Memory Subsystem

Ashish Kuvelkar
Hardware Technology Development Group
C-DAC, Pune

Memory

Why is the memory required?

- Temporary storage
- Break the problem in time domain

How the data can be stored ?

- Charge
- Voltage
- Mechanical action
- Light
- Magnetisation

Memory

What is the choice criteria ?

- Basic
  - Reliability
  - Size
  - Speed
  - Power
  - Type of use
- Secondary
  - Package type,
  - cost, availability,
  - Scalability
  - voltage compatibility(3.3/5V)

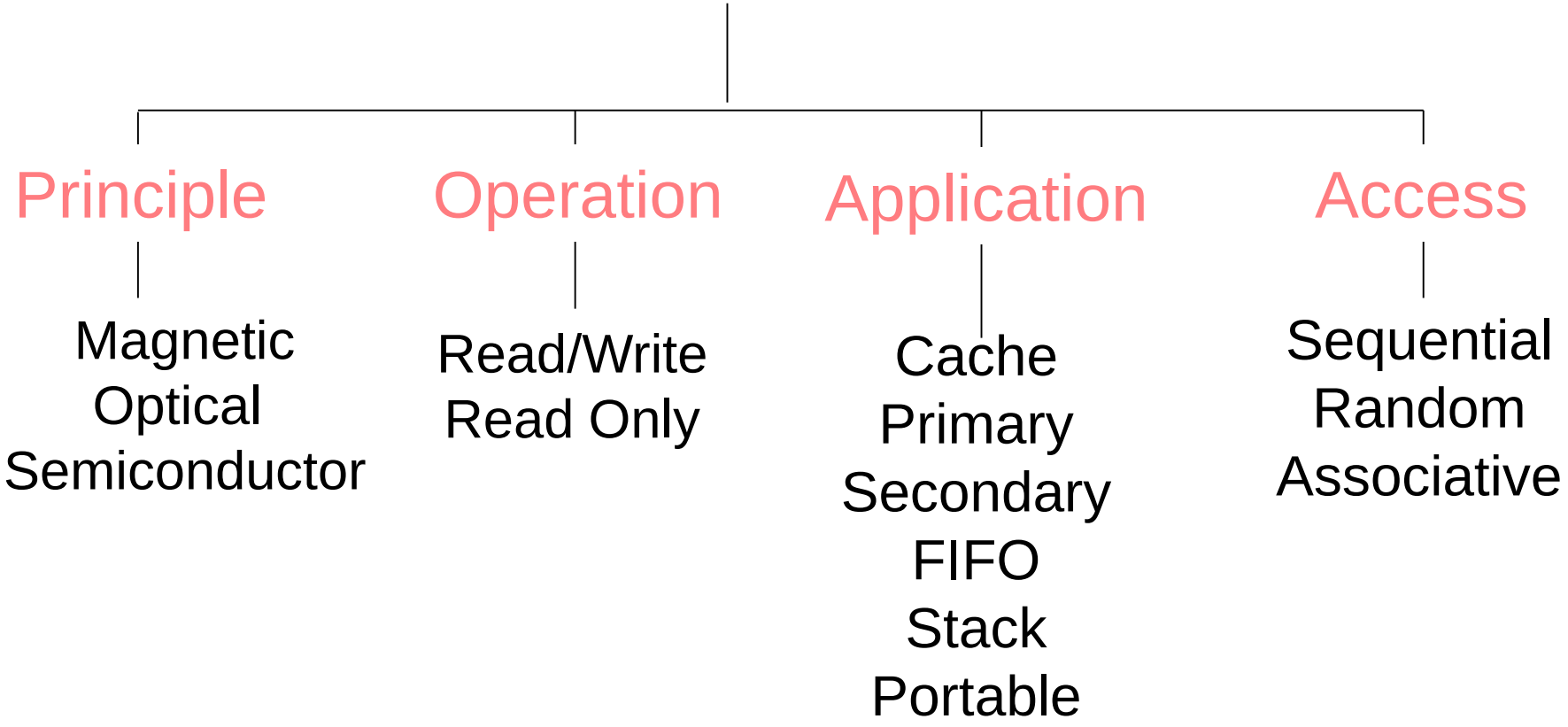
Memory

What are the components ?

- Basic
  - Storage element
  - Address decoder
  - Data line driver

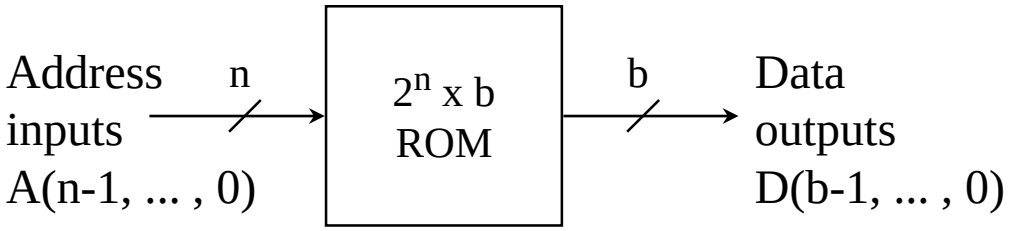
- Optional
  - Data multiplexers
  - Sense amplifiers
  - Voltage equalizers
  - Refresh circuits

Memory Types



Read-Only Memory (ROM)

- A combinational circuit with n inputs and b outputs:



- Programmable - values determined by user
- Nonvolatile - contents retained without power

Read-Only Memory (ROM)

- Two views:
  - ROM stores 2^n words of b bits each, or
  - ROM stores an n-input, b-output truth table

Example:

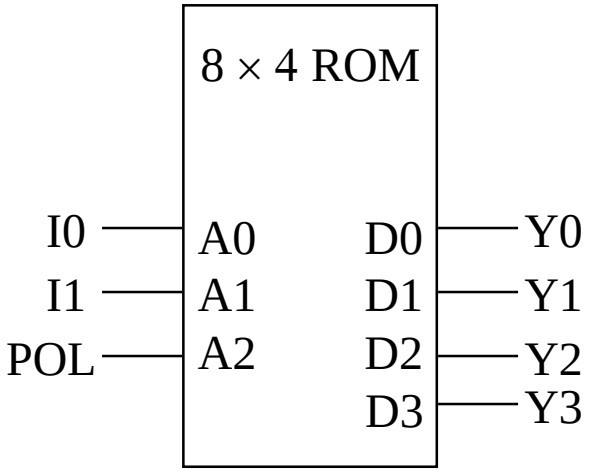
n = 2		b = 4			
A1	A0	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	1
1	1	1	0	1	0

⇦ Stores 4 4-bit words, or stores 4 functions of 2 input variables

Using ROMs for Combinational Logic

A 3-input, 4-output combinational logic function:

Inputs			Outputs			
A2	A1	A0	D3	D2	D1	D0
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



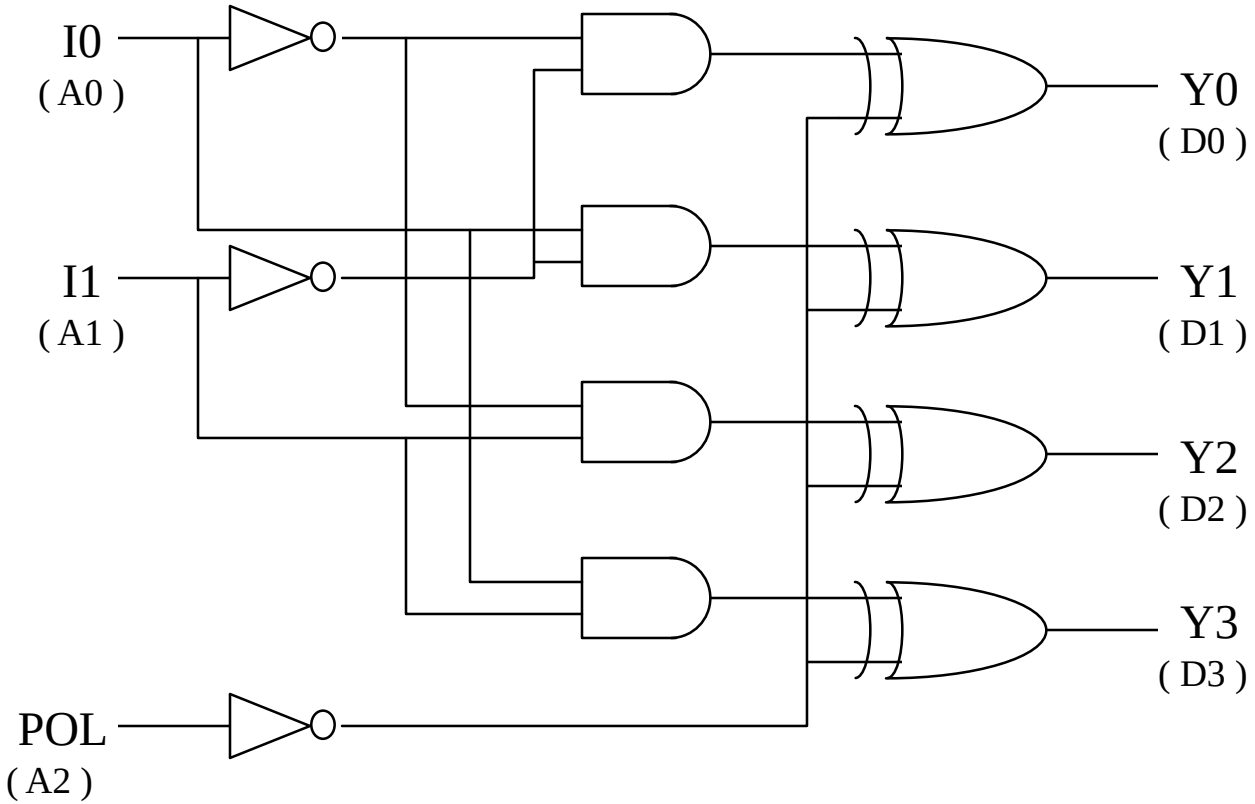
Function: 2-to-4 Decoder with Polarity Control

A2 = Polarity (0 = active Low, 1= active High)

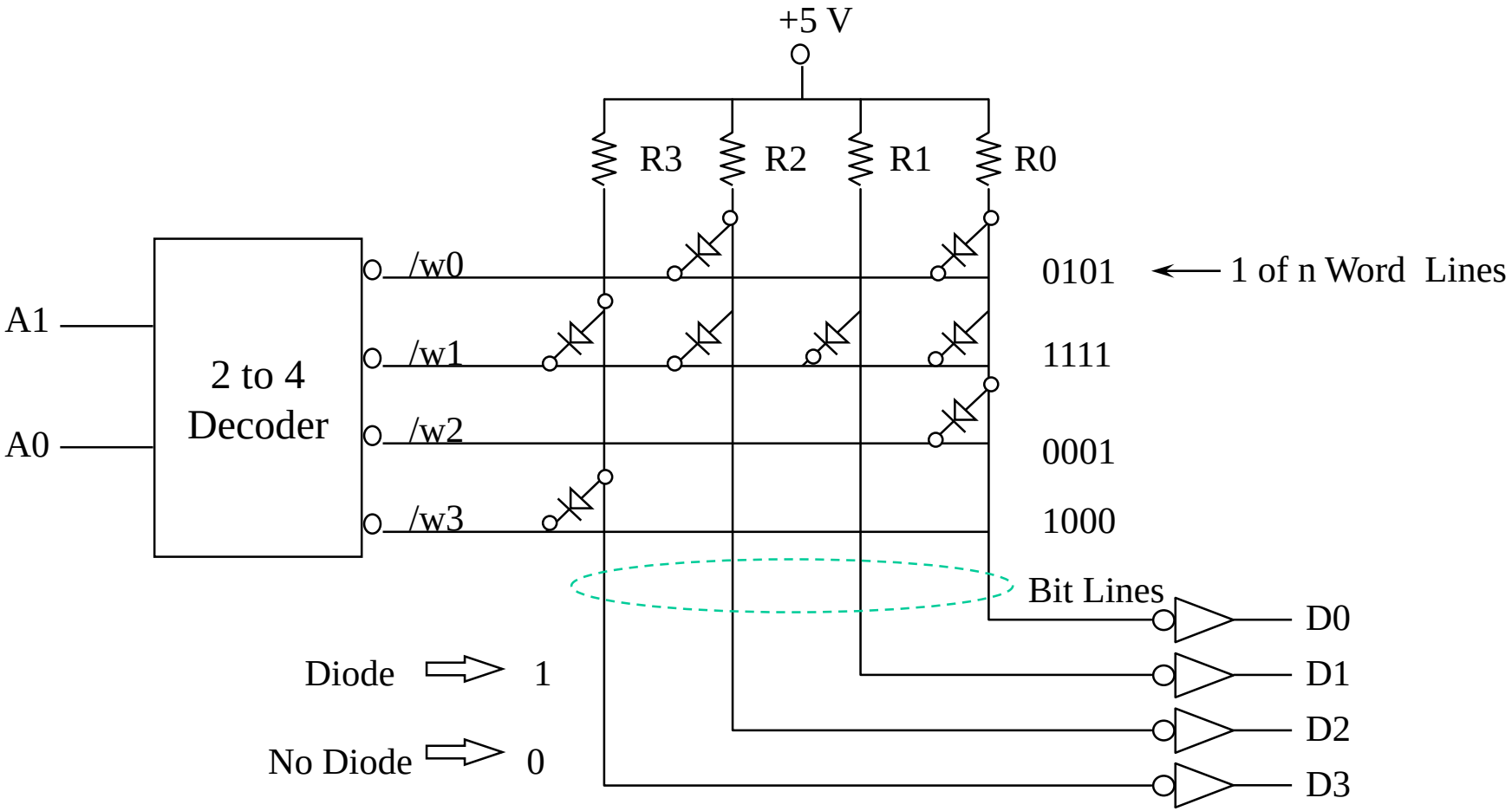
A1, A0 = I1, I0 (2-bit input)

D3...D0 = Y3...Y0 (4-bit decoded output)

Equivalent Decoder in Discrete Logic

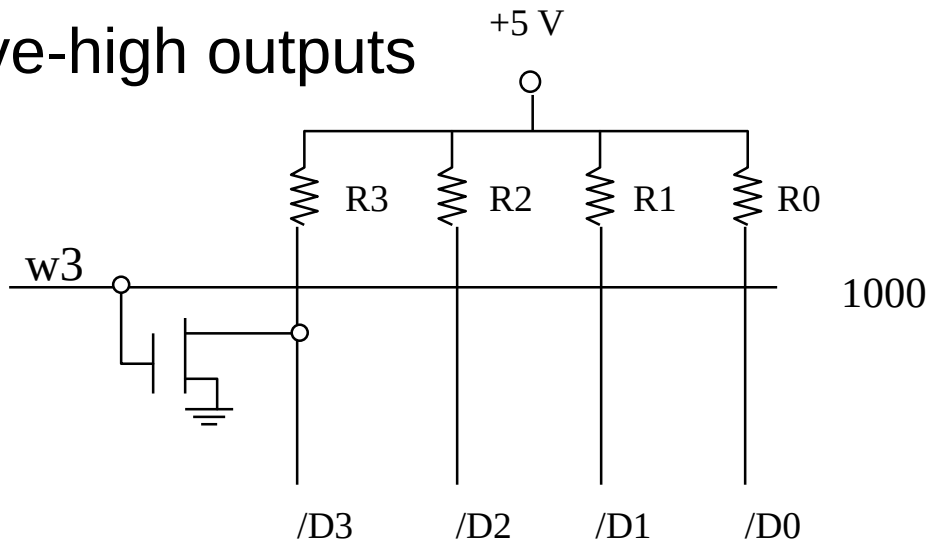


Internal Structure of 4×4 Diode ROM



Internal Structure of MOS Transistor ROM

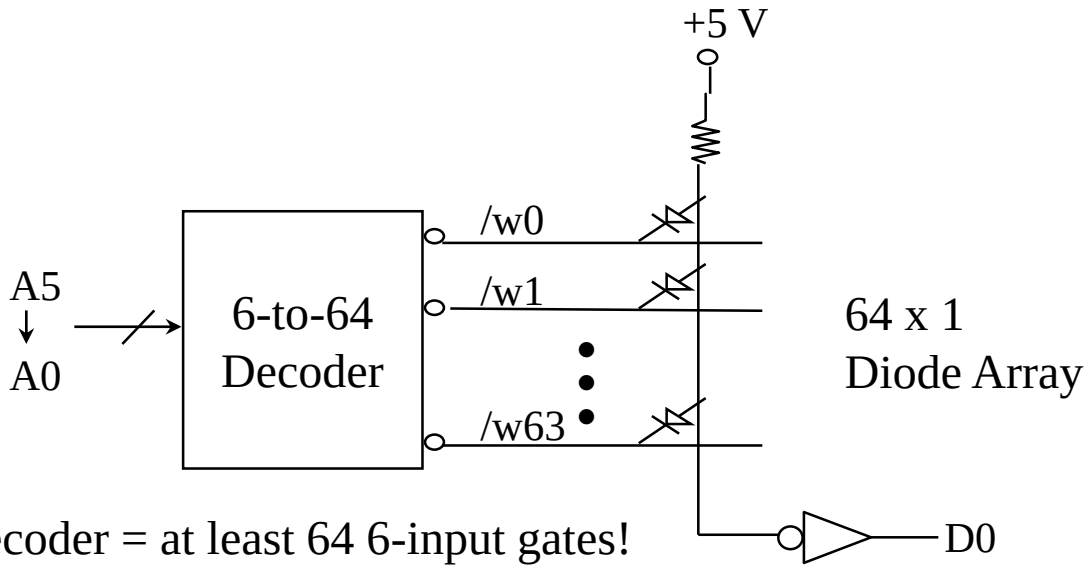
- Replace diodes with MOS transistors
- Change decoder to active-high outputs



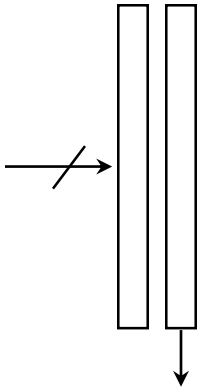
Transistor ⇨ 1

No transistor ⇨ 0

Consider a 64 x 1 ROM



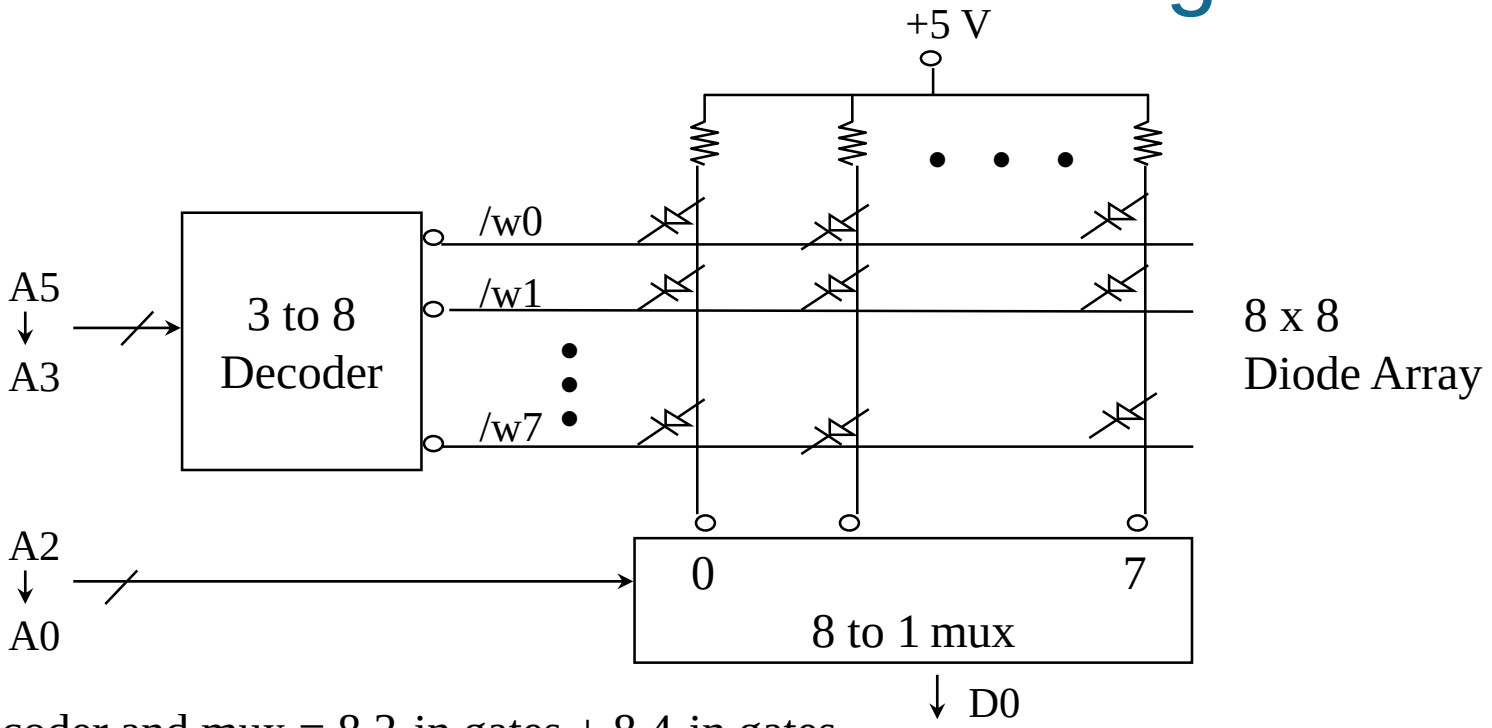
Decoder = at least 64 6-input gates!
Avoidable



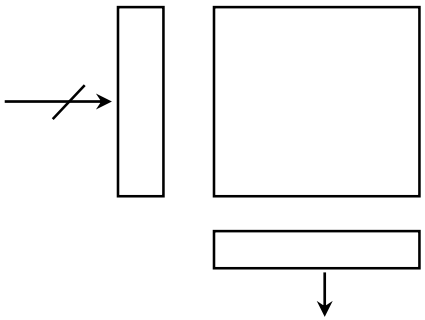
-> Very tall, narrow chip (BAD)
Even worse for larger chips!

How to make it more square?

2-Dimensional Decoding

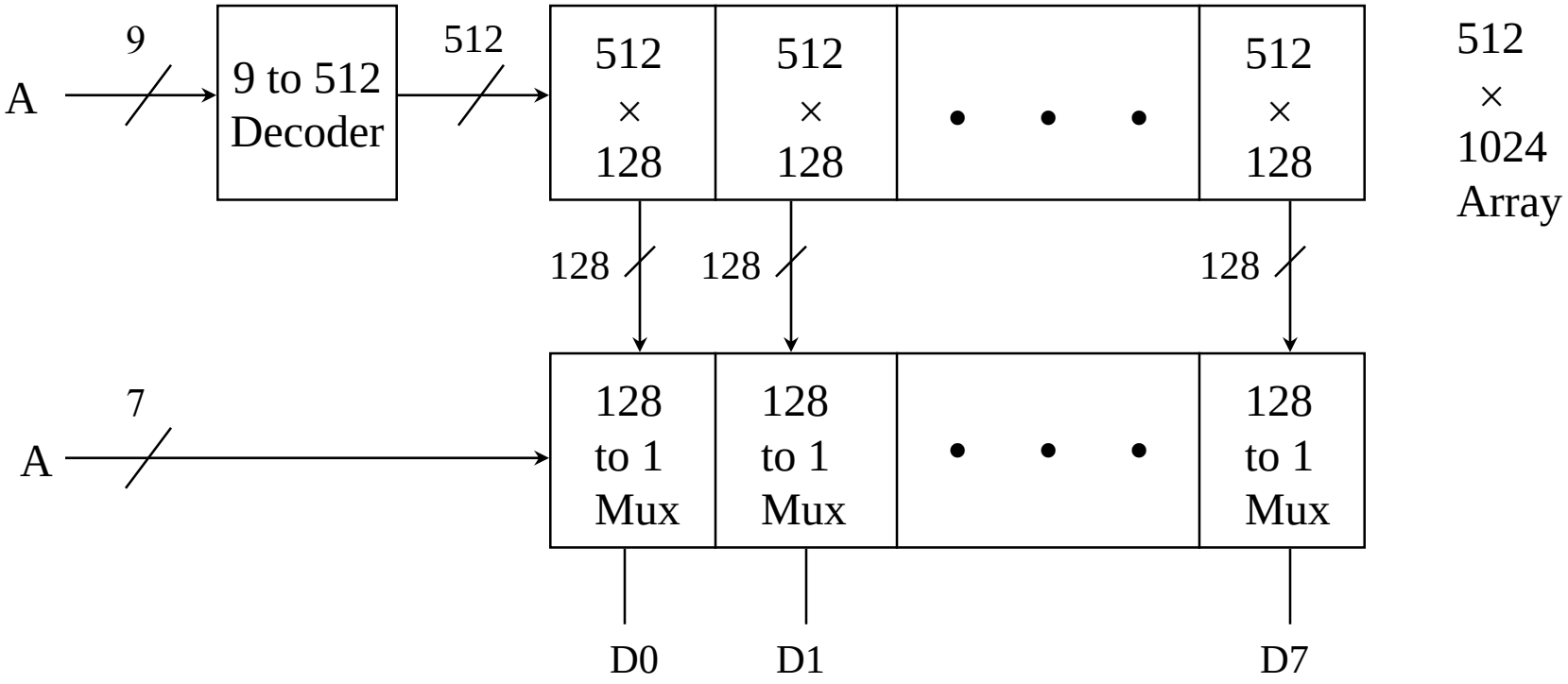


Decoder and mux = 8 3-in gates + 8 4-in gates



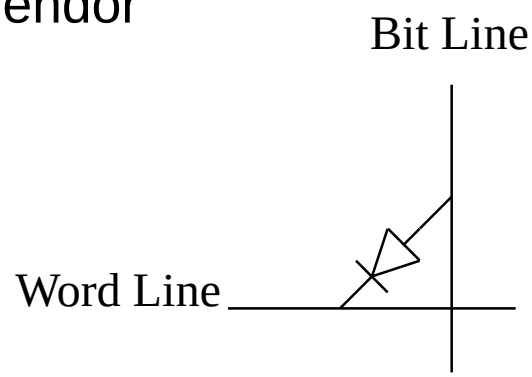
—> Almost square chip (good!)

64K x 8 ROM with 2-D Decoding

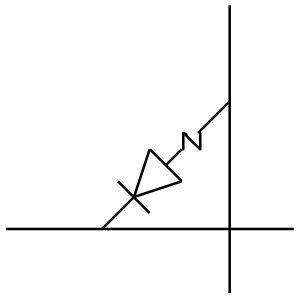


Types Of ROMs (1)

- Mask ROM
  - Connections made by the semiconductor vendor
  - Expensive setup cost
  - Several weeks for delivery
  - High volume only
  - Bipolar or MOS technology

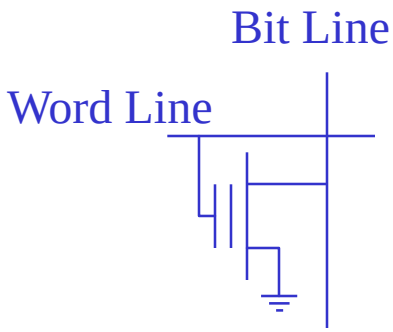


- PROM
  - Programmable ROM
  - Connections made by equipment manufacturer
  - Vaporize (blow) fusible links with PROM programmer using high voltage/current pulses
  - Bipolar technology
  - One-time programmable

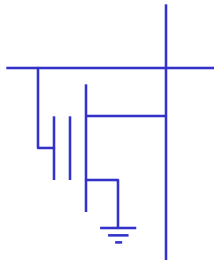


Types of ROMs (2)

- EPROM
  - Erasable Programmable ROM
  - Charge trapped on extra “floating gate” of MOS transistors
  - Exposure to UV light removes charge
    - 10-20 minutes
    - Quartz Lid = expensive package
  - Limited number of erasures (10-100)



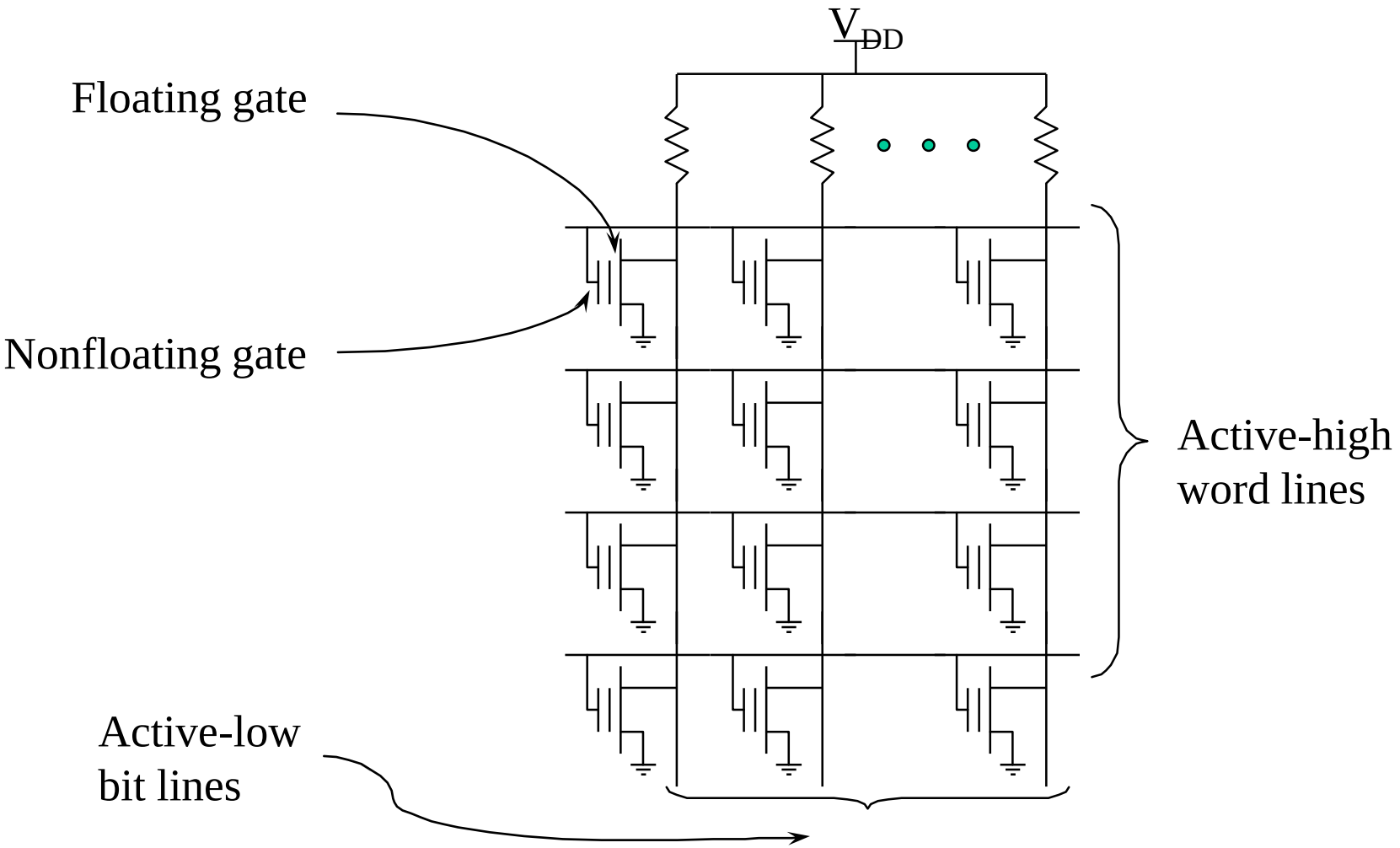
- EEPROM (E^2ROM)
  - Electrically Erasable ROM
  - Floating gates charged/discharged electrically
  - Not RAM! (relatively slow charge/discharge)
  - limited number of charge/discharge cycles (10,000)



Types of ROMs (3)

- Flash Memory
  - Electronically erasable in blocks
  - 100,000 erase cycles
  - Simpler and denser than EEPROM

EPROM and EEPROM Structure



NOR Vs. NAND Flash

Introduced by	Intel in 1988	Toshiba in 1989
Capacity	1 to 16MB	8 to 128MB
Code Execution	Yes	No
Erase	Very Slow(5 Sec)	Fast (3 msec)
Write	Slow	Fast
Read	Fast	Fast
Erase cycle	10k – 100k	100K to 1M
Interface	Memory like	I/O only
Access	Random	Sequential
Ideal usage	Code Storage	Basic data storage

ROM Type Summary

Type	Technology	Read Cycle	Write Cycle	Comments
Mask ROM	NMOS, CMOS	20-200 ns	4 weeks	Write once; low power
Mask ROM	Bipolar	<100 ns	4 weeks	Write once; high power; low density
PROM	Bipolar	<100 ns	5 minutes	Write once; high power; no mask charge
EPROM	NMOS, CMOS	25-200 ns	5 minutes	Reusable; low power; no mask charge
EEPROM	NMOS	50-200 ns	10 μs/byte	10,000 writes/location limit
FLASH	CMOS	25-200 ns	10 μs/block	100,000 erase cycles

Read/Write Memory (RWM / RAM)

- RWM = RAM (Random Access Memory)
- Highly structured like ROM's and PLD's
- Can store and retrieve data at the same speed

- Static RAM (SRAM) retains data in latches (while powered)
- Dynamic RAM (DRAM) stores data as capacitor charge; all capacitors must be recharged periodically.

- Volatile Memory: Both Static and Dynamic RAM
- Nonvolatile Memory: Data retained when power lost
  ROM's, NVRAM (w/battery), FlashMemory

Random Access Memory

Types

SRAM , DRAM

Application

Primary memory, Cache , FIFO , STACK

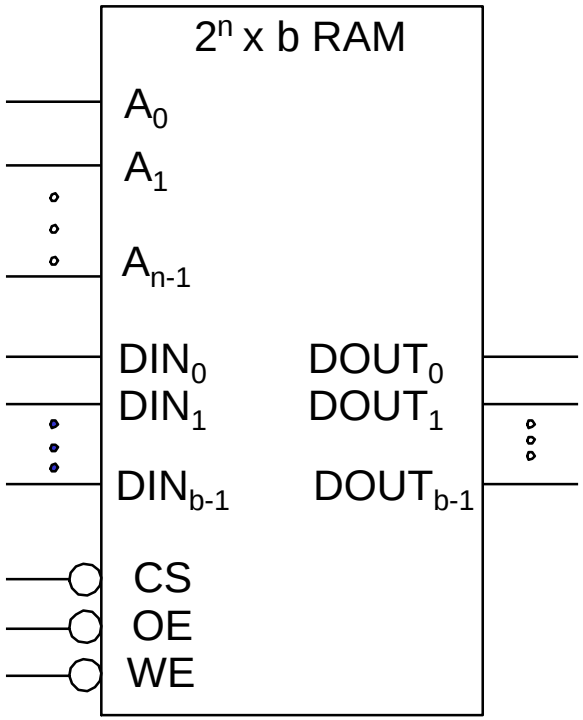
Speed

Access Typical – 10 – 100 ns

Size

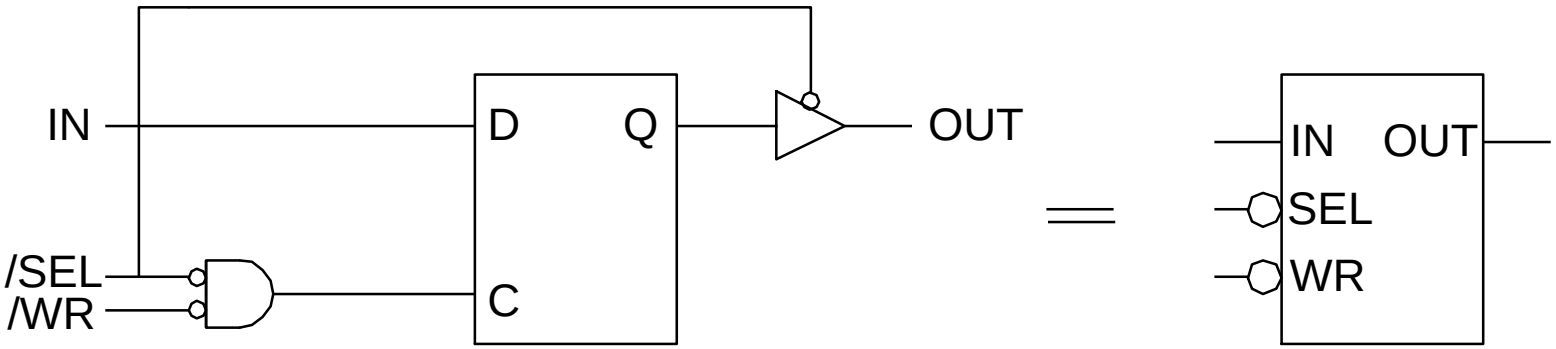
100’s of KBs to 10’s of MBs

Basic Structure of SRAM



- Address/Control/Data Out lines like a ROM (Reading)
  + Write Enable (WE) and Data In (DIN) (Writing)

One Bit of SRAM

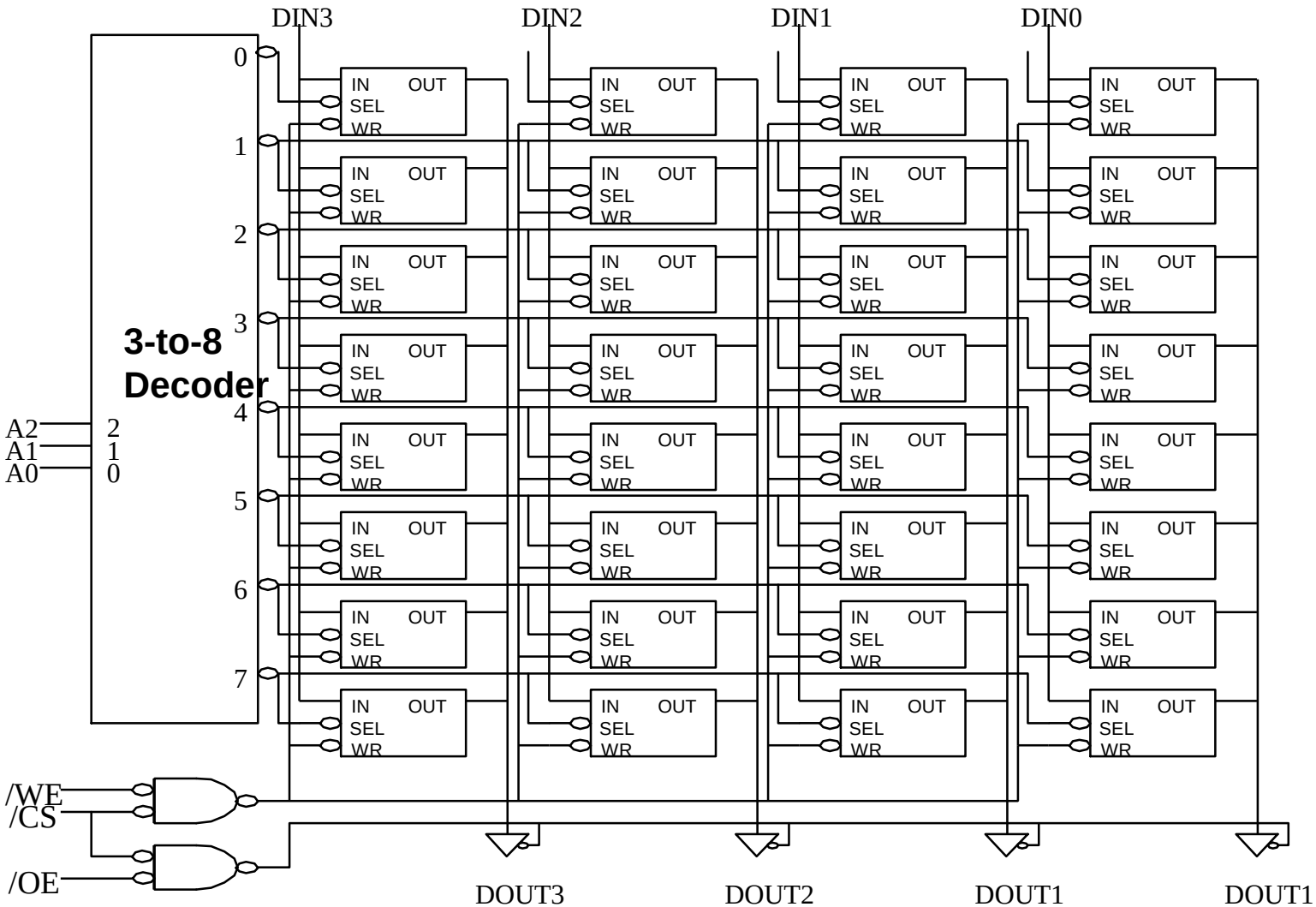


- SEL and WR asserted → IN data stored in D-latch (Write)
- SEL only asserted → D-latch output enabled (Read)
- SEL not asserted → No operation

Structure of 8x4 SRAM

- 3-to-8 Decoder selects one row of cells for Read or Write
- If ADDR + CS + OE are asserted, DOUT (3-0) is enabled. (Read)
- If ADDR + DIN + CS + WE are asserted, DIN → latches (Write)

8 x 4 SRAM



SRAM Timing

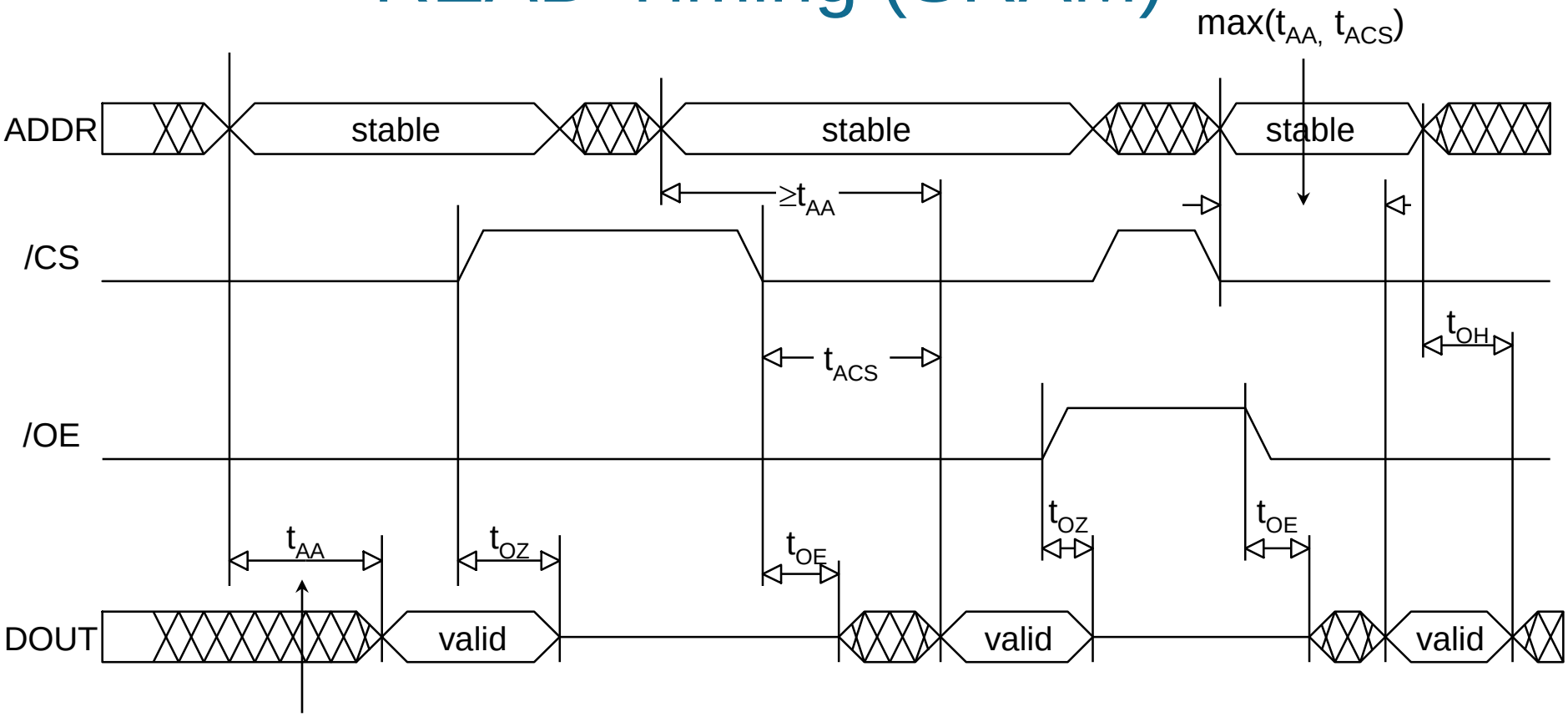
- During READ, outputs are combinational functions of ADDR, CS, OE (like ROM)
  - Inputs can freely change without problems (except for propagation delay from last input change to output)

- During WRITE, data stored in latches
  - Thus, Setup & Hold on Data IN relative to trailing edge of /WR

- ADDR must be stable
  - for setup time before /WR asserted, and
  - for hold time after /WR deasserted
    - to prevent “spraying” data to multiple rows

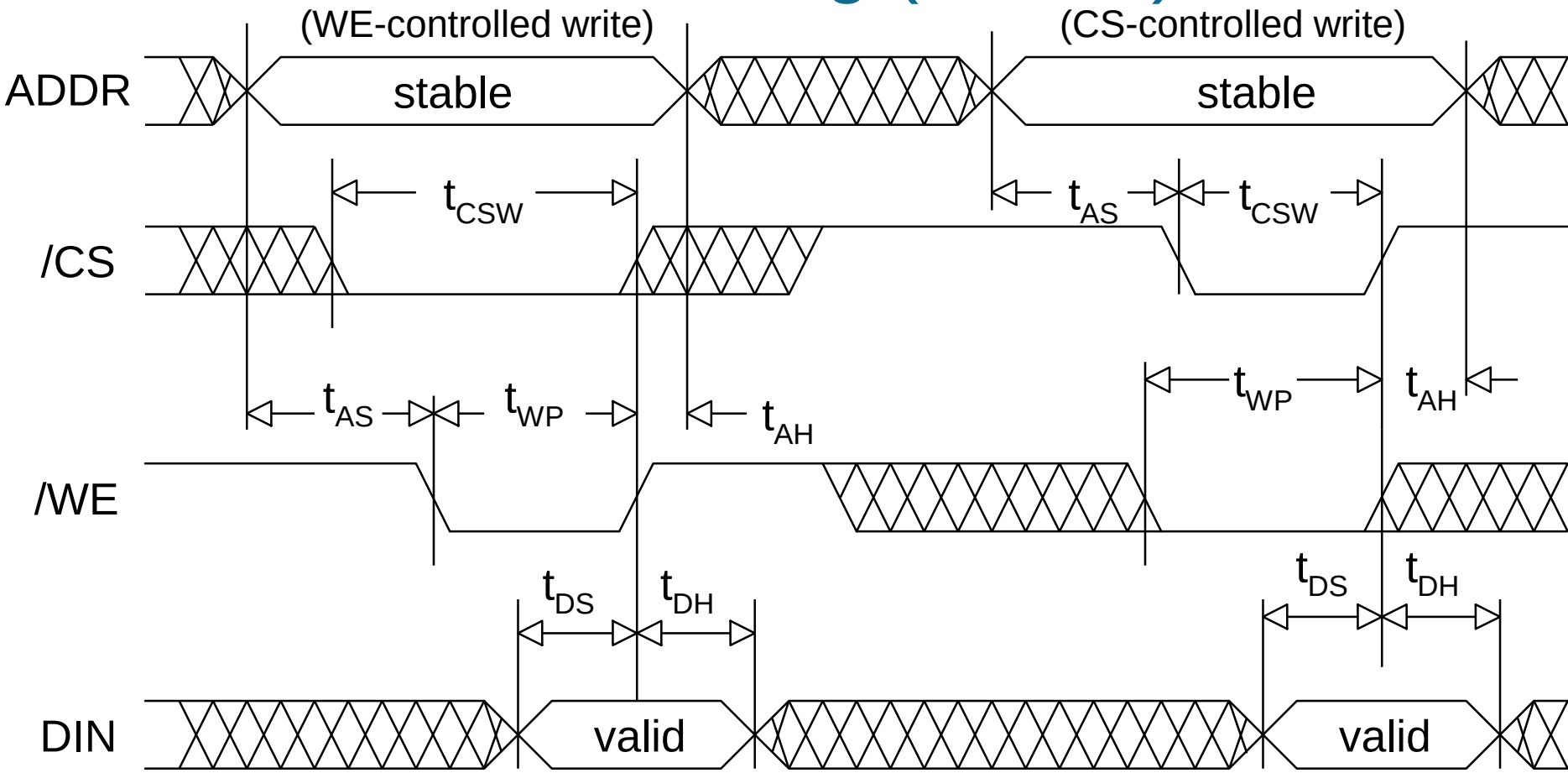
- /WR asserted when BOTH /CS and /WE asserted
- /WR deasserted when EITHER /CS or /WE deasserted

READ Timing (SRAM)

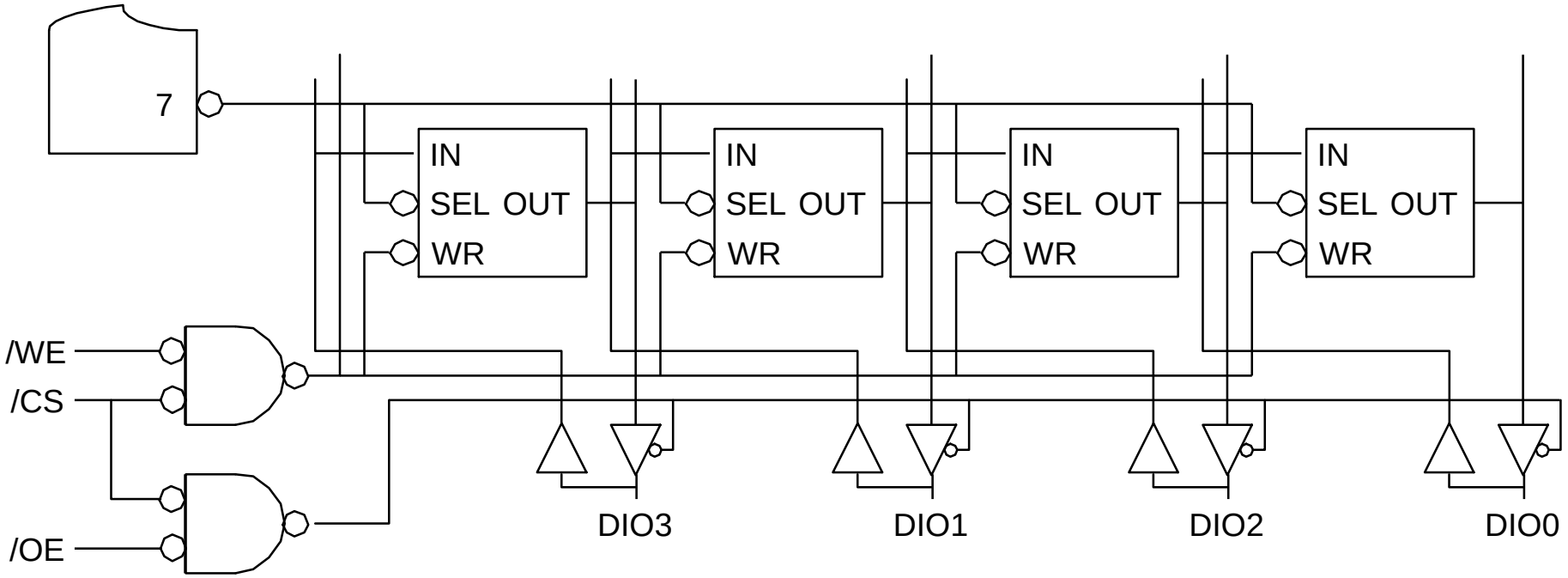


Primary Spec
for SRAMs

WRITE Timing (SRAM)



SRAM with Bidirectional Data Bus



SRAM-Read/Write

Asynchronous SRAM
- Address Before Cycle Start

Read data valid only after
- /OE Asserted

Write Initiated by either
- /CE or /WE

Write Terminated by either
- /CE or /WE

SRAM Types

Asynchronous

Read , Write Cycles controlled by /CE,/WE,/OE.

Synchronous and Sync-Burst

Clock with /WE , /OE controls data cycles.

Double-Data SRAM

Data read and written on both edges of clock.

Quad-Data SRAM

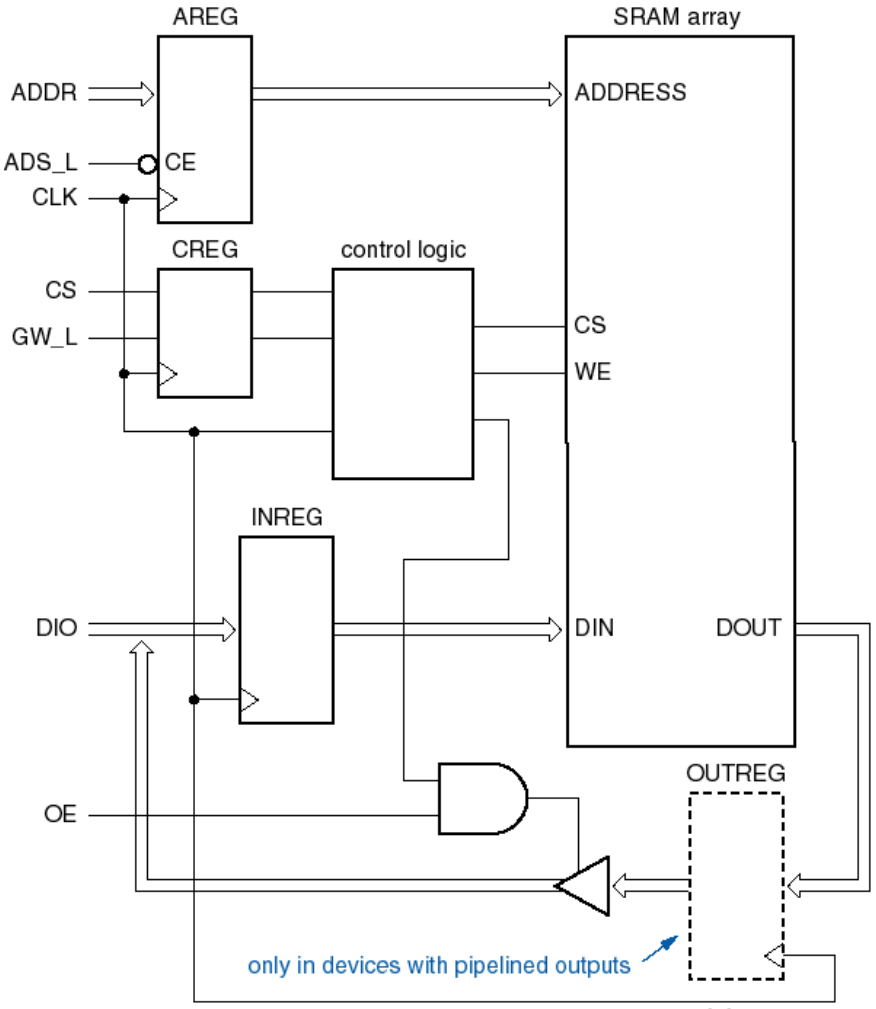
Dual-Unidirectional Ports with both edges to read/write.

Zero Bus Turn-around RAM

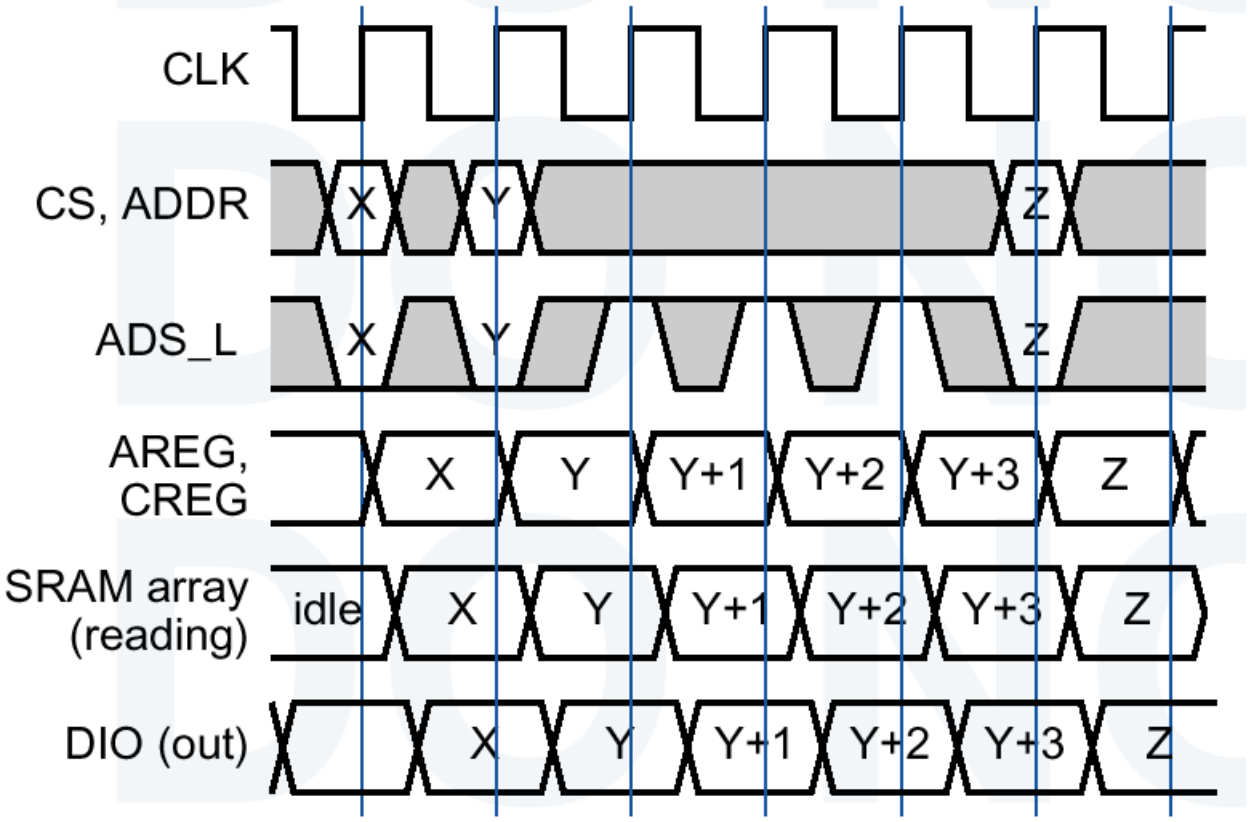
No Additional wait required between read and write.

Synchronous SRAMs

- Use latch-type SRAM cells internally
- Put registers in front of address control and data for easier interfacing with synchronous systems at high speeds
- OUTREG is present for pipelined output
  - Absent for flow through outputs



SSRAM Read –Flow through



Note: GW_L is HIGH throughout

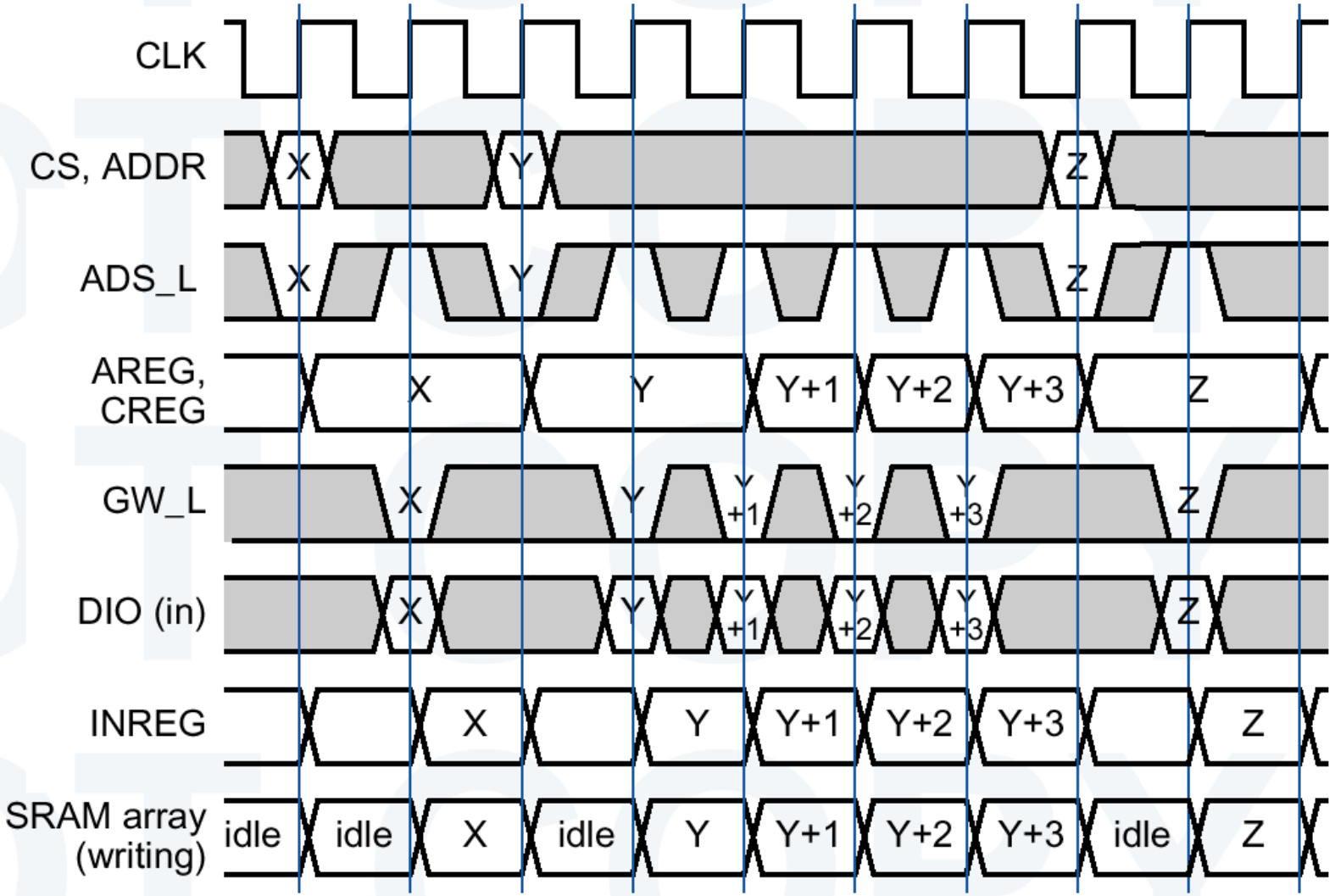
Read Operation

- Control and address inputs sampled at rising edge
- AREG loaded only if ADS_L is asserted
- During the next clock the internal SRAM is accessed and data is delivered to DIO pins
- Burst mode is supported
  - AREG behaves like a counter

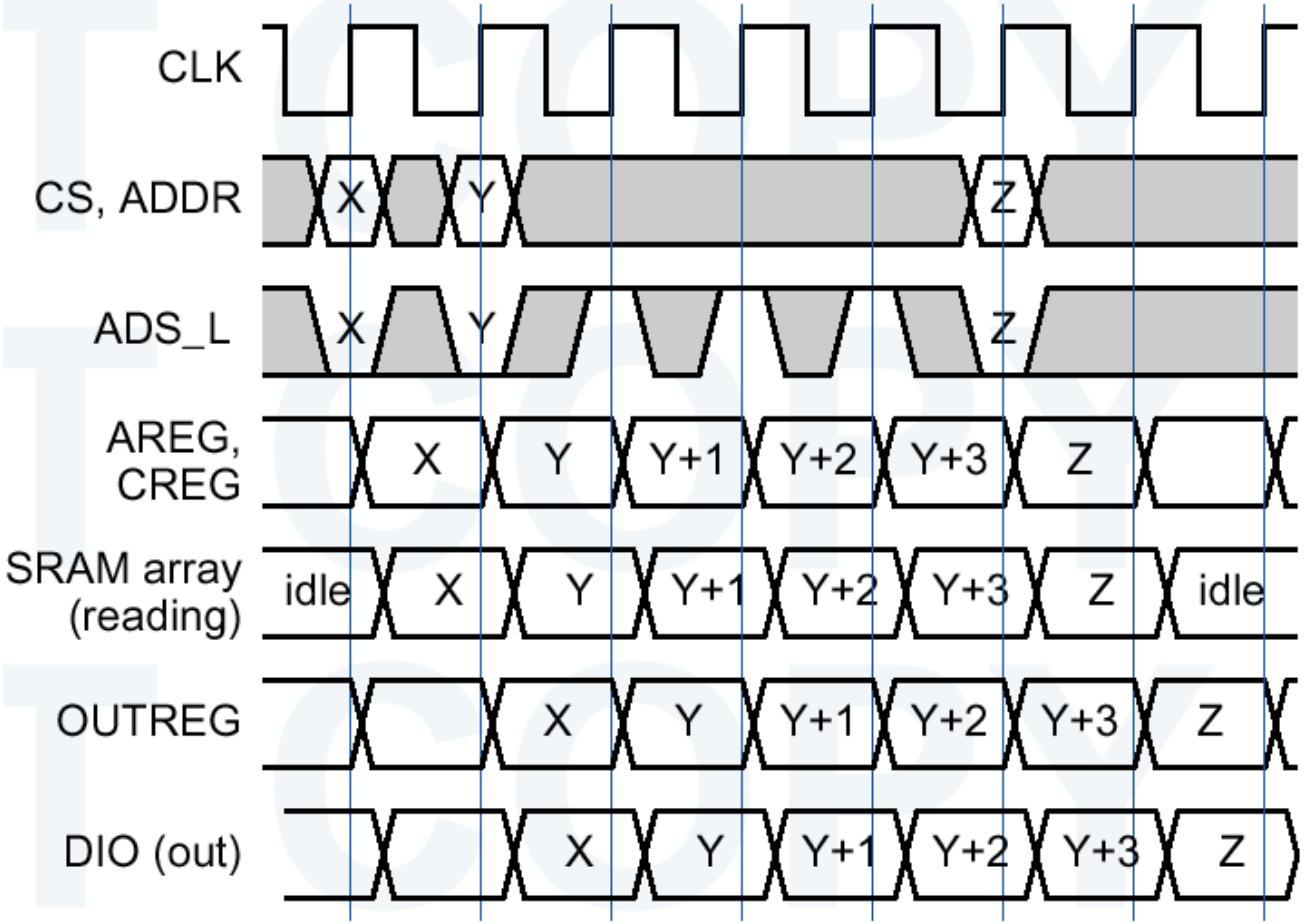
Write Operation

- Control and address inputs sampled at rising edge
- Data is stored temporarily in INREG
- During the next clock the internal SRAM is written when the control signal GW_L is asserted
  - ADS_L should be inhibited for one clock after loading the address
- Burst mode is supported
  - AREG behaves like a counter
- It is not possible to write at two non-sequential addresses in successive clocks.

SSRAM Write – Flow Through



Read SSRAM - Pipelined



SSRAM - Pipelined Output

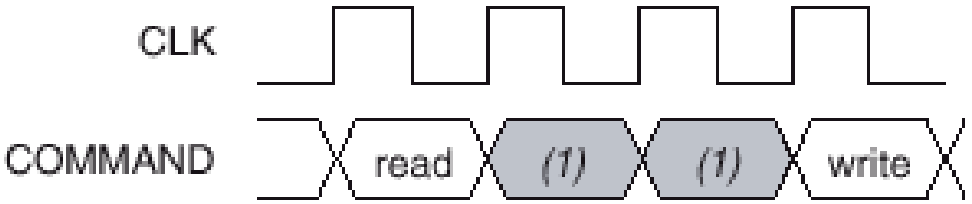
- OUREG is placed between array and device outputs
  - Disadvantage: Output data is delayed by one clock
  - Advantage: Data is available for one clock period
- No change in write cycles
- Provides better setup time for device performing reads

SSRAM -ZBT

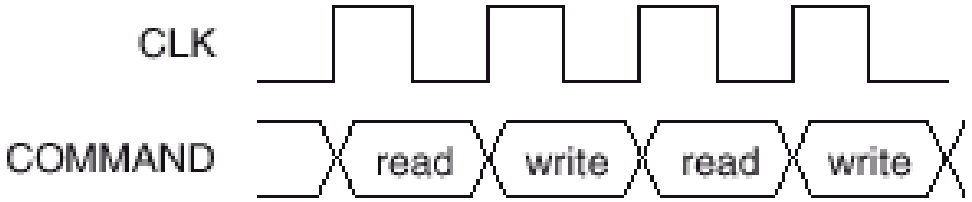
- Late write SSRAMs suffer from one clock delay when a read follows a write or vice-versa
  - Internal SRAM array remains idle
- Turn around penalty is avoided in Zero Bus Turnaround (ZBT) SSRAMs
- The type of operation is selected by R/W\ that is sampled at the same edge of the address
- DIO bus is used for both operations
  - OE avoids bus conflicts between successive cycles

Comparison of read to write transition

Standard SyncBurst SRAM

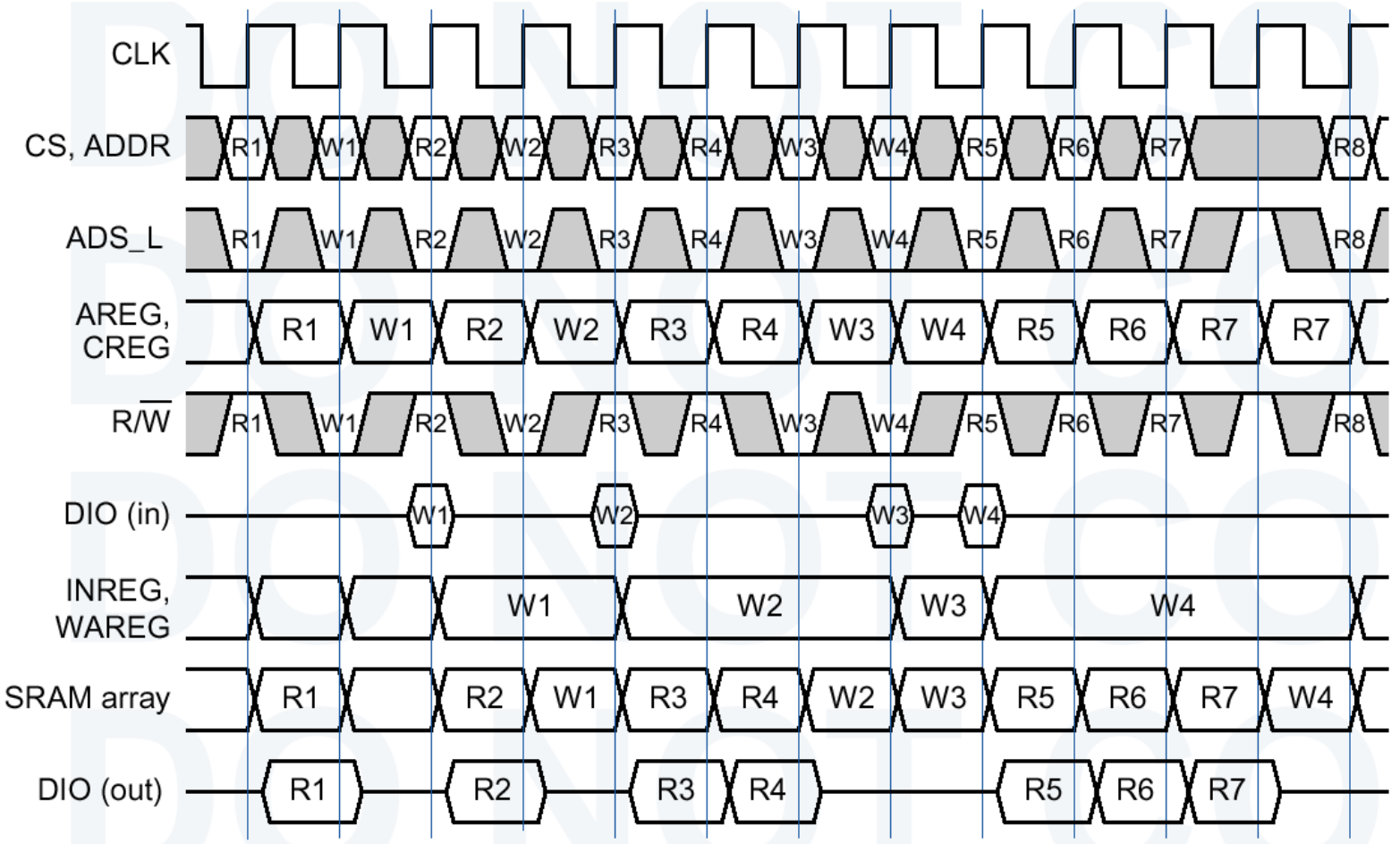


ZBT SRAM



(1) The shaded sections represent no-operation cycles.

SSRAM - ZBT



ZBT SSRAMs

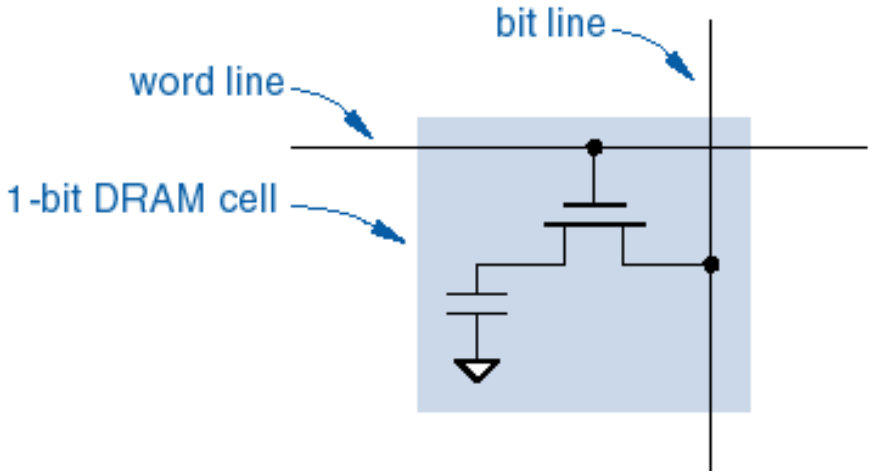
- If a write is followed by read, resource conflict for the array occurs.
  - So write operation is delayed till next available cycle
  - Opportunity occurs during a write or no operation
- While write is pending, address and data needs to be stored in another set of registers
- A write operation may be delayed indefinitely, if a series of reads follows.
  - Care should be taken if one of the reads has same address as a pending write

QDR SSRAM

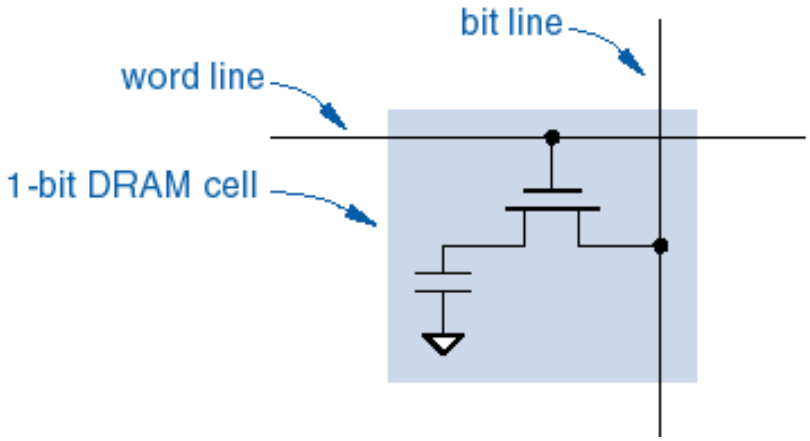
- Capable of transferring up to 4 words of data per clock
  - Both – rising and falling edges of clock
  - two clocks, one for read data and one for write data
  - separate read and write data buses
- Designed for high-speed communications and networking applications
  - Data throughput is more important than cost, power efficiency or density
  - Useful where reads and writes are interleaved

DRAM (Dynamic RAMs)

- SRAMs typically use six transistors per bit of storage.
- DRAMs use only one transistor per bit:
- 1/0 is capacitor charged/discharged



DRAM read operations

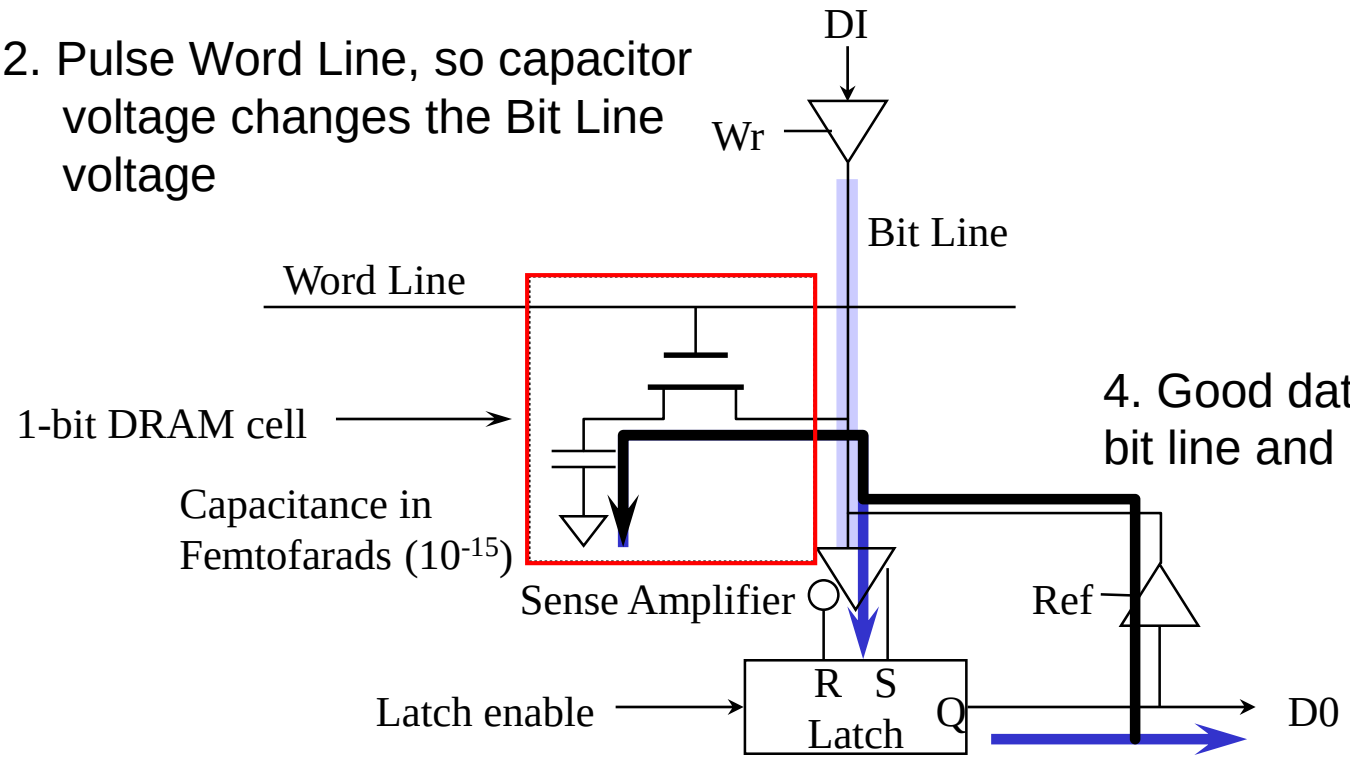


- Precharge bit line to $V_{DD}/2$.
- Take the word line HIGH.
- Detect whether current flows into or out of the cell.
- Note: cell contents are destroyed by the read!
- Must write the bit value back after reading.

DRAM Internal Structure - Read

1. Bit Line precharged to half-high voltage

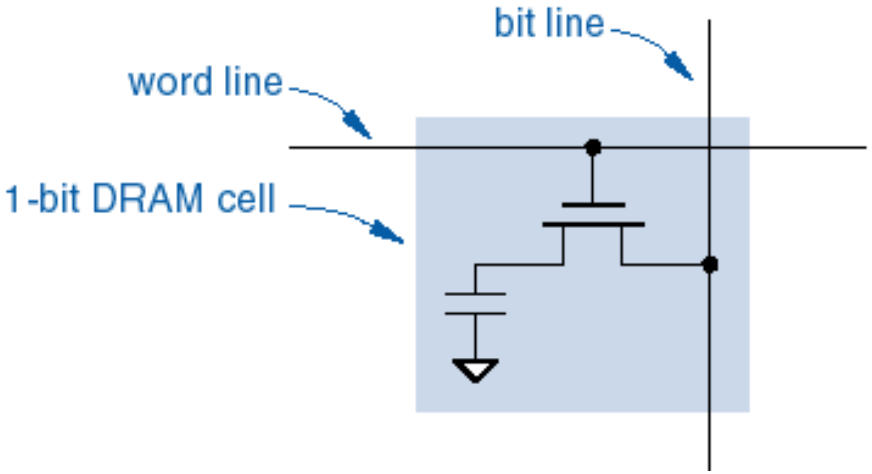
2. Pulse Word Line, so capacitor voltage changes the Bit Line voltage



4. Good data forced back on bit line and into capacitor

3. Sense Amplifier detects voltage change and latches

DRAM write operations

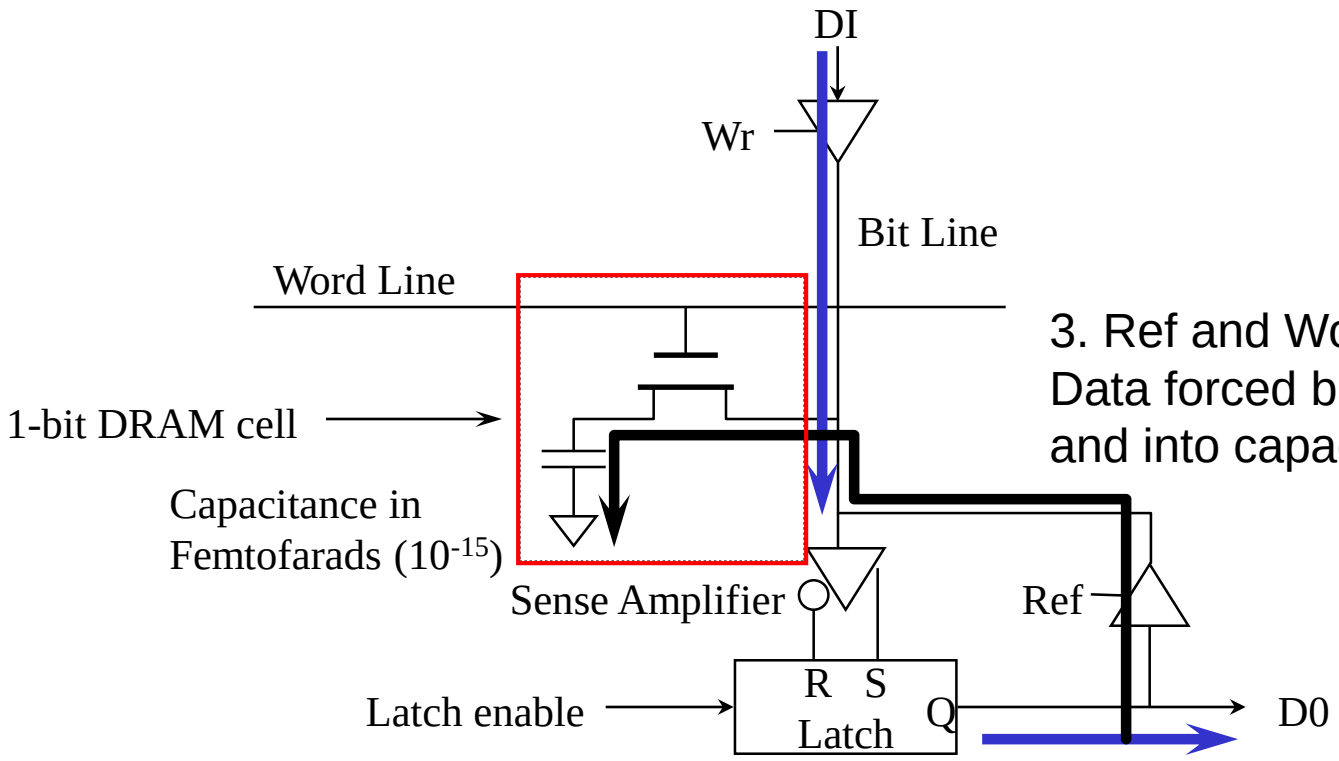


- Take the word line HIGH.
- Set the bit line LOW or HIGH to store 0 or 1.
- Take the word line LOW.

- Note: The stored charge for a 1 will eventually leak off.

DRAM Internal Structure - Write

1. Drive Data onto Bit Line (enable Wr)

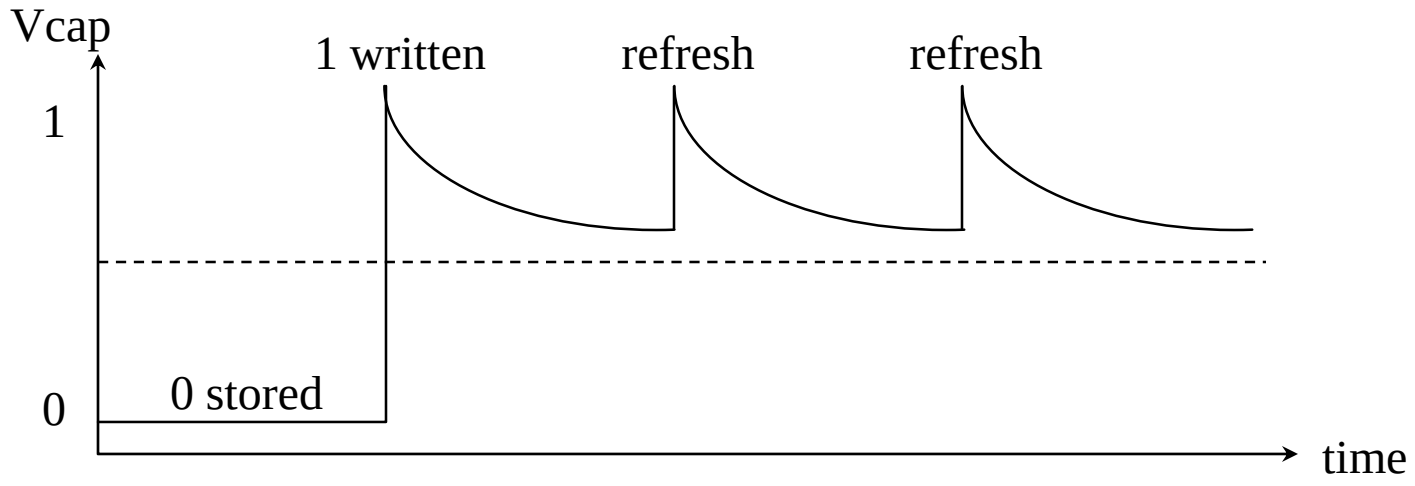


3. Ref and Word Line pulsed. Data forced back on bit line and into capacitor

2. Sense Amplifier detects Bit Line and latches new data

Cell Voltage Discharge and Refresh

- Capacitor stores a charge for several milliseconds. Why so short?
- So we must re-write or refresh every cell voltage within 4 ms (typically)

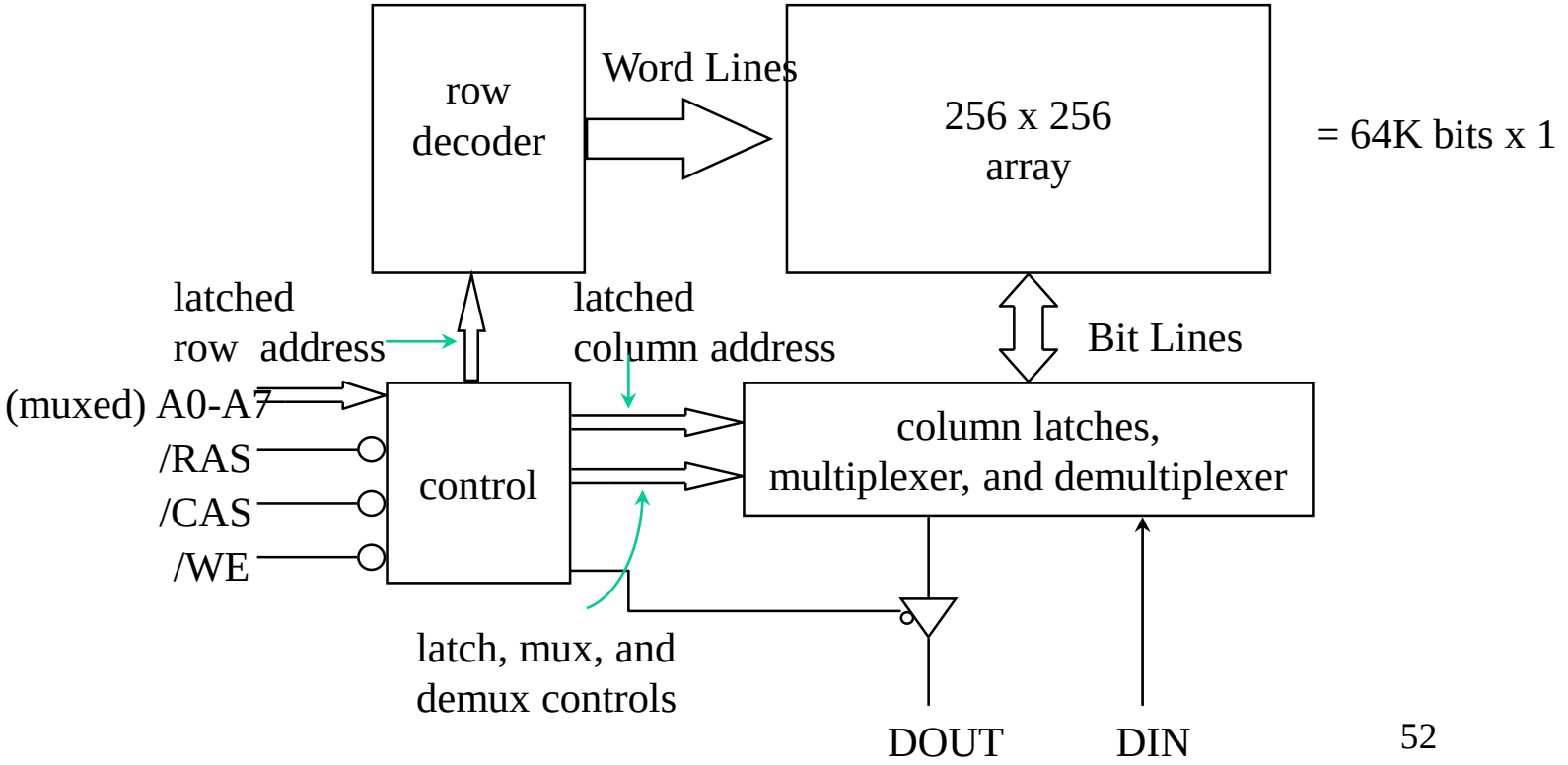


Refresh Cycles overheads

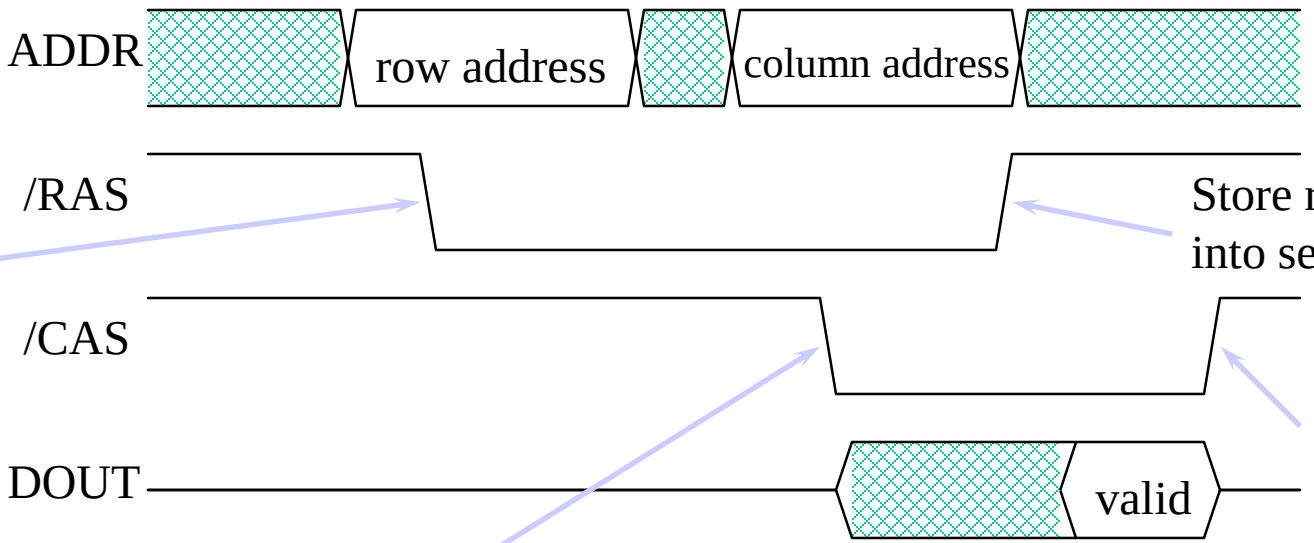
- Each refresh cycle takes time away from useful read/write cycles
- Any useful time left for reading or writing data?
  - Refresh entire row in 1 cycle
  - Typically 256 rows in a DRAM refreshed sequentially
  - Assume 1 refresh cycle takes ≈100 ns
  - What percentage of time is available for read/write cycles?
    - 4ms/256 » 16 us per row refresh
    - 0.1/16 < 1 % of time used for refresh cycles
    - so » 99 % of time remains for read/write cycle

Structure of 64K x 1 DRAM

- 16 bit Address multiplexed into two 8-bit load operations
- 8 bit Row Address latched during /RAS pulse
- 8 bit Column Address latched during /CAS pulse



DRAM Timing: Read



Load row-address register, read selected row, and store in row latch

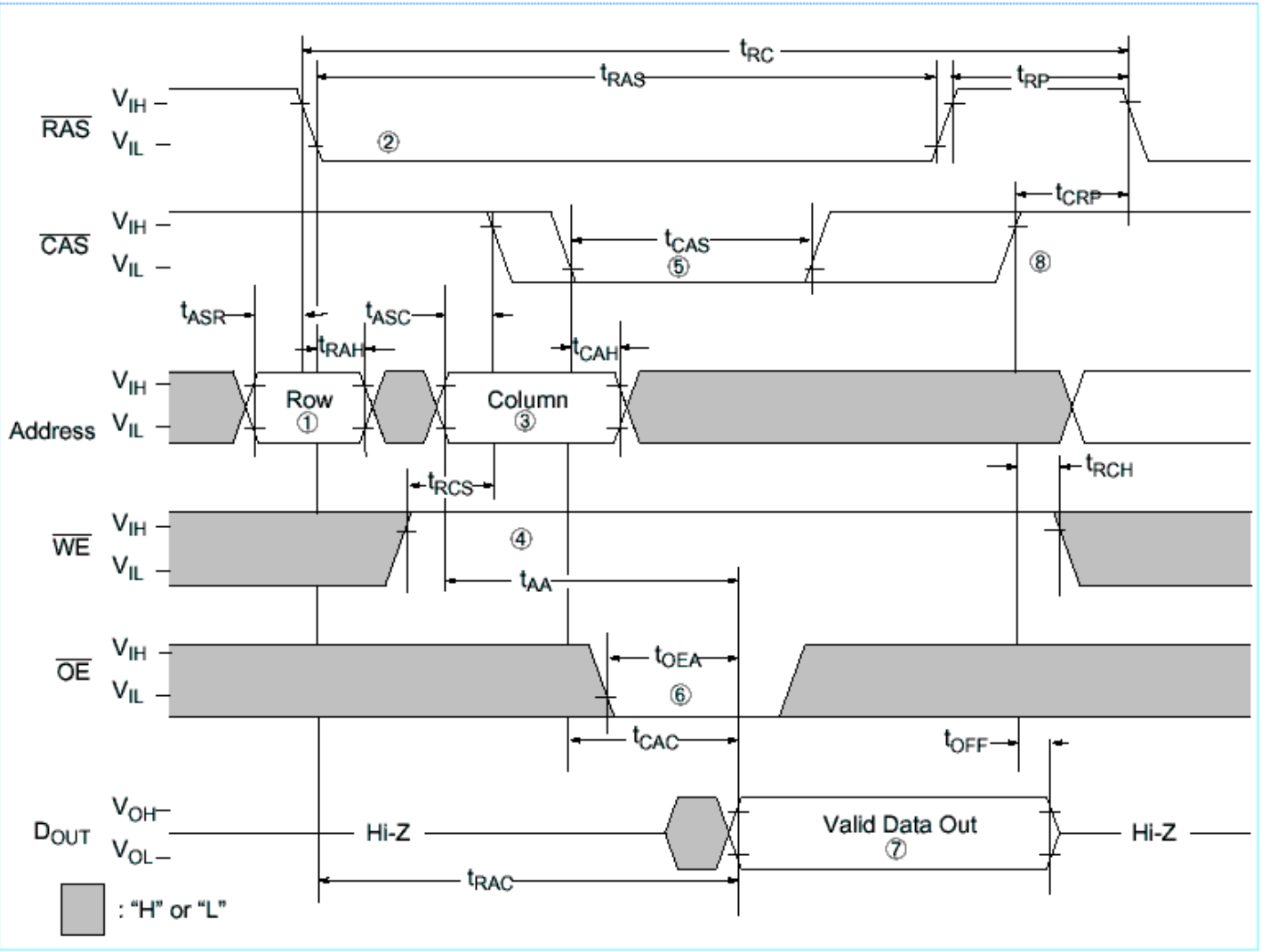
Store row latch into selected row

Disable DOUT

Note:
/WE = High

Load column-address register, enable DOUT, drive with selected bit

DRAM-Read



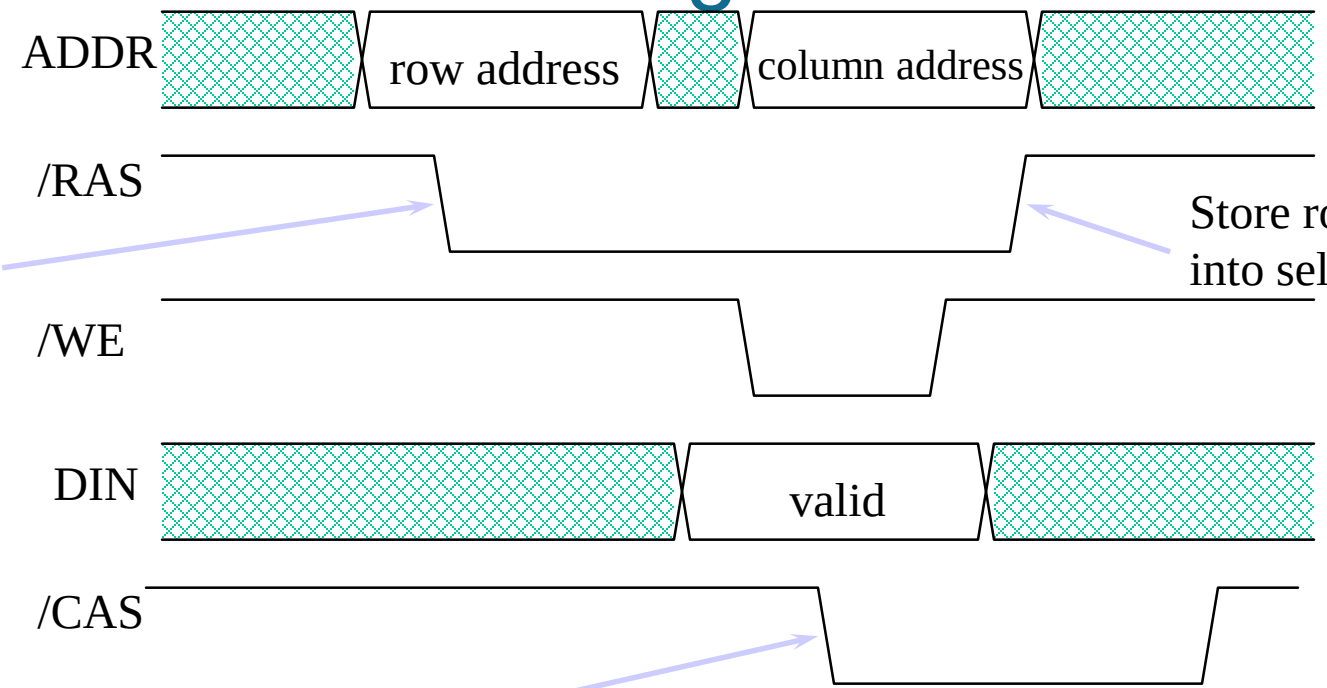
DRAM Timing: Read Cycle

- /WE kept high
- /RAS (falling edge)
  - Latches address into Row Address Latch
  - All bit lines precharged to intermediate voltage
  - 1 word line activated from Row Address
  - Bit Line voltage slightly changed +/- by capacitor charge
  - Sense amplifiers store data in Row Data Latch when enabled
- /CAS (falling edge)
  - Latches column address into Column Address Latch
  - Selects 1 bit of Row Data Latch with a MUX to drive DOUT
  - Enable DOUT tri-state buffer to output data (like OE in SRAM)

DRAM Timing: Read Cycle

- /RAS (rising edge)
  - Row Data Latch driven back onto bit lines when **ref** signal active
  - Selected row of cells re-written High or Low
  - Word Line de-activated
- /CAS (rising edge)
  - Disable DOUT tri-state buffer

DRAM Timing: Write



Load row-address register, read selected row, and store in row latch

Store row latch into selected row

Load column-address register, drive DIN into selected column of row latch

Note: DOUT = Hi-Z

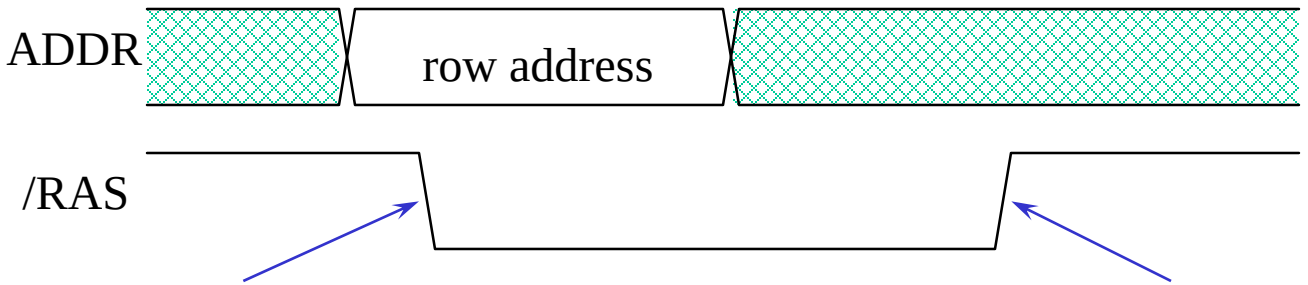
DRAM Timing: Write Cycle

- /RAS (falling edge)
  - Latches address into Row Address Latch
  - All bit lines precharged to intermediate voltage
  - 1 word line activated from Row Address
  - Bit Line voltage slightly changed +/- by capacitor charge
  - Sense amplifiers store data in Row Data Latch when enabled
- /CAS (falling edge) with /WE true
  - Latches column address into Column Address Latch
  - Drives DIN onto the selected Bit Line and into one Row Data Latch

DRAM Timing: Write Cycle

- /RAS (rising edge)
  - Row Data Latch driven back onto bit lines when **ref** signal active
  - Selected row of cells re-written High or Low, with one new bit
  - Word Line de-activated
- /CAS (rising edge)
  - End of write cycle

DRAM Timing: RAS-only Refresh



Load row-address register, read selected row, and store into row latch

Restore row latch back into selected row of cells

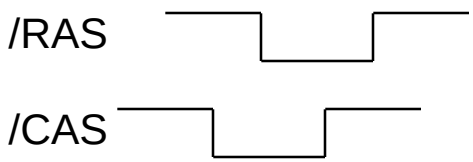
(Note: /CAS = /WE = High)

DRAM: RAS-only Refresh Cycle

- /CAS, /WE kept high throughout cycle
- /RAS (falling edge)
  - Latches address into Row Address Latch
  - All bit lines precharged to intermediate voltage
  - 1 word line activated from Row Address
  - Bit Line voltage slightly changed +/- by capacitor charge
  - Sense amplifiers store data in Row Data Latch when enabled
- /RAS (rising edge)
  - Row Data Latch driven back onto bit lines when **ref** signal active
  - Selected row of cells re-written High or Low
  - Word Line de-activated

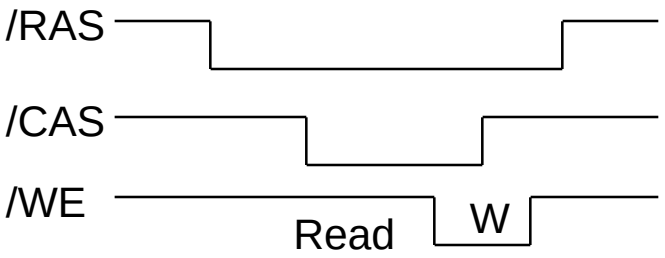
Other Types of DRAM Cycles (1)

- CAS-before-RAS Refresh
  - Internal Row-Addr counter used and incremented
  - Eliminates external addr counter and part of Addr mux



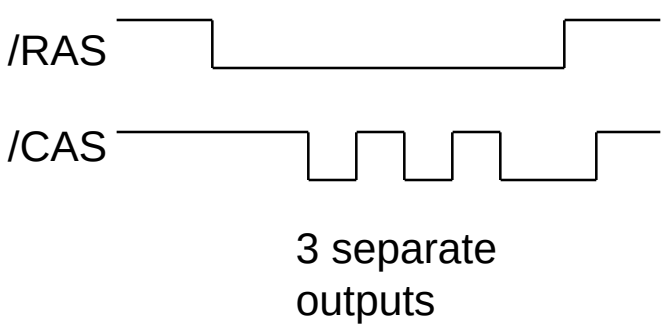
- Read-Modify-Write

  Normal Read ending with WE pulse to write new data in same location - Faster



Other Types of DRAM Cycles (2)

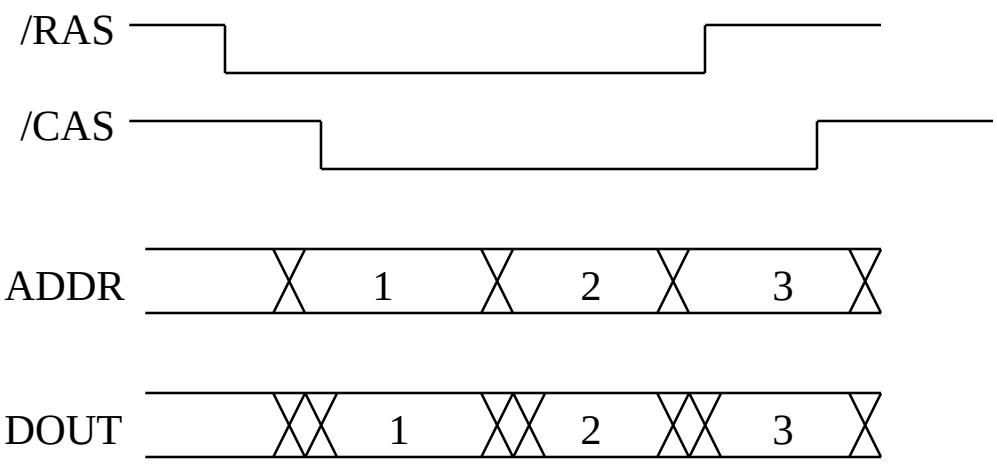
- Page Mode Read
  - Allows entire DRAM row to be output **FAST**
  - 1 RAS pulse
  - 1 CAS pulse per output bit



- Page Mode Write
  - Allows multiple bits per DRAM row written **FAST**
  - 1 CAS pulse per input bit (with /WE true)

Other Types of DRAM Cycles (3)

- Static-Column-Mode Read
  - 1 long RAS pulse, 1 long CAS pulse
  - Multiple column addresses (not latched) supplied for multiple output bits



DRAM Types

DRAM – Dynamic Random Access Memory
- Normally Asynchronous

EDORAM – Extended Data Output RAM
- Outputs are registered

FPMDRAM – Full Page Access DRAM
- Give row only once go on changing column

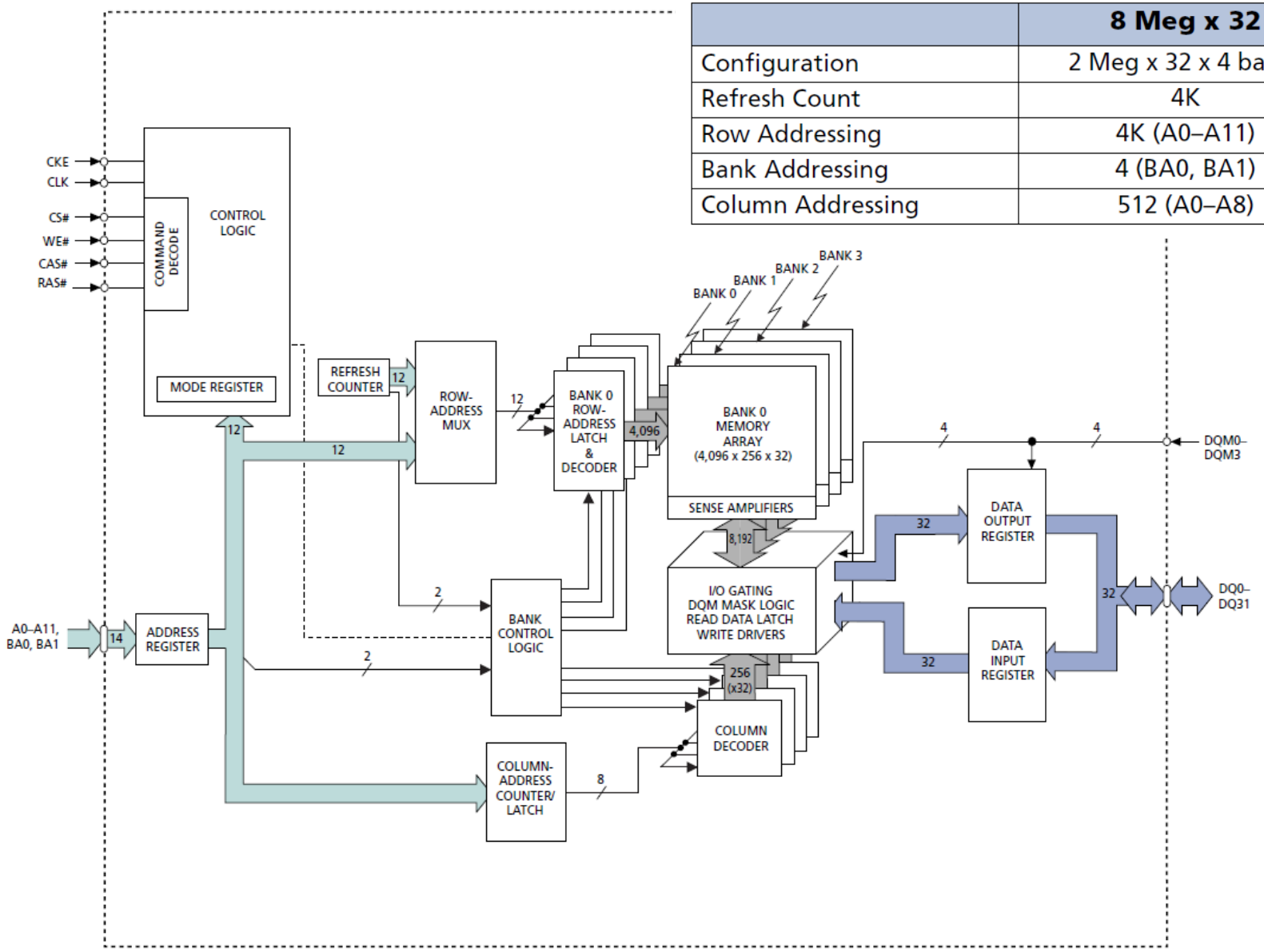
SDRAM – Synchronous DRAM
- DRAM based on Clock

ESDRAM – Enhanced Synchronous DRAM
- Small SRAM on same chip

RDRAM – Rambus® DRAM
- Protocol based proprietary

Functional Block Diagram – 8 Meg x 32 SDRAM

	8 Meg x 32
Configuration	2 Meg x 32 x 4 banks
Refresh Count	4K
Row Addressing	4K (A0–A11)
Bank Addressing	4 (BA0, BA1)
Column Addressing	512 (A0–A8)



SDRAM

- All signals are latched on positive edge of clock
- Internally, divided into multiple banks
- Read and write access are burst oriented
- Access begins with registration of ACTIVE command
  - Followed by a READ or WRITE command
- The address bits registered coincident with the ACTIVE command are used to select the bank and row to be accessed
- The address bits registered coincident with the READ or WRITE command are used to select the starting column location for the burst access

SDRAM

- Provision for programmable READ or WRITE burst lengths of 1, 2, 4, or 8 locations, or the full page, with a burst terminate option.
- Precharging one bank while accessing one of the other banks will hide the precharge cycles and provide seamless, highspeed,random-access operation.
- Prior to normal operation, the SDRAM must be initialized.

SDRAM Commands

NAME (FUNCTION)	CS#	RAS#	CAS#	WE#	DQM	ADDR	DQS
COMMAND INHIBIT (NOP)	H	X	X	X	X	X	X
NO OPERATION (NOP)	L	H	H	H	X	X	X
ACTIVE (Select bank and activate row)	L	L	H	H	X	Bank/Row	X
READ (Select bank and column, and start READ burst)	L	H	L	H	L/H[8]	Bank/Col	X
WRITE (Select bank and column, and start WRITE burst)	L	H	L	L	L/H[8]	Bank/Col	Valid
BURST TERMINATE	L	H	H	L	X	X	Active
PRECHARGE (Deactivate row in bank or banks)	L	L	H	L	X	Code	X
AUTO REFRESH or SELF REFRESH (Enter self refresh mode)	L	L	L	H	X	X	X
LOAD MODE REGISTER	L	L	L	L	X	Op-Code	X
Write Enable/Output Enable	–	–	–	–	L	–	Active
Write Inhibit/Output High-Z	–	–	–	–	H	–	High-Z

DDR SDRAM

- Double Data Rate SDRAM is advantageous for systems that require higher bandwidth than can be obtained using SDRAM.
- Enhancements to the SDRAM core
  - Prefetching
  - double transition clocking
  - strobe-based data bus,
  - SSTL_2 low voltage signaling.
- At 400 MHz, DDR increases memory bandwidth to 3.2 GB/s, which is 400 percent more than original SDRAM.

Pre-fetching

- In SDRAM, one bit per clock cycle is transferred from the memory cell array to the input/output (I/O) buffer or data queue (DQ).
- The I/O buffer releases one bit to the bus per pin and clock cycle
- Pre-fetching transfers two bits from the memory cell array to the I/O buffer in two separate pipelines.
- Then the I/O buffer releases the bits in the order of the queue on the same output line.
- This is known as a 2n-prefetch architecture because the two data bits are fetched from the memory cell array before they are released to the bus in a time multiplexed manner.

Double transition clocking

- Standard DRAM transfers one data bit to the bus on the rising edge of the bus clock signal,
- DDR SDRAM uses both the rising and falling edges of the clock to trigger the data transfer to the bus
- This, double transition clocking, delivers twice the bandwidth of SDRAM without increasing the clock frequency.

SSTL_2 & Stobe-based data bus

- Instead of using a 3.3-V operating voltage, DDR SDRAM uses a 2.5-V signaling specification known as Stub Series-Terminated Logic_2
- This low-voltage signaling results in lower power consumption and improved heat dissipation.
- SSTL_2 signaling allows DDR SDRAM to run at faster speeds than traditional SDRAM.
- DDR SDRAM uses a delay-locked loop (one for every 16 outputs) to provide a data strobe signal as data becomes valid on the SDRAM pins.
- The memory controller uses the data strobe signal to locate data more accurately and resynchronize incoming data from different DIMMs.

Summary of DDR SDRAM

Type	Component naming convention	Module naming convention	Bus speed	Peak bandwidth
DDR	DDR200	PC1600	100 MHz	1.6 GB/s
	DDR266	PC2100	133 MHz	2.1 GB/s
	DDR333	PC2700	166 MHz	2.7 GB/s
	DDR400	PC3200	200 MHz	3.2 GB/s
DDR-2	DDR2-400	PC2-3200R	200 MHz	3.2 GB/s
	DDR2-533	PC2-4300	266 MHz	4.3 GB/s
	DDR2-667	PC2-5300	333 MHz	5.3 GB/s
	DDR2-800	PC2- 6400	400 MHz	6.4 GB/s

SDR and DDR

- Single Data Rate SDRAM can accept one command and transfer one word of data per clock cycle.
  - 168-pin DIMMs read or write 64 (non-ECC) or 72 (ECC) bits at a time.
- DDR SDRAM uses the same commands, accepted once per cycle, but reads or writes two words of data per clock cycle.
- The supply voltage is reduced from 3.3 to 2.5 V.
- It doubles the minimum read or write unit; every access refers to at least two consecutive words.
- Typical DDR SDRAM clock speeds are 133, 166 and 200 MHz generally described as DDR-266, DDR-333 and DDR-400
  - Corresponding 184-pin DIMMS are known as PC2100, PC2700 and PC3200.

DDR2 and DDR3

- DDR2 doubles the minimum read or write unit again, to 4 consecutive words.
- The bus speed of the SDRAM is doubled without increasing the speed of internal RAM operations
  - internal operations are performed in units 4 times as wide as SDRAM.
- Typical DDR2 SDRAM clock speeds are 200, 266, 333 or 400 MHz generally described as DDR2-400, DDR2-533, DDR2-667 and DDR2-800
  - Corresponding 240-pin DIMMS are known as PC2-3200 through PC2-6400
- DDR3 continues the trend, doubling the minimum read or write unit to 8 consecutive words. This allows another doubling of bandwidth and external bus speed without having to change the speed of internal operations.

Cache

Memory Requirements

- Programmers want unlimited amount of fast memory
- Hardware aids programmer by creating an illusion of unlimited fast memory
- Realities
  - Fast memory is costly
  - Programs tend to follow principle of locality
    - Temporal: same location is repeatedly accessed
    - Spatial: Close by locations are accessed soon

Processor and memory speeds

- When microprocessors were introduced, standard memory parts were faster than contemporary microprocessors
- Over a period
  - Microprocessors have become faster
  - Memories have become little faster but mostly have higher capacity
- The memories are not able to keep up with the speed achieved by microprocessors

How to bridge the gap

- Interleaving the memory
  - High bit interleaving (mainly used to increase memory size)
  - Low bit interleaving (used to overlap memory delays and speedup the operations)
- Wider memory paths
- Cache memory system

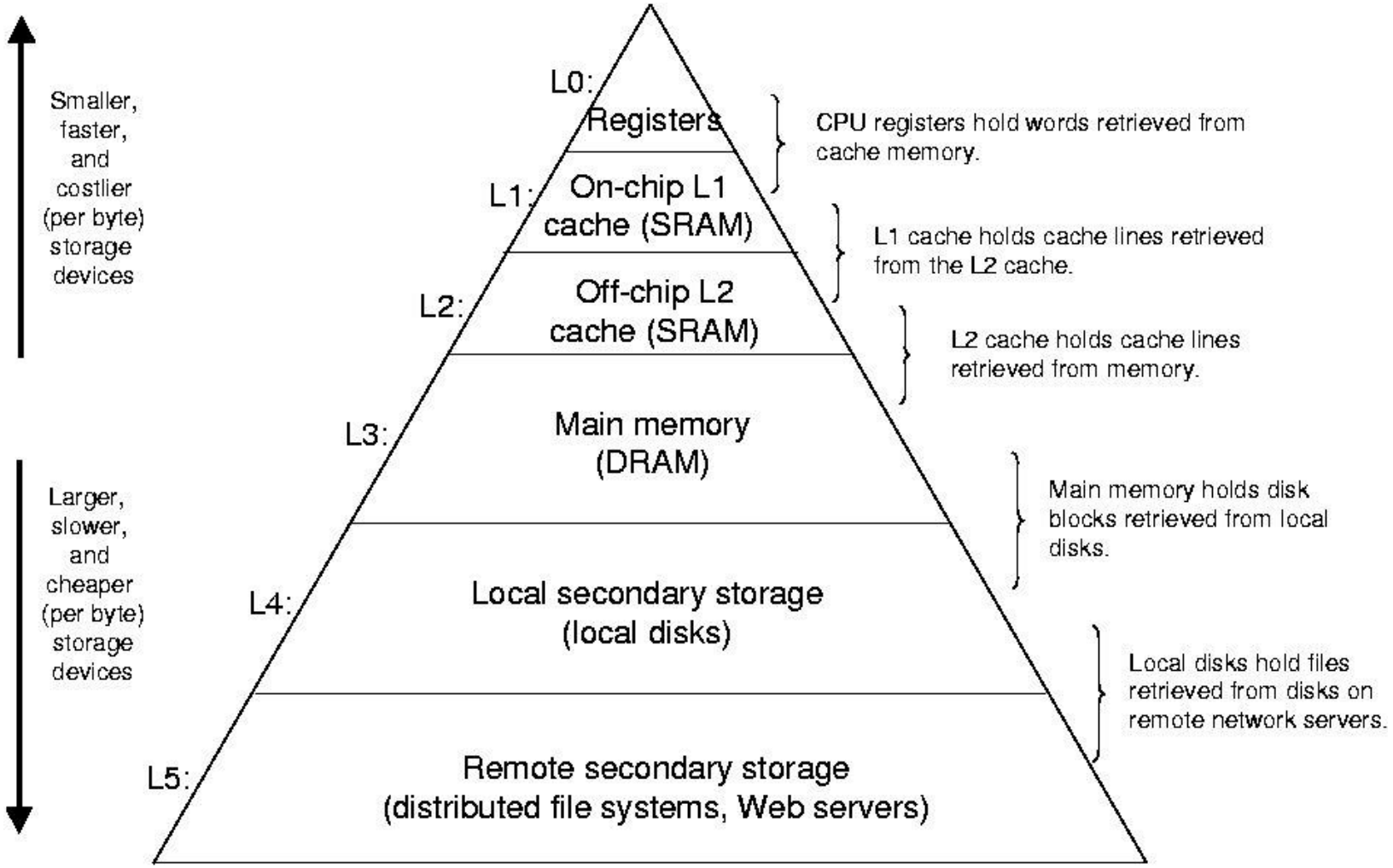
Memory Hierarchy

- Takes advantage of principle of locality
- Memory subsystem consists of multiple levels with different speeds and sizes.
- Small fast memory is put close to processor and slower away
- The goal
  - Provide maximum memory which is cheapest
  - Provide access at the speed offered by the fastest
- Data is copied only between two adjacent levels
- If data is found in level closest to CPU, it's a hit
- Hit rate is a measure of the performance

Speed of Memory

- Registers
  - 10s in number ; 1 - 5 ns access time
- Fast SRAMs
  - 1k - 10k ; 10-20ns
- SRAMs
  - 10k to few MB ; 20-40ns
- Main memory
  - 100MB to 1GB; 50-100 of ns
- Magnetic storage
  - 10G to 500G; 10 msecs

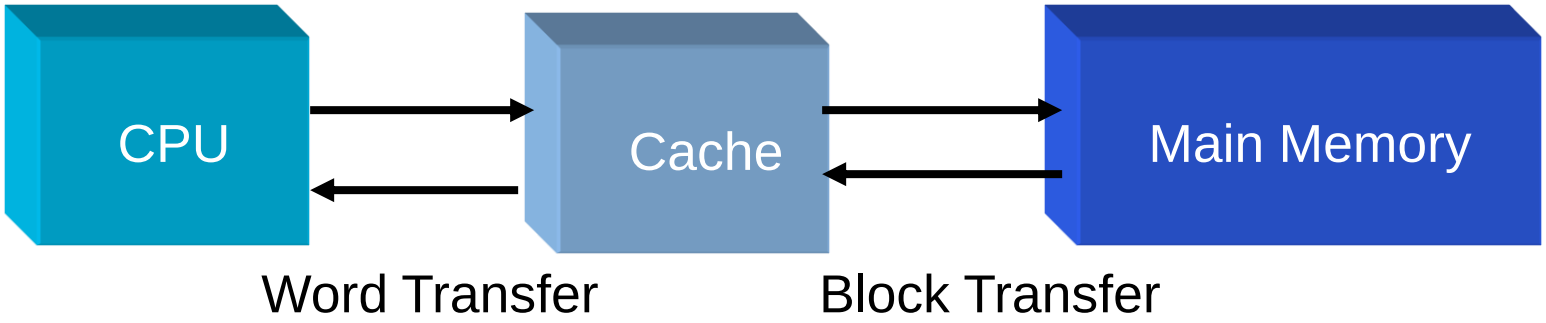
Memory Hierarchy



Cache

Cache: as safe place for hiding or storing things

-Webster's New world dictionary

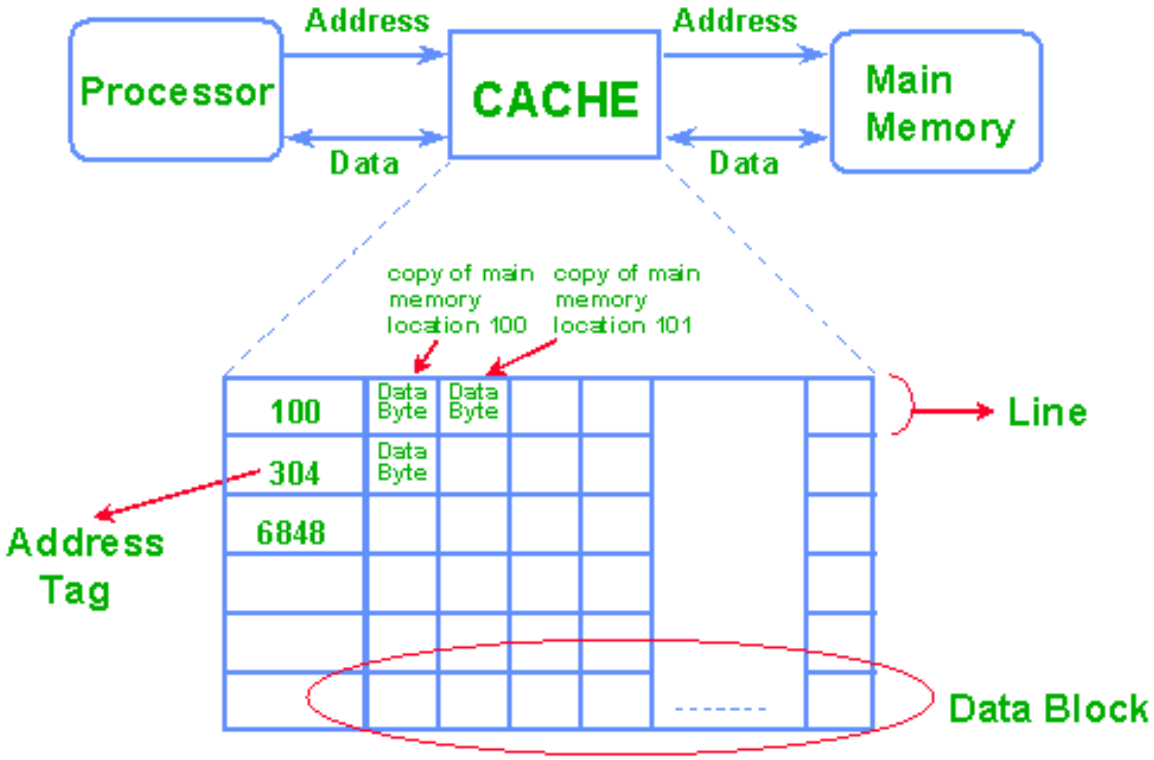


- Cache contains a copy of portions of main memory
- When CPU attempts to read a word, cache is checked first. If found, it is presented, else a line of words is brought from main memory to the cache

Basic Architecture of Cache

- A cache consists of 3 main parts
  - Directory store
    - The cache must know where the information in cache originated from
    - Each directory entry is called as cache-tag
  - Data section
    - Holds the data copied from main memory
  - Status information
    - Valid bid indicates live data
    - Dirty bit indicates non-coherency

Inside a Cache

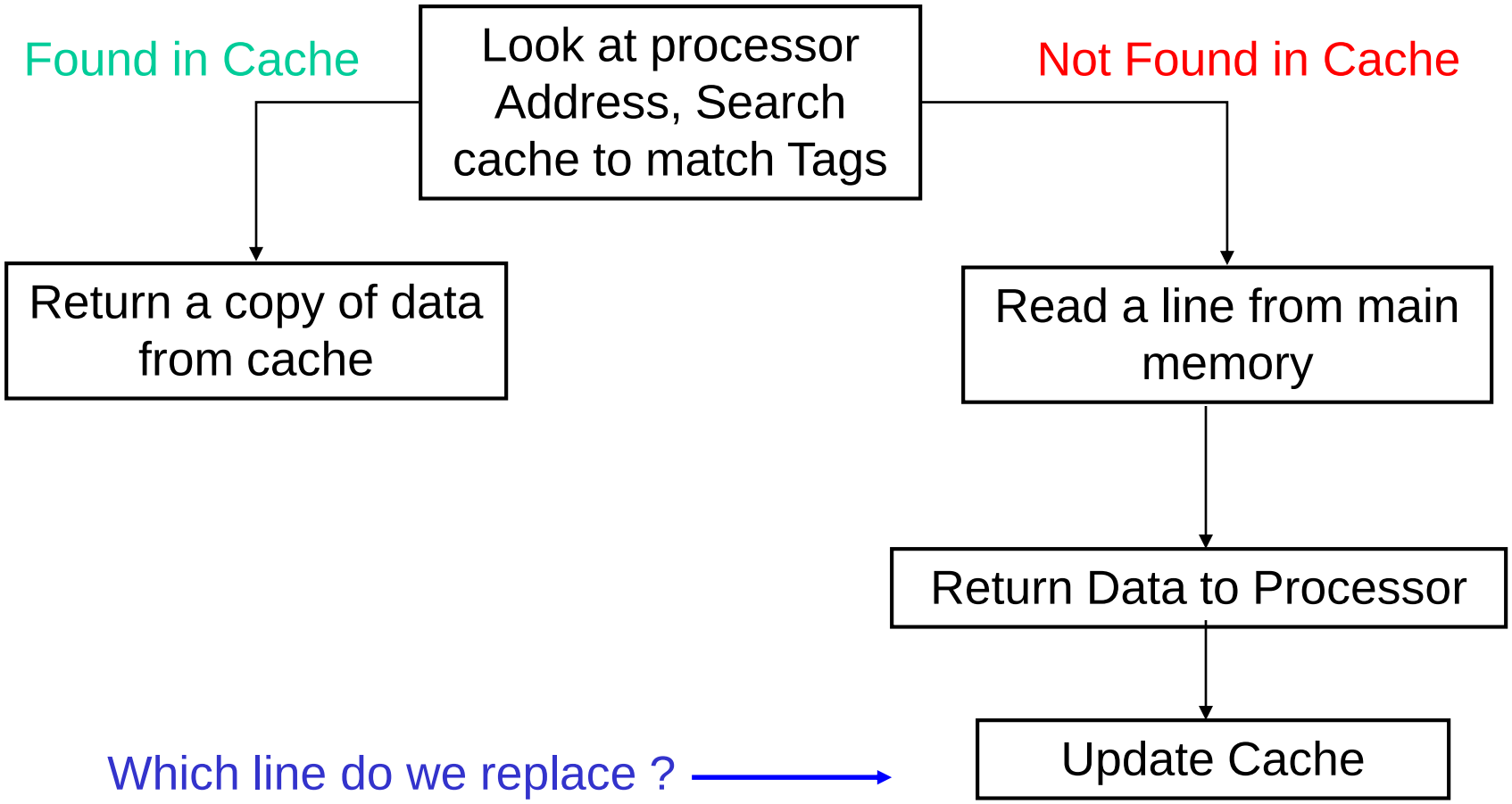


8

Cache Controller

- It’s the hardware that copies code or data from main memory to cache memory automatically.
- The operation is transparent to CPU and the software
- It intercepts reads and writes before passing them on to the memory controller.
- The relationship between main memory and cache is called as mapping

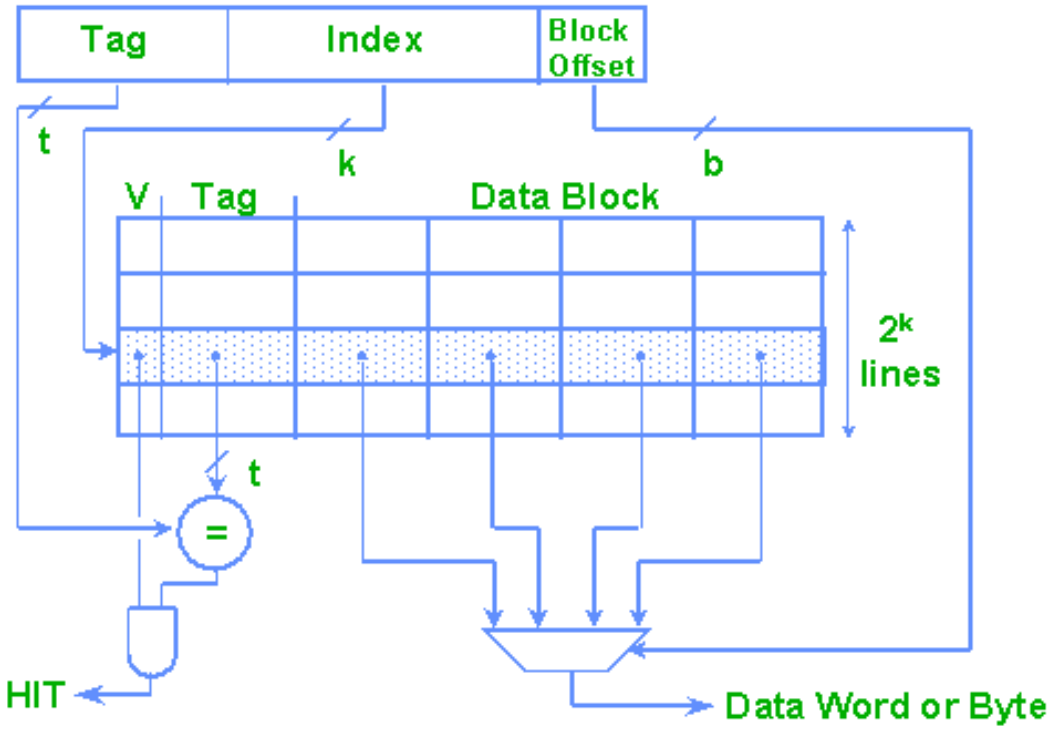
Cache Read Algorithm



Cache Mapping

- Mapping between main memory locations and cache
  - Direct mapped
  - N-way set-associative (n=2,4)
  - Fully set-associative

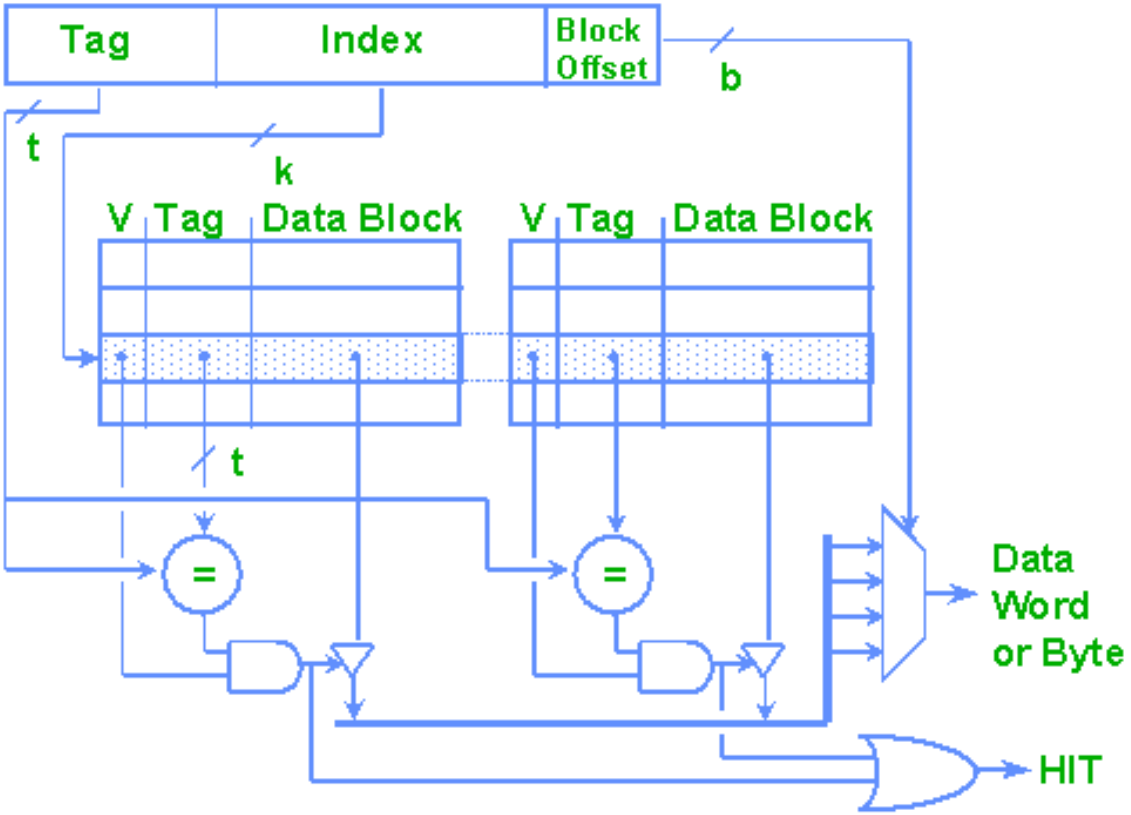
Direct mapped



Direct Mapped Cache

- Address is broken into 3 parts
  - Bottom ‘b’ bits correspond to offset with the line
  - Next ‘k’ bits access an entry into the cache
  - Top ‘t’ bits act as a tag
- Replacement policy is simple to implement
  - Each memory location has only destination in cache
- Disadvantage is – if two memory lines are used alternatively, which map onto the same cache line, the hit ratio will sharply drop

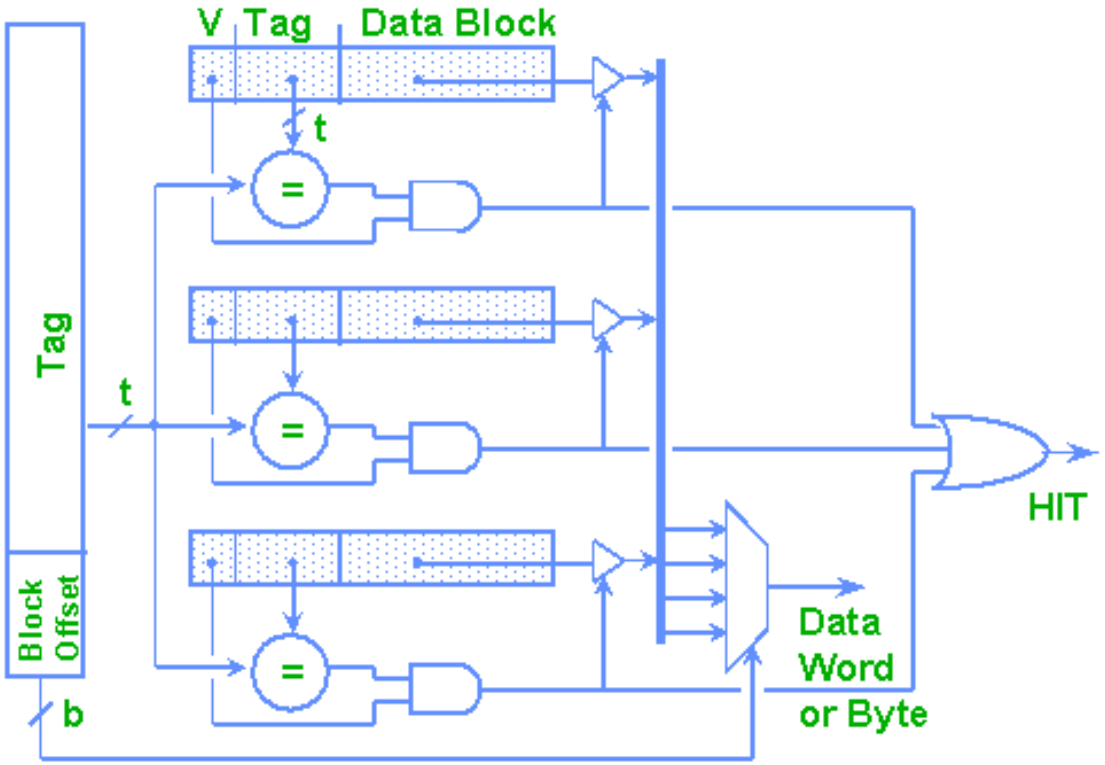
2-way Set Associative Cache



2-way Set Associative Cache

- Improved direct mapped cache
- Each memory line has two destinations in cache
- Extra logic is required for comparing tags
- Replacement policy needs consideration

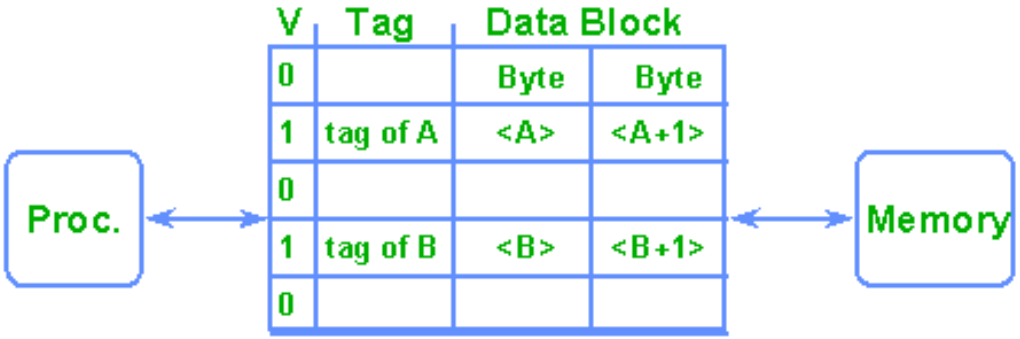
Fully Associative Cache



Fully Associative Cache

- This scheme represents other extreme
  - Any line can be placed in any location in cache
- Gives highest hit ratio
- Search is slowest
- Content addressable Memories (CAM) are used for storing Tags

Valid bits



- Valid bit must be 1 for cache line to HIT.
- At power-up or reset, we set all valid bits to 0.
- Set valid bit to 1 when cache line is first replaced.
- Flush cache by setting all valid bits to 0, under external program control.

Cache Efficiency

- Two terms are use to characterise cache efficiency of a program
  - Cache hit rate
  - Cache miss rate

$$\text{Hit rate} = \frac{\text{cache hits}}{\text{memory requests}} \times 100$$

  - Rates can measure read, writes or both
  - Other performance measurement terms are hit time and miss penalty

Cache Policy

- Three policies determine cache operation
  - Write policy
    - Determines where data is stored during processor write operation
  - Replacement policy
    - Determines the line that will be used for next line fill during a cache miss
  - Allocation policy
    - Determines when the cache controller allocates a line

Cache Coherency Problem

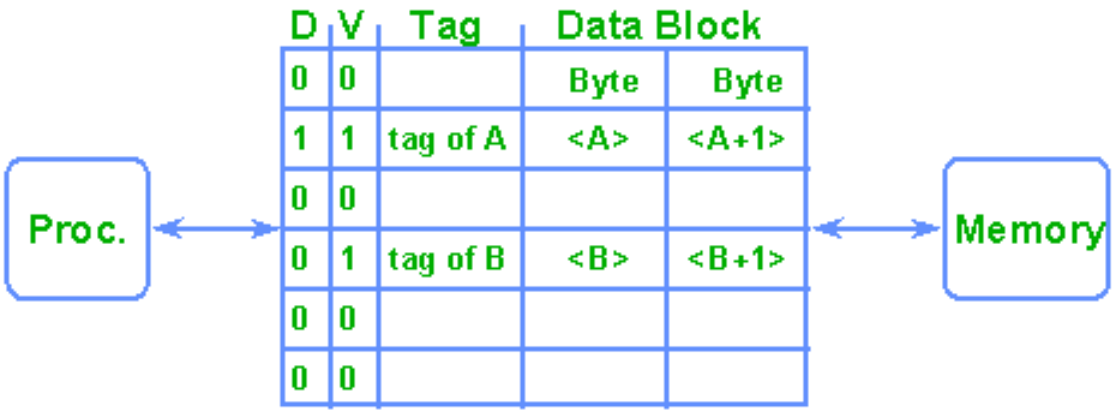
- It arises when the data in the cache is different as compared to the data in the main memory. Stale data.
- This problem is of concern only when there is more than one entity using the memory. e.g. multiprocessor systems, systems with DMA agents.

- To avoid the problem write-through policy is used.
- The cache controller snoops the memory bus for memory operations and updates the cache whenever required. This is called as snooping.

Write Policy

- Two alternatives when processor issues a write
  - Update both cache and main memory: write-through
  - Update only the cache: write-back
- Write-through
  - Both memories remain coherent all the time
  - Writes are slower
- Write-back
  - Valid cache lines and memory may not be coherent
  - Dirty bits are used to indicate non-coherency
  - Dirty cache line are updated before eviction

Dirty bits for write-back caches



When the line corresponding to A is replaced, its data block has to be written to main memory since its dirty bit is set.

18

Replacement Policy

- After a read miss, which cache line should be replaced with the data line read from main memory
  - Direct mapped cache: unique line
  - Associative cache
    - Least Recently Used(LRU): Replace line in set corresponding to address which has not been read or written for the longest time
    - First-in First Out(FIFO): Select line which has been in the set for the longest time
    - Random: Select line in the set randomly

Allocation Policy

- Two strategies to allocate cache line on a cache miss
  - Read-allocate
    - Allocation is done during read from main memory
    - A write does not update cache unless a line was allocated on previous read
  - Read-write-allocate
    - Allocation is done during either a read or write
    - On write, if cache line is not valid, it does a cache fill before updating the cache.
  - Write policy determines whether main memory should be updated

Thank You

ashishk @ cdac.in