EEE 2306 Digital Electronics II

Dr. Wawerũ Njeri

Dedan Kimathi University of Technology

Electrical & Electronic Engineering Department

EEE 2309 Digital Electronics II

Aim

The aim of this course is to enable the student to;

1. Understand applications of different electronic devices like BJTs and FETs.
2. Understand working principle of digital measuring instruments.

Learning outcome

At the end of this course, the student should be able to;

1). Design ROMs, RAM and PLAs

2). Use different digital measuring instruments

Course Content

1. Switching logic families; Bipolar and MOSFET,Transistor Logic (TTL), Emitter Coupled Logic (ECL), Current Injected Logic (CIL), PMOS and NMOS. Characteristics of TTL, ECL, CIL and PMOS technologies.
2. Operation and application of memory devices: Read Only Memory (ROM), Programmable ROM (PROM), Read Access Memory (RAM), Programmable Logic Array (PLA), Electrically Erasable ROM (EEROM)and PLA base design.
3. Design of Digital Systems: State diagrams design, state machine chart representation and implementation using Programmable Logic Devices (PLD).
4. Introduction to VHDL.

Teaching Methodology

2 hour lectures and 1 hour tutorial per week, and at least five 3-hour laboratory sessions per semester organized on a rotational basis.

Course Assessment

Continuous assessment and written University examinations shall contribute 30% and 70%, respectively of the total marks.

Course Books

1). Puri (1997), Digital Electronics: Circuits and Systems, Tata McGraw-Hill.

2). Roger L. Tokheim (2007), Digital Electronics: Principles and Applications, McGraw- Hill Education, 7th Ed.

3). Anil Kumar Maini (2007), Digital electronics: principles, devices and applications, McGraw-Hill, illustrated Ed.

Contents

1 LOGIC FAMILIES AND THEIR CHARACTERISTICS

1.1 Introduction

A logic family comprises a set of devices manufactured using a particular process. Each member of a particular logic family has the same basic characteristics that are unique to that logic family. The logic families include:

1. Direct Coupled Transistor Logic (DCTL) - the oldest
2. Resistor Transistor Logic (RTL)
3. Diode Transistor Logic (DTL)
4. Transistor Transistor Logic (TTL)
5. Emitter Coupled Logic (ECL)
6. Integrated Injection Logic (IIL or I2L)
7. P or N channel Metal Oxide Semiconductor Logic (PMOS or NMOS)
8. Complementary Metal Oxide Semiconductor Logic (CMOS).

All these logic families have different characteristics. The first six in the list use bipolar transistors (BJTs) while the other two use field effect transistors (FETs). The first three logic families are obsolete though you may come across DTL devices in some old electronic equipment. In this course, we are going to look at the most common logic families, the TTL and CMOS.

1.2 Characteristics

- **Propagation Delay (speed)**

  The speed of a logic family is measured by the propagation delay time of the basic inverter or the NAND gate. This is illustrated in Figure 1.1.

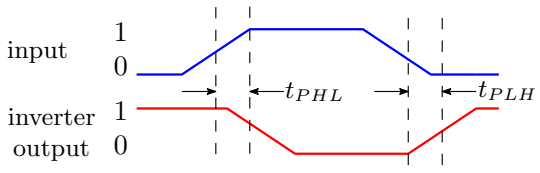

Figure 1.1: Propagation delay

t_{PHL} - propagation delay time for the HIGH to LOW transition.

t_{PLH} - propagation delay time for the LOW to HIGH transition.

The propagation delay time for a logic family is given as:

$$tP = \frac{t_{PHL} + t_{PLH}}{2}$$

- **Power requirements**

  Every gate requires a certain amount of power to operate, and this differs between the logic families. Generally, a high power consumption means higher operation costs. Battery operated appliances should be designed using logic families with low power consumption.

- **Fan-out (loading factor)**

  This is the maximum number of standard logic inputs that a logic output can reliably drive e.g. if the fan-out of a logic device is specified to be 10, then it can drive a maximum of 10 standard logic inputs. If this number is exceeded, the output logic-level voltages cannot be guaranteed. Buffers can be used to increase the fan-out of a gate. (The number of input terminals to a gate is referred to as the fan-in).

- **Noise immunity**

  Stray electrical and magnetic fields induce voltages on the connecting wires between logic circuits. These unwanted signals can sometimes cause the input voltages to change, and this could produce unpredictable outputs. Noise immunity of a logic family refers to the circuits' ability to tolerate noise voltages on its inputs. A quantitative measure of the noise immunity is called noise margin.

- Speed-Power product

  $$= \text{Propagation delay} \times \text{Power consumption}$$

1.3 Definitions

- V_{CC} - Supply voltage.
- V_{IH} - Input HIGH voltage - range of input voltages that represent logic 1 in the system.
- $V_{IH(min)}$ - Minimum input HIGH voltage - this is the minimum allowed input HIGH in a logic system.
- V_{IL} - Input LOW voltage - range of input voltages that represent a logic 0 in a system.

- $VIL(max)$ - Maximum input LOW voltage - The maximum allowed input LOW in a system.
- $V_{OH(min)}$ - Minimum output HIGH voltage - the minimum HIGH voltage at an output terminal in the HIGH state.
- $V_{OL(max)}$ - maximum voltage at an output in its LOW state.

All logic families are designed such that:

$$\begin{aligned} V_{OH(min)} &> V_{IH(min)} \\ V_{OL(max)} &< V_{IL(max)} \end{aligned}$$

This gives us the noise margin, which is computed as follows:

$$\begin{aligned} \text{Noise margin(HIGH state)}; V_{NH} &= V_{OH(min)} - V_{IH(min)} \\ \text{Noise margin(LOW state)}; V_{NL} &= V_{IL(max)} - V_{OL(max)} \end{aligned}$$

The smaller of the values between V_{NH} and V_{NL} is taken to be the noise margin for a logic family. The voltages defined above are illustrated in Fig. 1.2, using a NAND gate and an inverter.

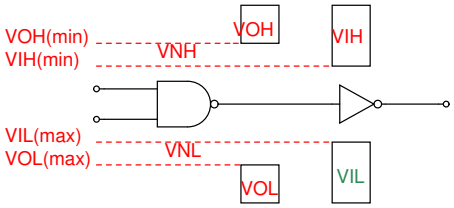

Figure 1.2

1.4 Current Sourcing and Current Sinking

Consider the circuit shown in Fig. 1.3, where the output of the driving gate is HIGH. The current flows in the direction shown by the arrow, so in this case, the output of the driving gate is acting as a current source for the load gate. This is known as **current sourcing**.

Now consider the circuit shown in Fig. 1.4, where the output of the driving gate is LOW. The current flows in the direction shown by the arrow, hence the output of the driving gate is acting as a current sink for the load gate. This is known as **current sinking**.

- I_{CC} - Supply current.

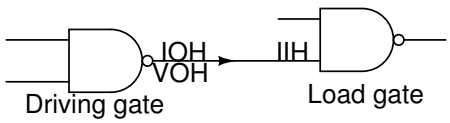

Figure 1.3: Current sourcing

Figure 1.4

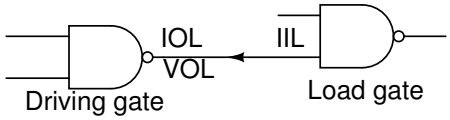

Figure 1.5: Current sinking

- I_{IH}- Input HIGH current - Current flowing into an input when a HIGH level voltage is applied.
- $I_{IL(min)}$ - Minimum LOW current - Current owing out of an input when a LOW voltage is applied.
- I_{OH} - Output HIGH current - Current flowing out of an output in the HIGH state under specified load conditions.
- $I_{OL(max)}$ - Output LOW current - Current flowing into an output which is in the LOW state under specified load conditions

1.5 DTL gates

NOTE: *[(a)]The basic circuit in each IC digital logic family is either a NAND gate or a NOR gate. This basic circuit is the primary building block from which all other more complex digital components are obtained. In all the logic gates, transistors are operated only in two modes: saturation and cut-off. That way, the transistor operates as a switch: open (cut-off), and closed (saturation).*

Figure 1.6 shows a Diode-Transistor Logic (DTL) NAND gate. Note that although this type of NAND gate is no longer manufactured, the understanding of how it works will assist in understanding the operation of TTL gates that we shall come across later.

When A or B or both are both LOW, D_1 or D_2 or both will get forward biased and will conduct, and the voltage V_X is approximately zero. This is insufficient to turn on D_3, D_4 and Q hence the transistor is cut-off. The output is therefore HIGH.

When $A = B =$ HIGH, both D_1 and D_2 are reverse-biased. Current flows from the supply (V_{CC})

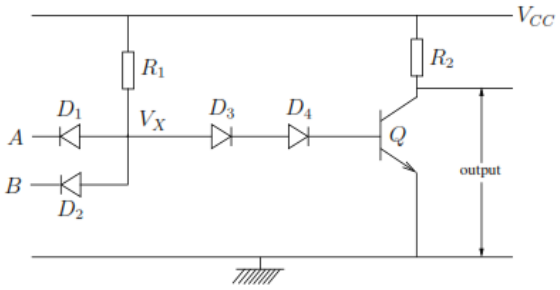

Figure 1.6: DTL NAND gate

through $R1$, D_3 and D_4, and forward-biases the base-emitter junction of the transistor Q. The transistor will get into the saturation mode and the output of the gate is LOW.

The truth-table for this gate is shown below:

A	B	output
0	0	1
0	1	1
1	0	1
1	1	0

This is the truth-table for a NAND gate. Assuming $V_{BE(sat)} = 0.7V$ and a diode drop = 0.7 V , then V_X should be at least 2.1V to turn Q on. When either A or B or both are zero, V_X = 0.7 V . This gives us a noise margin of $2.1V - 0.7V = 1.4V$. If we had used only one diode in place of D_3 and D_4, then V_X will need to be at least 1.4V to turn on the diode and transistor Q. The noise margin in this case would be $1.4V - 0.7V = 0.7V$. This means that using more diodes increases the noise margin. If both D_3 and D_4 were omitted, the voltage $V_X \geq 0.7V$ and hence transistor Q would be permanently on regardless of the inputs A and B (hence would not act as a NAND gate).

1.6 TTL gates

1.6.1 Introduction

Fig. 1.6 shows a TTL NAND gate. Q_1 is a multi-emitter transistor. The transistor is equivalent to the circuit shown in Fig. 1.7. (Note that the gate shown in Figure 1.7 is a 2-input NAND gate. A 3-input NAND gate would have a multi-emitter transistor with 3 emitters etc.)

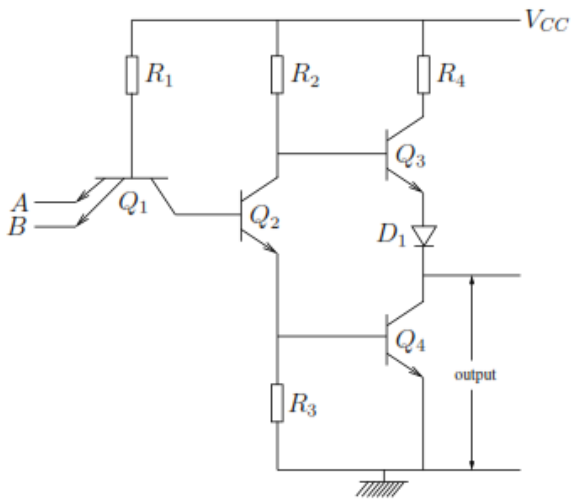

Figure 1.7: TTL NAND gate

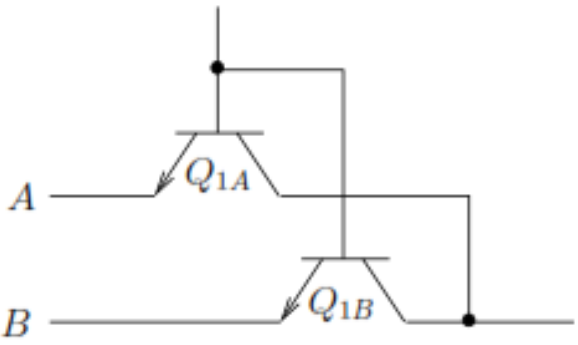

Figure 1.8

1.6.2 Operation of the TTL NAND gate

When both inputs are HIGH, the base-emitter junctions of the multi-emitter transistor are both reverse-biased, while the base-collector junctions are forward-biased. Q_1 is in cut-off mode. The V_{CC} supply pushes current through R_1 and the base-collector junction of Q_1, and this turns Q_2 on (saturation mode). Current from the emitter of Q_2 will ow into the base of Q_4 and this turns Q_4 on (saturation). At the same time, the flow of Q_2 collector current produces a voltage across resistor R_2 that reduces Q_2 0s collector voltage to a level that is insufficient to turn on Q_3 because of diode D_1. The diode D_1 is needed to keep Q_3 OFF under these circumstances. Since Q_3 is OFF, the current from the supply does not pass through Q_4. Transistor Q_4 0s collector current comes from the output that the gate is connected to (current sinking). The ow of current in this mode can therefore be drawn as shown by the

arrows in Figure 1.9.

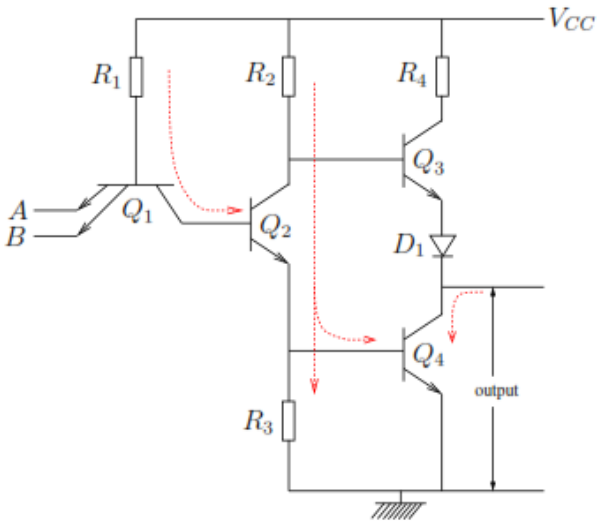

Figure 1.9

When one or both inputs are LOW, one or both of Q_1 0s base-emitter junctions will get forward-biased, hence Q_{1A} or Q_{1B} or both turn on so Q_1 is in saturation mode. In this case, the collector voltage of Q_1 is insufficient to turn on Q_2 and Q_4 so both of these transistors are in cut-off mode. Since Q_2 0s collector current is zero, the voltage at Q_3 0s base is sufficient to forward-bias Q_3 and D_1 so that Q_3 gets into saturation mode. The flow of current is shown by the arrows in Figure 1.10.

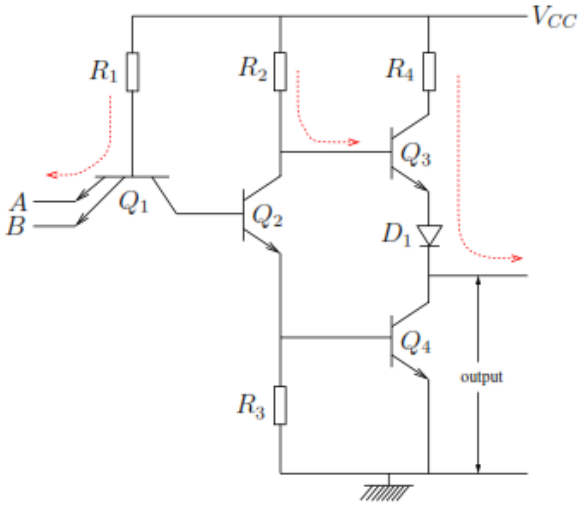

Figure 1.10

The operation of the gate can be summarized by the truth-table below;

A	B	output
0	0	1
0	1	1
1	0	1
1	1	0

Exercise:

Explain the working of the logic gate shown in Figure 1.11 and hence identify the logic gate.

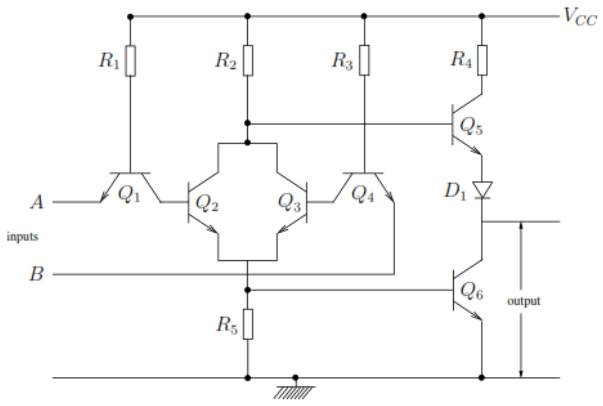

Figure 1.11

1.6.3 Current Sourcing and Current Sinking-Revisited

The idea of current sourcing and current sinking can now be more appropriately illustrated using the NAND gate circuit. The case when the output of the driving gate is LOW is shown in Figure **??**. (This is equivalent to the situation illustrated in Figure 1.5.)

Figure 1.11 From the Fig. 1.11, it can be observed that Q4 of the driving gate acts as a current sink, deriving its current from the load gate (current sinking). Now consider the situation when the output of the driving gate is HIGH. This is illustrated in Fig. 1.12 (equivalent to Fig. 1.3). Figure 1.12 This time, transistor Q3 of the driving gate acts as a current source for the load gate (current sourcing). Note that in the case of current sourcing, the current is driven into a reverse-biased p-n junction in the load gate so this current (IIH) is very small, in the order of μA.

1.6.4 TTL Loading (Fan-Out)

Consider the case where the output of the driving gate is HIGH (current sourcing). The circuit shown in Fig. 1.13 is used to determine the fan-out. Figure 1.13: Fan out (HIGH) Assuming that the driving gate is connected to n load gates, then; IOH = nIIH If too many gates are connected, IOH will cause a higher voltage drop on R4 (Fig. 1.9), and this reduces VOH. This is undesirable as it reduces the noise margin and it could make VOH get into the indeterminate range. fan out (HIGH) = IOH(max) IIH(max) In the case where the output of the driving gate is LOW, Fig. 1.14 shows the diagram used to compute the fan-out. Figure 1.14: Fan out (LOW) 15 For n gates connected to the load gate; IOL = nIIL If too many gates are connected, IOL will increase and cause VOL to increase (can you explain this?). This reduces the noise margin and VOH could get to the indeterminate range. fan out (LOW) = IOL(max) IIL(max) The fan out of a logic family is the lower of the two values between, (fan-out (HIGH) and fan-out (LOW)).

Example 1.1:

The data shown in table 2.1 below applies to logic families A and B. Use the table to determine;

a). The DC noise margins for logic families A and B.

b). The number of logic family B gates that can be satisfactorily driven by a logic family A gate.

Note: Typ stands for Typical value.

Table 1.1: Table for Example 1.1 Logic Family parameter Min A Typ Max Min B Typ Max Units VIH Input Voltage (HIGH) 2 - - 2 - - Volts VIL Input Voltage (LOW) - - 0.8 - - 0.8 Volts VOH Output Voltage (HIGH) 2.4 3.4 - 2.7 3.4 - Volts VOL Output Voltage (LOW) - 0.2 0.4 - 0.35 0.5 Volts IIH Input Current (HIGH) - - 40 - - 20 Microamps IIL Input Current (LOW) - - -1.6 - - -0.95 Milliamps IOH Output Current (HIGH) - - 400 - - 400 Microamps IOL Output Current (LOW) - - -16 - - -8 Milliamps

Solution:

a).

$$\begin{aligned} V_{NH} &= V_{OH(min)} - V_{IH(min)} \\ V_{NL} &= V_{IL(max)} - V_{OL(max)} \end{aligned}$$

For logic family A;

$$\begin{aligned} V_{NH} &= 2.4V - 2V = 0.4V \\ V_{NL} &= 0.8V - 0.4V = 0.4V \end{aligned}$$

The two values are the same so noise margin for logic family A = 0.4V: For logic family B;

$$V_{NH} = 2.7V - 2V = 0.7V$$
$$V_{NL} = 0.8V - 0.5V = 0.3V$$

The smaller of the two values is 0.3V, which is the noise margin for logic family B.

b). In this case, the logic family A gate is going to be the driving gate and logic family B gates the load gates.

In the HIGH state, the number of B gates that can be driven by A are given by;

$$\frac{I_{OH(max)} \text{ for logic family A}}{I_{IH(max)} \text{ for logic family B}} = \frac{400\mu A}{20\mu A} = 20$$

In the LOW state, the number of B gates that can be driven by A are given by;

$$\frac{I_{OL(max)} \text{ for logic family A}}{I_{IL(max)} \text{ for logic family B}} = \frac{-16mA}{-0.95mA} = 16.8 \simeq 16$$

The number of gates of logic family B that can be satisfactorily driven by logic family A gate is the smaller of the two numbers = 16.

1.6.5 Totem Pole Outputs

The output stage of the NAND gate described above is called a totem-pole output. It is illustrated in Fig. 1.15. All TTL gates have this type of output stage. Note that the NAND gate circuit shown in Fig. 1.6 would still operate as a NAND gate even if D1 and Q3 were removed (and R4 connected directly to Q4). However, removing D1 and Q4 is not desirable as R4 is usually a small resistor (about 100) so a large current would flow through Q4 when it is in saturation mode. When Q3 is in the circuit, and Q4 is in saturation mode, Q3 is in cut-off mode hence there will be no current through R4, and this keeps the power dissipation of the circuit low. Note also that the transistors Q3 and Q4 operate such that when Q4 is saturated, Q3 is cut-off, and when Q4 is cut-off, Q3 is saturated. 17 Figure 1.15: Totem pole output Totem pole outputs should not be connected directly together (connecting two outputs directly together is known as a wired-AND operation). Consider the case shown in Fig. 1.16 where two totem-pole outputs are connected together directly. Figure 1.16: Wired-AND connection No particular problems arise if both inputs are HIGH or are both LOW. A problem arises if one input is HIGH and the other one LOW. In the case shown in Fig. 1.16, it is assumed that the output of gate A is HIGH while that of gate B is LOW. A large current flows in the direction shown by the arrow. This current is usually

large enough to damage transistors Q3A and Q4B, thereby destroying the gates. Even if by some chance the gates did not get damaged, the high current drawn will lead to a very high power consumption (and make the ICs get very hot), and the voltage at the point where the two outputs are connected together will get into the indeterminate region.

1.6.6 TTL Open-Collector Outputs

Figure 1.17: TTL open-collector NAND gate When the output is LOW, the transistor Q4 is in saturation mode hence Vout = VCE(sat) ' 0. When the output is HIGH, Q4 is cut-off. For a HIGH level voltage to appear, an external resistor RP (known as a pull-up resistor) is connected to the collector of Q4 as shown in 1.17. Without RP , there would be no output voltage when Q4 is cut-off. Open-collector outputs allow us to perform the wired-AND operation, which is not possible with totem-pole outputs. As an example, if we were to implement the function;

$$x = \overline{AB} \cdot \overline{CD} \cdot \overline{EF}$$

Using open-collector TTL gates, the circuit would look as shown in Fig. 1.18. Figure 1.18: Wired-AND operation using open-collector gates The AND gate shown in dotted in Fig. 1.18 is just symbolic-it means that connecting the three (open-collector) outputs together is equivalent to an AND gate. The pull-up resistor RP is necessary to observe the output x. The open-collector gate circuit design therefore generally results in the use of fewer gates than totem-pole gates design, as we can directly connect open-collector outputs together, which is not possible with TTL outputs. The open-collector gates are also designed to handle a relatively large current and are used for driving devices like LEDs and relays.

1.6.7 Tristate Devices

Tristate devices have outputs that can be get to three possible states: HIGH, LOW and High Impedance (Hi-Z). An example of a tristate TTL inverter is shown in Fig. 1.19. Figure 1.19: Tristate TTL inverter When E = 1, the circuit operates as an inverter since E has no effect on Q1 or Q3 . We say that the gate is enabled when E = 1. When E = 0, Q1 is in saturation mode, hence Q2 and Q4 are cut-off. At the same time, Q3 is also in cut-off mode. Since both transistors on the totem-pole output are in cut-off mode, the output is essentially an open circuit. We say that when E = 0, the gate is disabled, and the output is in the High Impedance state, hence the input to the gate has no effect on the output. These situations are illustrated in Fig. 1.20. Figure 1.20: Tristate inverter 20 Note from Fig. 1.19 that when E = 0, the output of the gate is in the Hi-Z state, which is equivalent to having the output being

disconnected from the input. Tristate ability allows many devices to be connected on the same output line, with one device enabled at a time.

1.6.8 TTL 74 Series

a. 74XX e.g. 7400, 7432 e.t.c.-This is the standard TTL. In this case (Fig. 1.6, R1 = 4K;R2 = 1K6;R3 = 1K and R4 = 130.

b. 74LXX e.g. 74L00, 74L32 e.t.c.- Low power TTL - Uses resistors of higher values than standard TTL. R1 = 40K;R2 = 20K;R3 = 12K and R4 = 5K.

c. 74HXX e.g. 74H00, 74H32 e.t.c. - High speed TTL - Uses resistors of lower values than standard TTL, has a higher power consumption than standard TTL. R1 = 2K;R2 = 760 ;R3 = 470 and R4 = 58.

d. 74SXX e.g. 74S00, 74S32 e.t.c.-Uses Schottky transistors for high-speed switching-can work up to speeds of 100 MHz.

e. 74LSXX e.g. 74LS00, 74LS32 e.t.c.-Low power Schottky - low power version of 74SXX series.

f. 74ALSXX e.g. 74ALS00, 74ALS32 e.t.c.- Advanced Low Power Schottky - has a lower power consumption than 74LSXX series and can operate at higher speeds than 74SXX series.

1.7 CMOS

1.7.1 Comparison between TTL and CMOS

a. Power Consumption: In TTL devices, power is consumed at constant rate. CMOS devices have a low power consumption since they only consume a significant amount of power when switching (HIGH to LOW or LOW to HIGH) - very little power is consumed when the gate is in a stable HIGH or stable LOW state. (Note that among all logic families, ECL has the highest power consumption.)

b. **Noise Margin:** With a supply voltage of VCC = 5 V, TTL devices have a noise margin of about 0.8 V. CMOS devices have a noise immunity ' 1 3VCC, which is generally higher than that of TTL devices. (ECL has the lowest noise margin among the logic families.)

c. **Power Supply Voltage:** TTL devices should be powered with a supply voltage of 5V±5%. The 4XXX and 74C series of CMOS devices can be powered with voltages between 3V and 15 V, while

the 74HC and the 74HCT CMOS series can operate with power supply voltages of 2V to 6 V. CMOS devices are therefore less sensitive to power supply voltages.

d. **Speed:** TTL devices have higher speed than CMOS devices. TTL devices have an av- erage propagation delay of about 10 ns, compared to about 50 ns for the 4XXX CMOS series. (ECL has the highest speed among the logic families - used where speed is the main consideration.)

e. **Fan-Out:** CMOS devices have a higher fan-out than TTL devices.

Exercise:

Fig. 1.21 shows a CMOS inverter. Explain its operation. Figure 1.21: CMOS inverter

1.7.2 CMOS Handling Precautions

CMOS devices are easily damaged by static charges, making them electrically delicate and hence the following precautions should be taken when working with these devices;

i. Do not handle CMOS ICs unless your body is connected with a wrist strap to earth ground.

ii. Tools used to straighten pins should be grounded.

iii. Do not remove or replace a CMOS IC in a socket with power ON.

iv. All unused CMOS gate inputs must be connected to an active gate input, VCC or ground. If allowed to float, an input can take on enough static charges to damage the IC.

1.7.3 CMOS Logic Sub-families

a. CD 4XXX e.g. CD 4001 Quad 2-input NOR gate IC - this is the oldest and it is no longer manufactured. It is also the slowest among the CMOS logic families.

b. 74CXX - Standard CMOS - these are pin-compatible with TTL devices (e.g. the CMOS 74C08 and the TTL 7408 are both AND gate ICs and have the same pin configuration. However, they are not electrically compatible.

c. 74HCXX - High Speed CMOS - has a speed comparable to the TTL 74LSXX series. This is also pin compatible but not electrically compatible with TTL devices.

d. 74HCTXX - High Speed, TTL compatible CMOS - This is pin comparable and electrically compatible with TTL devices. This means that you can construct a circuit having TTL and 74HCT ICs.

2 SEMICONDUCTOR MEMORY DEVICES

2.1 Introduction

Memory devices are used in digital systems to store digital information (1's and 0's). The ability of digital systems to store large quantities of information is one major advantage over analog systems. It makes digital systems very versatile and adaptable to many situations e.g. in a digital computer, the memory stores instructions to tell the computer what to do so that the computer can carry out a large variety of tasks with a minimum amount of human intervention.

The digital information stored is usually;

i). **Data** - data to be operated on, intermediate and final results.

ii). **Instructions** - An instruction tells the digital system (computer) to perform a specific operation, such as adding two numbers together, comparing one number to another number, or moving a number to a specific memory location.(A series of instructions is usually referred to as a program).

As an example, in an expression like $3 + 4$, 3 and 4 is data while $+$ is the instruction. (The data and instruction in this example would have to be binary-coded first before being stored in a memory system).

There are many different types of semiconductor memory devices and they are suited for different applications.

2.2 Memory Terminology

Memory Cell: A device or an electrical circuit used to store a single bit (0 or 1) e.g. a flip-flop. A capacitor can also be used to store 1 bit of information (presence of charge representing one logic level, absence of charge signifying the other logic level).

Memory Word: A group of 1's and 0's which may represent a number, an instruction, one or more alphanumeric characters, or any other binary coded information.

Byte: Special term for an 8-bit word.

Capacity: A way of specifying how many bits can be stored in a particular memory device or a complete memory system. E.g. suppose a certain memory device can store 4096 20-bit words. Capacity $= 4096 \times 20 (= 81920)$ bits. The first number represents the number of words which can be stored in

a particular memory device, while the second number represents number of bits per word. In memory devices, 210 = 1024 = 1K(1 Kilo), so we can write that the capacity of the memory device above as $4K \times 20$. Further, 2^{20} = 1M(1 Mega), 2^{30} = 1G (1 Giga), 2^{40} = 1T (1 Tera) e.t.c.)

Address: A number that identifies the location of a word in memory. Each word stored in a memory device has a unique address.

Read Operation: Operation whereby a binary word is stored in a specific memory is sensed and then transferred to another device. Also known as the fetch operation.

Write Operation: The operation whereby a new word is placed into a particular memory location. Also known as the store operation. Whenever a new word is written into a particular memory location, it replaces the word previously stored there.

Access Time: Time required to perform a read operation. It is a measure of speed for memory devices - the higher the access time, the slower the memory and viceversa. A more specific de

nition of access time will be given when we cover Read Cycle Timing.

Volatile Memory: A type of memory that requires the application of electrical power in order to store information. If electrical power is removed, all the information stored in the memory will be lost. An example of a volatile memory device is a flip-flop: if the power supply to a flip-flop is interrupted, the data stored will be lost.

Random Access Memory (RAM): A type of memory where the access time is the same for all locations i.e. access time does not depend on the address of the word. Example: Semi-conductor memories.

Sequential Address Memory: A type of memory where the access time varies with the location of the word. A particular word is found by sequencing through all address locations until the desired address is reached. Sequential Access Memories generally have larger access than RAMs. Example of a Sequential Access Memory device: Magnetic tape.

Read Write Memory (RWM): Any type of memory device that can be written into or read from with equal ease.

Read Only Memory (ROM): A type of memory where the ratio of READ to WRITE operations is very high. Some ROMs can be written into only once after which information can only be read from them. Other types of ROM can be written into more than once but the WRITE operation is more complicated than the READ operation and is not performed very often.

Static Memory Devices: Semiconductor memory devices in which data will remain permanently stored as long as power is supplied, without the need for periodically rewriting the data into memory e.g. flip-flops.

Dynamic Memory Devices: Semiconductor memory devices in which the stored data will not remain permanently stored, even with power applied, unless data is periodically rewritten into memory. The rewriting of data is called a refresh operation.

2.3 Semiconductor Memory Technologies

On the basis of their manufacturing technologies, semiconductor memory devices are divided into two categories: bipolar memories (made using bipolar transistors) and MOS memories (made using MOSFETS). The two categories may be subdivided as shown in Fig. 2.1. Figure 2.1: Semiconductor Memory Technologies The memory devices manufactured using various technologies are usually compared according to the following criteria;

i). Density - Capacity in bits per chip.

ii). Speed - usually specified using access time - the smaller the access time, the faster the memory device.

iii). Power requirements (μW or mW per bit) - Power required to store data/operate.

iv). Cost of fabrication per bit.

v). Noise immunity - Ability of a memory device to tolerate noise voltages at its inputs.

The technologies shown in Fig. 2.1 compare as shown in table 2.1 below:

i). TTL memories are the most commonly used. They are suitable for applications that demand relatively high speed and medium capacity and where power consumption is not a critical factor.

ii). ECL memories are used only where very high operating speed is required without regard to cost and power consumption.

Parameter	Speed	Power Consumption	Capacity	Noise Immunity	Cost per Bit
TTL/S-TTL	Fast	High	Low	Low	High
ECL	Fastest	Highest	Lowest	Lowest	Highest
PMOS/NMOS	Medium	Low	Highest	High	Lowest
CMOS	Slowest	Lowest	High	Highest	Low

iii). CMOS memories are best suited for applications in high noise environments or where power consumption is extremely critical e.g. in battery operated systems.

2.4 Semiconductor Memory Operation

A semiconductor device can be represented as shown in Fig. 2.2. Figure 2.2: Semiconductor Memory - General case

Address Inputs: The address of the location being read from or being written into is entered through this inputs. N address inputs address 2N memory locations. (For the memory device shown, the number of locations that can be addressed is $2^6 = 64$).

Data inputs: Used to enter data during a WRITE (or STORE) operation. Data outputs: Data appears on this lines during a READ operation. The data output lines are usually tristate i.e. have the states HIGH, LOW and High-Impedance (Hi-Z) states. This enables many memory devices to share a common data bus.

Memory Enable input: (Usually abbreviated ME) Used to enable or disable the memory chip. If the ME input is not at the correct logic level, the memory chip will not respond to address and READ/WRITE inputs and the data output lines will be in the Hi-Z state. Note that the memory enable input comes under several names e.g. CHIP ENABLE (CE) or CHIP SELECT (CS). Note also that some chips may have more than one enable input, and that some enable inputs may be active-LOW or active-HIGH.

READ/WRITE input: (Usually abbreviated R/W. Determines whether a READ or WRITE operation will take place.

$$\text{R/W} = 1 \rightarrow \text{READ OPERATION}$$

$$\text{R/W} = 1 \rightarrow \text{WRITE OPERATION}$$

Note that some of the older semiconductor devices had two separate inputs for READ and WRITE operations.

Note: Modern semiconductor memories use the same set of data pins for writing and reading the device. This is done to save the number of pins in the IC package. When R/W = 1 (READ OPERATION), the data pins act as outputs, while R/W = 0 (WRITE OPERATION) sets the pins to act as inputs. Generally speaking, semiconductor memory devices operate as follows;

WRITE operation:

a). Using the address input pins, apply the address of the location to be written into.

b). Apply the data to be written on the data input lines.

c). Enable the memory chip by setting the Memory Enable Input(s) to the appropriate logic levels.

d). Set the R = W input to LOW for a WRITE operation.

READ operation:

a). Using the address input pins, apply the address of the location to be read from.

b). Enable the memory chip by setting the Memory Enable Input(s) to the appropriate logic levels.

c). Set the R = W input to HIGH for a READ operation.

d). Data is read from the data output lines

2.5 Read-Only Memories (ROMs)

ROMs are used to store data that is not to change during the operation of a system. One use of such memories is in the start-up programs of digital computers, printers, electronic photocopiers etc. A ROM IC does not have a R = W input since the ROM cannot be written into under normal operating conditions.

2.5.1 ROM Timing

A ROM READ cycle is shown in Fig. 2.3. Figure 2.3: ROM READ operation timing diagram

i). t_0 : New address is applied to the ROM. The ROM circuitry begins to decode the address inputs to select the register which is to send data to the outputs.

ii). t_1 : The Chip select input CS is activated to enable the output buffers.

iii). t_2 : Data outputs change from high impedance state to the valid data stored at the specified address.

iv). t_{OE} : Output Enable Time - Delay time between the chip being enabled and valid data appearing at the data outputs.

v). t_{ACC} : Access Time - Time interval between the address being valid and the data outputs being valid.

Exercise:

A ROM has the following timing parameters: tACC = 250ns, tOE = 120 ns. The ROM has an active-LOW chip select input CS. Assume that a new address is applied to the ROM x nanoseconds before CS is driven LOW. Determine the minimum duration for which CS must be kept LOW for a reliable READ operation if; (i) x = 50 ns, (ii) x = 100 ns, iii) x = 500 ns **Ans:** (200ns, 150ns, 120ns).

2.5.2 Mask-Programmed ROM (MROM)

This is programmed by the manufacturer according to the customer's speci

cations (usually in the form of a truth table). A photographic negative called a mask is used to control the electrical interconnections in the chip to produce 0's and 1's to satisfy the given truth table. Since the masks are expensive, this type of ROM is only economical if a large quantity of the same ROM is needed.

The main disadvantage of this ROM is that it cannot be reprogrammed. A new ROM would have to be programmed if the original program needs a modi

cation. This type of ROM is generally used for data that does not change e.g. math tables, and character generator for CRT displays.

2.5.3 (User) Programmable ROM (PROM)

As the name suggests, this type of ROM is user programmable. The memory consists of an array of cells such as the one shown in Fig. 2.4. Figure 2.4: PROM Cell

Each of the connections between the SELECT inputs and the transistor bases is made with a thin fuselink that comes intact from the manufacturer. By applying a certain voltage, the user can selectively blow any of the fuselinks to produce the desired truth table. Once a fuselink is broken, it cannot be reprogrammed. Special kits are available for programming PROMs.

To read a cell, the SELECT input is set HIGH. If the fuselink is intact, output voltage = 0, hence the cell is storing a 0. If the fuselink has been broken, the transistor will be in cut-off mode hence output voltage will be approximately equal to VCC, meaning the cell is storing a 1.

2.5.4 Erasable Programmable ROM (EPROM)

An EPROM is user programmable, and it can also be erased and reprogrammed as often as desired. To see how an EPROM cell works, let us

rst look at the n-channel enhancement type MOSFET shown in Fig. 2.5. Figure 2.5: MOSFET The drain is biased positive with respect to the source. No conduction occurs if no voltage is applied to the gate.

By applying a voltage such that the gate is positive with respect to the substrate, electrons in the substrate are attracted towards the gate and the substrate near the gate effectively changes from p-type to n-type. A low impedance channel is thus formed between the gate and the source and the drain and current flows from the drain to the source. (The minimum voltage which causes a drain-source current ow is known as the *threshold voltage*).

Fig. 2.6 shows the EPROM cell. Figure 2.6: EPROM Cell The storage cell in an EPROM is a field-effect transistor with a silicon gate that has no electrical connections i.e. a floating gate. In the unprogrammed state, the memory cell has no charge stored in the floating gate and this state represents one logic state (usually logic 1 state). In the programmed state, the cell has charge stored in the floating gate and this represents the other logic state (logic 0).

Programming a '0' in the cell involves the application of about +20V between the drain and the source (the drain +) and about +25V on the select gate. The drain source potential causes electrons to flow from the source to the drain. Due to the high select gate voltage, some of the electrons crossing between the source and the drain acquire suffcient energy to pass through the silicon dioxide insulation to get to the floating gate. When the programming voltages are removed, the charge in the floating gate remains trapped since silicon dioxide is a good insulator. The charge can remain trapped for 10-20 years.

For an EPROM to be programmed, it has to be removed from the circuit board and placed in an

EPROM programming kit. The chip's address and data pins are used to determine which memory cells will be affected by the programming voltages. To read an EPROM cell, the select gate is set to a voltage slightly higher than the threshold voltage for the MOSFET. In the unprogrammed state (no charge stored in the floating gate), the cell will conduct like a MOSFET. When charge is stored in the floating gate, the threshold voltage for the MOSFET increases and the transistor will not conduct when a voltage slightly grater than the threshold voltage is applied to the select gate. Once a memory cell has been programmed, it can be erased only by exposing it to ultra-violet (UV) light. For this reason, EPROM ICs have a quartz window through which UV light can be applied. The UV light causes a flow of photocurrent from the oating gate to the p-substrate, thereby restoring the gate to its unprogrammed status. There is no way to expose just a single cell to the UV light without exposing all the cells. The UV light will therefore erase the entire memory. The erasure process requires 15-30 minutes of exposure to the UV rays. (For this reason, the EPROM is sometimes referred to as UV-EPROM).

Example 2.1:

The 2716 $2K\times8$ EPROM Figure 2.7: 2716 EPROM During normal operation, both VCC and VPP are connected to a $+$5V supply. CE and OE have to be LOW in order for the internal circuitry to select the register which is being addressed and to route the data from that register to the outputs. Programming the 2716 EPROM;

a). Connect both VCC and OE to a $+$5V supply, and V_{PP} to a $+$25V supply.

b). Apply the desired address to the address inputs.

c). Apply the desired 8-bit data word to the data input pins $D_0 - D_7$. Since the OE input is HIGH, these data pins function as inputs.

d). Apply a 50ms LOW to HIGH pulse at CE. At the termination of the pulse, the selected address location should be storing the applied data word.

e). Disconnect the data inputs from the data output pins.

f). To verify if data has been stored correctly, the address location should be read. This is done by connecting V_{CC} and V_{PP} to $+$5V supply, applying the address of the location to be read, setting OE and CE LOW and then reading the data appearing at the data output pins.

Other examples of EPROM ICs are the $4K \times 82732$, $8K \times 82764$ and the $32K \times 827256$.

2.5.5 Electrically Erasable Programmable ROM (EEPROM or E2PROM)

(Some books refer to this as Electrically Alterable Programmable ROM). Some disadvantages of the UV-EPROM are;

i). The device must be removed from the circuit to be programmed and to be erased.

ii). Erasure removes the entire memory contents.

iii). The quartz window package is expensive.

These problems were solved by the development of the EEPROM in the early 1980s. Figure 2.8: EEPROM Cell The EEPROM cell is shown in Fig. 2.8. The EEPROM is a modi

ed version of the UV-EPROM to allow electrons to tunnel through the silicon dioxide insulation in both directions depending on the applied voltages. By applying about +21V between the cell's select gate and the drain, charge can be induced in the oating gate, where it will remain even when power is removed. Reversal of the same voltage will cause the removal of the trapped charges from the oating gate and this erases the cell. This process requires very low currents so the EPROM can be programmed in-circuit (without removing it from the circuit where it is being used).

Advantages of EEPROM over UV-EPROM include;

i). With EEPROMs, it is possible to delete individual words in the memory array, which is not possible with UV-EPROMs.

ii). The EEPROM can be programmed in-circuit.

iii). The complete EEPROM can be erased in 10ms compared to about 30 minutes required for the UV-EPROM.

2.5.6 ROM Applications

a). Microcomputer Program Storage:

Microcomputers use ROMs to store start-up programs. These microcomputer programs that are stored in ROM are called firmware because they are not subject to change. Products that include a microprocessor to control their operation use ROMs to store their control programs e.g. electronic cash registers, electronic photocopiers, printers, electronic games e.t.c.

b). Storage of data tables:

ROMs are used to store tables of data that does not change e.g. trigonometric tables.

A	B	C	f_1	f_2	f_3	f_4
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	0	1	1	0
0	1	1	0	1	1	1
1	0	0	0	0	1	0
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	1

c). Character generators:

Alphanumeric characters displayed on cathode ray tubes are made up of dots. Each character is made to fit into a pattern of dot positions, usually arranged as a 5×7 or a 7×9 matrix. To display a character, some dot positions are made bright and others dark. A character generator ROM stores the dot pattern for each character.

d). Implementing combinational logic functions:

When used to implement combinational logic functions, the ROM acts like a combinational circuit that has a number inputs equal to its address inputs and the number of outputs equal to its data lines. The ROM is programmed so that each data output represents a specific function of the inputs. As an example, suppose we would like to implement the following functions using a ROM:

$$\begin{aligned} f_1(A,B,C) &= \overline{A}\overline{B}\overline{C} + AB \\ f_2(A,B,C) &= AB\overline{C} + AC + B \\ f_3(A,B,C) &= A\overline{B}\overline{C} + ABC + \overline{A}B \\ f_4(A,B,C) &= C \end{aligned}$$

First, these Boolean expressions have to be converted to the truth-table format, and the truth-table is given below: There are three variables A, B and C and 4 outputs f_1, f_2, f_3 and f_4. To implement the truth-table, we therefore need an 8×4 ROM IC (which has 3 address lines). The variables A, B and C are connected to the address lines. The address 000 is then programmed to store 1000, address 001 to store 0001 e.t.c. You can think of this process as the ROM being made to memorize the truth-table.

A	B	C	f1	f2
0	0	0	1	1
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	1	0
1	1	1	0	1

2.6 Programmable Logic Devices (PLDs)

A PLD is an integrated circuit (IC) with internal logic gates that are connected through electronic fuses. Programming the device involves blowing of the fuses along the paths that must be disconnected so as to obtain a particular function. Like a ROM, once a PLD has been programmed for a particular purpose, it cannot be erased and reprogrammed. An example of a PLD is a Programmable Logic Array (PLA).

A PLA is a logic circuit made up of AND and OR gates which can be interconnected to generate one or more outputs that are sum of product functions of several inputs. By selectively blowing the fusible links that interconnect the logic inputs to the AND gates, and the And gates to the OR gates, any desired sum-of-products output function can be generated. In block diagram form, a PLA looks as shown in Fig. 2.9. Figure 2.9: Block diagram representation of a PLA

The symbols shown in Fig. 2.10 are used in PLAs. Figure 2.10: Block diagram representation of a PLA

An example of a 3-input, 2-output PLA is shown in Fig. 2.11.

Figure 2.11: 3-input 2-output PLA **Example 2.2:**

Use the PLA shown in Fig. 2.11 to implement the truth-table shown below: **Solution:**

From the given truth-table, we can write that:

$$f1 = \overline{ABC} + \overline{A}BC + AB\overline{C}$$

$$f2 = \overline{BC} + ABC$$

These functions can be implemented as shown in Fig. 2.12

Figure 2.12: Example: using a PLA used to implement a combinational logic circuit A shorthand way of drawing Fig. 2.12 is shown in Fig. 2.13. Figure 2.13: Shorthand notation of Fig. 2.12 In the

implementation of large combinational logic circuits, PLDs are preferred to SSI or MSI devices for the following reasons;

i). PLDs use less circuit board area since one PLD package is equivalent to several packages of SSI/MSI devices.

ii). PLDs shorten design time since less packages are used.

iii). Design changes can be done by reprogramming new PLDs, which is less time consuming than redesigning a circuit board made up of SSI/MSI devices.

iv). Since PLD-based circuits use fewer ICs, there are fewer interconnections to be made and this makes PLD-based circuits more reliable than SSI/MSI circuits.

2.7 Semiconductor RAMs

This refers to random-access read-write memory devices. RAMs are used in microcomputers for the temporary storage of programs and data. RAMs are volatile and will lose all information stored if power is turned off.

There are two categories of RAM: static RAM (SRAM) and dynamic RAM (DRAM). STATIC RAM is made up of memory cells which are capable of retaining stored information indefinitely, as long as power supply is maintained. DYNAMIC RAM memory is only capable retaining the stored information for a few milliseconds (typically 2ms). The dynamic memory cell is a capacitor which stores charge, and this charge inevitably leaks with time. The DRAM must therefore be refreshed periodically by recharging the capacitors in order to retain the information.

2.7.1 Static RAM Architecture

This is illustrated using a 64×6 RAM shown in Fig. 2.14. Figure 2.14: SRAM Architecture

READ Operation: The address code selects one register for READ or WRITE operation. For READ operation, the R/W input and the CS input must be HIGH. (Chip Select input is active-HIGH in this example). This combination enables the output buffers so that the contents of the selected register will appear at the data outputs. Setting the R/W signal HIGH disables the input buffers so that the data inputs do not affect the memory during a READ operation.

WRITE Operation: R/W = 0 and CS = 1. This combination enables the input buffers so that the 6-bit word applied to the data inputs will be loaded into the selected register. Setting R=W LOW disables the output buffers so that the data outputs are in their High-Impedance State during a write operation.

Note:Since the READ and WRITE operations are not carried out simultaneously, most RAM ICs use the same data pins for data input and output (such pins are known as data input/output lines, abbreviated data I/O lines). The R/W input controls the functions of these pins i.e. R/W = 0, data I/O lines act as inputs and for R/W = 1, data I/O lines act as outputs. This helps to reduce the number of pins on an IC package. For larger RAMS, the registers are arranged in a matrix similar to that of DRAM architecture (Figure 7.20).

2.7.2 Static RAM memory Cell

Consider the circuit shown on Fig. 2.15 where Q_1, Q_2, Q_3 and Q_4 are field-effect transistors. (In the figure S = Source, D = Drain and G = Gate). Figure 2.15: Basic SRAM cell Q_3 and Q_4 are connected such that they act as high-value resistors (typically 1012 ohms - this type of connection is preferred to using resistors as it takes much less space than resistors of the same value, and the very high resistance reduces power consumption). Transistors Q_1 and Q_2 are cross-coupled, and act like a flip-flop i.e. suppose Q1 is fully conducting. Then its drain falls to nearly 0V and this causes Q_2 to become non-conducting and its drain rises to the supply voltage. This voltage is applied to the gate of Q_1 and since it is above the threshold voltage of Q_1, Q_1 is maintained in the conducting state. The circuit can be maintained in one of two states: Q_1 fully conducting and Q_2 off, or Q_1 off and Q_2 fully conducting. The complete memory cell is shown on Fig. 2.16. 39 Figure 2.16: Complete SRAM cell Data is written into the cell by enabling Q5 and Q6 (by setting ROW SELECT line HIGH), applying the desired data to the COLUMN SELECT line and the complement of the data on the $\overline{COLUMN}SELECT$ line: Data is read by enabling Q_5 and Q_6 and reading the COLUMN SELECT line.

2.7.3 Static RAM Timing

READ Cycle Figure 2.17: SRAM READ cycle

a). t_0: Address inputs change to a new address from which data is to be read. This is the beginning of a READ cycle.

b). $t1$: Chip Select input activated

c). $t2$: Data outputs change from High Impedance State to valid data outputs.

d). $t3$: Chip Select input is de-activated.

e). $t4$: Data outputs change from valid data outputs to the High Impedance State in response to the de-activation of the Chip Select signal. Any device that needs to read data from the memory device should do so between t_2 and t_4.

f). $t5$: End of READ cycle. Address inputs change to a different address for a another READ or WRITE cycle.

g). tCO : Minimum time taken for memory outputs to go from High Impedance state to valid data outputs after Chip Select input has been activated.

h). tOD : Minimum time taken for memory outputs to go from valid data outputs to Impedance state after Chip Select input has been de-activated.

i). tOD : READ cycle time.

WRITE cycle Figure 2.18: SRAM WRITE cycle

i). t_0 : Address inputs change to a new address to which data is to be written. This is the beginning of a WRITE cycle.

ii). t_1 : Chip Select input is activated and at the same time, the R/W input is set LOW for a WRITE operation.

iii). t_{AS} : Address set-up time - time taken by the RAM's address decoders to respond to the new address.

iv). t_2 : Data to be written into the addressed memory location is applied. The data has to be held stable for at least a time t_{DS}, data set-up time, before R/W and CS are returned HIGH. The data also has to be held stable for at least a time t_{DH}, data hold time, after R/W and CS have returned HIGH. Similarly, the address inputs have to be maintained for a time t_{AH}, address hold time after R/W and CS have returned HIGH. If any of these set-up or hold-time requirements are not met, the WRITE operation will not occur reliably.

v). t_{WC}: WRITE cycle time.

2.8 Dynamic RAM (DRAM)

2.8.1 DRAM Cell

A DRAM cell is shown on Fig. 2.19. Figure 2.19: DRAM Cell The FET acts as a switch while the capacitor is the actual storage element.

WRITE operation: The SELECT line is held HIGH to turn on the transistor. The voltage applied to the data input/output line charges or discharges the capacitor to store either a 1 or a 0. The SELECT line is then made LOW, turning off the transistor and opening the path to the capacitor. In practice, the charge in the capacitor leaks off after about 2ms.

READ operation: The SELECT line is made HIGH, connecting the capacitor to the data input/output line. The charge stored in the capacitor determines the voltage that appears on the data input/output line.

Advantages of DRAMs over SRAMs include;

i). High capacity due to its simple cell structure.

ii). Low cost per bit.

iii). Power consumption is low because the DRAM cell only draws a signi

cant amount of current when it is being charged.

Disadvantages of DRAM over SRAM include;

i). Slower than SRAMs.

ii). DRAMs require more external support circuitry than SRAMs because of the refreshing and address multiplexing operations that must take place.

2.8.2 DRAM Structure

A DRAM is usually arranged as a matrix of single bit registers as shown in Fig. 2.20. Figure 2.20: DRAM Structure In order to reduce the required number of address pins, a DRAMs memory address is split into two parts: a row address and a column address corresponding to the row and column of the required cells. the row address is first applied to the DRAM chip and then the column address is applied

using the same address pins as the row address. This is known as address multiplexing. This concept is illustrated using a 64×1 DRAM in Fig. 2.21.

Figure 2.21: DRAM address multiplexers In the case shown in Fig. 2.21, we have saved three address pins but we have an additional two pins, CAS and RAS, so overall, we save only one pin by using address multiplexing. The saving is higher with larger DRAMs, e.g. using address multiplexing with a $256K \times 1$ DRAM, the saving is seven pins. The timing diagram corresponding to Fig. 2.21 is shown in Fig. 2.22.
Figure 2.22: DRAM timing

a). t_0 : Row address is applied to the address inputs. After allowing time for the address inputs to stabilize, the Row Address Strobe RAS is driven LOW at t1. RAS clocks the row address latch to store the row address.

b). t_2 : Column address is applied to the address input pins.

c). t_3 : Column Address Strobe, CAS is driven LOW and the column address is loaded into the column address latch.

When both portions of the address are in their respective latches, the decoders will select the memory cell that is being accessed, and a READ or WRITE operation will take place just as in a static RAM.

2.8.3 DRAM refreshing

DRAMs are designed so that each time a READ or WRITE operation is performed on a memory cell, all the cells on the same row will be refreshed. However, it cannot be guaranteed that every row of a DRAM will be read from or written into every 2ms so the refresh operation has to be performed by other means.

Typical refresh circuitry contains a refresh counter that generates sequential row addresses for the refresh operation. The addresses from the refresh counter are used to access the various rows of the DRAM to perform a row refresh operation once in every 2ms.

2.9 Memory Expansion

Memory expansion refers to the combination of two or more memory ICs to;

i). Form longer words.

ii). Store more words.

iii). Both to form longer words and to store more words.

2.9.1 Expanding word size

This involves the combination of several ICs to form longer words.

Example 2.3:

Suppose we need to store 16 8-bit words but we only have 16×4 RAM ICs as shown on Fig. 2.23. Figure 2.23: 16×4 RAM The objective here is to create a 16×8 memory module using 16×4 ICs. Number of ICs needed $= (16 \times 8)=(16 \times 4) = 2$. Two 16×4 ICs can be connected as shown in Fig. 2.24 to form a 16×8 memory module. Figure 2.24: 16×8 RAM Module Any one of the 16 words is selected by applying the appropriate address code to the 4-line

address bus. Since each address bus line is connected to the corresponding address input of each IC, the same address code is applied to both chips so that the same location in each IC is accessed at the same time. The combination of the two ICs acts as a single 16×8 memory chip. READ operation: R/W = HIGH, CS = LOW. WRITE operation: R/W = LOW, CS = LOW.

Exercise

Construct a $1K \times 4$ DRAM module using $1K \times 1$ DRAM ICs.

2.9.2 Expanding Capacity

This refers to creating a memory module that has a bigger number of words than the ICs used to make it.

Example 2.4:

Suppose we need a memory that can store 32 4-bit words and we only have 16×4 RAM ICs as shown in Fig. 2.23. Number of ICs needed $= (32 \times 4)=(16 \times 4) = 2$. Two 16×4 ICs can be connected as shown in Fig. 2.25 to form a $32\times$ memory module. Figure 2.25: 32×4 RAM Module Each RAM is used to hold 16 4-bit words. Since the capacity of the memory is 32×4, 5 address lines are required. When AB4 = 0, the CS of RAM 1 enables this IC for a READ or WRITE operation. Then any address location in RAM 1 can be accessed by address bits AB3AB2AB1AB0. The range of addresses representing locations in RAM 1 are $A_{B4}A_{B3}A_{B2}A_{B1}A_{B0}$ = 00000 to 01111 (00 - 0F Hex).

When AB4 = 1, RAM 2 is enabled. The range of addresses located in RAM 2 is AB4AB3AB2AB1AB0 = 10000 to 11111 (10 - 1F Hex). In this example, only two ICs were used so it was convenient to use an inverter to select between the two ICs. When more than two ICs are used, it is more convenient to use a decoder, as the next example illustrates.

Example 2.5:

Construct a $1K \times 8$ ROM using 256×8 ROM ICs. Solution: The 256×8 ROMs are connected as

shown on Fig. 2.26 to achieve this purpose. Figure 2.26: $1K \times 8$ ROM Module The range of addresses in each ROM is as follows;

ROM 0: 0000000000 to 0011111111 (000 to 0FF Hex)

ROM 1: 0100000000 to 0111111111 (100 to 1FF Hex)

ROM 2: 1000000000 to 1011111111 (200 to 2FF Hex)

ROM 3: 1100000000 to 1111111111 (300 to 3FF Hex)

Note: It is also possible to increase both word size and capacity.

Exercise:

Using an IC such as the one shown on Fig. 2.23, construct a 32×8 memory module. 48

3 Design of Digital Systems

3.1 Introduction

3.2 State diagrams design

3.3 State machine chart representation and implementation using Programmable Logic Devices (PLD)

3.4 Introduction to VHDL