EEL 4746: Microprocessor-based System Design

Chapter 9: Computer Memory

Dr. Ming Yu
Dept. of ECE
FAMU-FSU College of Engineering
Fall 2006

Table of Contents

9.1 Introduction

- All computers have both RAM and ROM
- RAM: random access memory
  - The semiconductor RAM is volatile: anything stored in memory is lost when the power is removed!
- ROM: read only memory
  - Once it is programmed, either at the IC factory as part of the manufacturing process, or in the field, for field-programmable devices, it can only be read!
  - ROM is nonvolatile

RAM and ROM

- RAM:
  - For variable information
  - Can be data used by the programs
  - Can also be the programs themselves in general purpose systems
- ROM:
  - For constant information that must be retained while the power is disconnected
  - For the program in specific application systems
  - For “boot-up” programs for general-purpose systems when turning on power or reset
- The amount of RAM and ROM depends on the type of system

DRAM and SRAM

- **DRAM** is a type of RAM that stores each bit of data in a separate capacitor
- As real-world capacitors are not ideal and hence leak electrons, the information eventually fades
  - unless the capacitor charge is refreshed periodically.
  - Because of this refresh requirement, it is a *dynamic* memory as opposed to SRAM and other *static* memory.
- Advantage:
  - DRAM: one transistor and a capacitor are required per bit
  - SRAM: six transistors
  - This allows DRAM to reach very high density.
- Since DRAM loses its data when the power supply is removed, it is in the class of *volatile* memory devices.

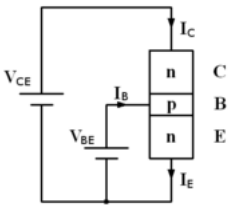
EEPROM

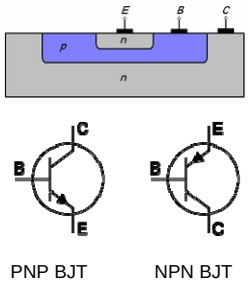
- EEPROM: electrically erasable programmable ROM, is a non-volatile storage chip used in computers and other devices to store small amounts of volatile (configuration) data.
  - The main advantage of EEPROMs over EPROMs is that they are erased electrically instead of by ultraviolet light; this is faster and can be done in-circuit.
  - While RAM has no limitations on rewrites to memory, EEPROMs are limited in that repeated write and erase cycles eventually damage the thin insulating layer, a process called 'wear out'.
- SEEPROM: serial EEPROM, is an EEPROM chip that uses a serial interface to the circuit board.

Flash Memory

- A non-volatile memory that can be electrically erased and reprogrammed
  - Unlike EEPROM, it is erased and programmed in blocks consisting of multiple locations (in early flash the entire chip had to be erased at once).
  - Flash memory stores information in an array of floating gate transistors, called "**cells**", each of which traditionally stores 1 bit of information.
  - multi-level cell devices, can store more than 1 bit per cell, by using more than two levels of electrical charge, placed on the floating gate of a cell.
- In NOR flash, each cell looks similar to a standard MOSFET, except that it has two gates instead of just one:
  - One is control gate (CG) like in other MOS transistors
  - Another is floating gate (FG) that is insulated all around by an oxide layer
  - The FG is between the CG and the substrate
  - Because the FG is isolated by its insulating oxide layer, any electrons placed on it get trapped there and thus store the information
- NAND Flash uses tunnel injection for writing and tunnel release for erasing. NAND flash memory forms the core of the removable USB interface storage devices known as USB flash drives.

BJT (**Bipolar Junction Transistor**)





PNP BJT NPN BJT

$$\alpha_T = \frac{I_{Cp}}{I_{Ep}}$$ (pnp device)

$$\beta_F = \frac{I_C}{I_B}$$

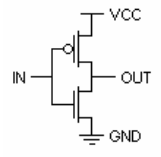
$$\beta_F = \frac{\alpha_F}{1 - \alpha_F}$$

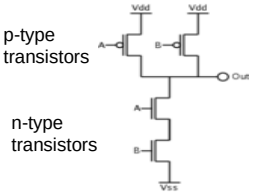
A BJT consists of three differently doped semiconductor regions, the *emitter* region, the *base* region and the *collector* region. These regions are, respectively, *p* type, *n* type and *p* type in a PNP, and *n* type, *p* type and *n* type in a NPN transistor. Each semiconductor region is connected to a terminal, appropriately labeled: *emitter* (E), *base* (B) and *collector* (C).

CMOS

- **Complementary metal–oxide–semiconductor** (**CMOS**) is a major class of ICs.
- Also explained as **complementary-symmetry metal–oxide–semiconductor**: the design uses complementary and symmetrical pairs of p-type and n-type MOSFET transistors for logic functions.
- **CMOS logic** uses a combination of p-type and n-type metal–oxide–semiconductor field-effect transistors (MOSFETs) to implement logic gates and other digital circuits
- As an example, shown on the right is a circuit diagram of a NAND gate in CMOS logic. If both of the A and B inputs are high, then: both the n-type transistors (bottom half of the diagram) will conduct, neither of the p-type transistors (top half) will conduct, and a conductive path will be established between the output and Vss, bringing the output low. If either of the A or B inputs is low, one of the n-type transistors will not conduct, one of the p-type transistors will, and a conductive path will be established between the output and Vdd, bringing the output high

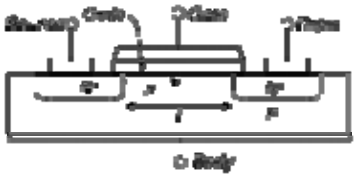
Static CMOS Inverter





NAND gate in CMOS logic

Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET)



Cross Section of an NMOS

- If the MOSFET is an N-Channel or nMOS FET, then the source and drain are 'N+' regions and the body is a 'P' region. When a positive gate-source voltage is applied, it creates an *N-channel* at the surface of the P region, just under the oxide. This channel spreads from the source to the drain and provides conductivity of the transistor. When zero or negative voltage is applied between gate and source, the channel disappears and no current can flow between the source and the drain.
- If the MOSFET is an P-Channel or pMOS FET, then the source and drain are 'P+' regions and the body is a 'N' region. When a negative gate-source voltage (positive source-gate) is applied, it creates a *P-channel* at the surface of the N region, just under the oxide. This channel spreads from the source to the drain and provides conductivity of the transistor. When no or a positive voltage is applied between gate and body, the channel disappears and no current can flow between the source and the drain

9.2 Computer Types and Memory Maps

- General purpose systems (multiple applications)
  - Use ROM for the basic I/O software and large amounts of RAM for programs and data
  - Application programs are loaded into RAM from the disk by disk operating system (DOS)
  - Firmware is in ROM, such as basic I/O software (BIOS)
  - Bootstrapping: BIOS loads the OS from the disk before other programs are executed

Figure 9-1 General-purpose computer system.

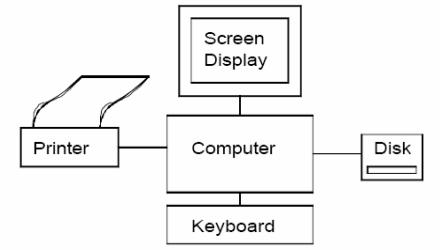
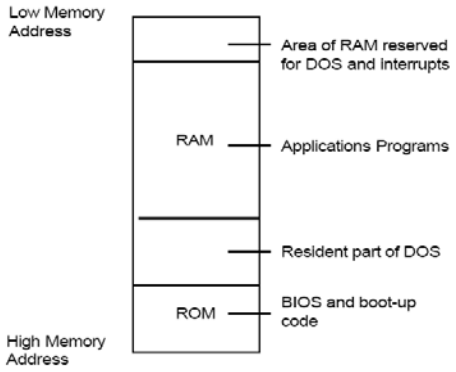


Figure 9-2 General-purpose system memory map.



Most of the memory is RAM used for the OS resident code and for application programs

11/16/2006 13

Dedicated-application systems

- Designed to do some particular job or jobs
  - Contain the least amount of hardware to accomplish the job at the least cost
  - Unless it is part of the application, there is little or no human-oriented I/O such as display, keyboard, etc.
  - The program is kept in ROM. There is no disk system from which the program can be loaded
  - Only data variables and the stack are kept in RAM
- A dedicated-application system contains
  - much more ROM for the program and
  - less RAM fro data storage than general purpose computers

11/16/2006 14

Figure 9-3 Dedicated-application system.

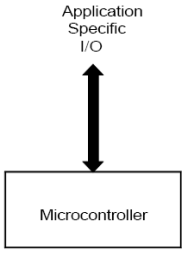
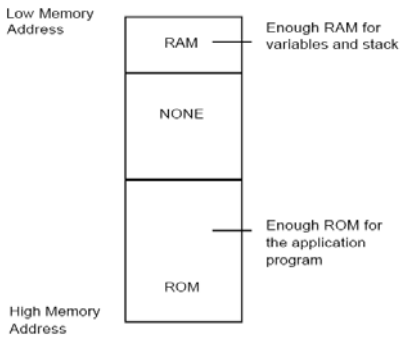


Figure 9-4 Dedicated system memory map.



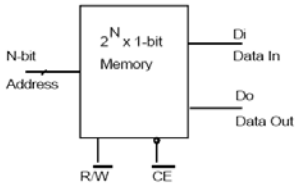
The entire memory map does not have to be filled

Memory addresses not used become "don't cares"

9.3 Semiconductor RAM

- RAM chip consists of
  - An array of memory cells
  - a decoder for addressing particular cell, and
  - signals to control the direction of data flow
- Memory Cell Types
- Static RAM chips
- Dynamic Memory
- DRAM Refresh
- Pseudostatic RAM

Figure 9-5 One bit of the 2^N x 1 bit memory.

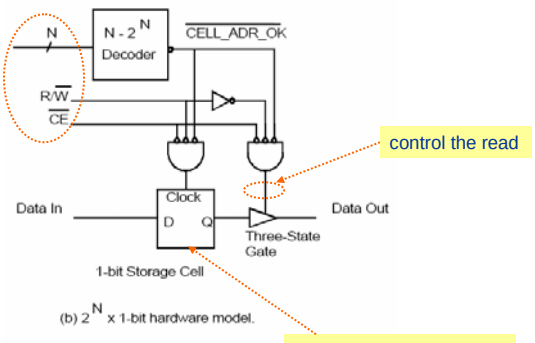


The N-bit address selects 1 of 2^N memory cells

controls whether the memory cell is being read from or written to

CE (chip enable) or CS (chip select) is derived by decoding the rest of the address bus

Figure 9-5 One bit of the 2^N x 1 bit memory.

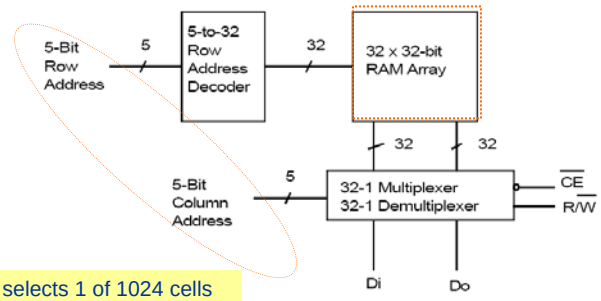


generate a clock signal for control

control the read

The 1-bit memory cell can be a D-type flip-flop or a MOS capacitor

Figure 9-6 32 x 32 bit RAM array.

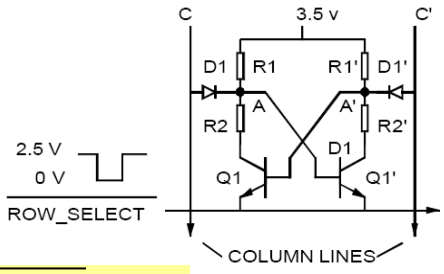


selects 1 of 1024 cells

Memory Cell Types

- Two kinds of memory cells: static memory and dynamic memory
- A static memory cell is a flip-flop
  - The transistors could be bipolar, as shown, or MOS devices.
- A dynamic memory cell is a capacitor

Figure 9-7 Static RAM (SRAM) cell.



WRITE:

asserting the **ROW_SELECT** and driving either C or C' to set Q1 or Q1' depending on whether a **0** or **1** is to be stored.

When **ROW_SELECT** is high and Q1 is on, D1 and D1' isolate the cell from the column lines C and C', the row is not selected.

Current flows through R1 and R2, make voltage at A higher than line C. Q1' is off, make the voltage at A' higher than C'.

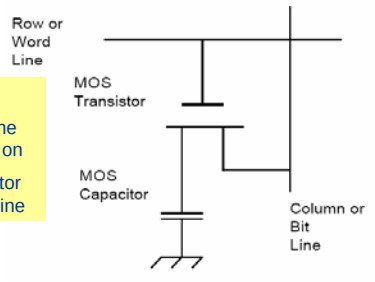
READ:

When the cell is selected, **ROW_SELECT** is low, A becomes lower than C, current flows in D1 from C. This current flow could signify a logic **1** stored in the cell.

A logic **0** is stored by turning Q1 off and Q1' on.

When **ROW_SELECT** is asserted, C will not have current flow and C' will.

Figure 9-8 Dynamic RAM (DRAM) cell.



write:

1. activate the row line to turn the transistor on
2. charge the capacitor through the column line

read:

1. turn the transistor on
2. sense a voltage on the column line

Dynamic memory cell: is a capacitor where the absence or presence of charge denotes a stored one or zero

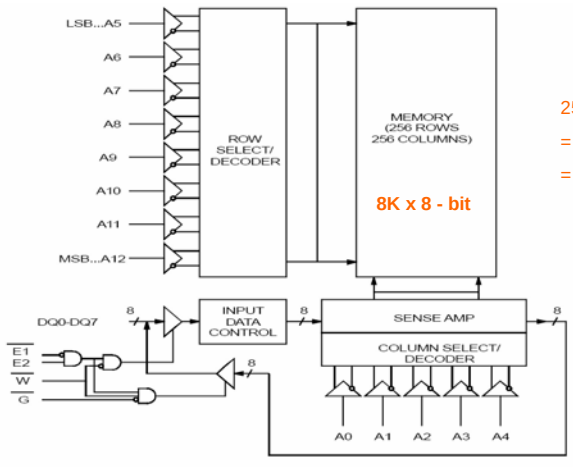
Problem: the charge stored on the capacitor leaks away to the substrate

Solution: the dynamic memory must be refreshed at periodic intervals by activating the ROW_SELECT line while holding all column lines at a particular voltage level. All cells in the row can have the capacitor's charge (or lack of charge) refreshed at once.

Static RAM Chips

- Static RAM (SRAM)
  - consists of arrays of flip-flops
  - SRAM has lower bit density and thus
  - lower storage capabilities than DRAM
  - simpler to use: no need to be refreshed.

Figure 9-9 Motorola MCM6064 SRAM (reprinted with permission of Motorola Corp.).

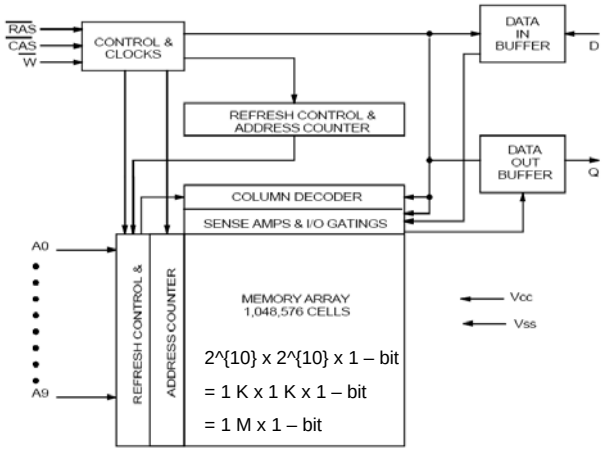


256 x 256 x 1 – bit
= 2^3 x 2^{10} x 2^3 – bit
= 8 K x 8 – bit

Dynamic Memory

- See an example:
  - The chip has 10 address bits
  - Separate data-in and data-out pins
  - A write enable: **W**
  - Two other control signals:
    - Row address strobe: **RAS**
    - Column address strobe: **CAS**
  - Control the multiplexing of the two 10-bit address fields that make up the full 20-bit address required for the 1M bits.

Figure 9-10 Intel 1,048,576 x 1-bit dynamic RAM (reprinted with permission of Intel Corp.).



DRAM Refresh

- The DRAM memory cell requires a periodic refresh operation
  - The refresh timing depends on the manufacturer and the chip, typically 2 ~ 4 ms.
- Refresh methods
  - RAS-only refresh:
    - The row addresses are strobed by asserting RAS while CAS is held high
    - The cycle must be repeated for every row address
  - CAS-before-RAS refresh:
    - If CAS is held low a specific time before RAS is asserted, on-chip refresh circuitry automatically furnishes the refresh address.
    - Eliminates the need for external refresh addresses
    - This method takes slightly longer than RAS-only refresh
  - Hidden refresh
    - This refresh is done while maintaining the latest valid data at the output and extending CAS and cycling RAS
- DRAM controller:
  - used for refreshing DRAM
  - interfaces the DRAM chips to the system bus and handle all refresh operations automatically

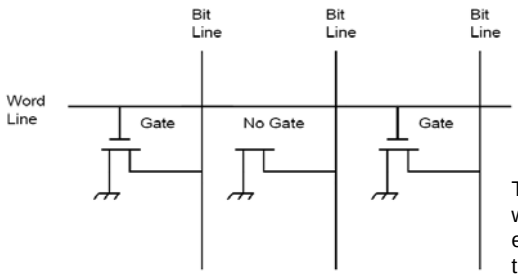
Pseudostatic RAM

- Pseudostatic RAM is dynamic RAM with on-chip refresh circuitry
  - Combines the high storage capacity of DRAM and ease of use of SRAM
  - Dynamic storage cells are used and on-chip refresh circuitry is included so that the devices appears to be the user as SRAM
- To avoid conflict when the system attempts to access the memory while an internal refresh is taking place:
  - Two strategies:
    - A separate pin may be included to tell the RAM when it can execute a refresh cycle without conflicting with an external access request. External logic can pulse this input to refresh the chip.
    - A “ready” or “wait” output from the RAM may be used for handshaking in a system where “wait state” can be generated.

9.4 ROM Memory

- Mask programmable ROMs
  - The system designer decides what is to go into the ROM and then specifies the *mask* used by the manufacturer
- Field programmable, called programmable ROM
  - UV-erasable PROMs (EPROMs)
  - One-time programmable EPROMs
  - Fusible-link PROMs
  - Electrically-erasable PROM (EEPROM)
- The least expensive ROM for large production runs is mask programmed at the factory
- For system development and small production runs, field programmable ROMs are preferred.

Figure 9-11 ROM cell. (Mask PROM Cell)



The ROM cell is simply a wire or connection that is either made or not made in the programming process.

The binary information is represented by the presence or absence of the gate on the MOS transistor.

Activating the word line put a **1** or a **0** on the bit line.

PROM

- Two categories: erasable and nonerasable
- Two types of erasable PROMs:
  - (ultraviolet) UV-erasable EPROM
  - electrically-erasable EEPROM
    - also called electrically-alterable, EAPROM.
- PROM programmer
  - To change the state of the gate, electrons are either injected an avalanche mechanism into the silicon floating gate or not.
- PROM eraser
  - This disperses any charges stored in the floating gate back into the substrate and erase the memory

Figure 9-12 EPROM storage cell.

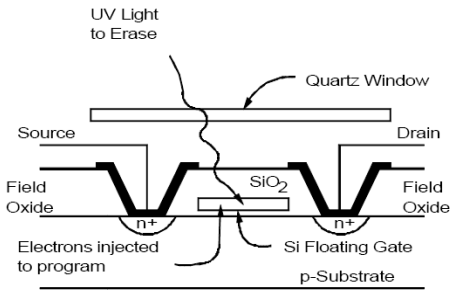
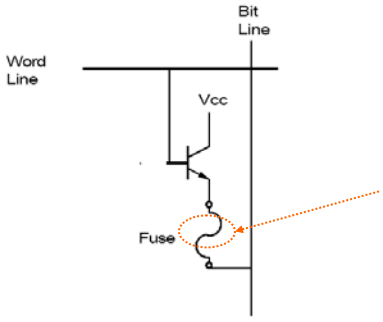


Figure 9-13 Fusible-link ROM.



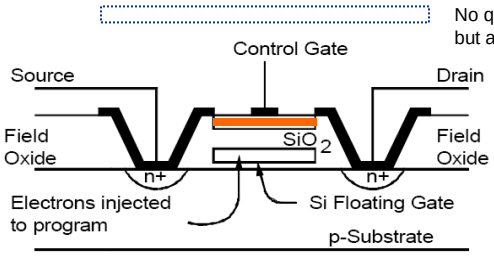
Fusible-link ROMs may be programmed once.

A bipolar transistor connects the word line to the bit line through the polycrystalline fuse.

Nonvolatile RAM – EEPROM, FLASH, and NVRAM

- EEPROM
- Flash memory
- NVRAM: shadow RAM or nonvolatile RAM
  - NVRAM is a standard RAM cell and one EEPROM cell for each memory location.
  - The RAM is read and write at full speed
  - No limit on the number of read/write cycles.

Figure 9-14 Electrically erasable PROM.



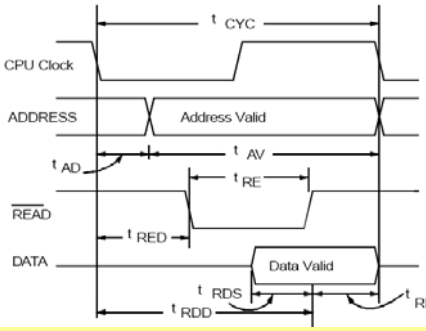
A control voltage may be applied to this gate to program and erase the cell by injecting or dispersing electrons in the floating gate

EEPROM can be erased and reprogrammed without having to be removed from the circuit

9.5 Memory Timing Requirements

- Must consider the timing requirements of both the CPU and the memory
  - CPU controls the information transfer in the system
    - Generates the control signals, such READ/ WRITE, and, in the absence of a handshaking signal such as WAIT or READY, takes data from or puts data on the bus at specific times,
  - The CPU clock controls the overall timing,
  - A processor may use more clock cycles for each read or write cycle.

Typical CPU Read Cycles



t_CYC: cycle time, the total time to complete a read or write cycle.

t_AD: address delay, the delay from the start of the write or read cycle until the address appears on the external address bus. This delay accounts for multiplexing and other CPU-generated delays.

t_AV: address valid, the time that the address is valid on the external address bus. The CPU takes it away or changes it at the end of write or read cycle.

t_RED: read enable delay, the delay from the start of the read cycle until the read enable signal is asserted. This is found in CPUs that have separate READ and WRITE control signals.

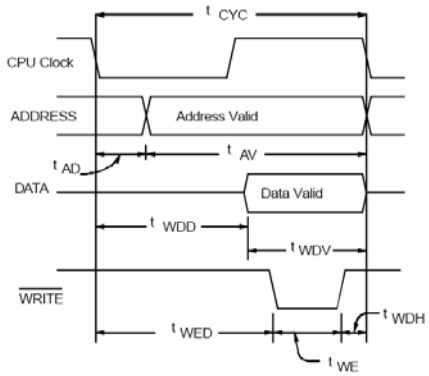
t_RE: read enable pulse length, the duration of the READ signal.

t_RDD: read data delay, the CPU waits for this time before it reads the data from the data bus.

t_RDS: read data setup, the time the data must be valid before they are read by the CPU.

t_RDH: read data hold, the CPU may require the data to be held after it reads them.

Typical CPU Write Cycles



t_WDD, write data delay, the CPU waits for this time before it places the data to be written to memory on the data bus.

t_WDV, write data valid, the time the CPU keeps the data on the data bus.

t_WED, write enable delay, the CPU waits fro this time before it asserts the write enable signal.

t_WE, write enable pulse length, the duration of the WRITE signal.

t_WDH, write data hold, the time the CPU holds the data bus after de-asserting the write enable signal.

Table 9-6: Timing Specifications for Motorola MC68HC11A8

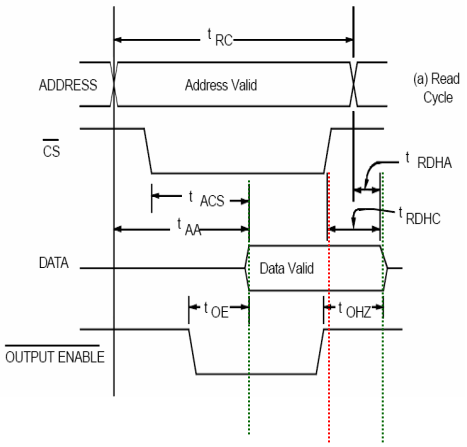
Symbol	Parameter	2 MHz Clock (ns)
t_CYC	Cycle time	500
t_AD	Address delay	123
t_AV	Address valid	379
t_RED	Read enable delay	247
t_RE	Read enable pulse length	222
t_RDD	Read data delay	469
t_RDS	Read data setup	30
t_RDH	Read data hold	30

t_WED	Write enable delay	133
t_WE	Write enable pulse length	389
t_WDD	Write data delay	375
t_WDV	Write data valid	147
t_WDH	Write data hold	0

Memory Read and Write Cycles

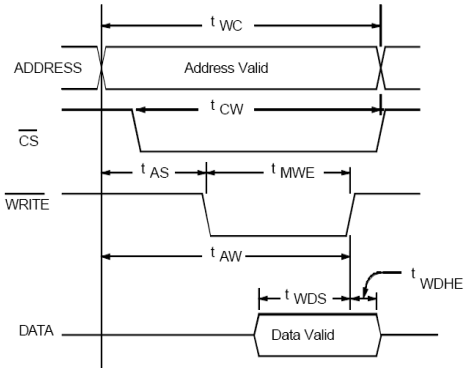
- Important times for reading data
  - **t_RC: the memory cycle time**, is the minimum time that the address must be stable (unchanging) at the chip.
  - **t_AA: the address access time**, is the maximum time required by the memory before the data are available.
- Important time for writing data
  - **t_WC, the write cycle time**, a minimum time that the address must be present and stable at the chip.
    - For some memories, the chip select signal must go low at least t_CW (chip selection to the end of write) ns before the time the CPU takes the data away.
    - In other memories, t_CW is not an important parameter.
  - The write enable signal, WRITE, may be asserted t_AS (address setup time) after the address are valid.
    - The data being written into the memory must be valid at least **t_WDS (write data setup)** ns and
    - must be held for **the data hold time, t_WDHE**, after the WRITE goes high.

Memory Read Cycles



t_RC: read cycle, this is the total time for the read cycle.
t_ACS: chip select access, the maximum time required by the memory for the **CS** to be asserted before the data are available.
t_AA: address access, this is the maximum time required by the memory for the address to be presented before the data are available.
t_RDHA: read data hold after address, the time the memory may hold the data at the output after the address is changed.
t_RDHC: read data hold after chip select, the minimum time the chip will hold the data after being-deselected.
t_OE: output enable access, on chips that have an output enable, this parameter gives the maximum time for the chip to respond with the data.
t_OHZ: output enable to output high Z, on chips that have an output enable, this parameter specifies the time the data will remain valid before going into three-state (high impedance).

Memory Write Cycles



t_WC: write cycle. The minimum total time required by the memory to complete a write cycle. This may or may not be the same as the read cycle time **t_RC**.

t_CW: chip select to end of write. The minimum time the **CS** signal must be asserted.

t_AS: address setup. The minimum time the address must be valid before the **WRITE** signal is asserted.

t_MWE: write enable. The minimum time **WRITE** must be asserted.

t_AW: address valid to end of write. The minimum time the address must be valid.

t_WDS: write data setup. The minimum time the data must be valid before the end of write enable.

t_WDHE: write data hold after enable. The minimum time the data must be valid after the **WRITE** signal is de-asserted.

Table 9-7: MCM6064-12 Timing

Read Cycle

Symbol	Parameter	ns Min	ns Max
t_RC	Read cycle time	120	
t_ACS	CS access time		120
t_AA	Addr access time		120
t_RDHA	Read data hold after address	10	
t_RDHC	Read data hold after CS	10	
t_OE	Output enable access time		40
t_OHZ	Output enable to output high Z		40

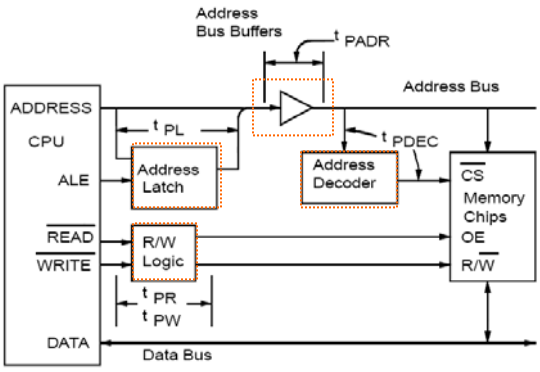
Write Cycle

Symbol	Parameter	ns Min	ns Max
t_WC	Write cycle time	120	
t_CW	CS to end of write	NA	
t_AS	Address setup time	0	
t_MWE	write enable width	60	
t_AW	Address valid to end of write	85	
t_WDS	Write data setup	50	
t_WDHE	Write data hold time	0	

9.6 Putting It All Together

- In a system with buffers and other logic, there is a gap between the time when signals are generated by CPU and when they arrive at the memory.
- For example:
  - The address delay, **t_AD**, is the delay from the start of the memory cycle to the time the address latch enable, **ALE**, strobes the address into the latch.
  - The actual time the memory receives the address will be delayed by the propagation time of the latch, **t_PL**, and address bus buffers, **t_PADR**.
  - Similarly, the time at which the CS signal is asserted will be delayed by the propagation time of the address decoder, **t_PDEC**.
  - In some systems, there may be bidirectional data bus buffers between the memory and the data bus. This causes a delay between the output of the data from the memory and when the data arrive at the CPU.

Figure 9-18 CPU - memory interface.



t_PL: propagation time for the address latch.

t_PADR: address buffer propagation time.

t_PDEC: address decoder propagation time.

t_PW: logic propagation time for a write control signal.

t_PR: logic propagation time for a read control signal.

A CPU with a multiplexed address bus and static memory

Principles of data exchange

- The CPU clock controls the overall timing
- CPU controls the information transfer in the system
- CPU Read
  - Memory comes in early
  - Memory stays late
    - To ensure that CPU gets the data
- CPU Write
  - Memory comes in early
  - Memory quickly gets its job done
  - CPU stays a little bit late
    - To ensure that memory gets the data

Figure 9-19 CPU reading data from memory.

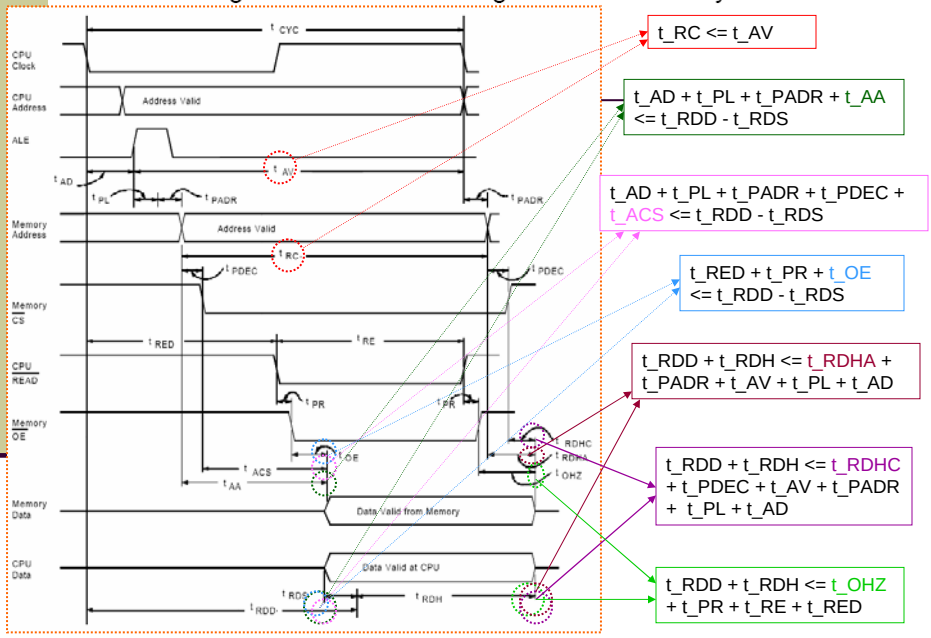
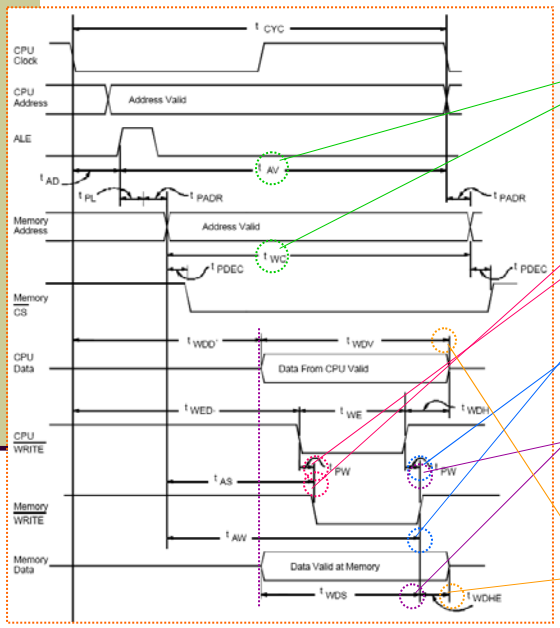


Figure 9-20 CPU writing data to memory.



t_WC <= t_AV

t_AD + t_PL + t_PADR + t_AS
<= t_PW + t_WED

t_AD + t_PL + t_PADR + t_AW
<= t_PW + t_WE + t_WED

t_WDD + t_WDS
<= t_PW + t_WE + t_WED

t_WDD + t_WDV
>= t_WED + t_WE + t_PW + t_WDHE

49

Example 9-1

- Using the information for a Motorola MC68HC11A given in Table 9-6 and the MCM 6064-12, 8K x 8 – bit SRAM given in Table 9-7, check to see if the 6064-12 can be used in a system with the MC68HC11A8. Assume the propagation delays are:
  t_PADR = 0 (no address buffer),
  t_PR = 10 ns, t_PW = 0, t_PL = 10 ns,
  and t_PDEC = 20 ns.
- Solution:
  - Check read cycle times:
    - All the inequalities are met!
  - Check write cycle times:
    - All the inequalities are met!

9.7 Chapter Summary

- RAM and ROM
- EPROM and NVRAM
- Memory Timing Requirements
  - CPU read and write
  - Memory read and write
- Memory System Design