

Digital Logic Design

ECEN 3233

Module 10 - Memory

Weihua Sheng
School of Electrical and Computer Engineering
Oklahoma State University

Spring 2007

Portions © Copyright 2004, Gaetano Borriello
and Randy H. Katz

1

Read/Write Memories

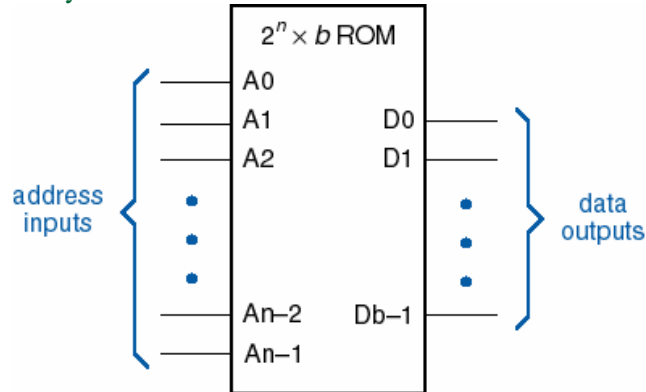
This module presents a brief introduction to the operation of read-only memory, programmable ROM, static and dynamic read/write semiconductor memory devices.

- a.k.a. “ROM” (Read Only Memory)
- a.k.a. “RAM” (Random Access Memory)
- Volatility
 - Most RAMs lose their memory when power is removed
 - NVRAM = RAM + battery
 - Or use ROM/EEPROM
- ROM
 - “Permanent”
 - Combinational – Lookup Table
- Programmable ROM (EPROM, EEPROM, Flash)
 - Can modify the data through special programming methods
- SRAM (Static RAM)
 - Memory behaves like latches or flip-flops
- DRAM (Dynamic Memory)
 - Memory lasts only for a few milliseconds
 - Must “refresh” locations by reading or writing

Portions © Copyright 2004, Gaetano Borriello and Randy H.
Katz

2

Read-Only Memories



A memory device stores binary data at an array of externally selectable addresses (indices)

Memories can be either Read-Only or Read-Write, and Volatile or Non-Volatile

Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

3

Why “ROM”?

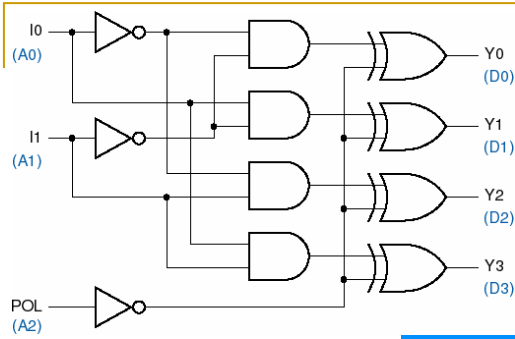
- Program storage
 - Boot ROM for personal computers
 - Complete application storage for embedded systems
- Actually, a ROM is a combinational circuit, basically a truth-table lookup.
 - Can perform any combinational logic function
 - Address inputs = function inputs
 - Data outputs = function outputs

- Thus, you can think of 2^m word, n -bit wide memory devices as look-up tables useful for realizing up to n Boolean functions of m variables

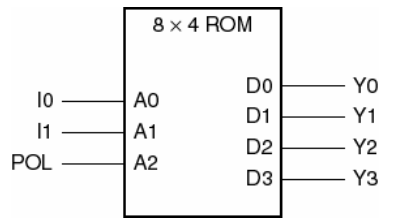
Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

4

Logic-in-ROM Example



Circuit inputs form the address
Desired outputs (D's) are stored at each address



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

Portions © Copyright Katz

Read-Only Memories

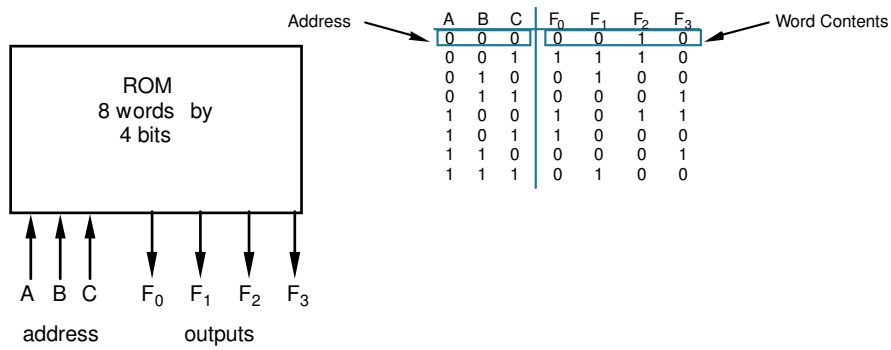
Example: Combinational Logic Implementation

$$F_0 = A' B' C + A B' C' + A B' C$$

$$F_1 = A' B' C + A' B C' + A B C$$

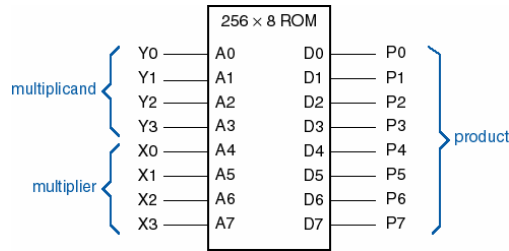
$$F_2 = A' B' C' + A' B' C + A B' C'$$

$$F_3 = A' B C + A B' C' + A B C'$$



Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

4x4 Multiplier Example



```

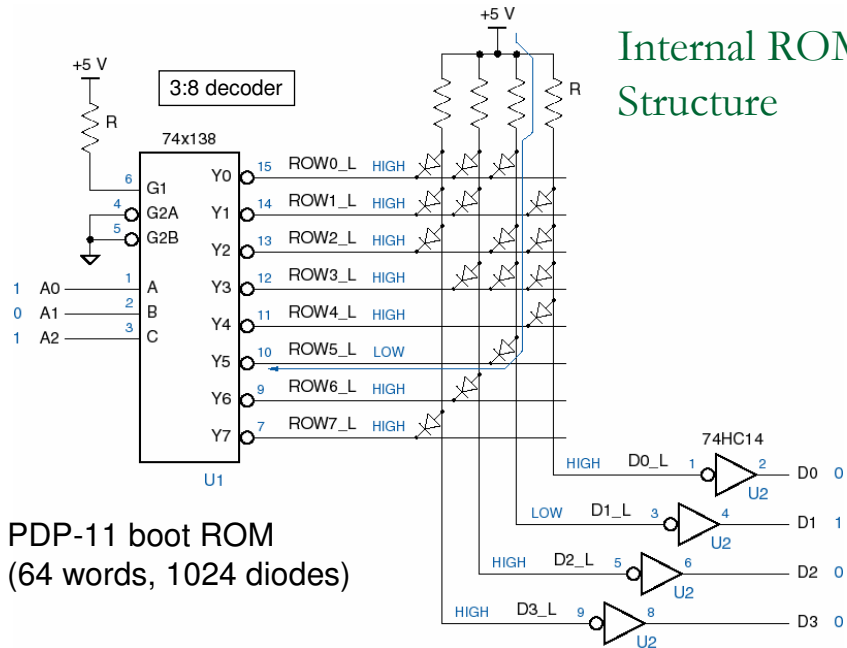
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
20: 00 02 04 06 08 0A 0C 0E 10 12 14 16 18 1A 1C 1E
30: 00 03 06 09 0C 0F 12 15 18 1B 1E 21 24 27 2A 2D
40: 00 04 08 0C 10 14 18 1C 20 24 28 2C 30 34 38 3C
50: 00 05 0A 0F 14 19 1E 23 28 2D 32 37 3C 41 46 4B
60: 00 06 0C 12 18 1E 24 2A 30 36 3C 42 48 4E 54 5A
70: 00 07 0E 15 1C 23 2A 31 38 3F 46 4D 54 5B 62 69
80: 00 08 10 18 20 28 30 38 40 48 50 58 60 68 70 78
90: 00 09 12 1B 24 2D 36 3F 48 51 5A 63 6C 75 7E 87
A0: 00 0A 14 1E 28 32 3C 46 50 5A 64 6E 78 82 8C 96
B0: 00 0B 16 21 2C 37 42 4D 58 63 6E 79 84 8F 9A A5
C0: 00 0C 18 24 30 3C 48 54 60 6C 78 84 90 9C A8 B4
D0: 00 0D 1A 27 34 41 4E 5B 68 75 82 8F 9C A9 B6 C3
E0: 00 0E 1C 2A 38 46 54 62 70 7E 8C 9A A8 B6 C4 D2
F0: 00 0F 1E 2D 3C 4B 5A 69 78 87 96 A5 B4 C3 D2 E1
    
```

The 4-bit multiplier + 4-bit multiplicand form an 8-bit address.

The ROM has the product for every combination of multiplier and multiplicand pre-stored.

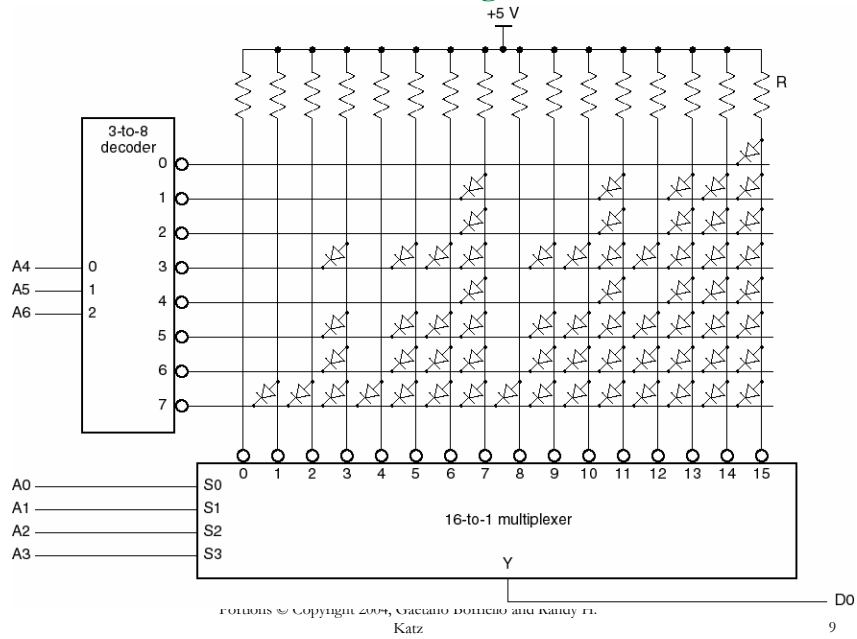
Applying a pair of multiplier and multiplicand as an address selects the corresponding 8-bit product stored at the corresponding storage location.

Internal ROM Structure



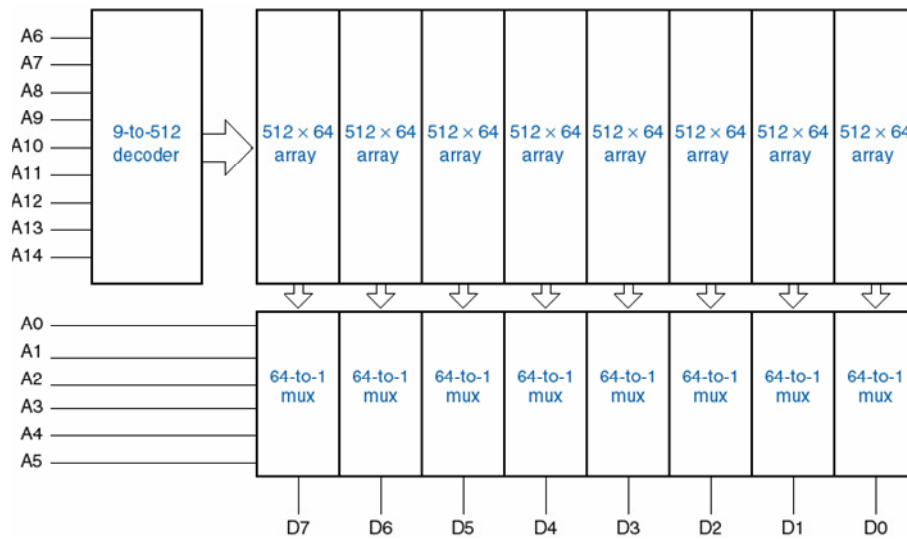
PDP-11 boot ROM
(64 words, 1024 diodes)

Two-Dimensional Decoding



9

Larger Example, 32Kx8 ROM

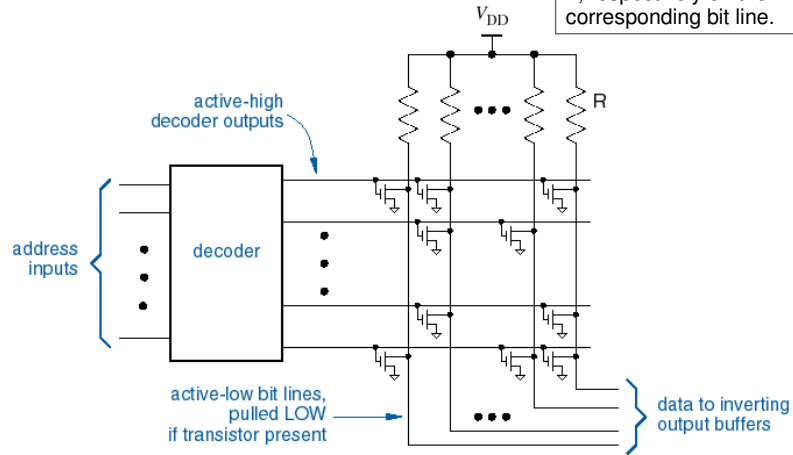


10

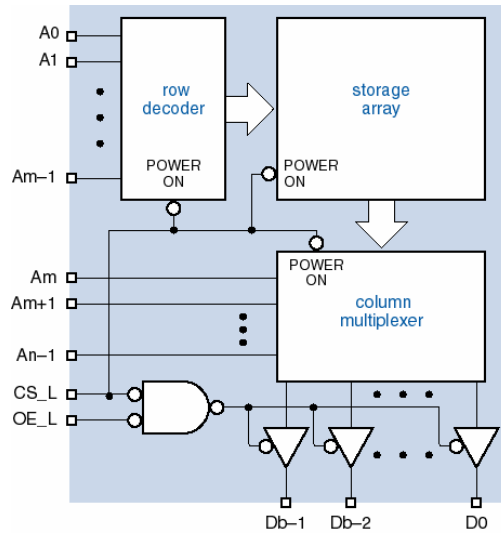
Today's ROMs

- 256K bytes, 1M byte, or larger
- Use MOS transistors

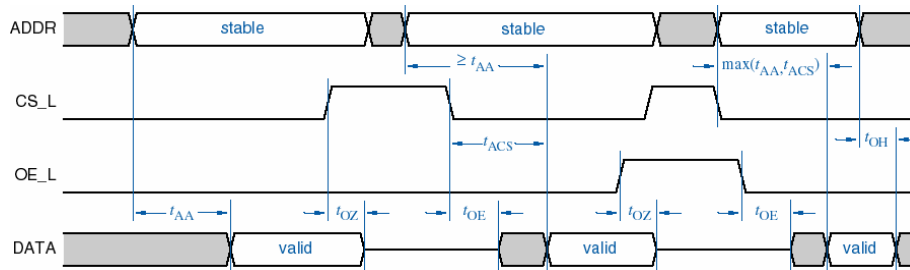
The transistor is present or absent if we want to store a logic-0 or a logic-1, respectively on the corresponding bit line.



ROM Control and I/O Signals



ROM Timing



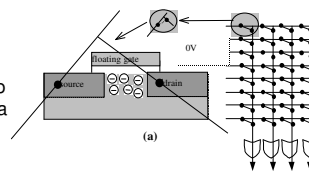
Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

13

EPROM: Erasable programmable ROM

■ Programmable component is a MOS transistor

- Transistor has "floating" gate surrounded by an insulator
- (a) Negative charges form a channel between source and drain storing a logic 1
- (b) Large positive voltage at gate causes negative charges to move out of channel and get trapped in floating gate storing a logic 0
- (c) (Erase) Shining UV rays on surface of floating-gate causes negative charges to return to channel from floating gate restoring the logic 1
- (d) An EPROM package showing quartz window through which UV light can pass



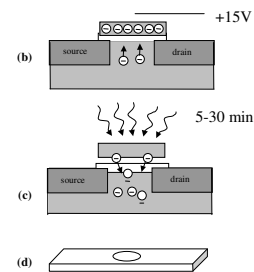
■ Better write ability

- can be erased and reprogrammed thousands of times

■ Reduced storage permanence

- program lasts about 10 years but is susceptible to radiation and electric noise

■ Typically used during design development



Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

14

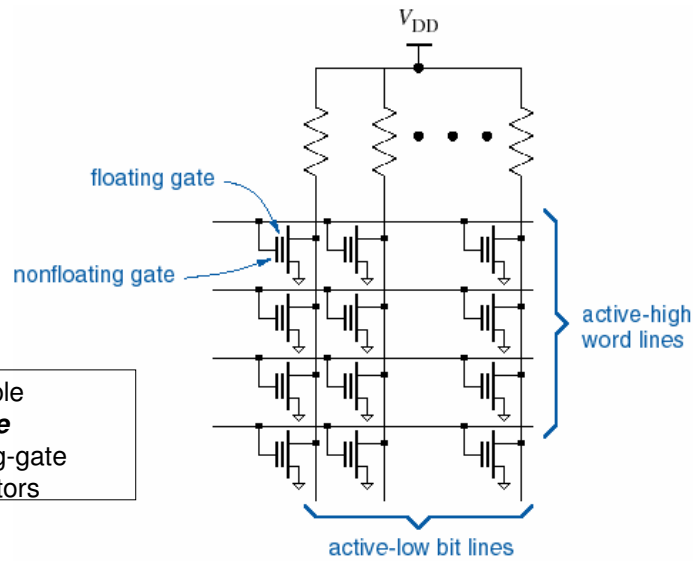
EEPROM: Electrically erasable programmable ROM

- Programmed and erased electronically
 - typically by using higher than normal voltage
 - can program and erase individual words
- Better write ability
 - can be in-system programmable with built-in circuit to provide higher than normal voltage
 - built-in memory controller commonly used to hide details from memory user
 - writes very slow due to erasing and programming
 - “busy” pin indicates to processor EEPROM still writing
 - can be erased and programmed tens of thousands of times
- Similar storage permanence to EPROM (about 10 years)
- Far more convenient than EPROMs, but more expensive

Flash memory

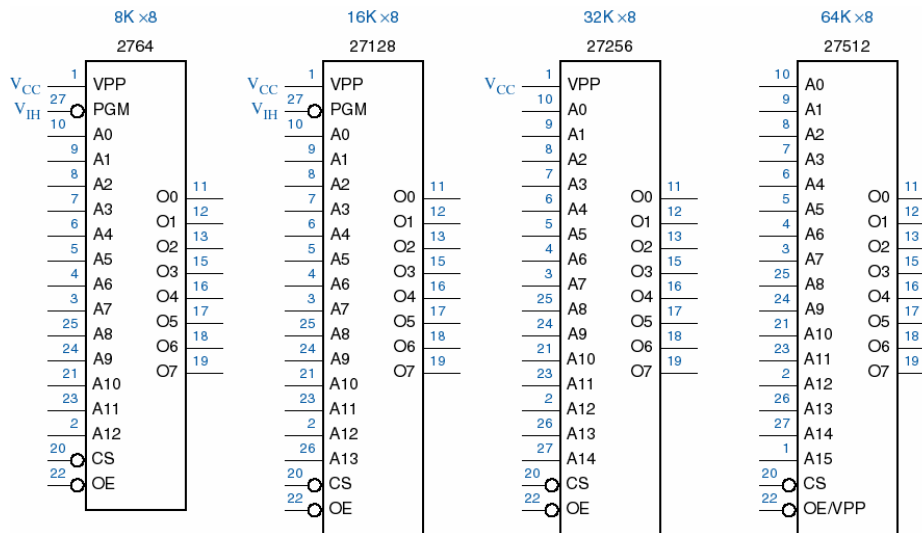
- Extension of EEPROM
 - Same floating gate principle
 - Same write ability and storage permanence
- Fast erase
 - Large blocks of memory erased at once, rather than one word at a time
 - Blocks typically several thousand bytes large
- Writes to single words may be slower
 - Entire block must be read, word updated, then entire block written back
- Used with embedded systems storing large data items in nonvolatile memory
 - e.g., digital cameras, TV set-top boxes, cell phones

EEPROMs, Flash PROMs

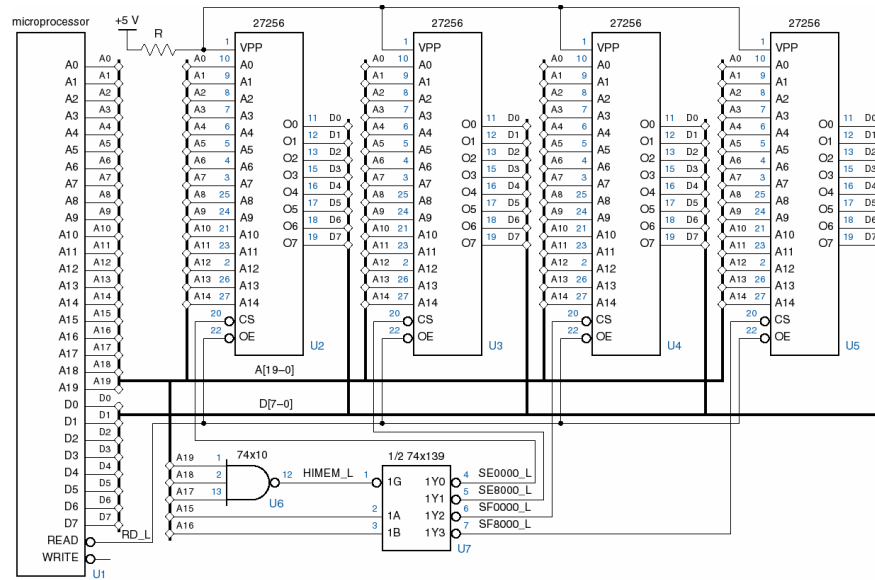


■ Programmable and **erasable** using floating-gate MOS transistors

Typical commercial EEPROMs



EEPROM Application

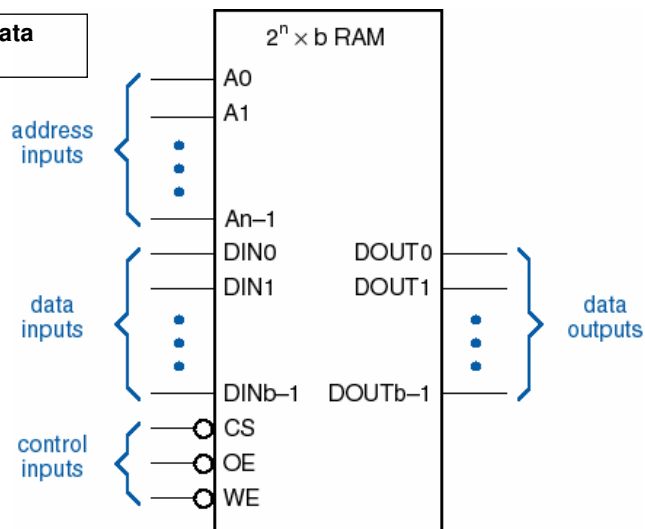


Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

19

SRAM

Typical address, data and control layout

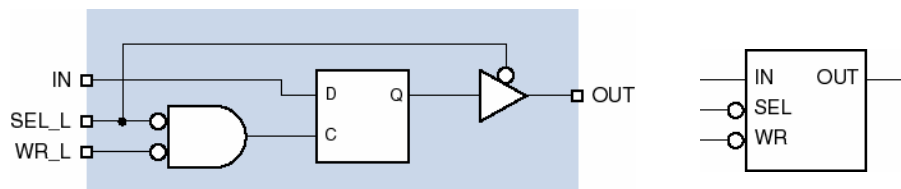


Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

20

SRAM operation

- Individual bits are D latches, (**not** edge-triggered D flip-flops.)
 - Fewer transistors per cell.
- Implications for write operations:
 - Address must be stable before writing cell.
 - Data must be stable before ending a write.

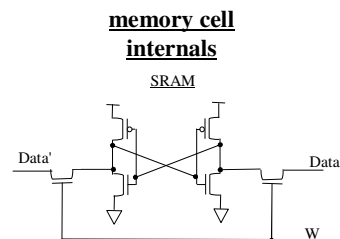


Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

21

6-Transistor SRAM Implementation

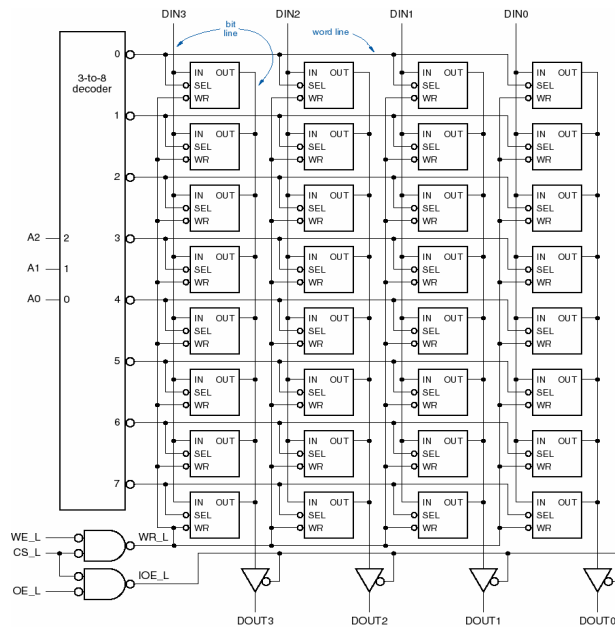
- SRAM:
 - Requires 6 transistors
 - Holds data as long as power supplied



Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

22

SRAM array

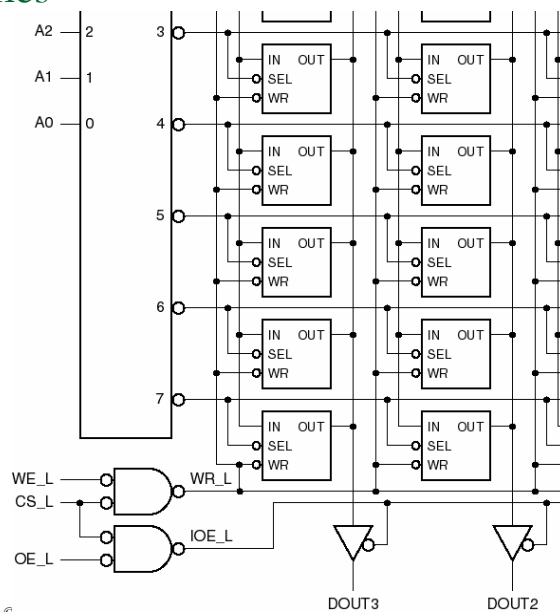


Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

23

SRAM control lines

- Chip select
- Output enable
- Write enable

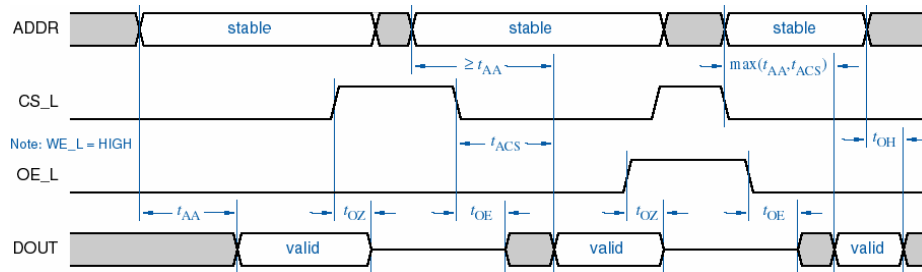


Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

24

SRAM read timing

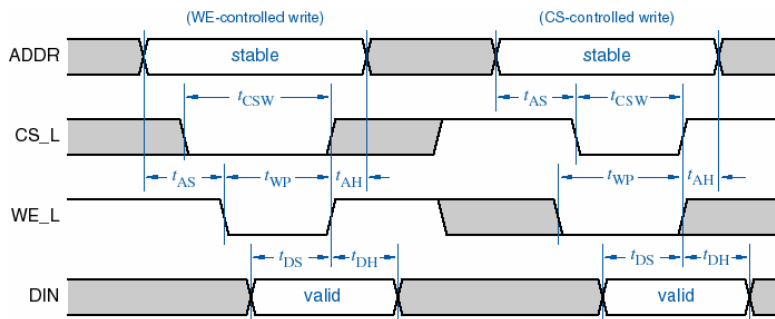
- Similar to ROM read timing



Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

25

SRAM write timing

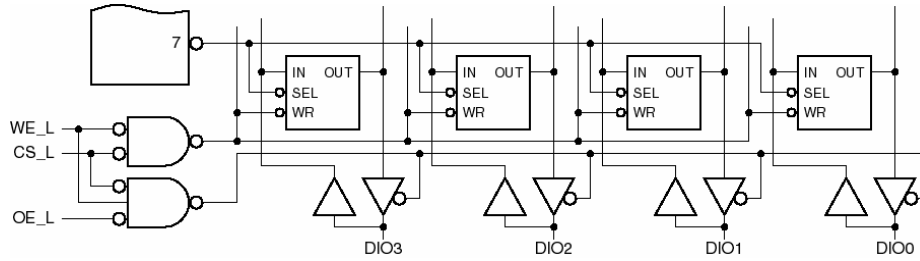


- Address must be stable before and after write-enable is asserted.
- Data is latched on trailing edge of (WE & CS).

Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

26

Bidirectional data in and out pins



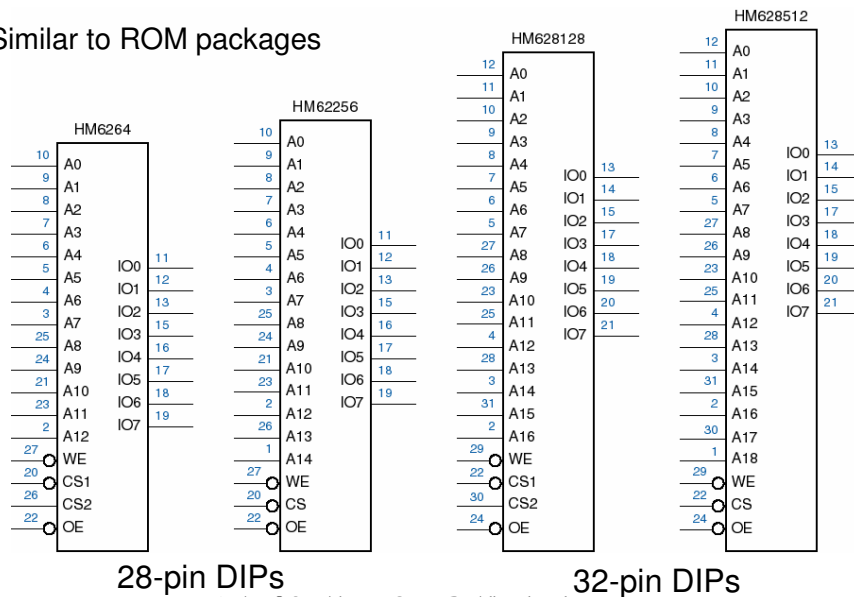
- Use the same data pins for reads and writes
 - Especially common on wide devices
 - Makes sense when used with microprocessor buses (also bidirectional)

Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

27

SRAM devices

- Similar to ROM packages



28-pin DIPs

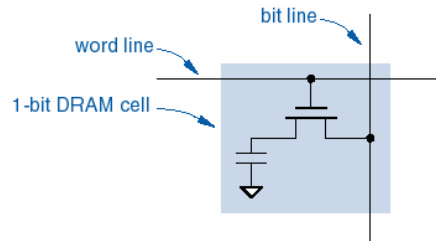
32-pin DIPs

Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

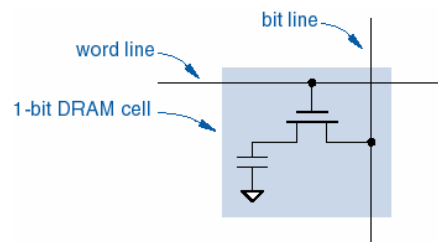
28

DRAM (Dynamic RAMs)

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

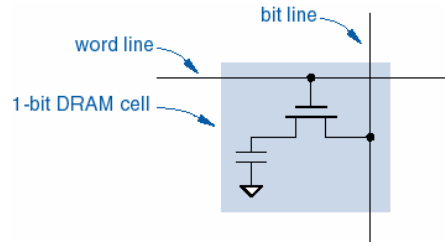


DRAM read operations



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

DRAM write operations

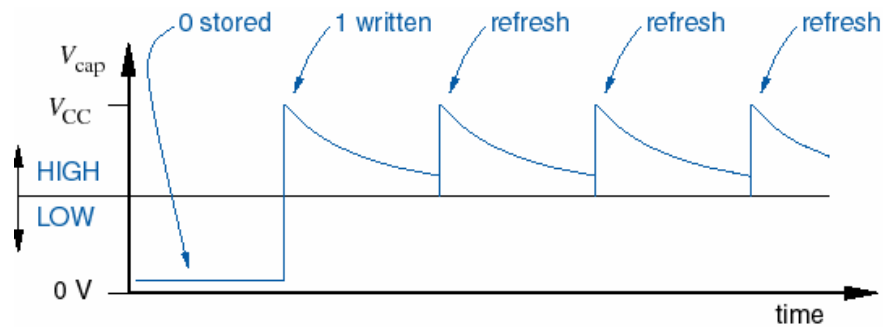


- Take the word line HIGH.
- Set the bit line LOW or HIGH to store 0 or 1.
- Take the word line LOW.
- Note: The stored charge for a 1 will eventually leak off.

Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

31

DRAM charge leakage



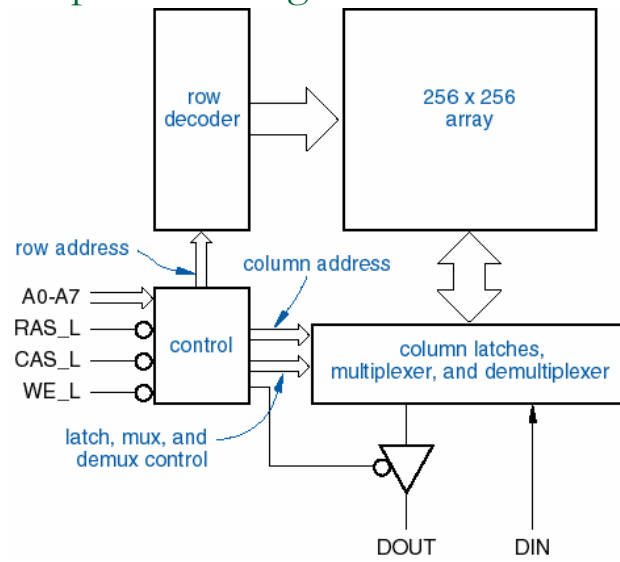
- Typical devices require each cell to be “refreshed” once every 4 to 64 mS.
- During “suspended” operation, notebook computers use power mainly for DRAM refresh.

Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

32

DRAM-chip internal organization

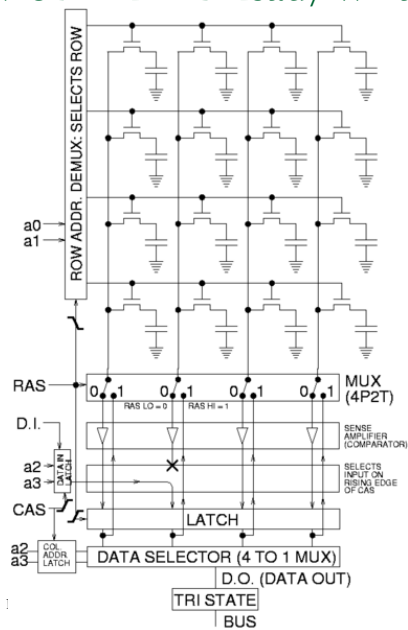
64K x 1
DRAM



Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

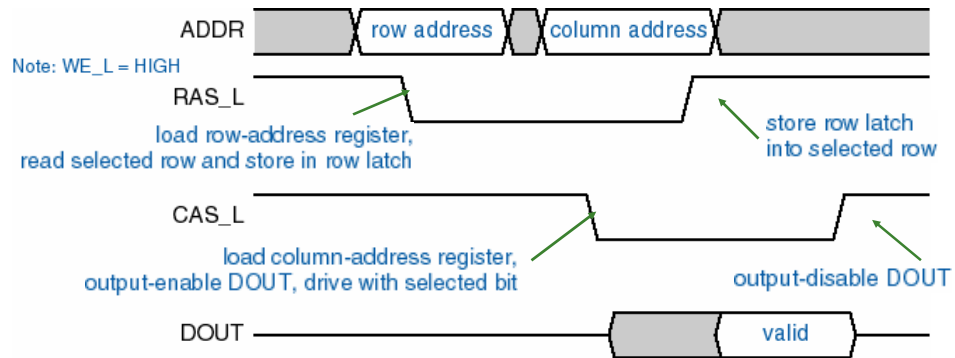
33

Detailed View of DRAM Read/Write



34

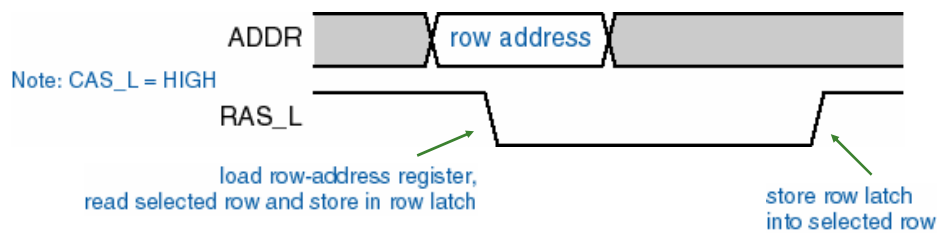
DRAM read timing



Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

35

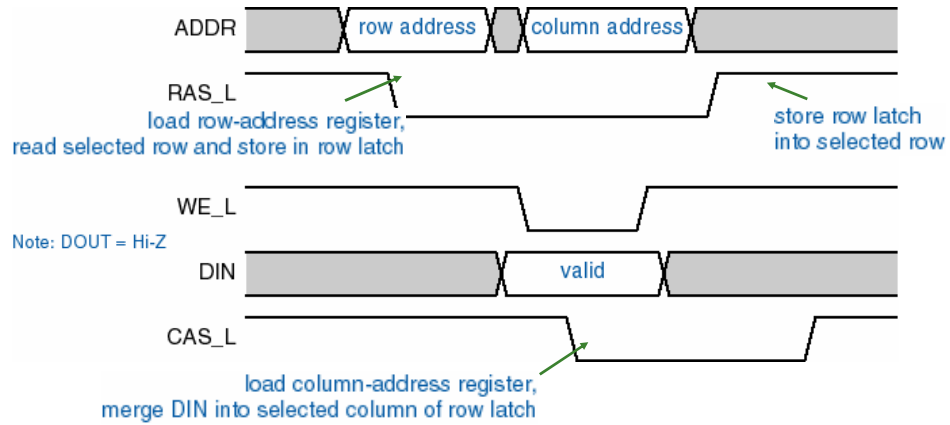
DRAM refresh timing



Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

36

DRAM write timing

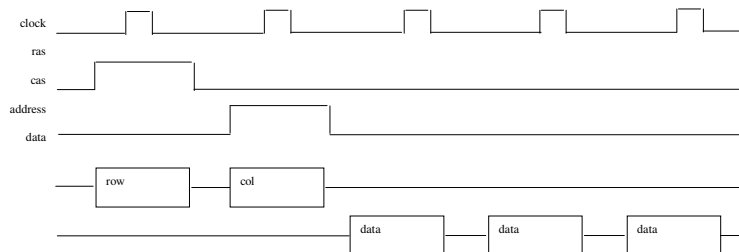


Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

37

Synchronous DRAM (SDRAM)

- SDRAM latches data on active edge of clock
- Eliminates time to detect *ras/cas* and *rd/wr* signals
- A counter is initialized to column address then incremented on active edge of clock to access consecutive memory locations



Portions © Copyright 2004, Gaetano Borriello and Randy H. Katz

38

SRAM vs DRAM

- DRAM access time is typically 10 times of SRAM
- The cost of SRAM is typically 100 times of DRAM
- SRAMs are usually used for memory cache
- DRAMs are usually used for main memories and frame buffers

Summary

- ROM
- EPROM/EEPROM/Flash ROM
- RAM
 - SRAM
 - DRAM (SDRAM)