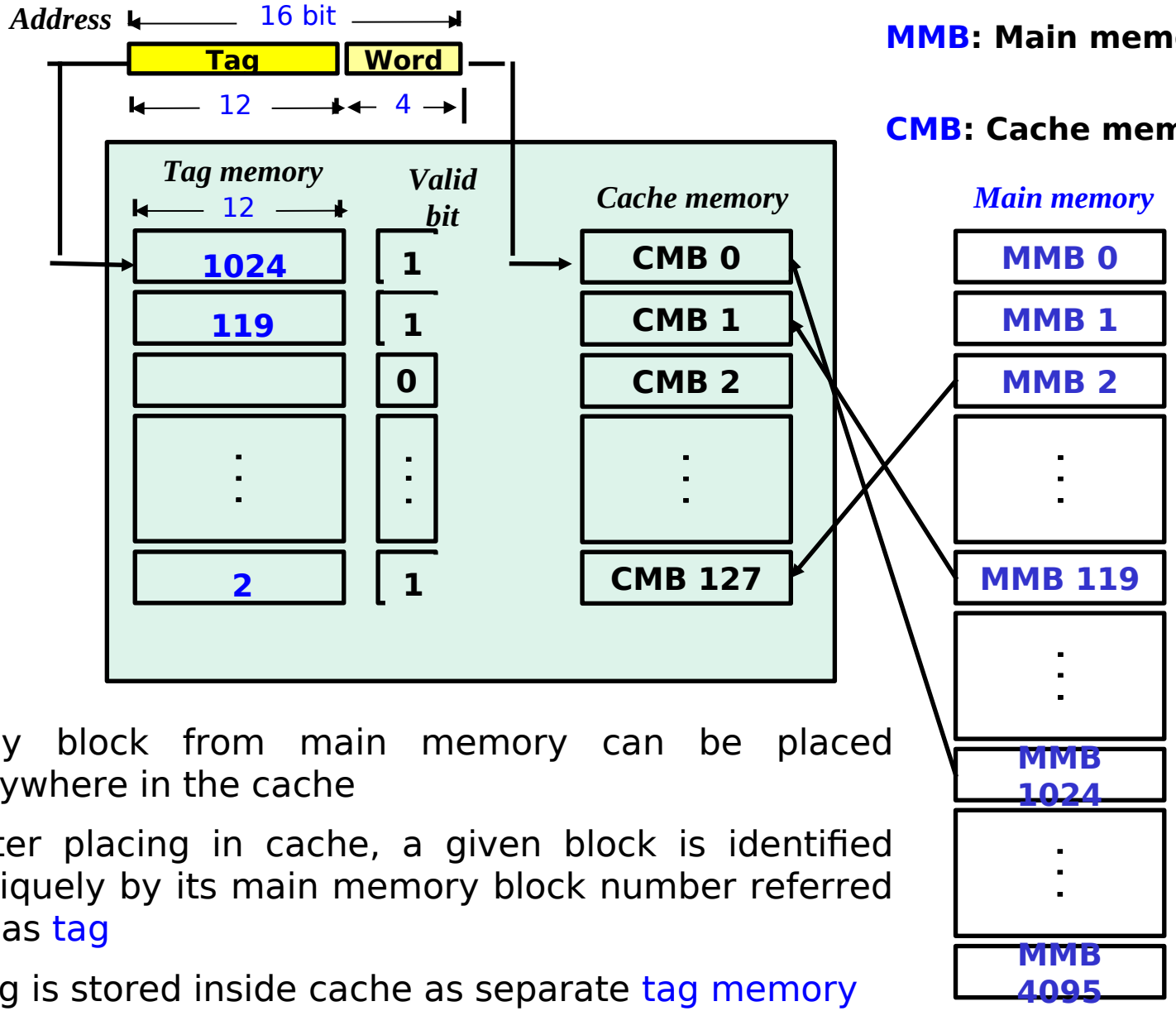
- Write operation:
  - Two techniques for write operation:
    - Write-through protocol
      - The cache location and main memory location are updated simultaneously
    - Write-back (copy-back) protocol
      - Update only the cache location and mark it as updated in a flag bit called dirty bit or modified bit
      - The main memory location is updated later when the block containing that modified word need to be removed from the cache
  - Handling write miss:
    - Write-through protocol:
      - The information is written directly into the main memory
    - Write-back (copy-back) protocol:
      - The block containing the addressed word is first brought into the cache
      - Then the desired word in the cache is overwritten with new information

Mapping Function

- Specifies where memory blocks are placed in the cache
- Cache and main memory are viewed as collection of fixed number of blocks
- Example: Consider a cache and main memory with 2K words and 64K words respectively and each blocks are of size 16 words
  - Block size: 16 words
  - Number of blocks in a cache, $N = 2K/16 = 128$
  - Number of blocks in main memory, $M = 64K/16 = 4K = 4096$
  - Addressable location in main memory: 2^{16}



Mapping Function:
Associative Mapping (Associative Mapped Cache)



MMB: Main memory block

CMB: Cache memory block

- Any block from main memory can be placed anywhere in the cache
- After placing in cache, a given block is identified uniquely by its main memory block number referred to as tag
- Tag is stored inside cache as separate tag memory