To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

RENESAS

Application Manual

Liquid Crystal Displays

Theory, Operation and Application

Global Document No. U18159EU1V0AN00 (1st edition)

Document no. 50542-2

Contents

1. Introduction 1-1
1.1 Contents of Section 2 1-1
1.2 Contents of Section 3 1-1
1.3 Contents of Section 4 1-1
1.4 About This Manual 1-1
2. Fundamentals of LCDs and NEC Electronics LCD Controllers 2-1
2.1 LCD Material 2-1
2.2 Basic LCD Operation 2-6
2.3 LCD Characteristics 2-10
2.4 Static Mode LCDs and LCD Controllers 2-14
2.5 LCD Boosters 2-18
2.6 NEC Electronics Microcontrollers 2-18
2.7 Multiplexed LCDs and Controllers 2-25
2.8 Duplex Mode LCDs and LCD Controllers 2-29
2.9 Triplex Mode LCDs and LCD Controllers 2-34
2.10 Quadruplex Mode LCDs and LCD Controllers 2-44
2.11 Summary of Operating Voltages for an NEC Electronics LCD Controller 2-49
3. LCD Visual and Ambient Factors 3-1
3.1 Specifications 3-1
3.1.1 Operating Temperature 3-2
3.1.2 Contrast Fundamentals 3-3
3.2 Operating Voltage, Contrast and Viewing Angle 3-4
3.3 Matching LCDs with Their Controllers 3-10
3.4 Drive Voltages vs. Contrast and Viewing Angle 3-12
3.5 Viewing Angles 3-13
3.6 Effect of Temperature on Contrast 3-15
3.7 Static Mode Operation 3-18
3.8 LCD ON/OFF Response Times 3-28
3.9 LCD Operating Frequency and Current 3-31
3.10 Ghosting and Dimming 3-36
4. Programming of NEC Electronics LCD Controllers 4-1
4.1 75X Programming Examples 4-1
4.1.1 Structure of 75X Programming Examples 4-2
4.1.2 ASCII to Seven-Segment Data Conversion 4-4
4.1.3 Icons 4-12
4.1.4 Programming Example for a 75X Microcontroller in Static Mode 4-15
4.1.5 Programming Examples for 75X Microcontrollers in Duplex Mode 4-22

4.1.6 Programming Examples for 75X Microcontrollers In Triplex Mode 4-29
4.1.7 Programming Examples for 75X Microcontrollers in Quadruplex Mode 4-35
4.2 78K0 and 78K0S Microcontrollers 4-40
4.2.1 Programming Examples for 78K0S LCD Controllers 4-42
4.2.2 Programming Example for 78K0 Microcontrollers 4-51

Figures

Figure 2-1. Liquid Crystal Molecules 2-3
Figure 2-2. Three Types of Liquid Crystal Structures 2-4
Figure 2-3. Twisted Liquid Crystal Molecules 2-5
Figure 2-4. Phases of Twisted Nematic LC Material 2-5
Figure 2-5. Optical Polarizer 2-6
Figure 2-6. Examples Using Two Optical Polarizers 2-7
Figure 2-7. LCD OFF State 2-8
Figure 2-8. LCD ON State 2-8
Figure 2-9. LCD ON/OFF States 2-9
Figure 2-10. Typical Transmissive LCD Voltage Characteristics 2-12
Figure 2-11. Typical Reflective LCD Voltage Characteristics 2-13
Figure 2-12. RMS Calculation of Square Wave 2-14
Figure 2-13. Typical Seven-Segment Static LCD with Decimal Point 2-15
Figure 2-14. Static LCD: Element OFF (Light) 2-16
Figure 2-15. Static LCD: Element ON (Black) 2-16
Figure 2-16. Static Resistor Ladder 2-18
Figure 2-17. μPD7530x/31x LCD Controller Block Diagram 2-20
Figure 2-18. LCDM Register Format 2-21
Figure 2-19. LCDC Register Format 2-22
Figure 2-20. LCD Data Memory 2-23
Figure 2-21. On-Chip Resistor Ladder Mask Option for μPD753xx LCD Controller 2-24
Figure 2-22. Typical Four-Digit, Seven-Segment Static LCD 2-26
Figure 2-23. Typical Eight-Digit, Seven-Segment Duplex LCD 2-27
Figure 2-24. Typical 10-Digit, Seven-Segment Triplex LCD 2-28
Figure 2-25. Typical 16-Digit, Seven-Segment Quadruplex LCD 2-29
Figure 2-26. Duplex LCD Timing 2-30
Figure 2-27. 1/2 Bias Resistor Ladder Configuration 2-31
Figure 2-28. Multiplexing Segments b and g in Duplex Mode 2-33
Figure 2-29. RMS Calculation for Segment b OFF and Segment g ON 2-34
Figure 2-30. Triplex Mode Timing with 1/2 Bias 2-35
Figure 2-31. Triplex Mode Timing with 1/3 Bias 2-36
Figure 2-32. 1/3 Bias Register Ladder Configuration 2-37
Figure 2-33. RMS Voltage Calculation for 1/2 Bias LCD in Triplex Mode 2-38
Figure 2-34. RMS Drive Voltage Calculations for 1/3 Bias LCD in Triplex Mode 2-41
Figure 2-35. 1/3 Bias Timing and RMS Voltage Calculation in Triplex Mode 2-43
Figure 2-36. Quadruplex Mode Timing 2-45
Figure 2-37. Quadruplex Mode LCD RMS ON Voltage Calculation 2-46

Figure 2-38. Quadruplex Mode Timing OFF Voltage Using In-Phase Technique 2-48
Figure 2-39. LCD RMS OFF Voltage for Quadruplex Mode Using In-Phase Technique 2-49

Figure 3-1. Effect of Viewing Angle on Contrast 3-6
Figure 3-2. Contrast vs. Applied Voltage 3-7
Figure 3-3. LCD Contrast Curves with Negative Viewing Angles 3-9
Figure 3-4. Contrast Curves for Typical Wristwatch with LCD 3-14
Figure 3-5. Effect of Temperature on Contrast 3-16
Figure 3-6. Contrast Temperature Compensation Using Diodes 3-17
Figure 3-7. Temperature Compensation Using an Op-Amp 3-20
Figure 3-8. LCD Response Time 3-30
Figure 3-9. LCD Segment Equivalent Circuit 3-32
Figure 3-10. Bodie Plot of fF vs. Typical Segment Impedance and Operating Current 3-34
Figure 3-11. Multiple LCD Segments 3-38
Figure 3-12. Voltages Resulting in Ghosting 3-40

Figure 4-1. Flowchart for All Four Modes 4-2
Figure 4-2. CLRLCD Subroutine 4-5
Figure 4-3. Programming Example for Four-Digit Triplex Mode 4-5
Figure 4-4. Packing of Seven-Segment Data in Triplex Mode 4-7
Figure 4-5. Sequence of Subroutine Instructions for Data Packed as Shown in Figure 4-4a 4-7
Figure 4-6. Sequence of Subroutine Instructions for Data Packed as Shown in Figure 4-4b 4-8
Figure 4-7. ROM Addressing for MOVT Instruction 4-9
Figure 4-8. MOVT Instruction Example 4-10
Figure 4-9. List of Locations for the Icons 4-12
Figure 4-10. Bit Manipulation Instructions That Turn On the BICON, DASH, and ONE Icons 4-13
Figure 4-11. LCD for Static Mode Programming Example 4-15
Figure 4-12. STATMAIN Routine for 75X Microcontroller in Static Mode 4-16
Figure 4-13. STATCONV Subroutine for 75X Microcontrollers in Static Mode 4-17
Figure 4-14. STATMOVE Subroutine for75X Microcontrollers in Static Mode 4-18
Figure 4-15. SICONADJ Subroutine for 75X Microcontrollers in Static Mode 4-19
Figure 4-16. ASCII to Seven-Segment Data Conversion in Static Mode 4-20
Figure 4-17. Storing of Seven-Segment Data in LCD Memory in Static Mode 4-21
Figure 4-18. Transfer of Seven-Segment Data to LCD Display Area in Static Mode 4-22
Figure 4-19. LCD for Programming Examples in Duplex Mode 4-23
Figure 4-20. Main Routine for 75X Microcontrollers in Duplex Mode 4-25
Figure 4-21. DUPLCONV Subroutine for 75X Microcontrollers in Duplex Mode 4-26
Figure 4-22. DICONADJ Subroutine for 75X Microcontrollers in Duplex Mode 4-27
Figure 4-23. Convert and Store Operations in Duplex Mode 4-28
Figure 4-24. Storing of One-Digit, Seven-Segment Data in LCD Display Area in Duplex Mode 4-28
Figure 4-25. MAIN Routine for 75X Microcontrollers in Triplex Mode 4-30
Figure 4-26. TRICONV Subroutine for 75X Microcontrollers in Triplex Mode 4-31
Figure 4-26 Continued 4-32

Figure 4-27. TICONADJ Subroutine for 75X Microcontrollers in Triplex Mode 4-33
Figure 4-28. Convert and Store Operations in Triplex Mode 4-34
Figure 4-29. Storing of One-Digit of Seven-Segment Data in LCD Display Area in Triplex Mode 4-34
Figure 4-30. 6-Digit LCD in Quadruplex Mode 4-35
Figure 4-31. MAIN Routine for 75X Microcontrollers in Quadruplex Mode 4-36
Figure 4-32. QUADCONV Subroutine for 75X Microcontrollers in Quadruplex Mode 4-37
Figure 4-33. QUICONADJ Subroutine for 75X Microcontrollers in Quadruplex Mode 4-38
Figure 4-34. Convert and Store Operations in Quadruplex Mode 4-39
Figure 4-35. Storing of One-Digit, Seven-Segment Data in Quadruplex Mode 4-40
Figure 4-36. Memory Address Space in μPD780308 78K0 Microcontroller 4-41
Figure 4-37. LCD Program Flowchart 4-44
Figure 4-38. Assembly Language Program Example of LCD Controller Using a 78K0S Microcontroller 4-45
Figure 4-39. C Language Program Example of LCD Controller Using a 78K0S Microcontroller 4-48
Figure 4-40. LCD Program Flowchart 4-53
Figure 4-41. Assembly Language Program Example of LCD Controller Using a 78K0 Microcontroller 4-54
Figure 4-42. C Language Program Example of LCD Controller Using a 78K0 Microcontroller 4-57

Tables

Table 2-1. Typical Specifications of Static LCDs 2-10
Table 2-2. Configurations of NEC Electronics LCD Controllers 2-19
Table 2-3. S9 Segment Controls 2-31
Table 2-4. Typical Specifications for Multiplexed LCDs 2-39
Table 2-5. Summary of RMS Voltages for NEC Electronics' LCD Controller 2-50

Table 4-1. ASCII Code for Numbers 0–9 4-11
Table 4-2. Number vs. ROM Location with Seven-Segment Data 4-12
Table 4-3. LCD Controller Configurations 4-40
Table 4-4. Program Specifications 4-43
Table 4-5. Program Specifications 4-52

1. INTRODUCTION

This application manual explains the theory, operation and application of liquid crystal displays (LCDs) and serves as a reference for design engineers using NEC Electronics microcontrollers with on-chip LCD controllers/drivers.

1.1 Contents of Section 2

- How to select an LCD whose voltage is compatible with the LCD controllers in NEC Electronics' 75X, 75XL, 78K0S, and 78K0 product lines
- Basics of LCDs and LCD controllers
- Static, multiplex, triples and quadruplex operation of LCDs and LCD controllers
- Complex waveforms generated by the LCD controller
- Description of NEC Electronics LCD controllers

1.2 Contents of Section 3

- LCD contrast, viewing angles, operating temperatures, on and off times, and operating frequencies
- How to interpret manufacturers' LCD specifications
- What to do if certain LCD specifications are missing from the data sheet

1.3 Contents of Section 4

- How to program NEC Electronics' 75X, 75XL, 78K0S, and 78K0 LCD controllers to achieve desired results when used with LCDs
- Four detailed program examples using seven-segment LCD displays controlled by an NEC Electronics 75X microcontroller operating in static, duplex, triplex and quadruplex modes
- Programming examples for NEC Electronics 78K0S and 78K0 microcontrollers
- Use and programming of icons

1.4 About This Manual

NEC Electronics' LCD microcontrollers can directly drive an LCD and technically are *LCD controllers/drivers*. For the sake of brevity, they are called LCD controllers in this manual.

The rest of this section provides suggestions for using this manual effectively.

1. Decide whether to use a static, duplex, triplex, or quadruplex LCD. The term multiplex refers to a duplex, triplex, or quadruplex LCD. Choosing one of the four modes normally depends on the number of LCD segments to be driven. The 75X controller can drive up to 32 segments in static mode, 64 in duplex mode, 96 in triplex mode, and 128 in quadruplex mode. The 78K0S and 78K0 families have several microcontrollers with LCD controllers, the largest of which drive 112 and 160 segments, respectively.
2. If you need basic information, read section 2 to understand how LCDs operate in each of the four modes. You will need to understand the duplex mode to understand triplex and quadruplex modes, so concentrate on the duplex mode first. You should also familiarize yourself with NEC Electronics' various LCD controllers, especially to learn how their generated voltages will determine some of the operating voltage requirements of the LCD used in your design.
3. Use section 3 as an aid for designing visual and ambient factors.
4. Use section 4 as an aid to initialize and program the LCD controllers for the 75X, 75XL, 78K0S and 78K0 products.

2. FUNDAMENTALS OF LCDs AND NEC ELECTRONICS LCD CONTROLLERS

LCDs have been used in large volumes for approximately 20 years. Their small size, low power consumption, and low cost make them ideal for calculators, wrist watches, laptop and notebook computers, exercise equipment, telephones, glucosometers, power meters, and cameras. In recent years, manufacturers have been producing color LCDs.

In their infancy, LCDs had limited applications because of poor contrast and limited viewing angles. Manufacturers overcame the contrast problem by making LCDs with implanted transistors (active-matrix LCDs) and multiple layers (super-twisted LCDs) and by backlighting the displays. The viewing angle problem was minimized through other technological improvements.

Controlled and powered by integrated circuit LCD controllers/drivers, LCDs require very low operating power, typically 5 µA, because they don't emit light. Instead they use ambient light as their viewing light source and are brightest in direct sunlight.

There are two types of LCDs: segmented and dot-matrix. The former can be a seven- or 14-segment display. The seven-segment versions display the numbers 0–9 and letters a–f and contain several seven-segment digits plus a decimal point for each digit. Occasionally these LCDs also contain custom icons (special symbols) to handle additional characters.

The 14-segment versions display the numbers 0–9, letters A–Z, and a few other ASCII characters. Most applications requiring an alphanumeric display use dot-matrix LCDs, which usually contain the complete ASCII character set (20H–7FH) and produce a better-looking display because each displayed character uses a five-column by seven-row matrix of dots.

2.1 LCD Material

LCD operation is dependent on the chemistry of the liquid crystal material and the physics of optics and electric fields as they relate to liquid crystals. Liquid crystal molecules in an LCD are arranged so that the LCD can either pass a beam of light unchanged, or rotate it by 90 degrees. An electric field applied to a liquid crystal layer is used to control how the light beam is affected as it passes through the liquid crystal.

Liquid crystals are organic compounds with long, thin cylindrical molecules (Figure 2-1a). The crystals have some of the fluid characteristics of a liquid and some of the molecular orientation of a solid.

There are many types of organic compounds. Some exist in three phases (Figure 2-1b):

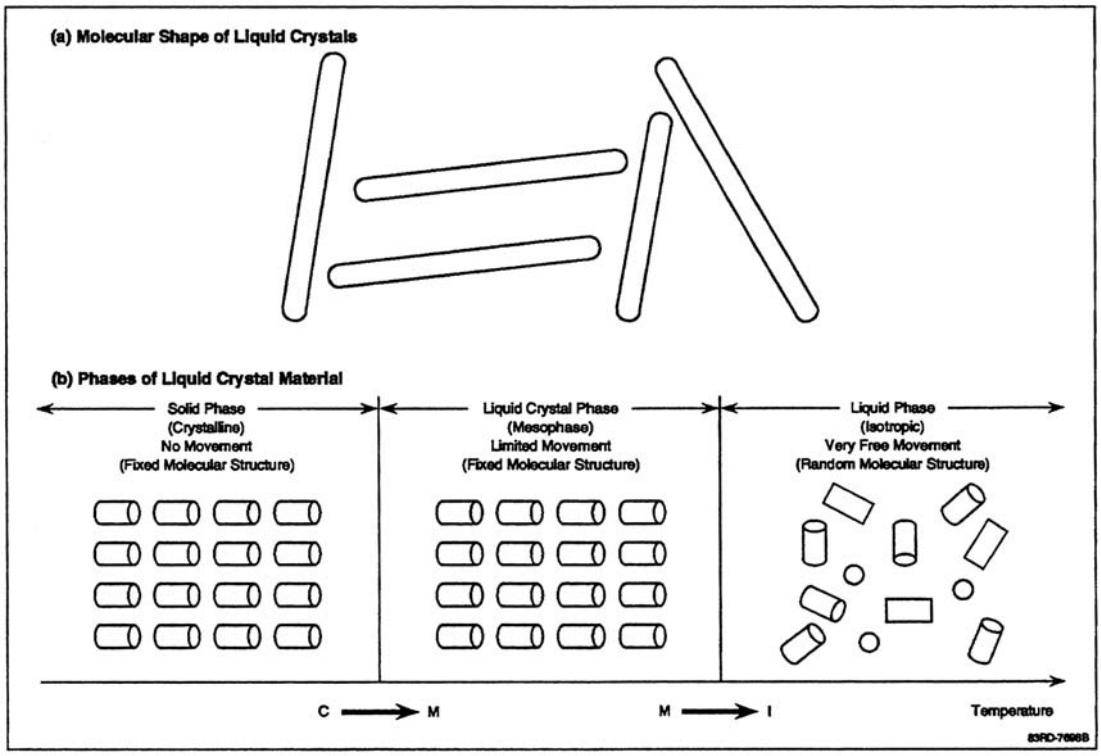
- Solid (crystalline)
- Liquid crystal (mesophase)
- Liquid (isotropic)

In the solid phase, molecules have no movement and remain in a fixed molecular structure that varies according to the liquid crystal compound being used. When heated in the solid phase, many organic materials melt and enter a liquid crystal phase called the *mesophase*. These compounds are called *thermotropic liquid crystals*. The transition into the mesophase is known as the crystal-to-mesophase (C → M) point, and the temperature when this occurs is called the C → M temperature.

The mesophase is an intermediate liquid crystal phase in which the material is neither a solid nor a liquid. It has a definite range of stability, with boundaries at the C → M and mesophase-to-isotropic (M → I) temperatures (Figure 2-1b).[1] The liquid crystal molecules maintain some positional order and limited movement of the molecules occurs when external forces are applied. The liquid crystal phase has macroscopic properties such as viscosity, electrical conductivity, and refractive indexes that are anistropic (have different values when measured along different directions).

1 Soref, Richard A., "Liquid Crystals," Proceedings of the Society of Photo-Optical Instrumentation Engineers *V38*, Seminar In-Depth Electro-Optics Principles and Applications, Boston, 30 April and 1 May, 1973, 23–28.

Figure 2-1. Liquid Crystal Molecules



The liquid crystal phase is stable up to a higher temperature (about 50 to 90 degrees Celsius), called the M → I temperature (Figure 2-1b). At this point, the material becomes isotropic fluid (a liquid phase in which properties such as velocity or light transmission have the same value when measured along axes in all directions) with very free molecular movement.[2] In this phase, the molecules can move freely and do not retain their structure.[3] When cooled, the material rapidly returns to the structure in the mesophase state.

The molecular organization of a liquid crystal is a repeating molecular pattern inherent to the liquid crystal compound being used. For simplicity, the diagrams here show the liquid crystal in three rows or planes (Figure 2-1b); actual liquid crystal material has many planes. Figure 2-2 shows three structurally different types of liquid crystals: nematic, smectic, and cholesteric.

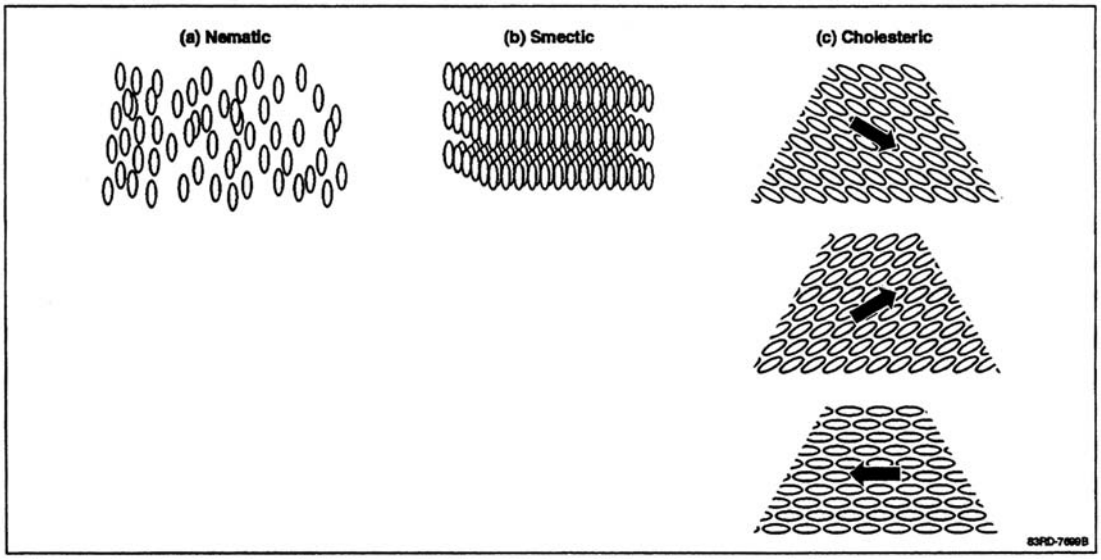
The nematic structure is the one-dimensional ordering of local parallelism of the long axis (Figure 2-2a). The smectic structure consists of very dense layers and is similar to a solid (Figure 2-2b). The cholesteric structure consists of a series of nematic planes (Figure 2-2c), in which the

[2] Shanks, I.A., "Properties and Prospects of Liquid-Crystal Displays," *Electronics and Power*, 20 March 1975, 301–305.

[3] Standisch Industries, Hamlin LCD Division, *Liquid Crystal Displays*, C93500A.

alignment direction progressively changes from one plane to the next, creating a spiral-like structure.[4]

Figure 2-2. Three Types of Liquid Crystal Structures



LCDs consist of a liquid crystal organic material that is manufactured in such a way as to produce molecular groups in a 90-degree twisted structure (Figure 2-3). The 90-degree twist in the liquid crystal phase allows incident light to be rotated 90 degrees when it passes through the LCD. Magnetic and electrical fields, surface pressure, and mechanical forces can alter the orientation of the molecular groups.

The twisted liquid crystal structure typically uses a nematic-type liquid crystal and two pieces of glass specifically prepared so that the liquid crystal material stays in one direction on the surface. The glass pieces have a conductive coating so that they can be used as electrodes. The glass plates are assembled so that the liquid crystal at each surface is orthogonal.[5] Cholesteric compounds are added to the nematic liquid crystal to help twist the molecules. When a liquid crystal is introduced between the plates, the molecules are formed into a structure having a smooth twist through 90 degrees from one plate to the other (Figure 2-3).[6]

[4] Soref, "Liquid Crystals," 23–28.

[5] Williams, Edward L., *Liquid Crystals for Electronic Devices*, 1975.

[6] Standisch

Figure 2-3. Twisted Liquid Crystal Molecules

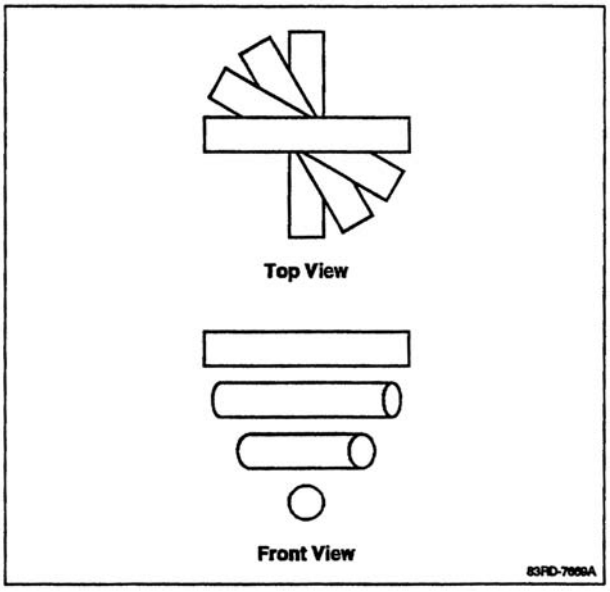
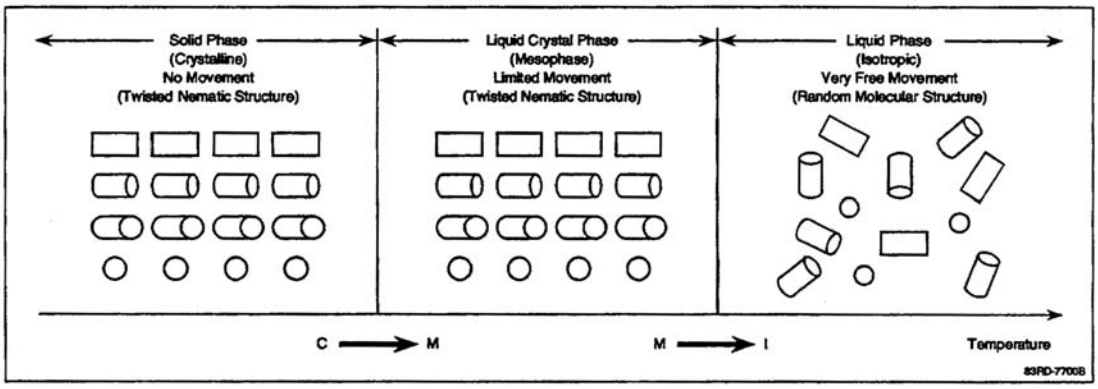


Figure 2-4 shows the twisted nematic liquid crystal material in the three phases described earlier. LCDs operate in the intermediate mesophase. At low temperatures, it is the viscosity of the liquid crystal (which increases as the temperature decreases) that determines an LCD's limit of operation. A temperature that produces a viscosity high enough so that the liquid crystal cannot respond, or cannot respond fast enough, is the temperature that defines the lowest operating temperature of the LCD.

Figure 2-4. Phases of Twisted Nematic LC Material



The upper temperature limit is defined by the point at which the material goes through the M → I transition. In an isotropic state, the molecular orientation of the material is random, and the

twisted structure is destroyed. When the material cools, it rapidly returns to a twisted nematic structure in the mesophase state.[7]

2.2 Basic LCD Operation

Liquid crystal compounds are used in displays because their twisted nematic configuration can twist light beams by 90 degrees and their molecular structure can be altered by applying an electric field. LCDs that use these principles are known as *twisted nematic field effect* (TNFE) LCDs.

In its basic operation, an LCD makes use of optical polarizers: a thin material (such as plastic film) that polarizes light. The polarizers can be horizontal or vertical, depending on the way in which it is physically oriented. Figure 2-5 shows a beam of light traveling in the X-axis that has light components in the Y-axis and Z-axis. This figure shows how a vertical polarizer (Z-axis) passes only the vertical component of light from a randomly polarized light source.

Figure 2-5. Optical Polarizer

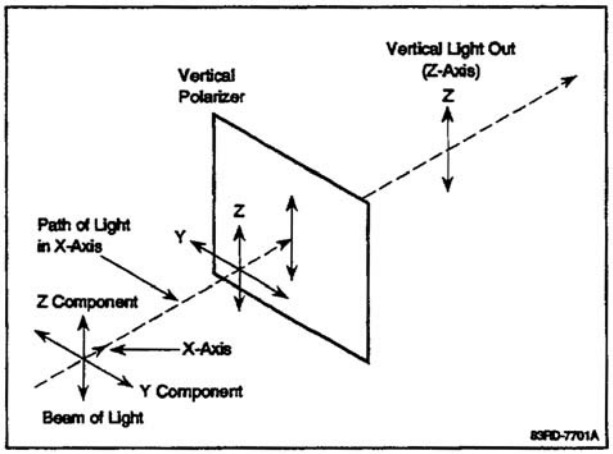
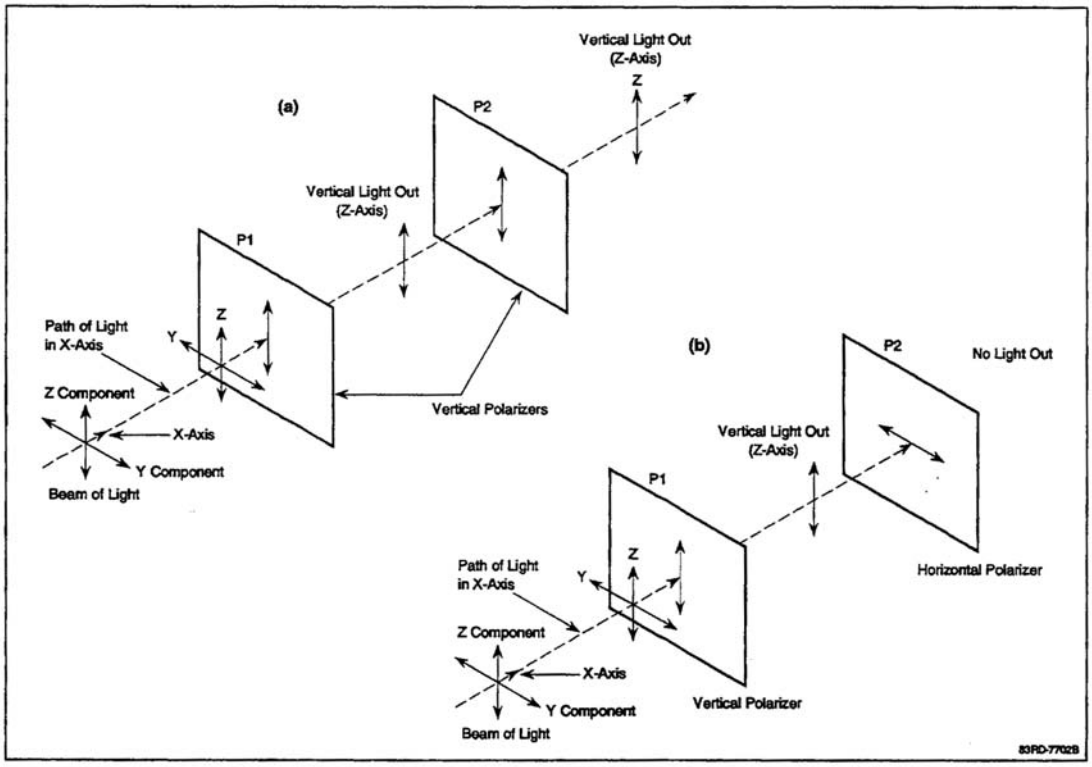


Figure 2-6 shows two vertical polarizers, where polarizer P1 filters out all light except light in the Z-axis, which passes through vertical polarizer P2. Figure 2-6b shows one vertical polarizer and one horizontal polarizer. Vertical polarizer P1 passes Z-axis light, which is then blocked by horizontal polarizer P2. Since horizontal polarizer P2 can pass Y-axis (horizontal) light only, no light is transmitted.[8]

[7] Standish Industries

[8] Seiko Instruments USA, Inc., "Liquid Crystal Displays," Technical Manual, 3–16.

Figure 2-6. Examples Using Two Optical Polarizers



Figures 2-7 and 2-8 show the basic structure of an LCD, which consists of vertical and horizontal polarizers, two transparent electrodes, and a liquid crystal layer. These four layers correspond to the four layers of the twisted liquid crystal molecules shown in Figures 2-1, 2-3, and 2-4.

Figure 2-7 shows an LCD in its OFF state and how 0 voltage across the electrodes of the LCD controls light in the LCD. Randomly polarized light is incident on the LCD at the left of the diagram while an observer is on the right. When randomly polarized light is incident on the vertical polarizer, only the vertical (Z-axis) light passes through the polarizer and transparent electrode to the liquid crystal layer. Because there is no electric field (0 volts) across the liquid crystal layer, it remains in a twisted configuration and twists the plane of polarization of light by 90 degrees. Polarized light is output in the Y-axis, passing through the second transparent electrode and through the horizontal polarizer. Thus, the observer on the right sees light and the LCD is OFF.[9] [10] [11] [12]

[9] Standish Industries
[10] Williams, Edward L., *Liquid Crystals for Electronic Devices*, 1975.
[11] Seiko Instruments.
[12] Jones, M., "Liquid Crystal Displays: Principles and Applications," *Proceedings of the Technical Conference of Electro-Optics/Laser International '84UK*, Brighton, England, 20–22 March 1984, 324–334.

Figure 2-7. LCD OFF State

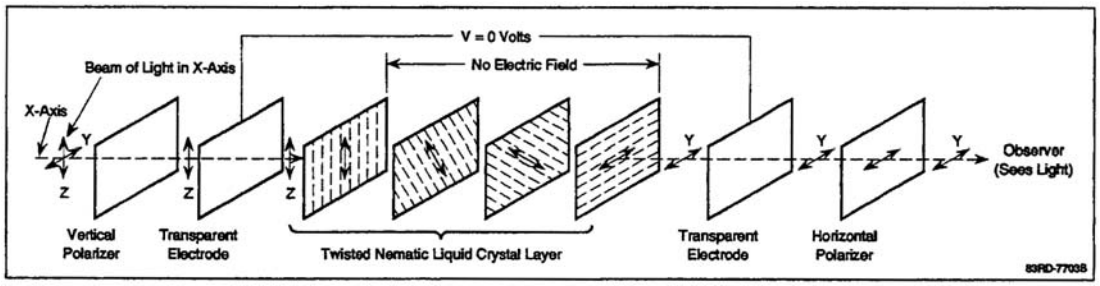
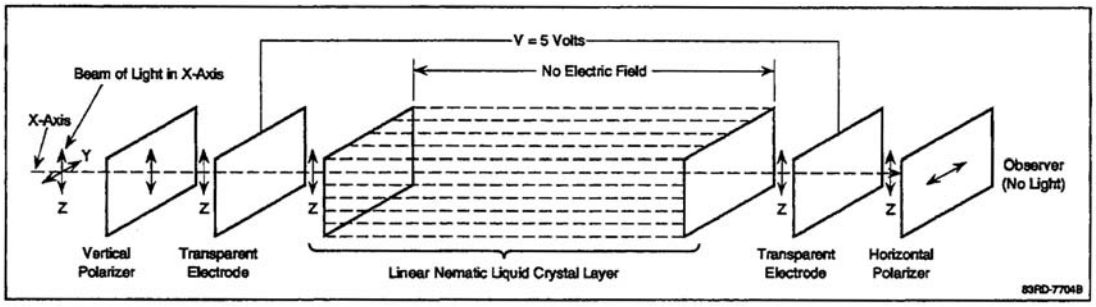


Figure 2-8 shows an LCD in its ON state and how 5 volts across the electrodes of the LCD control light in the LCD. In this example, placing 5 volts across the liquid crystal layer forces the liquid crystal molecules out of their twisted configuration and into a linear nematic alignment in the X-axis. Randomly polarized light is incident on the LCD at the left of the diagram and there is an observer on the right. When randomly polarized light is incident on the vertical polarizer, only the vertical (Z-axis) light passes through it. This light passes through the transparent electrode, the liquid crystal layer, and the second transparent electrode to the horizontal polarizer. Since the liquid crystal is not twisted, the light is not twisted and remains in the Z-axis. The second polarizer is horizontal (Y-axis), blocking the light and preventing the observer on the right from seeing any light. The observer sees only black; this is the LCD's ON state.[13] The magnitude of the voltage required to orient the molecules fully in the X-axis, and consequently in the electric field across the electrodes, depends on the liquid crystal used.

Figure 2-8. LCD ON State



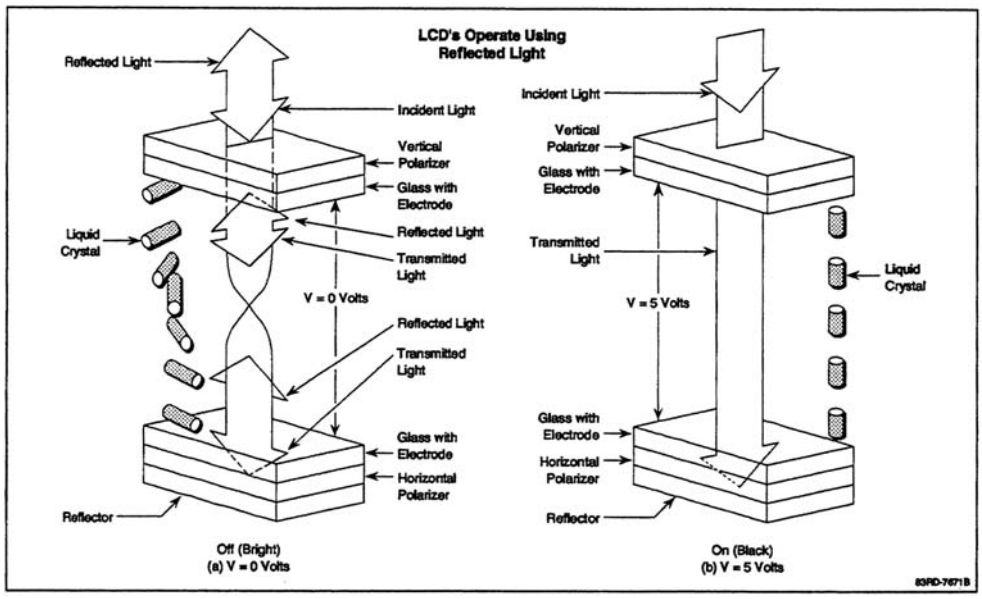
Figures 2-7 and 2-8 show how the liquid crystal in a transmissive LCD either passes or blocks the light at one end of the display, depending on the voltage across the electrodes. Power is consumed because of the active light source required.[14] However, most LCDs have a reflector attached to the second horizontal polarizer and require very low power (typically 5 μA in a one-

[13] Standish

[14] Amperex, "Introduction to Liquid Crystal Displays," *Technical Manual*, 1–21.

half-inch-high, four-digit display). The power is low because ambient light provides the light source (Figure 2-9).[15] Such an LCD is called a *reflective LCD*, the type described in this manual unless otherwise stated.

Figure 2-9. LCD ON/OFF States



Reflective LCDs are widely used in battery-powered applications because they have good brightness and contrast in high ambient light environments.[16] If a reflective display has zero volts across the electrode, the liquid crystal remains in a twisted configuration (Figure 2-9a). Incident light passes through the vertical polarizer and the liquid crystal layer, where it is twisted 90 degrees, transmitted through the horizontal polarizer, and then reflected by the reflector. The reflected light passes back through the horizontal polarizer, is twisted again by 90 degrees, passes through the vertical polarizer, and is seen as a light area on the display. This is the OFF state.

If there is 5 volts across the electrodes, light passes through the liquid crystal untwisted and is blocked by the horizontal polarizer (Figure 2-9b). There is no light incident on the reflector and no reflected light; a black area is seen on the display. This is the ON state.[17] Other LCDs combine transmissive and reflective characteristics, and are known as transflective LCDs.[18] Transflective LCDs are not covered in this manual.

[15] Seiko Instruments

[16] Amperex

[17] Standish Industries

[18] Amperex

2.3 LCD Characteristics

Many characteristics must be considered when selecting and using an LCD, including operating voltage, response time, operating frequency, type of liquid crystal fluid, contrast ratio, operating and storage temperatures, and viewing angle. This section partially covers the operating voltages of LCD. Section 3 contains information about operating voltages and other LCD characteristics.

One of the most important characteristics of an LCD is its operating voltage. The voltage used to drive an LCD has a squarewave-like waveform and its voltage is expressed as a root mean square (RMS) value. During LCD operation, the average voltage across any LCD segment must be zero.

LCDs are low-voltage AC devices that operate from approximately 2 to 6 volts RMS. The DC component of the AC signal must be kept very low, typically less than 100 mV. If it is not kept low, the liquid crystal fluid degrades and the life of the display is reduced.[19] An AC-driven LCD has a life expectancy of about ten years. A DC-driven LCD has a life expectancy of several hundred hours.[20] Table 2-1 shows the typical specifications for a static LCD.[21]

Table 2-1. Typical Specifications of Static LCDs[22]

Parameter	Fluid Type 2 Note 1			Fluid Type 5 Note 2			Fluid Type 7 Note 3			Fluid Type 8 Note 4			Unit
	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Operating voltage	4	7	15	3	5	12	5	7	15	5	7	15	V_{RMS}
DC drive component allowable		50				50			50			50	mV
Operating frequency range	30	60	100	30	60	100	30	60	100	30	60	100	Hz
Current (all segments on) Note 5		5	20		5	20		5	20		5	20	µA
Capacitance (all segments on) Note 5		3000			3000			3000			3000		pF
DC resistance (all segments on)		50			50			50			50		MΩ
Visual threshold (V_{th}): 10% on at 25°C		2.2			1.7			2.3			1.8		V_{RMS}
Visual threshold (V_{th}): 10% on at 0°C		2.4			2.0			2.6			2.4		V_{RMS}
Visual threshold (V_{th}): 90% on at 25°C		3.1			2.8			3.4			2.6		V_{RMS}
Visual threshold (V_{th}): 90% on at 0°C		3.2			3.1			3.7			3.2		V_{RMS}
Typical T_{ON} and T_{OFF} response time		< 50			< 225			< 60			< 55		ms

[19] Smith, Paul, "Multiplexing Liquid Crystal Displays," *Electronics*, 25 May 1978, 113–121.

[20] Amperex

[21] Williams, "Liquid Crystals for Electronic Devices."

[22] Standish

Parameter	Fluid Type 2 Note 1			Fluid Type 5 Note 2			Fluid Type 7 Note 3			Fluid Type 8 Note 4			Unit
	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
(T_A = 25°C) Note 5		total			total			total			total		
Typical segment on response time (T_A = 25°C) Note 5		15	20		50	100		20	30		15		ms
Typical segment off response time (T_A = 25°C) Note 5		30	60		175	300		30	30		40		ms
Typical segment on response time (T_A = 0°C) Note 5		50	180		150	600		70	90		35	75	ms
Typical segment off response time (T_A = 0°C) Note 5		150	320		1000	1500		80	110		110	150	ms
Contrast ratio		20:1			20:1			20:1			20:1		
Operating temperature range	–20 Note 6		85	–10		55	–30 Note 7		105	–40 Note 8		85	°C
Storage temperature range	–55		85	–55		55	–55		105	–55		85	°C
Expected life		100K			100K			100K			100K		Hrs
Viewing angle from normal		± 75° at 7V			± 75° at 5V			± 75° at 7V			± 75° at 12V		
		± 60° at 6V			± 60° at 4V			± 60° at 6V			± 60° at 10V		
		± 45° at 5V			± 45° at 3V			± 45° at 5V			± 45° at 6V		

Notes:

1. A wide temperature fluid with enhanced speed of response at extremely low temperatures
2. The most economical fluid type, intended for use in applications not requiring storage or operation within extreme humidity and temperature levels
3. An extremely wide temperature fluid (–30°C to +105°C) for extreme outdoor applications; closest material available for MIL-SPEC-type applications
4. A fluid with very low viscosity and an extreme temperature range. The fluid is combined with array process LCD technology and offers a combined T_{ON} and T_{OFF} time of 7 seconds at –40°C
5. Applies to a 4-digit (½-inch) instrument LCD (3906)
6. Typical combined on and off times of 0.8 seconds at –20°C
7. Typical on and off times of 0.4 seconds at –10°C; 1.0 seconds at –20°C; 2.5 seconds at –30°C
8. Typical combined on and off times of 0.7 seconds at –20°C; 3 seconds at –30°C; 7 seconds at –40°C

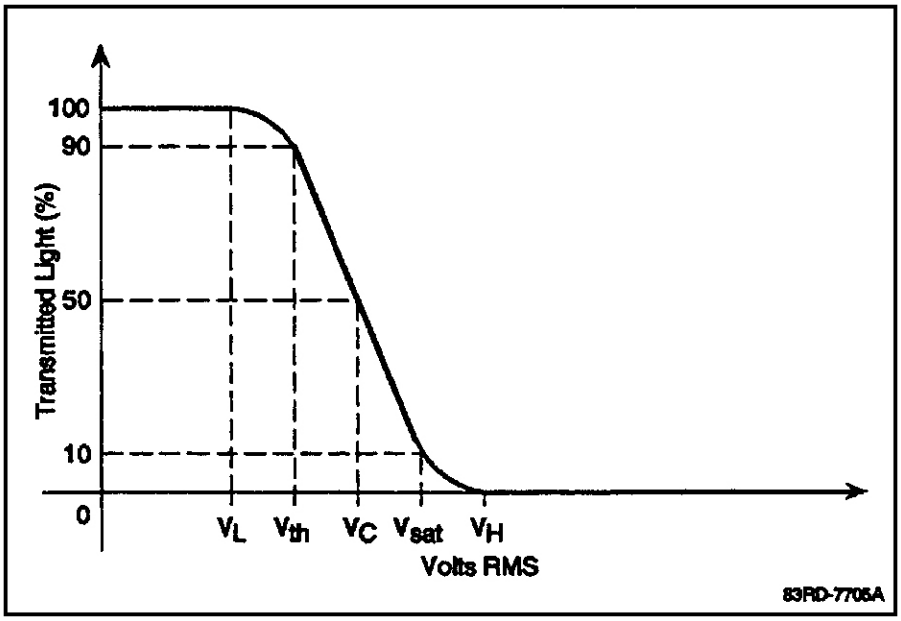
Figure 2-10 shows a curve with typical voltage characteristics for the transmissive LCD shown in Figures 2-7 and 2-8. When voltage is very low (V_L or 0 volts), 100% of the light passes through the LCD (Figure 2-7). When voltage is high (V_H or 5 volts), 0% passes through (Figure 2-8). For a typical transmissive LCD, voltage characteristics curves have three critical values:[23]

- V_{th} = 90% of the light passed through the LCD
- V_C = 50% of the light passed through the LCD
- V_{sat} = 10% of the light passed through the LCD

Where:

- V_{th} = the LCD threshold voltage
- V_C = the LCD center voltage (approximately halfway between V_{th} and V_{sat}
- V_{sat} = the LCD saturation voltage

Figure 2-10. Typical Transmissive LCD Voltage Characteristics



The reflective LCD shown in Figure 2-9 is an ideal model. It is OFF when V = 0 volts RMS and displays a bright area (all light reflected). It is ON when V = 5 volts RMS and displays a black area (no light reflected). However, manufactured reflective LCD devices have characteristics such as

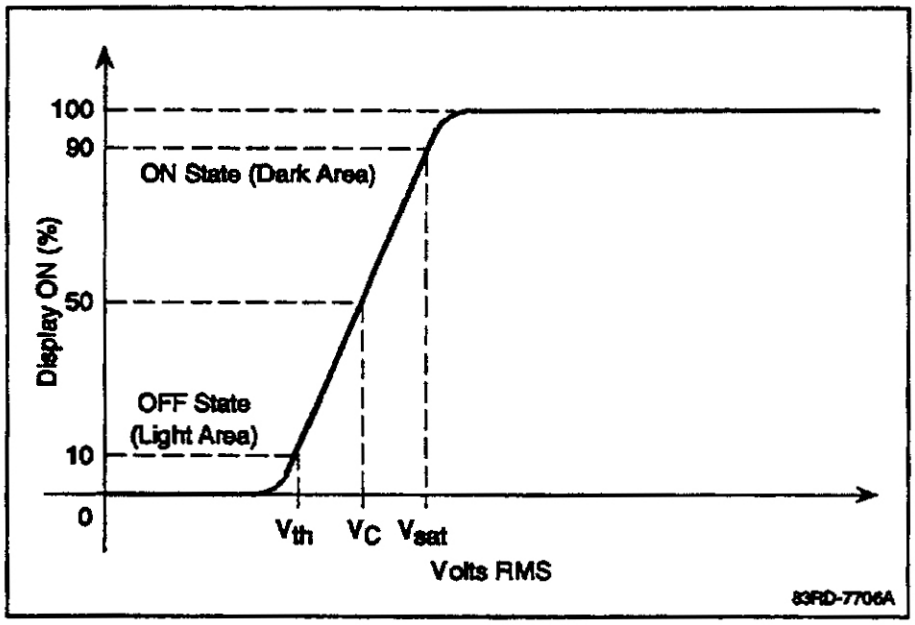
[23] Shanks, 301–305.

those in Figure 2-11, where the OFF point is not 0 volts and the fully ON point is not 5 volts. Three voltages are specified for reflective LCDs (Figure 2-11):[24]

- V_{th} = display 10% ON (transmitted light reduced by 10%)
- V_C = display 50% ON (transmitted light reduced by 50%)
- V_{sat} = display 90% ON (transmitted light reduced by 90%)

Table 2-1 refers to V_{th} as the 10% ON voltage and V_{sat} as the 90% ON voltage. In this case, for fluid type 2 at 25°C, V_{th} is 2.2 volts RMS and V_{sat} is 3.1 volts RMS. V_C normally is enough voltage to give reasonable contrast.[25] V_{th} is the threshold ON voltage (barely visible) and V_{sat} is the fully ON voltage (maximum contrast).[26] In all cases, the best way to evaluate any LCD characteristic is by visual inspection.

Figure 2-11. Typical Reflective LCD Voltage Characteristics



An RMS value is calculated as

$$V_{RMS} = \sqrt{\frac{1}{T}\int_0^t f^2(t)\,dt}$$

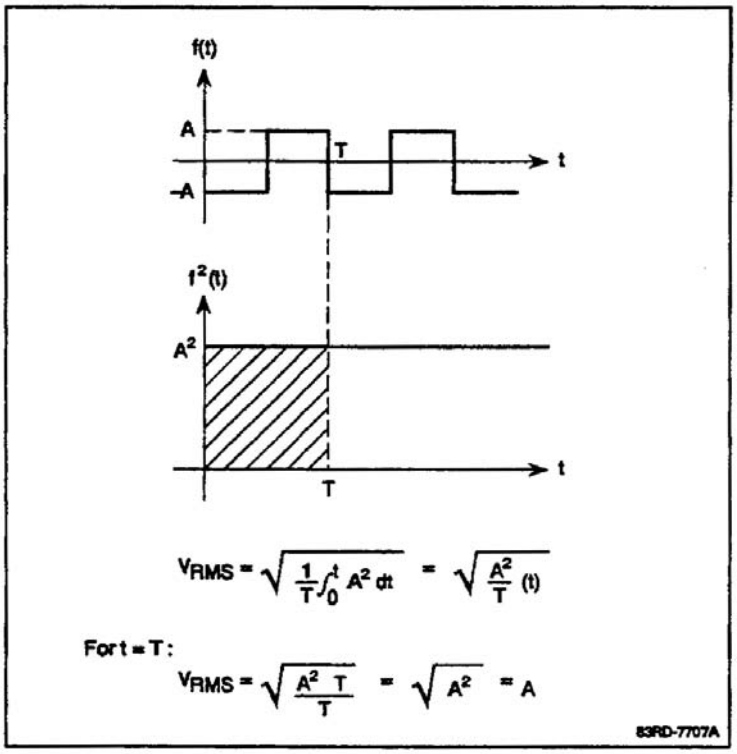
[24] Seiko Instruments, 3–16.

[25] Seiko.

[26] Jones, 324–334.

Figure 2-12 shows the procedure for calculating RMS values, where the RMS voltage of a square wave of amplitude A is A volts RMS.

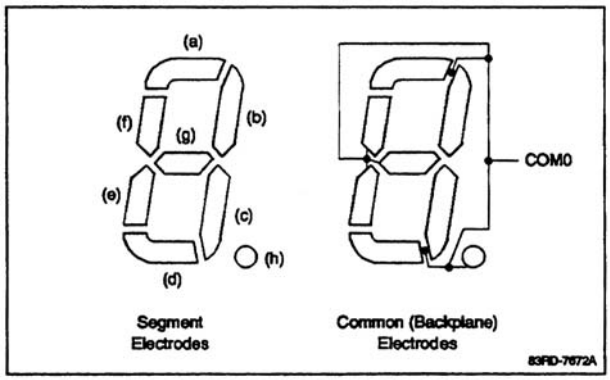
Figure 2-12. RMS Calculation of Square Wave



2.4 Static Mode LCDs and LCD Controllers

The simplest type of LCD is a static mode seven-segment display. Figure 2-13 shows a typical configuration consisting of separate segments (a–h) on one glass surface. The other glass surface has an electrode connected to all segments (a–h). This electrode is called the *backplane* or *common electrode*. In a static LCD, COM0 is connected to the backplane and each separate electrode segment is connected to a separate voltage source. The voltage difference between the segment and backplane electrode is the voltage across a segment. Each segment can be envisioned as an LCD element with voltage across the electrodes as shown in Figures 2-7 and 2-8.

Figure 2-13. Typical Seven-Segment Static LCD with Decimal Point



As an example, assume the number 3 is to be displayed on the seven-segment static LCD and each segment has an RMS OFF voltage of 0 volts and an RMS ON voltage of 5 volts. To display the number 3, segments a, b, c, d, and g must be ON and segments e, f, and h must be OFF. If the RMS voltage across a segment is zero, the segment is OFF and displayed as a light area. If the RMS voltage is 5, the segment is ON and the segment is displayed as a black area.

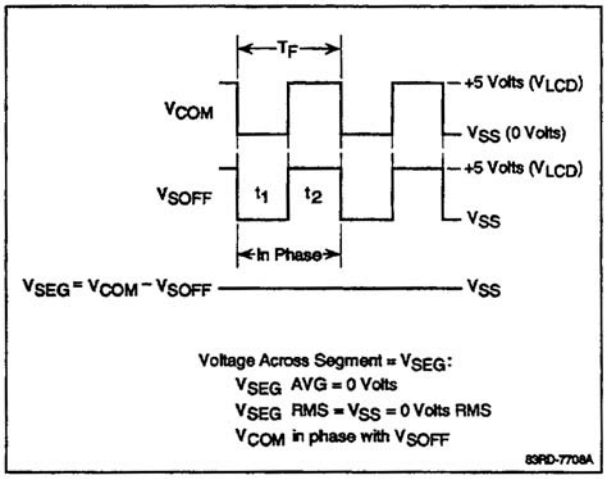
Figure 2-14 shows the waveforms required for an OFF segment. V_{COM} has amplitude of 5 volts and is connected to COM0. The frame period (T_F) has two time slots, t_1 and t_2. Each OFF segment has the V_{SOFF} waveform with amplitude of 5 volts connected to it. Therefore, the voltage across an OFF segment (V_{SEG}) is:

$V_{SEG} = V_{COM} - V_{SOFF} = 0$ volts. This voltage has an RMS value of 0 volts and an average value of 0 volts. Also note that V_{COM} is in phase with V_{SOFF}.

Figure 2-15 shows the waveforms required to have an ON segment. Each ON segment has the V_{SON} waveform with amplitude of 5 volts connected to it. The voltage across an ON segment (V_{SEG}) is:

$V_{SEG} = V_{COM} - V_{SON}$ = square wave amplitude ±5 volts = 5 volts RMS.

Figure 2-14. Static LCD: Element OFF (Light)

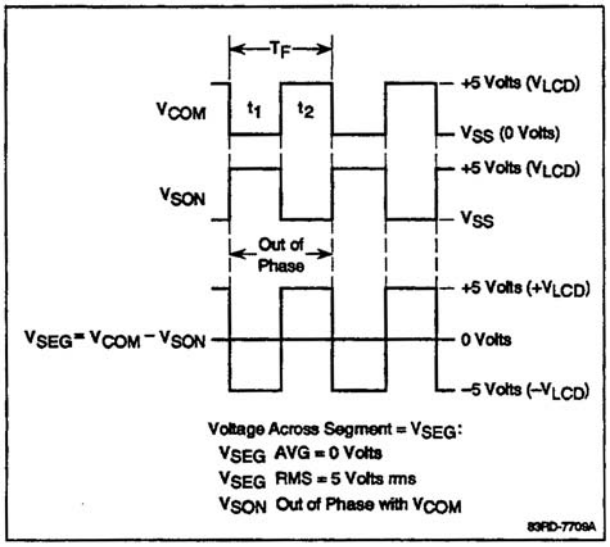


The average value is 0 volts and the RMS ON value is 5 volts. Note that V_{SON} is 180 degrees out of phase with V_{COM}. The difference between the RMS ON and OFF voltages is:

$$V_{DIFF} \text{ static} = 5 - 0 = 5 \text{ volts RMS.} \qquad (2.1)$$

The V_{COM}, V_{SOFF} and V_{SON} waveforms are identical. It is the phase relationship of these waveforms that produces ON or OFF segments. To turn OFF a segment, V_{COM} and V_{SOFF} must be in phase with each other to produce a zero voltage difference across the segment (Figure 2-14). To turn ON a segment, V_{COM} and V_{SON} must be out of phase with each other to produce a square wave of ±5 volts (Figure 2-15).

Figure 2-15. Static LCD: Element ON (Black)



If the static LCD specified in Table 2-1 is used, then the RMS ON and OFF voltages can be attained easily. The LCD controller described above outputs 0 volts RMS for the OFF segments and 5 volts RMS for the ON segments. Any of the fluid types in Table 2-1 can be used. The minimum OFF voltage is 1.7 volts RMS for fluid type 5. The maximum ON voltage is 3.7 volts RMS for fluid type 7 at 0°C. Thus, the voltages generated by the LCD controller are well below the minimum OFF voltage and well above the minimum ON voltage.

It should be noted from the specification in Table 2-1 that the maximum allowable DC voltage component is 50 mV for fluid types 5, 7 and 8 and 50 mV typical for fluid type 2. It is important that the LCD controller has ≤ 50 mV DC across any segment to maximize the life of the displays in Table 2-1.

The voltage drive levels produced by the common and segment lines in the LCD controller are set by a resistor ladder, which consists of up to four resistors connected in series to produce up to four voltage levels. The number of voltage levels required depends on whether static, duplex, triplex, or quadruplex mode is used.

In NEC Electronics' LCD controllers, the voltage levels from the resistor ladder are connected to pins V_{LC0}, V_{LC1}, V_{LC2}, and V_{SS}. The voltage source for the resistor ladder is usually the BIAS pin output from the LCD controller.

The static mode at full contrast is an exception, because no resistors are required to set the common and segment drive levels. In this case, the required voltage levels are obtained by connecting V_{LC0}, V_{LC1}, V_{LC2}, and V_{SS} as shown in Figure 2-16a. Sometimes the maximum voltage level from the LCD controller must be reduced to match the voltage level from the LCD controller to the voltage level required by the LCD. In static mode, adding resistors R× and R as shown in Figure 2-16b reduces the voltage. The maximum voltage at input V_{LC0} is:

$$(R \times V_{BIAS})/(R\times + R)$$

If the 5-volt RMS drive level in the example above is too large for fluid type 7, then adding resistors R× and R can reduce it.

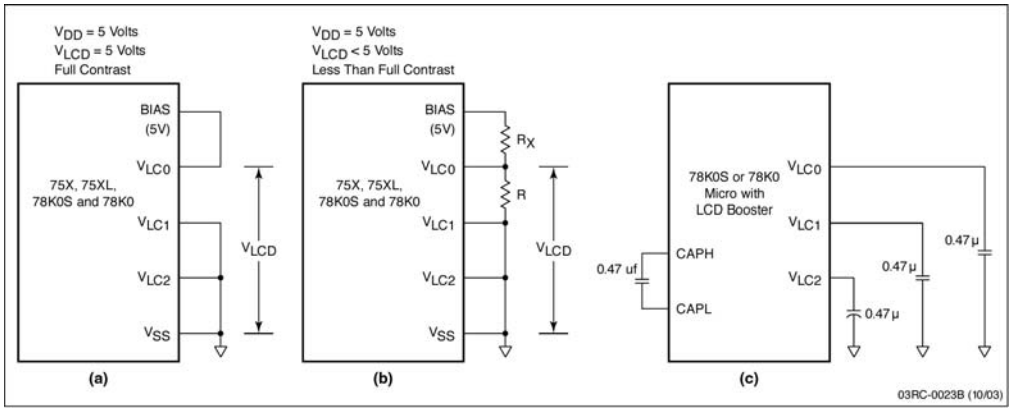
It is the job of the LCD controller to generate the common and segment waveforms. Duplex, triplex, and quadruplex modes use the same principles as static mode, but the waveforms are more complex since each segment line from the LCD controller controls multiple LCD segments. These waveforms are not the same for all LCD controllers. NEC Electronics' LCD controller waveforms are described later in this manual.

2.5 LCD Boosters

Once the operating voltage of the microcontroller drops below about 2.5 volts, the LCD supply voltage drops and the LCD becomes dim. There are two ways to solve the problem of operating an LCD at lower voltages. One way is to use low-voltage LCDs, but these tend to be expensive. The other is to make LCD controllers with voltage boosters. Using a booster is the most economical way to operate LCDs below 2.5V. The boosters use external capacitors in place of bias resistors and take about 0.5 seconds to get the boosters operating at full voltage after power is turned on. Figure 2-16c shows a typical configuration for connecting the capacitors to the microcontroller's booster. The description of how the boosters work is beyond the scope of this manual.

NEC Electronics' boosters have an internal regulator that can be programmed by a special function register (by the user) to output either 1 or 1.5 volts. When the booster regulator is set at 1 volt, the booster outputs 3 volts at V_{LC0}, 2 volts at V_{LC1}, and 1 volt at V_{LC2}. When the booster regulator is set at 1.5 volts, the booster outputs 4.5 volts at V_{LC0}, 3 volts at V_{LC1} and 1.5 volts at V_{LC2}.

Figure 2-16. Static Resistor Ladder



2.6 NEC Electronics Microcontrollers

NEC Electronics produces an extensive line of 4- and 8-bit microcontrollers consisting of three basic types of MCUs:

- General-purpose
- MCUs with fluorescent indicator panels (FIP® panels)[27]
- MCUs with LCD controllers

[27] FIP is a registered trademark of NEC Electronics America, Inc.

The 4-bit 75X MCUs operate from 2.7 to 6 volts and the 75XL from 1.8 to 5.5 volts. The 8-bit 78K0S and 78K0 MCUs operate from 1.8 to 5.5 volts. All of these devices contain a CPU, ROM, RAM, ports, timers, serial interface and interrupts. Many have flash memory, A/D converters, D/A converters, watchdog timers, real-time clocks, UARTs, I^2C buses, FIP controllers, and LCD controllers.

Figure 2-17 shows a block diagram of the 4-bit μPD7530x/31x LCD controller, which contains two control registers (LCDM and LCDC), a timing generator, four LCD common drivers (COM0–COM3), 32 segment drivers (S0–S31), and four bias connections (BIAS, V_{LC0}, V_{LC1}, and V_{LC2}). The LCD data display area in RAM for the 75X and 75XL devices is in locations 1E0H–1FFH. The timing generator uses the F_{LCD} clock source, LCDM register, and bias circuitry to generate the complex square wave-type waveforms output by the segment and common driving signals.

Since the LCD segment and common driver outputs can directly drive an LCD, NEC Electronics' microcontrollers with LCD controllers are usually referred to as *LCD controller/drivers*. To simplify the text, the term *LCD controller* is used throughout in place of LCD controller/driver.

The functional blocks of most NEC Electronics LCD controllers are very similar to what is shown in Figure 2-17. The numbers and names of the control registers, number of common and segment lines and locations of LCD RAM data may differ, but their basic functionality is the same.

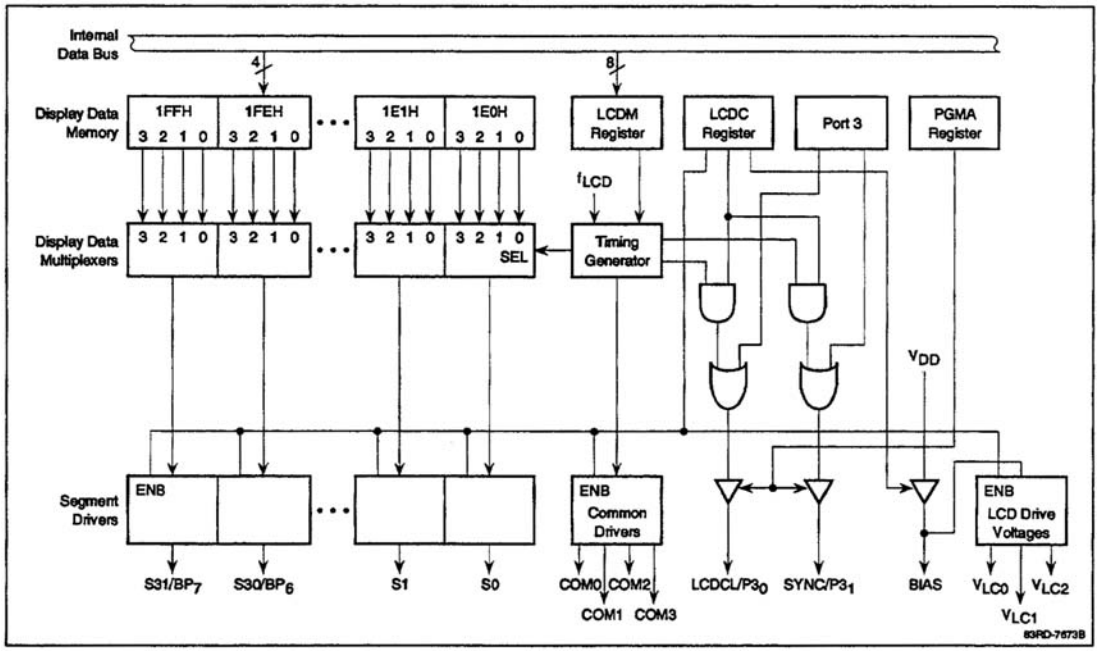
A 32-segment, 4-common LCD controller is normally described as having a 32 × 4 configuration. A 32 x 4 configuration can drive a maximum of 32 × 4 = 128 segments. The variety of configurations available from NEC Electronics LCD controllers is listed in Table 2-2.

Table 2-2. Configurations of NEC Electronics LCD Controllers

Configuration	Number of Segments
40 × 4	160 segments
32 × 4	128 segments
30 × 3	90 segments
28 × 4	112 segments
26 × 4	104 segments
24 × 4	96 segments
23 × 4	92 segments

Configuration	Number of Segments
20 × 4	80 segments
19 × 1	19 segments
15 × 4	60 segments
5 × 4	20 segments

Figure 2-17. μPD7530x/31x LCD Controller Block Diagram



The display mode (LCDM) register and display control (LCDC) register are two control registers located within the LCD controller that must be programmed to operate the LCD controller. The LCDM register sets one of four modes (static, duplex, triplex, or quadruplex), one of 16 LCD refresh rates, and controls segments S24–S31 as either LCD segment drivers or port outputs (Figure 2-18). The LCDC register is used to enable or disable the segment and common lines and, together with the PMGA register, to enable or disable the LCDCL or SYNC output signals (Figure 2-19).

Figure 2-18. LCDM Register Format

LCDM7	LCDM6	LCDM5	LCDM4	LCDM3	LCDM2	LCDM1	LCDM0

LCDM7	LCDM6	S24–S27	S28–S31
0	0	Segment output	Segment output
0	1	Segment output	Port output
1	0	Port output	Segment output
1	1	Port output	Port output

LCDM5	LCDM4	Refresh Rate (Hz)			
		Static	Duplex	Triplex	Quadruplex
0	0	64	32	21	16
0	1	128	64	43	32
1	0	256	128	85	64
1	1	512	256	171	128

LCDM3	LCDM2	LCDM1	LCDM0	Mode
0	X	X	X	Display off (segment lines off)
1	0	0	0	Quadruplex 1/3 bias
1	0	0	1	Triplex 1/3 bias
1	0	1	0	Duplex ½ bias
1	0	1	1	Triplex ½ bias
1	1	0	0	Static

Note: X = don't care.

Figure 2-19. LCDC Register Format

0	LCDC2	0	LCDC0

LCDC2	LCDCL and SYNC Signals
0	Disabled
0	Enabled

LCDC0	LCDM3	COM0–COM3	S0–S23	S24–S31		Bias Output
				Used as a Segment	Used as a Port	
0	X	Display OFF, low-level output	Display OFF, low-level output	Display OFF, low-level output	Outputs bit 0 of RAM (from corresponding display data segment location)	Bias output/high impedance
1	0	Common active, but voltage level is such that the display is OFF (nonselectable)	Segment active, but voltage level is such that the display is OFF (nonselectable mode)	Segment active, but voltage level is such that the display is OFF (nonselectable mode)	Outputs bit 0 or RAM (from corresponding display data segment location)	ON; high level
1	1	ON	ON	ON	Bit 0 of RAM (from corresponding display data segment location) is output	ON; high level

Note: X = don't care.

The display data memory is the area where information displayed by the LCD is stored (as shown in Figure 2-17). It is not part of the LCD controller, but is part of the RAM area contained in the microcontroller (Figure 2-20). Data for the μPD7530x/31x (a member of the 75X family) to be displayed by the 32-segment LCD controller is stored in 32 RAM nibbles at addresses 1E0H–1FFH (Figure 2-20a). Each nibble of the LCD data display area is assigned to a specific segment line (that is, location 1E0H is assigned to segment 0, 1E1H to segment S1, and so forth).

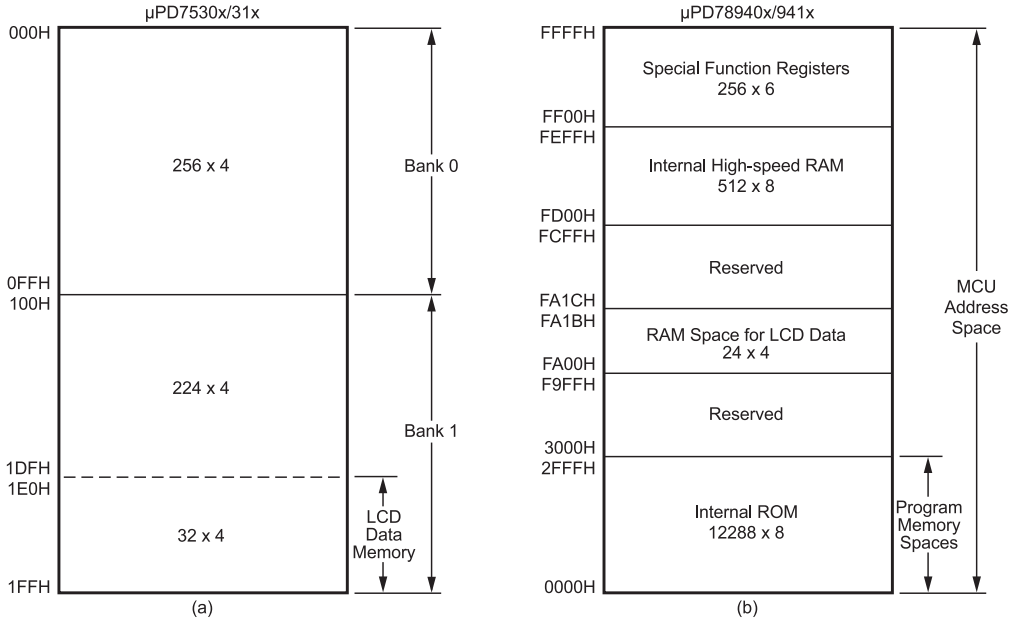
Figure 2-20b shows the typical LCD data display area for the 78K0S and 78K0 families. This area is always part of the microcontroller's 64 KB address space.

The user's program must fill this data area with the correct patterns of 1s and 0s to achieve the desired LCD readout. A one (1) in memory turns an LCD segment ON and a zero (0) turns it OFF. This is true for all NEC Electronics 4- and 8-bit microcontrollers with LCD controllers in

NEC's 4-bit and 8-bit microcontrollers. Note that in the 4-bit families, this area only may be written to in nibbles; writing to it in bytes will cause incorrect data to be displayed on the LCD.

Only bit 0 of the nibble is used in static mode; duplex mode uses bits 0–1, triplex mode bits 0–2, and quadruplex mode bits 0–3. Usage and programming of these bits and the two control registers are covered in Section 4.

Figure 2-20. LCD Data Memory



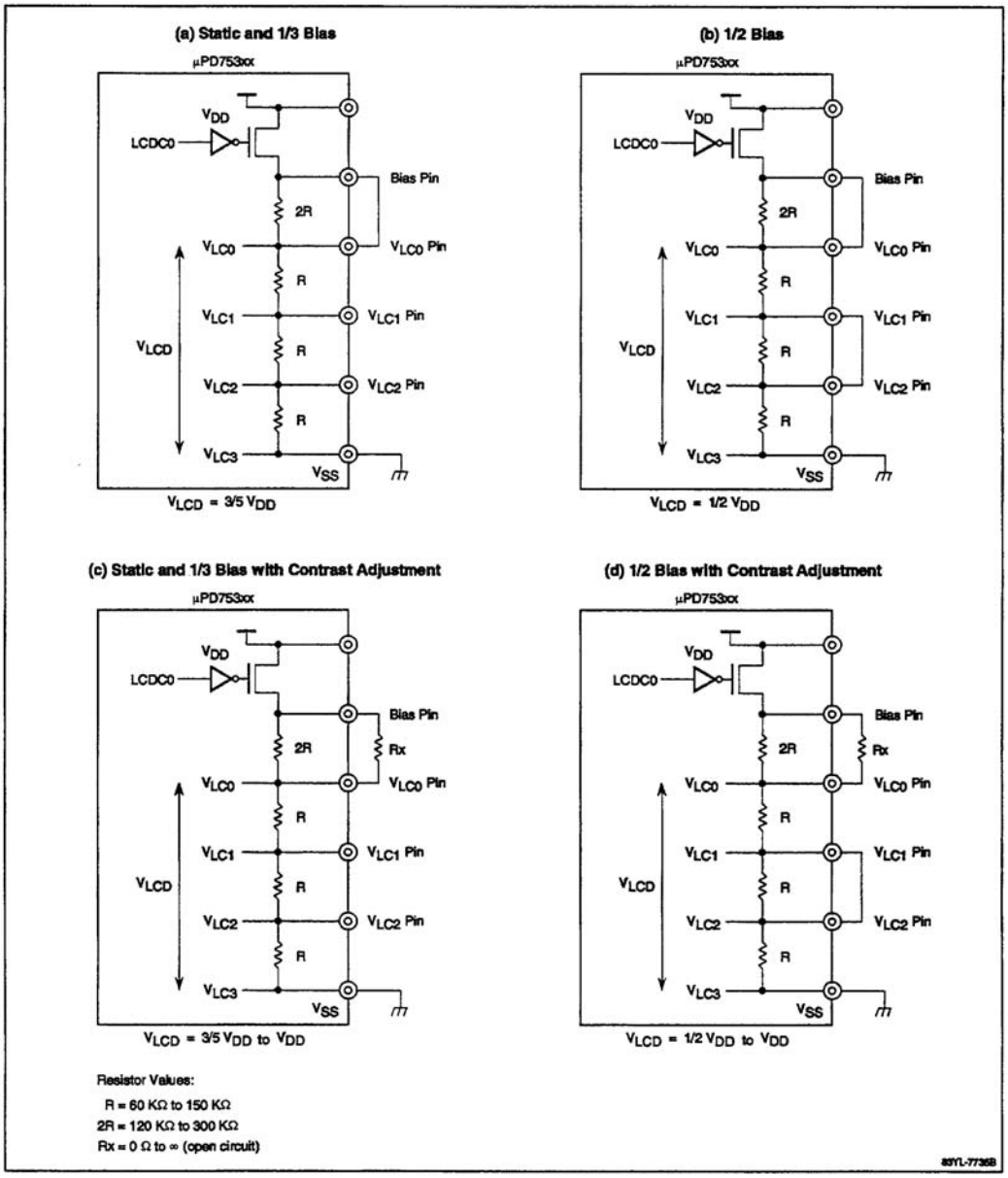
Operation of the LCD controller requires that the LCDM and LCDC registers be programmed and the LCD data display area filled with the correct 0 and 1 patterns. Once programmed, the LCD controller automatically generates all LCD timing waveforms, outputs the data from the LCD data area, and refreshes the display at the preprogrammed rate.

The voltage waveforms for static mode, shown in Figure 2-14 and Figure 2-15, are identical for most LCD controllers, although waveforms required for duplex, triplex, and quadruplex modes may differ among vendors. This manual only contains waveforms for the NEC Electronics LCD controllers.

The resistor ladder, described previously, can be external to the μPD753xx microcontroller (Figure 2-16, Figure 2-27, and Figure 2-32) or it can be provided on-chip as a mask option. Figure 2-21 shows the on-chip resistor ladder mask option for all bias modes. Note that one resistor ladder configuration can be used for all biasing modes using the various jumper wires shown in Figure

2-21. Also, the static mode resistor configuration in Figure 2-21 is different from the external configuration shown in Figure 2-16. Both schemes will work, but the scheme in Figure 2-16 uses fewer resistors than those shown using external resistors. The resistor ladder scheme for the 78K0 and 78K0S microcontrollers is the same. When using an LCD controller with a voltage booster, you must use capacitors in place of resistors, as shown in Figure 2-16c.

Figure 2-21. On-Chip Resistor Ladder Mask Option for μPD753xx LCD Controller



An off-chip resistor (R×) is required to adjust LCD contrast. Adding resistor R× between the BIAS and V_{LC0} pins reduces the maximum voltage at V_{LC0}. This provides a means for adjusting LCD contrast and also for lowering RMS voltage values to match the level required by the display.

2.7 Multiplexed LCDs and Controllers

So far, only static LCDs have been discussed. The LCD controller in a static LCD has a separate segment line for each LCD segment, as shown in Figure 2-22. LCD controllers typically have 5 to 40 segment lines. Hence, if there are only a few LCD segments to control, they can be controlled individually.

These types of LCD controllers and LCDs are called *multiplexed LCD controllers and multiplexed LCDs*. But how would a display having more than 32 segments be controlled with an LCD controller which has 32 segment lines? This is done by having one segment line from the LCD controller control multiple LCD segments.

Multiplexed LCD controllers and LCDs are usually available in duplex, triplex, and quadruplex modes. In a duplex display, one line from the LCD controller controls two LCD segments (Figure 2-23). In triplex mode, one line from the LCD controller controls three LCD segments (Figure 2-24); in quadruplex mode, one segment line controls four LCD segments (Figure 2-25). Static mode requires one common line, duplex mode two, triplex mode three, and quadruplex mode four.

Most LCD controllers can operate in static, duplex, triplex, and quadruplex modes. An LCD controller with 32 segment lines can control up to 128 segments when operated in quadruplex mode. If a seven-segment display is used, 16 digits can be displayed. When one LCD controller segment line controls more than one LCD segment, its waveform becomes more complex. Generally, the more LCD segments controlled by one LCD controller segment line, the more complex the waveform. Furthermore, the basic requirements for RMS and average voltages must always be attained for every type of LCD mode.

Figure 2-22. **Typical Four-Digit, Seven-Segment Static LCD**

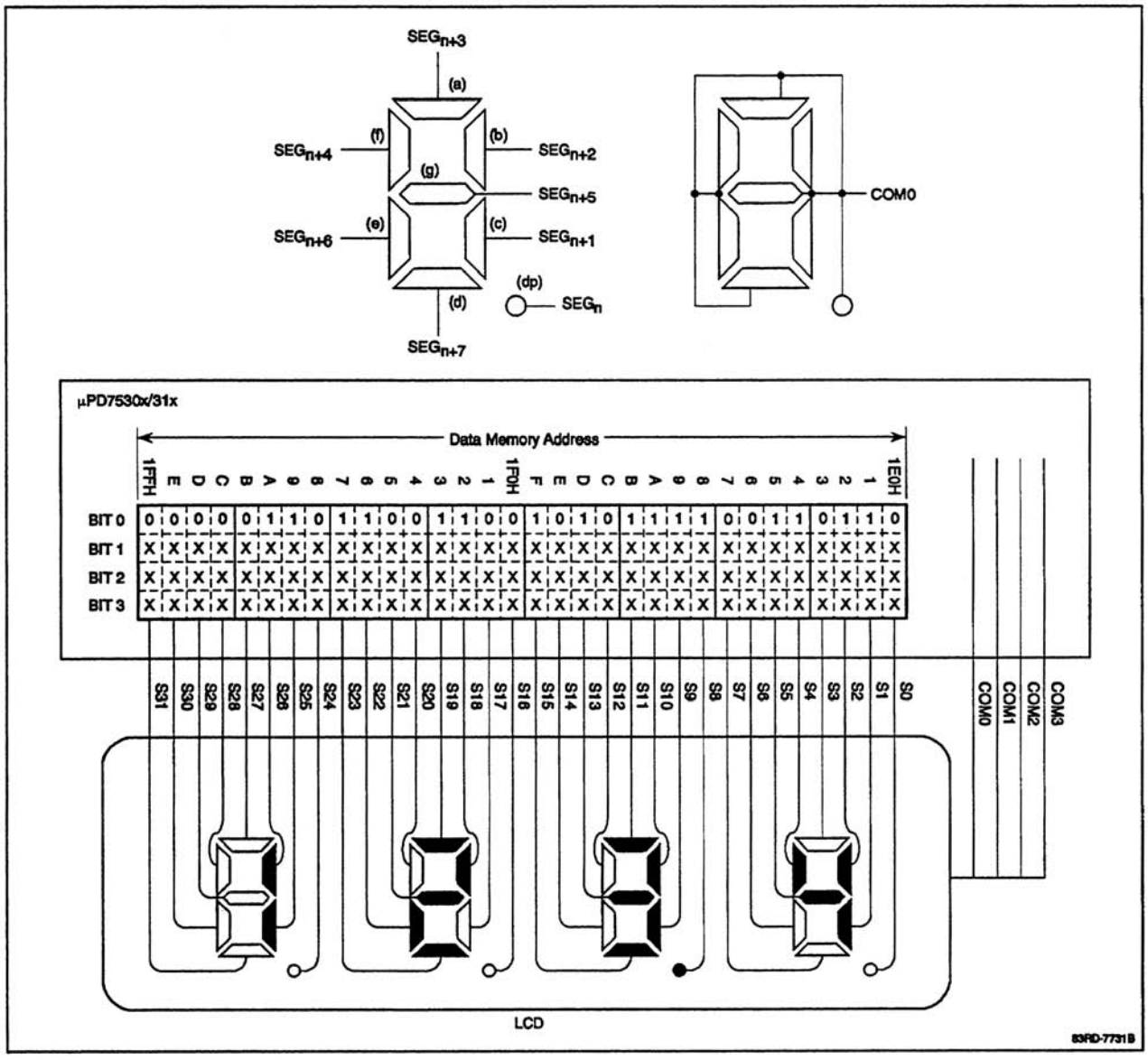


Figure 2-23. **Typical Eight-Digit, Seven-Segment Duplex LCD**

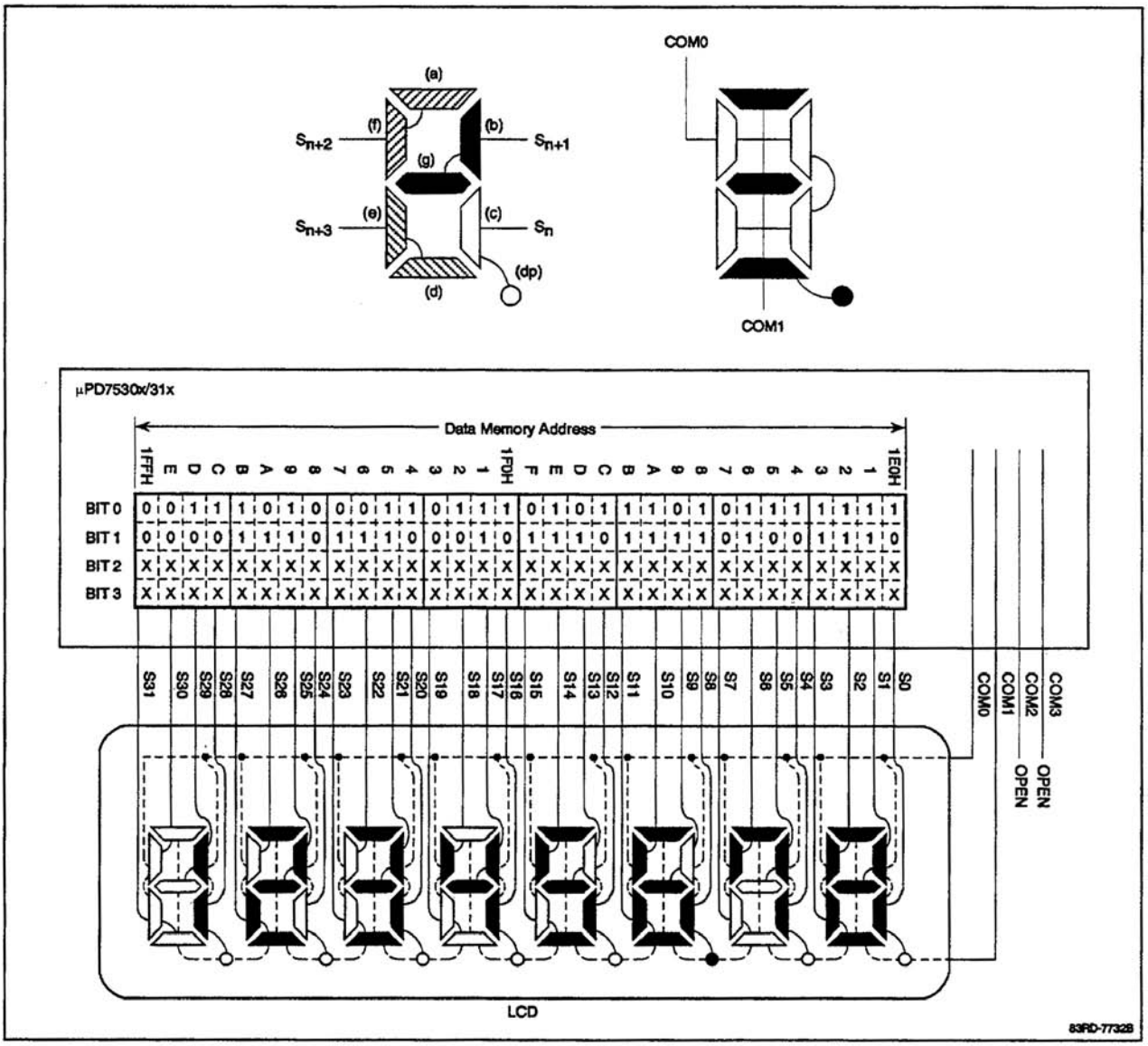


Figure 2-24. **Typical 10-Digit, Seven-Segment Triplex LCD**

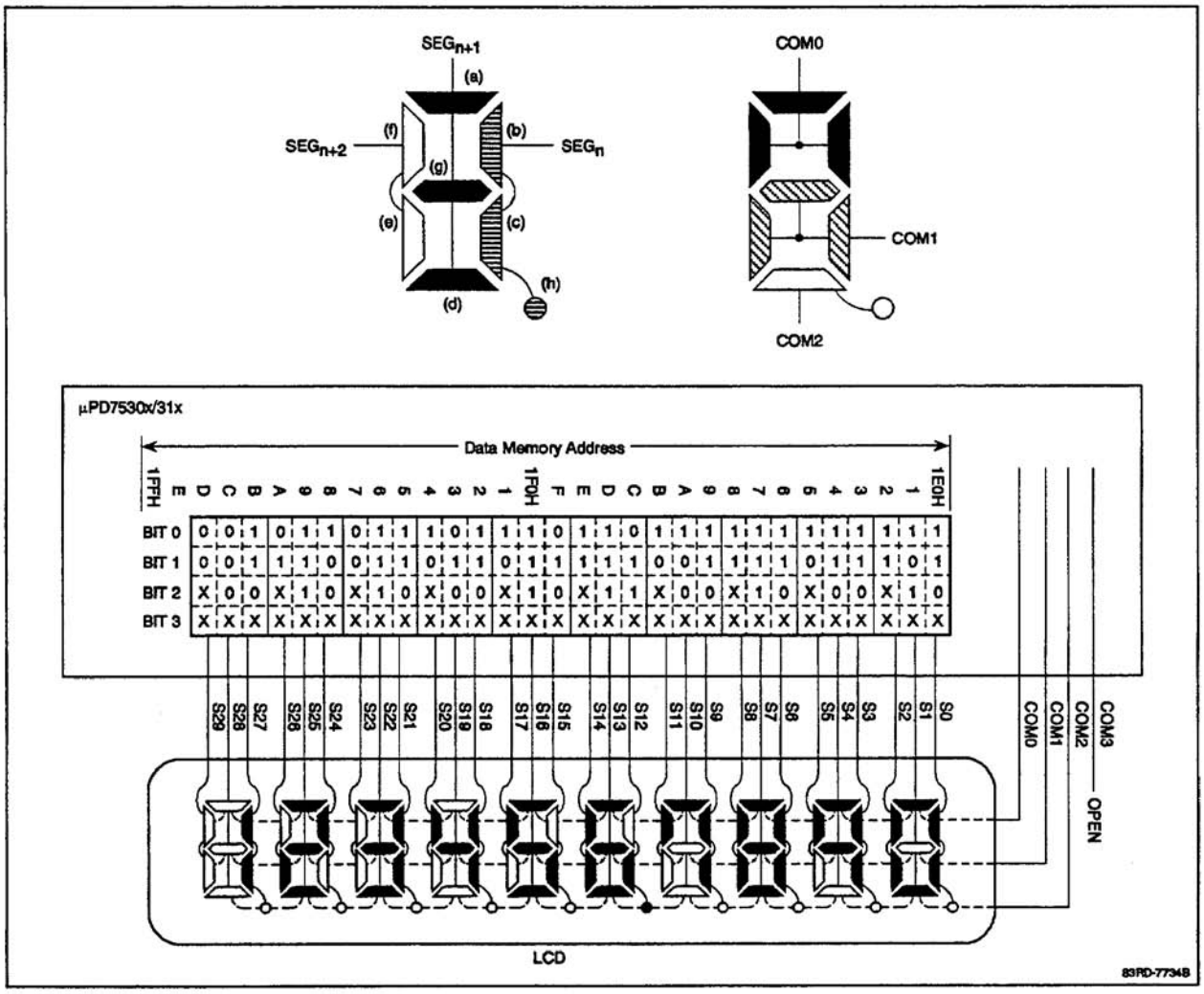
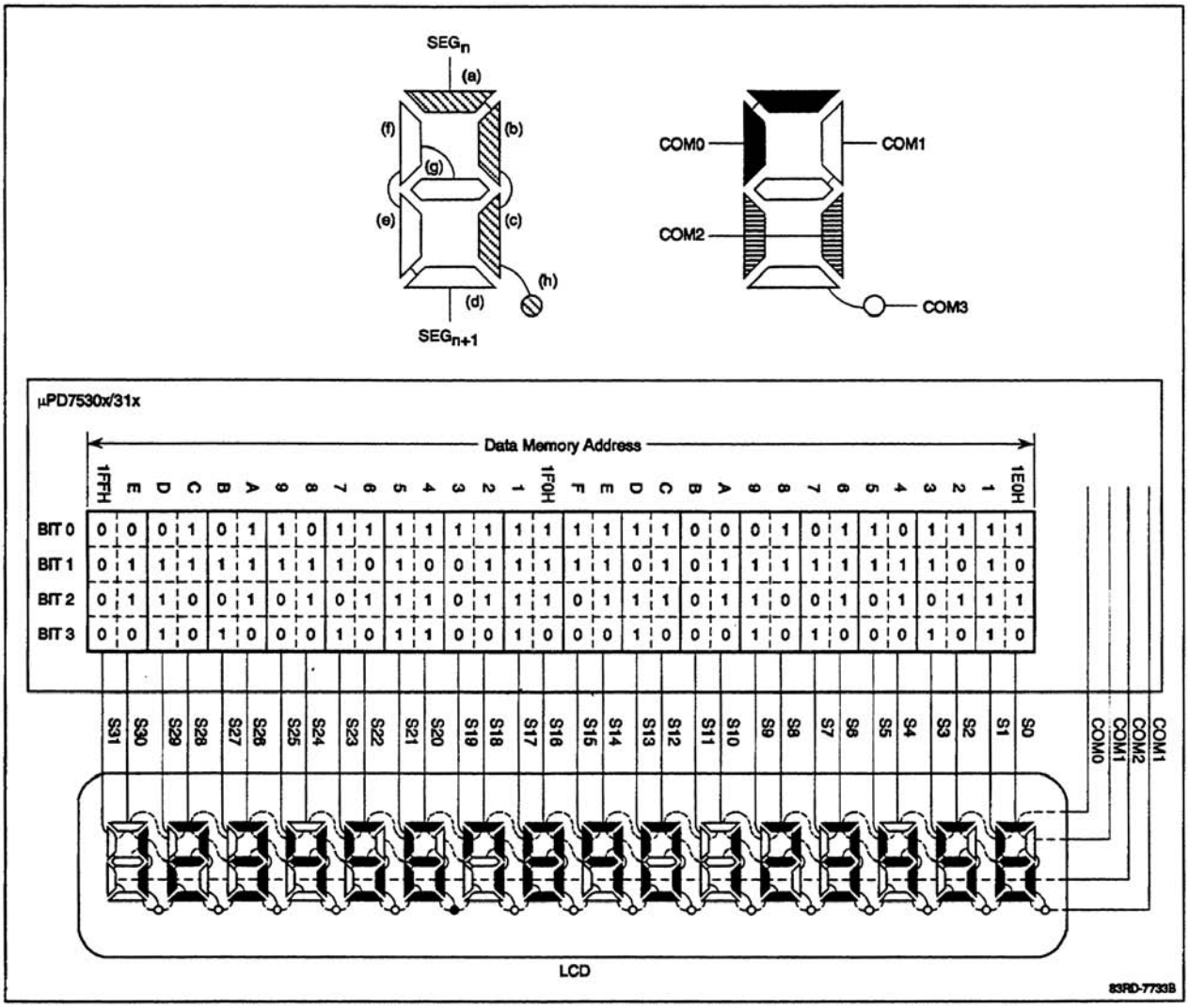


Figure 2-25. **Typical 16-Digit, Seven-Segment Quadruplex LCD**



2.8 Duplex Mode LCDs and LCD Controllers

As discussed earlier, a seven-segment duplex LCD requires two common lines and each of its segment lines controls two LCD segments. For example, to display the number "3" in the third digit (Figure 2-23), SEG_n, (S20), SEG_{n+1} (S21), and COM0 are used to control the vertical portions (segments b and c) of the number 3, while SEG_{n+1} (S21), SEG_{n+2} (S22), SEG_{n+3} (S23), and COM1 are used to control the horizontal portions (segments a, g and d) of the number. Bits 0 and 1 of the display data memory must be used.

The waveforms for the common lines in the duplex mode are shown in Figure 2-26. Note that two common lines are required (COM0 and COM1), and that the common waveforms have three

voltage levels: V_{SS}, V_{LC1} (= V_{LC2}), and V_{LC0}. This is different from the static mode, which has two voltage levels (V_{LC0} and V_{SS}) and one common line (COM0). Also note that the frame period (T_F) has four time slots: t_1, t_2, t_3, and t_4. This is different from static mode, which as two time slots, t_1 and t_2, in its frame period. The three voltage levels and four frame period time slots are necessary to control two LCD segments from one LCD controller segment line.

Figure 2-26. Duplex LCD Timing

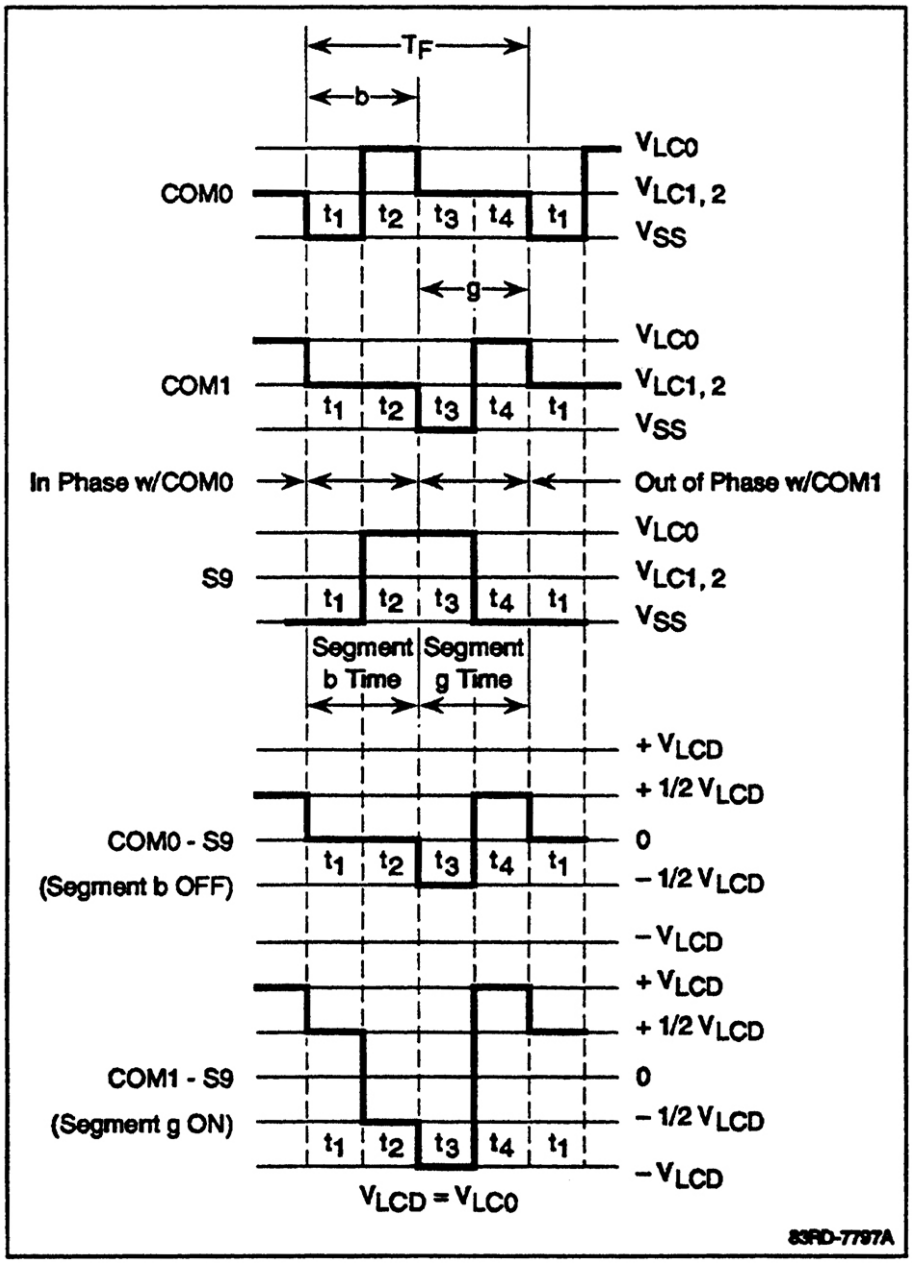


Figure 2-27 shows the 1/2 bias resistor ladder configuration used in duplex and triplex modes. This ladder generates the voltage levels [V_{LC0}, V_{LC1} (= V_{LC2}), and V_{SS}] required by the common and segment lines. Resistor Rx is used to raise or lower the voltage at the resistor ladder taps, raising or lowering the voltages from the segment and common outputs. Varying R× adjusts the RMS voltage values from the LCD controller to match the level required by the display. R× is also used to adjust LCD contrast.

Figure 2-27. 1/2 Bias Resistor Ladder Configuration

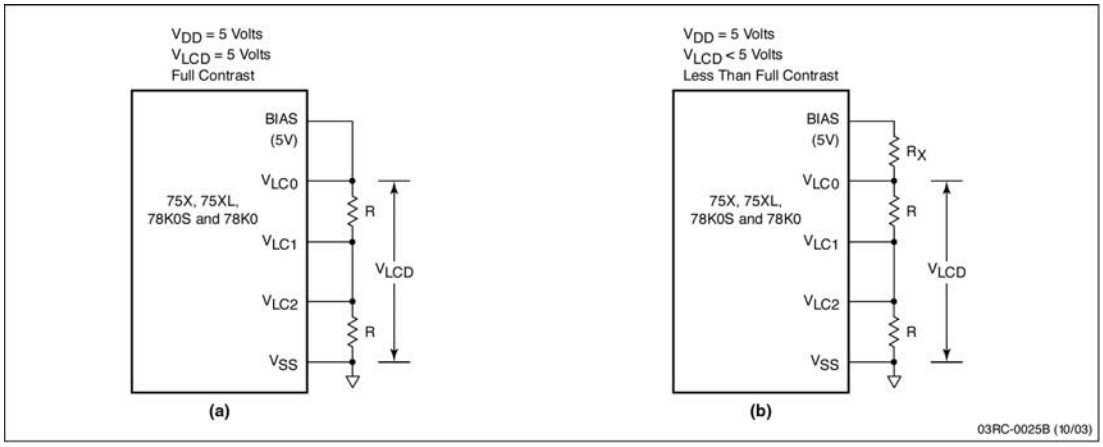


Figure 2-26 shows the timing of the S9 segment output from the LCD controller. Figure 2-23 shows that the LCD controller's S9 segment controls segments b and g by means of S_{n+1} and bits 0 and 1 of location 1E9H in RAM bank 1, as shown in Table 2-3.

Table 2-3. S9 Segment Controls

Control Signals	Operation
Bit 0, S_{n+1}, COM0	Control segment b
Bit 1, S_{n+1}, COM1	Control segment g
Time slots t_1 and t_2 in COM0 and S9	Select the ON/OFF time for segment b
Time slots t_3 and t_4 in COM1 and S9	Select the ON/OFF time for segment g

In this example, segment b is OFF and segment g is ON. During the segment ON/OFF selection times (in other words, t_1 and t_2 for segment b, and t_3 and t_4 for segment g), the common voltage always swings between its lowest (V_{SS}) and highest (V_{LC0}) voltages. This swing provides a mechanism for the segment voltage to be in, or out of phase with the common, thereby

producing a COM*n*–SEG*n* voltage of either 0 volts RMS (OFF) or a swing from V_{SS} to V_{LC0} volts = V_{LC0} volts RMS (ON) during the ON/OFF times slots.

As covered earlier, a segment is turned ON when the common and segment line waveforms are out of phase with each other, and OFF when the common and segment lines are in phase with each other. During the t_1 and t_2 time slots, segment line S9 is in phase with COM0 (Figure 2-26 and Figure 2-28a). This turns segment b OFF and the voltage across segment b is shown by waveform COM0–S9. During time slots t_3 and t_4, segment line S9 is out of phase with COM1. This turns segment g ON and the voltage across segment g is shown by waveform COM1–S9.

The maximum amplitude of COM0–S9 is 1/2 V_{LCD}, and the maximum amplitude of COM1–S9 is V_{LCD}. The RMS voltage across segment b is smaller than the voltage across segment g. The result is that segment b is OFF and segment g is ON. Observe that the average voltage across both b and g is 0 volts. This meets the average voltage requirement of LCDs outlined earlier in this application manual. If it were desired to have both segments b and g ON, S9 would have to be in phase with COM0 and COM1 (Figure 2-28b).

Figure 2-28. Multiplexing Segments b and g in Duplex Mode

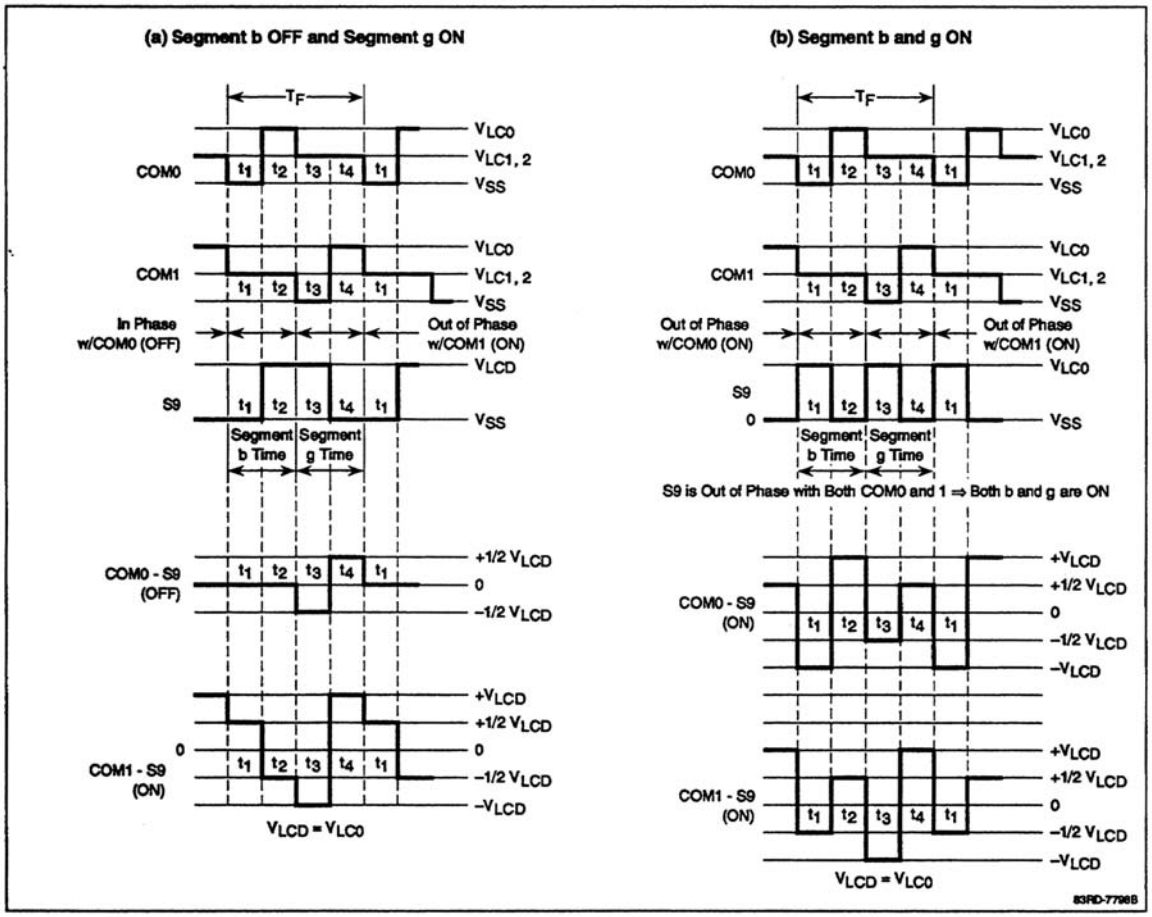


Figure 2-29 shows the RMS voltage calculation for segment b OFF and segment g ON. The RMS voltages across segment b (V_{OFF}) and segment g (V_{ON}) are:

V_{OFF} (segment b) = 0.35 V_{LCD} volts RMS, where V_{LCD} = supply voltage (typically 3 to 5 volts).

V_{ON} (segment g) = 0.79 V_{LCD} volts RMS, where V_{LCD} = supply voltage (typically 3 to 5 volts).

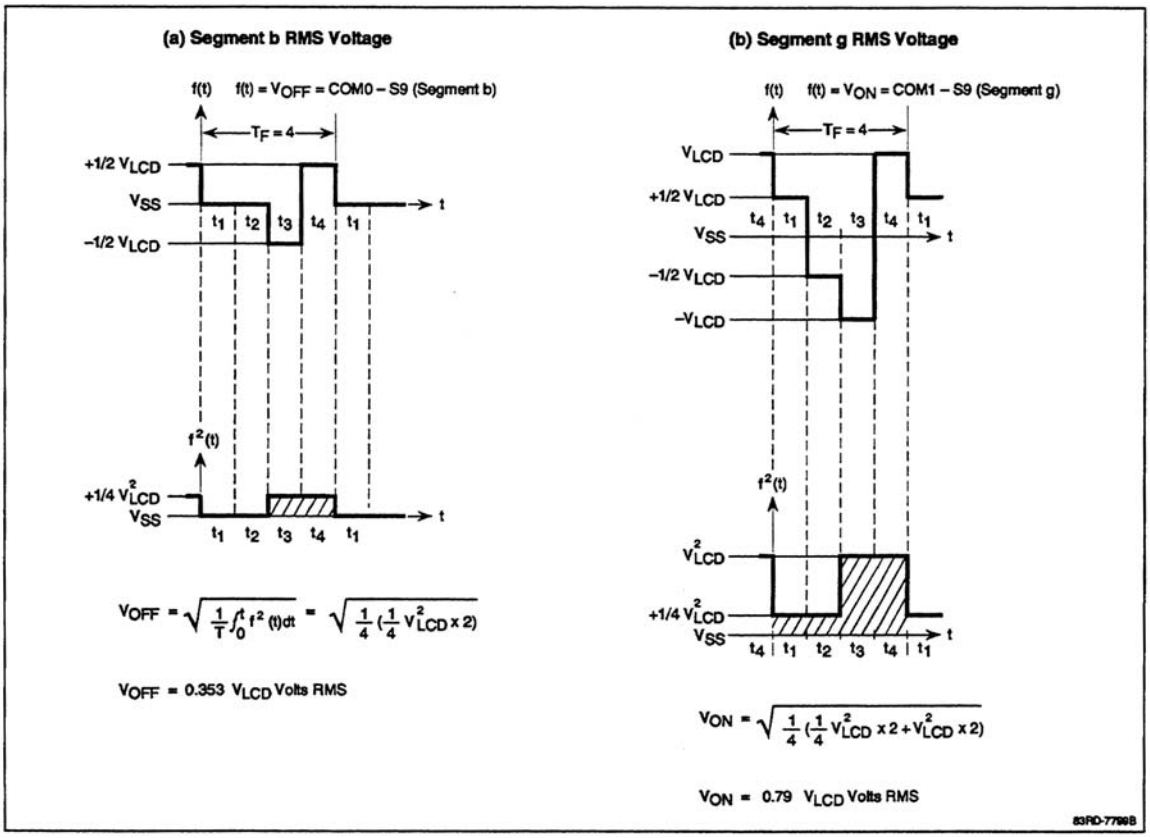
The RMS value of V_{ON} is more than twice that of V_{OFF}. Note that the RMS OFF value is not 0 volts and the RMS ON value is not 5 volts, as is the case in static mode. If V_{LCD} = 5 volts, then

$V_{OFF} = 0.35 \times (5) = 1.75$ volts RMS and $V_{ON} = 0.79 \times (5) = 3.95$ volts RMS

This means that the LCD must have an RMS OFF voltage specification of more than 1.75 volts and an RMS ON voltage of less than 3.95 volts. The difference between the RMS ON and OFF voltages is:

V_{DIFF} duplex = 3.95 – 1.75 = 2.2 volts RMS (2.2)

Figure 2-29. **RMS Calculation for Segment b OFF and Segment g ON**



2.9 Triplex Mode LCDs and LCD Controllers

An LCD controller can be configured to use either 1/2 or 1/3 bias in triplex mode. Figure 2-30 shows triplex mode timing with 1/2 bias; Figure 2-31 shows triplex mode timing with 1/3 bias. The 1/2 bias resistor ladder is shown in Figure 2-27 and the 1/3 bias resistor in Figure 2-32. One LCD controller segment line controls three LCD segments in the triplex mode. This means that six time slots (t_1 through t_6) are required (two time slots per segment). Figure 2-24 shows a triplex mode LCD used with the μPD7530x/31x LCD controller. Segment S12 (which corresponds to segment S12 in Figure 2-30 and Figure 2-31) controls segments b, c, and h of the sixth LCD digit using SEG*n* and COM0, COM1 and COM2. This displayed sixth digit is the number six. Segment b is OFF and segments c and h are ON. The 1/2 or 1/3 bias modes can control the same segments, but their waveforms are different.

Figure 2-30. **Triplex Mode Timing with 1/2 Bias**

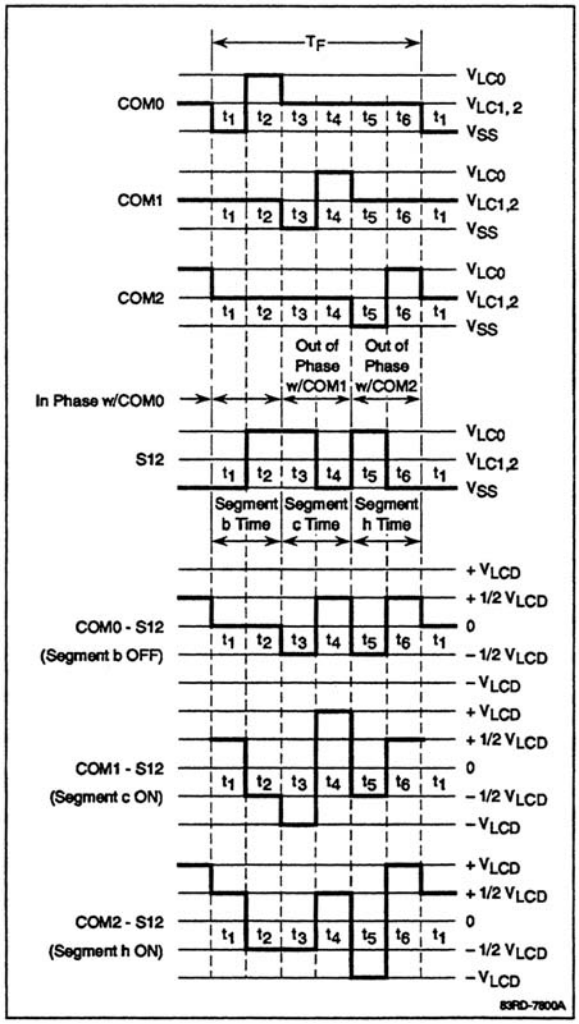


Figure 2-31. **Triplex Mode Timing with 1/3 Bias**

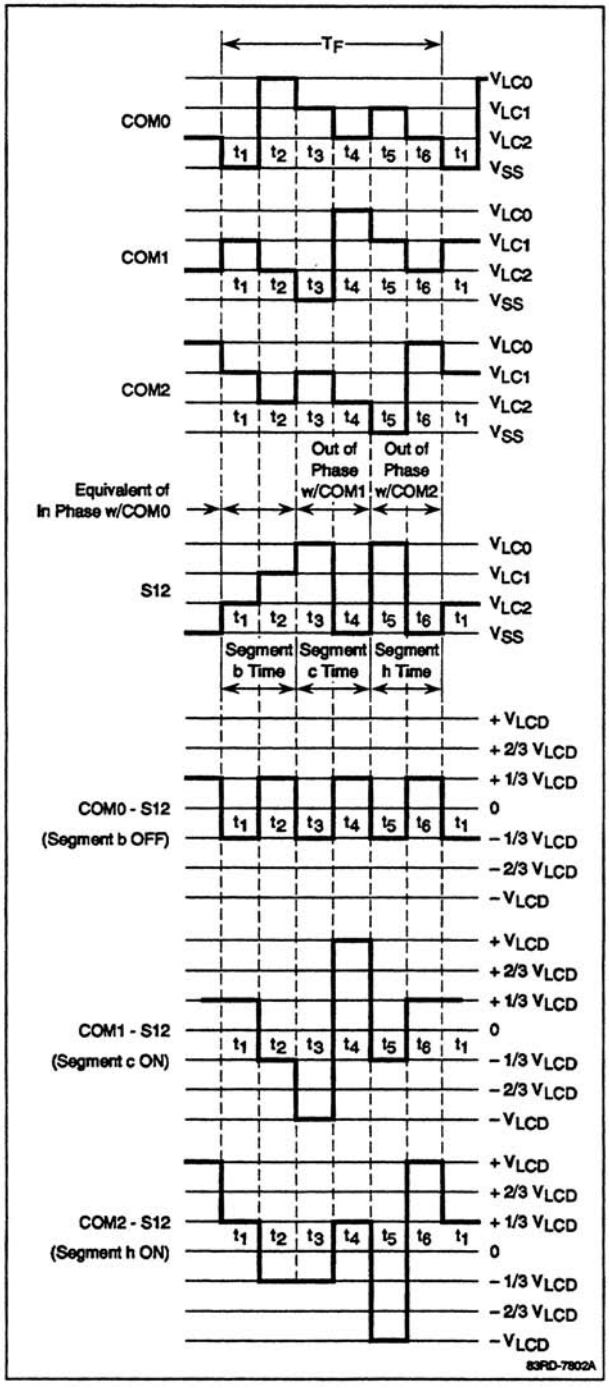
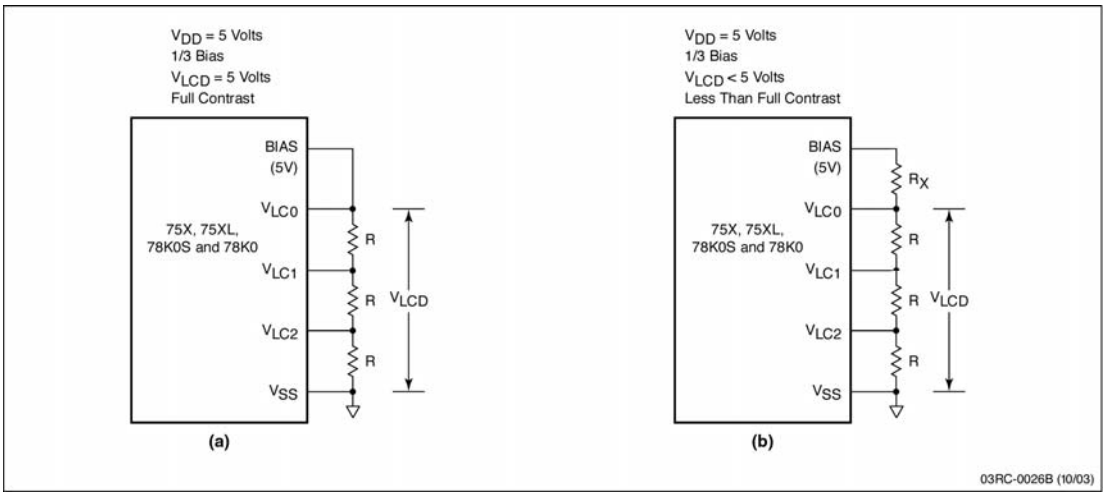


Figure 2-32. 1/3 Bias Register Ladder Configuration



The timing for a 1/2 bias configuration turns display segment b OFF because S12 is in phase with COM0 during the t_1 and t_2 time slots (Figure 2-30). Segment c is ON because S12 is out of phase with COM1 during the t_3 and t_4 time slots. Segment h is ON because S12 is out of phase with COM2 during t_5 and t_6. The calculations for 1/2 bias RMS OFF and ON voltages in triplex mode (Figure 2-32) are:

$V_{OFF} = 0.4\ V_{LCD}$ volts RMS and if $V_{LCD} = 5$ volts, then $V_{OFF} = 0.4 \times (5) = 2.00$ volts RMS

$V_{ON} = 0.7\ V_{LCD}$ volts RMS and if $V_{LCD} = 5$ volts, then $V_{ON} = 0.7 \times (5) = 3.50$ volts RMS

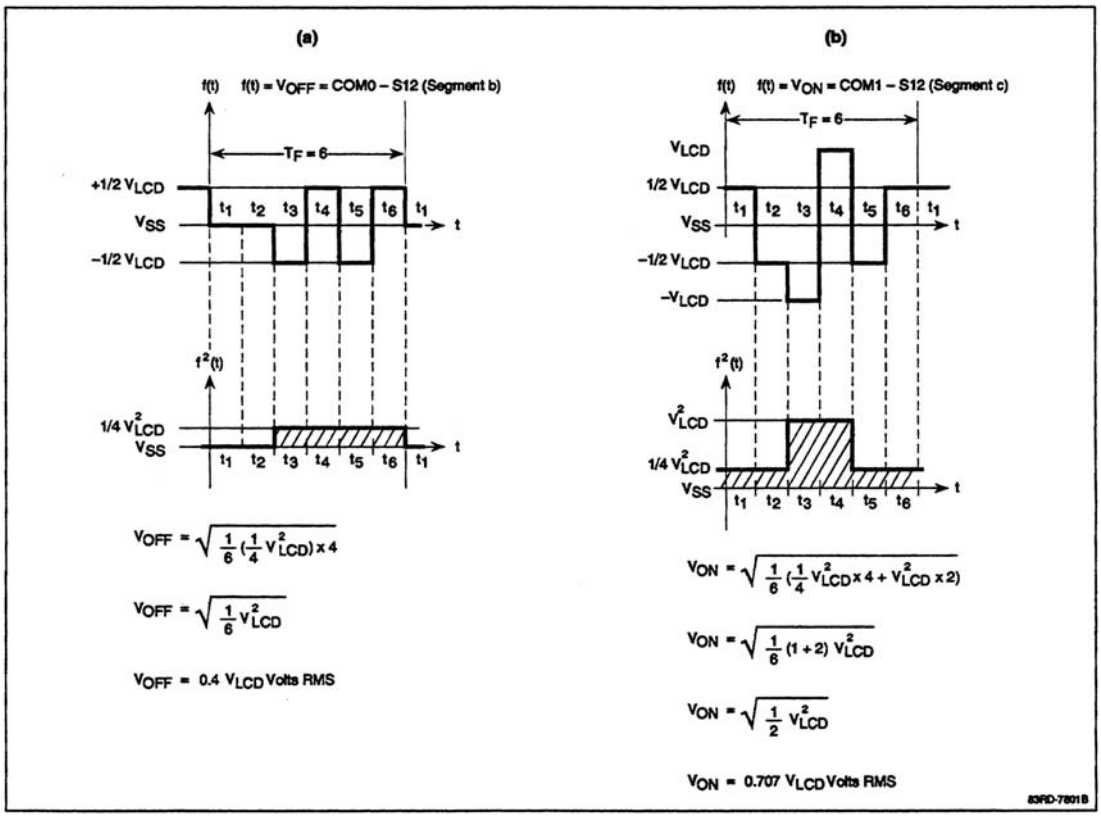
Consequently, the LCD must have an RMS OFF voltage specification of ≥2.0 volts and an RMS ON voltage of ≤3.5 volts. Table 2-4 is a typical specification for a multiplexed LCD. V_{th} and V_{sat} are referred to as V10 and V90, respectively.[28] V_{10} is the 10% display ON (light area) and V_{90} is the 90% display ON (black area). If $V_{LCD} = 5$ volts, then fluid type 2 can be used because V10 is 2.2 volts RMS and the LCD controller outputs 2.0 volts RMS. V_{90} is 3.1 volts RMS and the LCD controller outputs 3.5 volts RMS. However, fluid types 3 and 8 cannot be used when $V_{LCD} = 5$ volts because V_{OFF} is too large. By adding resistor R_x between BIAS and V_{LC0} such that $V_{LC0} = 3.0$ volts RMS, then:

$V_{OFF} = 0.4 \times (3) = 1.20$ volts RMS

$V_{ON} = 0.7 \times (3) = 2.10$ volts RMS

[28] Standisch

Figure 2-33. **RMS Voltage Calculation for 1/2 Bias LCD in Triplex Mode**



Although these values would allow fluid type 3 to be used, they may lack the necessary ON voltage for fluid type 8. Also note that the maximum DC voltage is 100 mV.

The difference between the RMS ON and OFF voltages at V_{LCD} = 5 volts is:

V_{DIFF} 1/2 bias triplex = 3.5 – 2.0 = 1.5 volts RMS. (2.3)

Table 2-4. Typical Specifications for Multiplexed LCDs[29]

Parameter	Fluid Type 2 Note 1			Fluid Type 3 Note 2			Fluid Type 8 Note 1			Unit
	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
DC drive component allowable	–	–	100	–	–	100	–	–	100	mV
Operating frequency range	30	–	250	30	–	250	30	–	250	Hz
Current consumption Note 3	–	3	5	–	3	5	–	3	5	$\mu A/cm^2$
Capacitance	–	1200	–	–	1200	–	–	1200	–	pF/cm^2
V_{10}	–	2.2	–	–	1.2	–	–	1.8	–	V_{RMS}
V_{90}	–	3.1	–	–	1.7	–	–	2.6	–	V_{RMS}
dv/dt	–	–9	–	–	–8	–	–	–6	–	mV/°C
Typical response times Note 3										
♦ T_{ON} at 25°C	–	15	–	–	45	–	–	15	–	ms
♦ T_{OFF} at 25°C	–	30	–	–	55	–	–	30	–	ms
♦ T_{ON} at 0°C	–	50	–	–	250	–	–	50	–	ms
♦ T_{OFF} at 0°C	–	150	–	–	350	–	–	150	–	ms
Operating temperature range Note 4	–20	–	80	–10	–	55	–20	–	80	°C
Storage temperature range	–55		80	–55		55	–55		80	°C

NOTES:

1. Fluid types 2 and 8 are wide-temperature fluids designed for applications using multiplex LCDs.
2. Fluid type 3 is a low-threshold voltage fluid designed for instrument-grade applications.
3. Test voltage = 6 volts RMS at 60 Hz.
4. Minimum operating temperature is the temperature at which one update per second is possible. Temperature compensation may be required to realize satisfactory operation over the entire operating temperature range.

[29] Williams, "Liquid Crystals for Electronic Devices"

The timing for 1/3 bias configuration turns segments c and h ON because S12 is out of phase with COM1 during t_3 and t_4, and S12 is out of phase with COM2 during t_5 and t_6 (Figure 2-31). The OFF method is different in the 1/3 bias mode. A segment is turned OFF by having S12 = V_{LC2} during t_1 and S12 = V_{LC1} during t_2 (see later sections for more about this). The calculations for RMS OFF and ON voltages in triplex mode are shown below and illustrated in Figure 2-34.

V_{OFF} = 0.33 V_{LCD} volts RMS and if V_{LCD} = 5 volts, then V_{OFF} = 0.33 × (5) = 1.65 volts RMS

V_{ON} = 0.64 V_{LCD} volts RMS and if V_{LCD} = 5 volts, then V_{ON} = 0.64 × (5) = 3.20 volts RMS

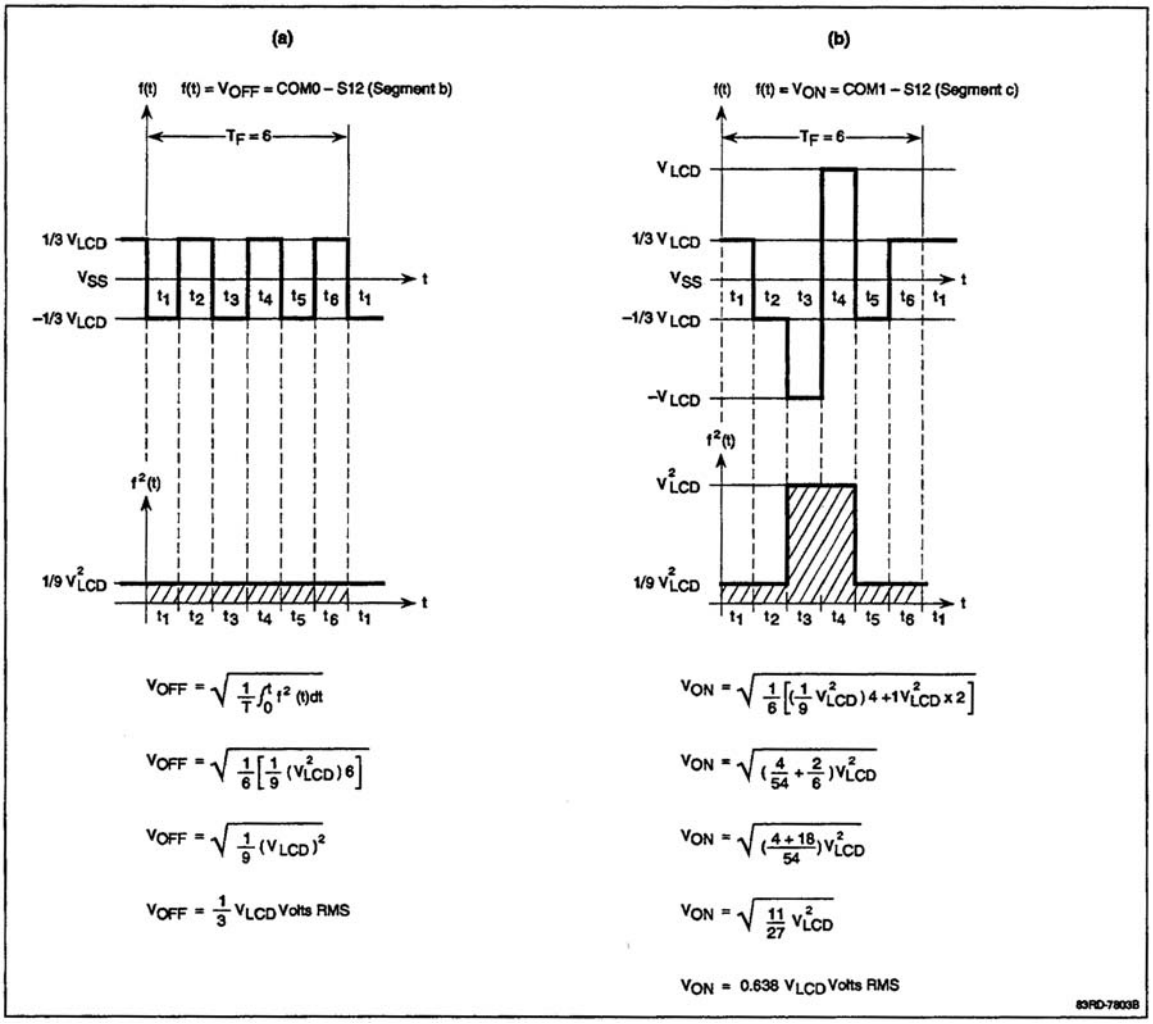
Consequently, the LCD must have an RMS OFF voltage specification of ≥1.65 volts and an RMS ON voltage of ≤3.2 volts. Fluid types 2 and 8 can be used with V_{LCD} = 5 volts (Table 2-4), while fluid type 3 cannot. If R× is added such that V_{LC0} = 3 volts RMS, then:

V_{OFF} = 0 × (3) = 0.99 volts RMS and V_{ON} = 0.64 × (3) = 1.92 volts RMS

These values would allow fluid type 3 to be used. At V_{LCD} = 5 volts, the difference between the RMS ON and OFF voltages would be:

V_{DIFF} 1/3 bias triplex = 3.2 − 1.65 = 1.55 volts RMS (2.4)

Figure 2-34. RMS Drive Voltage Calculations for 1/3 Bias LCD in Triplex Mode



The rest of this section explains the OFF method used in the 1/3 bias triplex mode. The problem with turning a segment OFF using the in-phase approach (used exclusively until now) is best illustrated by Figure 2-35. S12 turns OFF segments b, c and h using the in-phase method. The RMS value for the COM0–S12 waveform is:

$V_{OFF} = 0.5\ V_{LCD}$ volts RMS.

In this in-phase method, the difference between the OFF and ON voltages is reduced from:

$0.33\ V_{LCD} - 0.64\ V_{LCD}$ to $0.5\ V_{LCD} - 0.64\ V_{LCD}$.

If V_{LCD} = 5 volts, then:

$V_{OFF} = 0.5 \times (5) = 2.5$ volts RMS

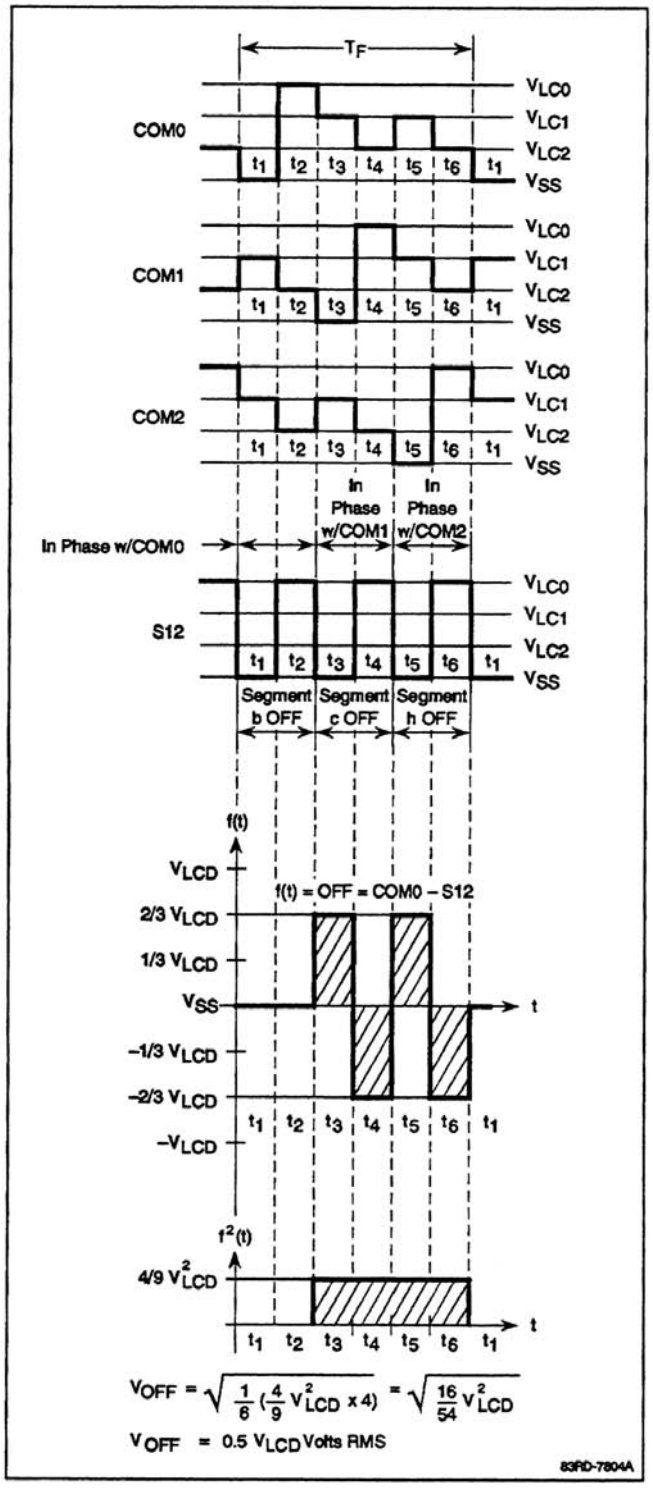
$V_{ON} = 0.64 \times (5) = 3.2$ volts RMS

The LCD's RMS OFF voltage must be ≥2.5 volts and the RMS ON voltage ≤3.2 volts. The difference between the two at V_{LCD} = 5 volts is:

V_{DIFF} 1/3 bias triplex = 3.2 − 2.5 = 0.70 volts RMS.

This difference severely restricts the possibility of finding an LCD with 1/3 bias triplex mode and makes it impossible to make one with quadruplex mode.

Figure 2-35. **1/3 Bias Timing and RMS Voltage Calculation in Triplex Mode**



2.10 Quadruplex Mode LCDs and LCD Controllers

The quaduplex mode uses a 1/3 bias configuration. One LCD controller segment line controls four LCD segments. Consequently, eight time slots (t_1 to t_8) are required (two time slots per segment). Figure 2-25 shows a quadruplex mode LCD used with the μPD7530x/31x LCD controller and Figure 2-36 shows the timing for quadruplex mode. Segment S20 corresponds to segment S20 in Figure 2-25, in which segment S20 controls segments a, b, c, and h of the sixth LCD digit using SEG*n* and COM0–3. The sixth digit is the number 6. S20 has segments a, c, and h ON and segment b OFF.

Only the waveforms for segments a and b are shown in Figure 2-36. Segment a is ON because S20 is out of phase with COM0 during t_1 and t_2. Segment b is turned OFF by having S20 = V_{LC2} during t_3 and V_{LC1} during t_4. The reason for this is explained later. The RMS voltage for the OFF segment is 0.33 V_{LCD} because the OFF waveform (COM1–S20) is a square wave of amplitude 0.33 V_{LCD}. The calculations for the 1/3 bias RMS ON and OFF voltages are illustrated in Figure 2-37 and shown below.

V_{OFF} = 0.33 V_{LCD} volts RMS and if V_{LCD} = 5 volts, then V_{OFF} = 0.33 × (5) = 1.665 volts RMS

V_{ON} = 0.58 V_{LCD} volts RMS and if V_{LCD} = 5 volts, then V_{ON} = 0.58 × (5) = 2.885 volts RMS

This requires an LCD with an RMS OFF voltage of ≥1.665 volts and an RMS ON voltage of ≤2.885 volts. As shown in Table 2-4, fluid type 8 definitely can be used. Fluid type 2 also can be used; however, the RMS ON voltage generated by the LCD controller would be less than the V90 specification, which may be a problem over the full operating temperature range of the particular application. This topic is covered more fully in Section 3.

Figure 2-36. **Quadruplex Mode Timing**

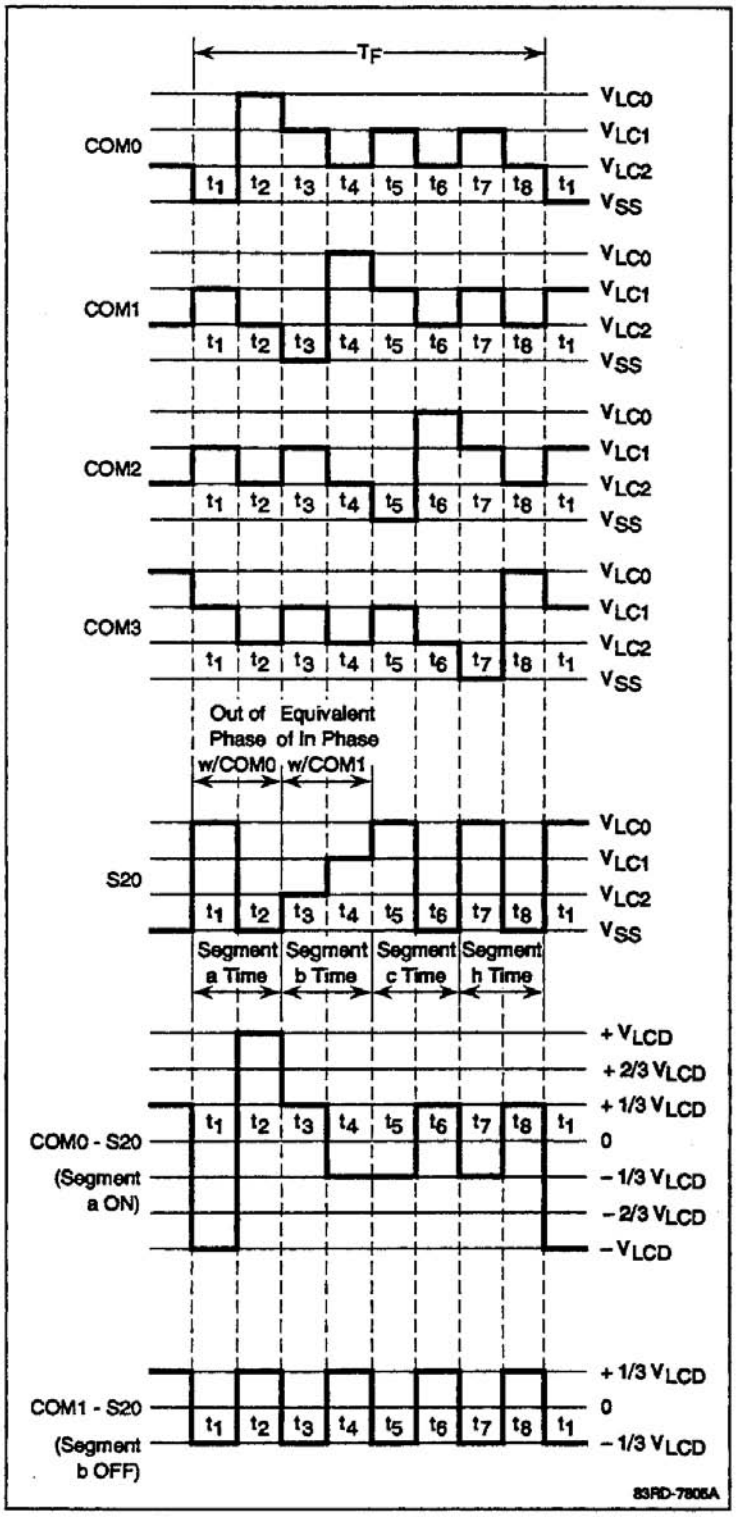
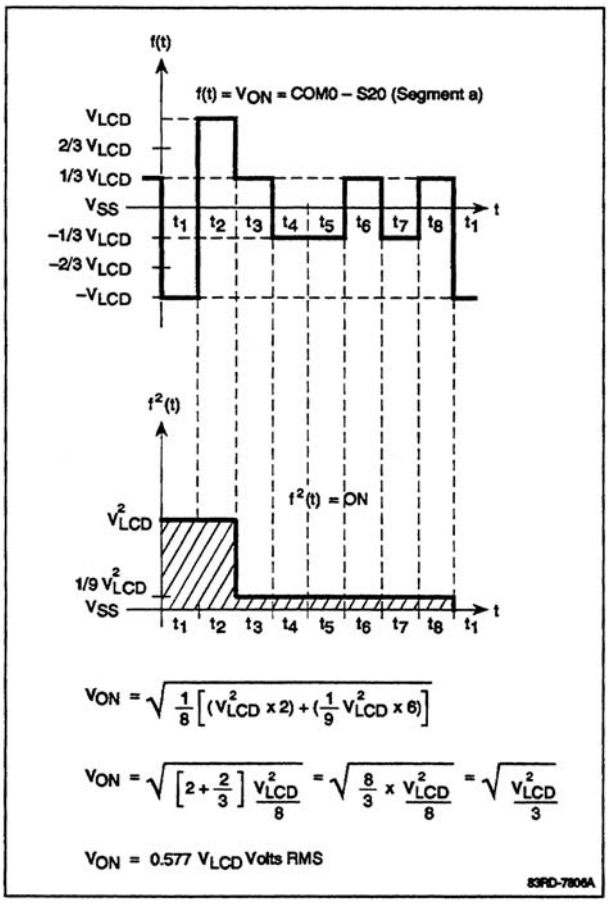


Figure 2-37. Quadruplex Mode LCD RMS ON Voltage Calculation



The difference between the RMS ON and OFF voltages at V_{LCD} = 5 volts is:

V_{DIFF} quadruplex = 2.88 – 1.66 = 1.22 volts RMS (2.5)

Equations 2.1 through 2.5 indicate that the more segments controlled by an LCD controller segment line, the smaller the difference between the V_{ON} and V_{OFF} voltages.

V_{DIFF} static	5 – 0 = 5.0 volts RMS	(2.1)
V_{DIFF} duplex	3.95 – 1.75 = 2.2 volts RMS	(2.2)
V_{DIFF} 1/2 bias triplex	3.5 – 2.0 = 1.5 volts RMS	(2.3)
V_{DIFF} 1/3 bias triplex	3.2 – 1.65 = 1.55 volts RMS	(2.4)
V_{DIFF} quadruplex	2.88 – 1.66 = 1.22 volts RMS	(2.5)

Going beyond quadruplex mode (in other words, controlling five or more LCD segments from one LCD controller segment time) would provide an even lower V_{DIFF} voltage. Hence, as the

multiplexing number increases, the difference between the V_{ON} and V_{OFF} voltage decreases, which makes it more difficult to design an LCD with good contrast.

The following example illustrates why a quadruplex segment cannot be turned OFF using the in-phase technique. Figure 2-38 shows segment line S20 with all four segments turned OFF using the in-phase technique. The bottom of Figure 2-38 shows the voltage across segment "a" (COM0–S20). Figure 2-39 shows the calculation of the RMS OFF voltage:

$V_{OFF} = 0.58\ V_{LCD}$ volts RMS.

The RMS OFF value is the same as the RMS ON value. The segment cannot be controlled, and the OFF technique described earlier must be used.

Figure 2-38. **Quadruplex Mode Timing OFF Voltage Using In-Phase Technique**

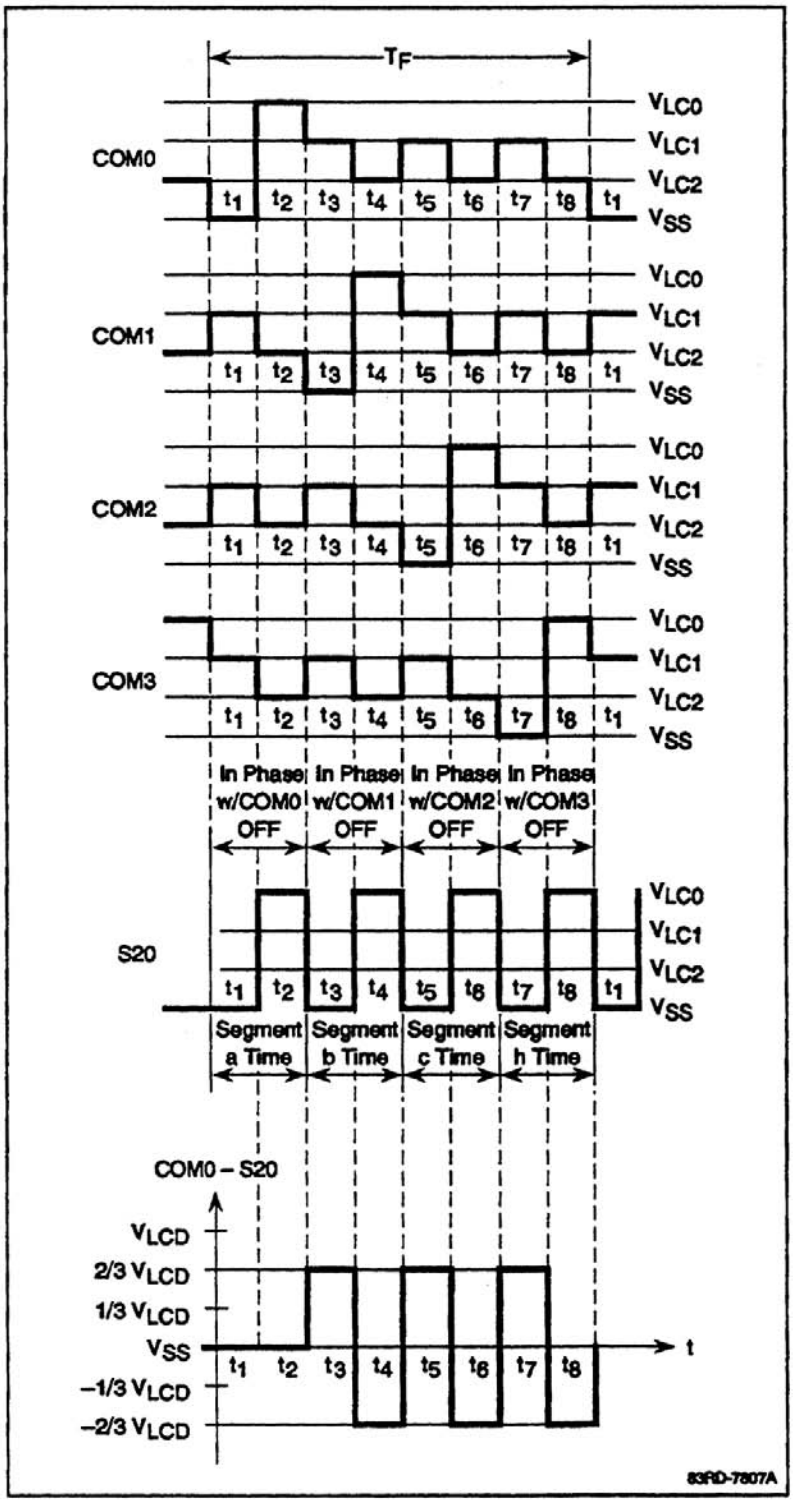
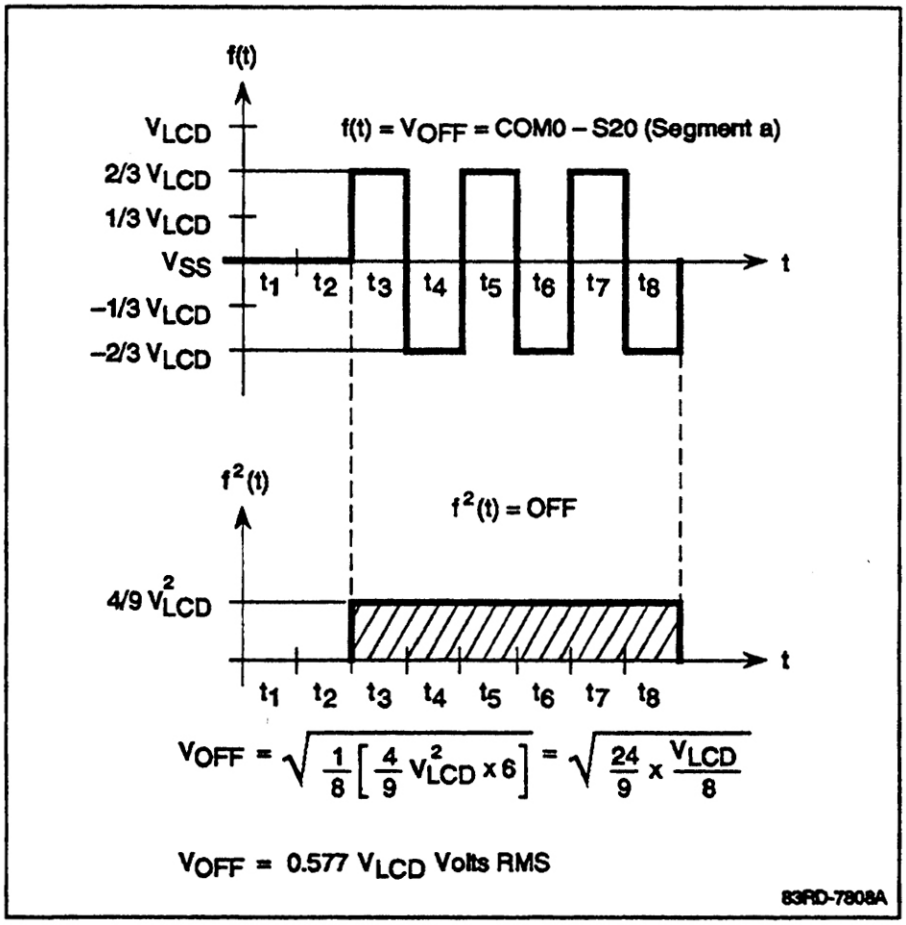


Figure 2-39. **LCD RMS OFF Voltage for Quadruplex Mode Using In-Phase Technique**



2.11 Summary of Operating Voltages for an NEC Electronics LCD Controller

Table 2-5 contains a summary of the RMS operating voltages for the NEC Electronics μPD75x LCD controller in static, duplex, triplex, and quadruplex modes.

Table 2-5. Summary of RMS Voltages for NEC Electronics' LCD Controller

Parameter	Static	Duplex ½ Bias	Triplex ½ Bias	Triplex 1/3 Bias	Quadruplex 1/3 Bias
V_{OFF} equation	V_{SS}	0.35 V_{LCD}	0.4 V_{LCD}	0.33 V_{LCD}	0.33 V_{LCD}
V_{ON} equation	V_{LCD}	0.79 V_{LCD}	0.7 V_{LCD}	0.64 V_{LCD}	0.58 V_{LCD}
V_{OFF} at 5V	0	1.75	2.00	1.65	1.66
V_{ON} at 5V	5	3.95	3.50	3.20	2.90
V_{OFF} at 3V	0	1.05	1.20	0.99	0.99
V_{ON} at 3V	3	2.37	2.10	1.92	1.74

Note: All values are in volts RMS.

3. LCD VISUAL AND AMBIENT FACTORS

This part of the application manual is intended to show designers how to include such factors as contrast, display viewing angle, operating temperature range, operating frequency, and display ON/OFF times in their design. It will also tell the designer how to interpret LCD manufacturer's specifications and what to do about some important LCD specifications, if they are missing.

In order to complete the hardware design, the following factors must also be discussed:

- Data sheet specifications
- Ambient temperature
- Contrast fundamentals
- LCD operating voltage, contrast and viewing angle
- Effect of temperature on contrast
- LCD ON/OFF response time
- LCD operating frequency and current
- Dimming and ghosting

These issues are reviewed from a specification and analysis point of view. However the best way to evaluate these factors is by extensive laboratory tests and visual inspection. Empirical data is the best.

3.1 Specifications

A complete set of specifications is necessary to complete an LCD design. A good example of LCD specifications is shown in Table 2-1. It contains nearly all of the necessary information for a static LCD, as follows:

- Operating voltage
- Allowable DC drive component
- Operating frequency range
- Segment capacitance
- ON/OFF voltage thresholds
- ON/OFF response times
- Contrast ratio

- Operating and storage temperature
- Expected life
- Viewing angle

For a multiplexed LCD (meaning one in duplex, triplex, or quadruplex modes), the data sheet contains most of the important specifications. An operating voltage vs. contrast specification is missing and should be obtained from the LCD manufacturer, who will probably recommend an optimal fluid type for the application and furnish the appropriate operating voltage vs. contrast information. This is an example where it is prudent for the design engineer to contact the LCD manufacturer for this information.

3.1.1 Operating Temperature

The ambient temperature affects an LCD in many ways. In addition to the operating and storage temperature of the LCD, the temperature also affects the LCD contrast, general appearance, response time, and viewing angle. This fact is reflected in the number of temperature specifications shown in Table 2-1. Since ambient temperature affects LCDs in so many ways, it is difficult to discuss all of its effects in one section. Hence, they will be discussed in multiple subsections of this section.

The easiest temperature requirements to meet are the operating and storage temperatures. To select the correct LCD, the designer need only be sure that the LCD operates within both the system operating and storage temperature requirements. Choosing the LCD fluid type will determine the type of LCD you need. For example, there is a large difference in operating temperature (Table 2-1) between fluid type 5 and fluid type 7. The same is true with the storage temperature. Hence, to fulfill the operating and storage requirements, the designer need only select the correct LCD fluid from the LCD specification tables.

It is important to choose the correct LCD fluid type for a particular application. If the LCD remains in the mesophase, it will perform well. If the temperature goes too low or too high, it will cause problems in operation. As the temperature drops, the LCD fluid will become more viscous and eventually cease to function. As the temperature increases, the LCD fluid will become more fluid and the LCD will become either very dark or very clear depending upon the orientation of the polarizers. LCDs usually recover well when the temperature drops to the normal range after exceeding its upper limit. If the LCD is not exposed to very low temperatures for a long time, it

will usually recover well when the temperature increases from the low value to its normal range. Improvements in LCD recovery from temperature extremes are being continually made.[30]

3.1.2 Contrast Fundamentals

The readability of an LCD, newspapers, copying machines and so forth is defined by the contrast ratio (C_R) or contrast (C). The background and different brightness levels of the characters primarily determine the optical appearance of any display. In the case of an LCD display, it is the different brightness levels of the ON segments against the background. The appearance is usually expressed by the contrast ratio. Values of C_R can vary from one to infinity. For twisted nematic LCD displays, the limit of readability is about $C_R = 2$ with good lighting and $C_R = 3$ under poor lighting. The contrast ratio of LCD displays is very similar to that of most types of copy media.[31]

Mathematically, the contrast ratio for a reflective LCD (positive image; dark area = symbol; bright area = background) is

$C_R = B_B/B_S$, where B_B = background brightness and B_S = symbol brightness and $B_B > B_S$[32]

For a transmissive LCD (negative image; light spot = symbol; dark spot = background), the contrast ratio is $C_R = B_S/B_B$, where $B_S > B_B$.[33] Because most applications use positive-image LCDs, only they will be described in the remainder of this section.

Another way to measure readability is to measure the contrast (C). Contrast is the ratio of the difference in brightness between the dark state and light state to the light state. For a positive-image LCD, the equation is:

$C = (B_B - B_S)/B_B$ [34]

For a negative-image LCD, the equation is $C = (B_S - B_B)/B_S$, where C has a value from 0 to 1. The relationship between C and C_r for a positive-image LCD is[35]

$$\begin{aligned} C &= (B_B - B_S)/B_B \\ &= B_B/B_B - B_S/B_S \\ &= 1 - B_S/B_B \\ &= 1 - 1/C_r \end{aligned}$$

[30] Smith, Paul, "Multiplexing Liquid Crystal Displays," *Electronics*, May 25, 1978, pp. 113–121.

[31] Amperex: 'Introduction To Liquid Crystal Displays', *Technical Manual*, pp. 1–21

[32] Smith, Paul

[33] Smith

[34] Standish

[35] Smith

Hence

$$C_r = 1/(1 - C)$$

In most cases, contrast ratio (C_r) is specified rather than contrast. C_r is usually 20 maximum for a positive-image LCD and 10 maximum for a negative-image LCD.[36] Typical values for C_r in various hard-copy media are shown in Table 3-1.[37]

Table 3-1. Values of C_r in Various Hard-Copy Media

Type of Hard Copy Media	Contrast Ratio (C_r)
Typewriter	4.75 to 7.7
Copying machine	2.5 to 5.3
Pencil written	2.5 to 3.7
Newspaper	6.7

3.2 Operating Voltage, Contrast and Viewing Angle

This section will discuss LCD operating voltage, contrast and viewing angle quantitatively. It should be emphasized again that for best results, empirical data from laboratory evaluations is the very best method for analyzing LCD contrast and viewing angle.

The voltage that drives an LCD contains an AC component and a small DC component. The AC components of the voltage are the AC-like voltages discussed extensively in Section 2 of the manual. The second voltage is the DC component of the waveform. In Section 2, we learned that the average voltages across an LCD segment must be 0V DC; hence, the DC component must be 0V DC. In actual practice, the DC voltage is not 0V DC. All LCDs have a DC voltage specification known as the *maximum allowable DC component* of the LCD drive waveform, as shown in the second entry of Table 2-1 and the first of Table 2-4. In Table 2-4, this value is 100 mV maximum for all three fluid types listed. If an LCD chosen for a design had the specifications in Table 2-4, then the LCD controller being used would have to have a DC component specification ≤100 mV. Remember from Section 2 that LCD life decreases as the DC component increases. The remainder of this section will discuss LCD square-wave-type RMS operating voltages, LCD contrast, and LCD viewing angle. (In most places in this section, "V" is used in place of "volts RMS" for simplicity.)

LCD voltages vs. contrast curves have well known shapes. The shape of the voltage vs. contrast curve is dependent on the angle at which the LCD is viewed. As will be seen in the next section,

[36] Smith

changes in temperature cause a shift in the voltages vs. contrast curves, which results in a change in contrast with temperature.

Figure 2-11 shows a basic curve of LCD voltage vs. "% display on." This curve could be renamed "LCD voltage vs. contrast." As stated earlier, V_{th} (10% ON) and V_{sat} (90% ON) are normally considered the OFF and ON voltages, respectively. The points are used because the 0% ON and 100% ON are approached asymptomatically and undefined.[38] Not all LCD contrast curves have the same shape shown in Figure 2-11. Figure 3-1b shows curves of voltage vs. contrast at specific viewing angles. The shape of curve 2 in Figure 3-1b is a generic-type curve whose shape is defined by points shown in Figure 3-2a. E_{10}, E_{50}, and V_{sat} in Figure 3-2a correspond to V_{th}, V_C, and V_{sat}, respectively, in Figure 2-11. The point on curve 2 in Figure 3-1b, where V = 2V, is not curve 2's V_{sat} point. Curve 2's 2V point corresponds to the E_P point in Figure 3-2a. V_{sat} of curve 2 (V_{sat2}) occurs at about 3.3V. Operating the LCD on curve 2 at V = 2V should be avoided because the slope of curve 2 is steep in that region. If this point is used as the LCD operating point, a small change in the RMS voltage will cause a large change in contrast.

[37] Smith

[38] Smith

Figure 3-1. Effect of Viewing Angle on Contrast

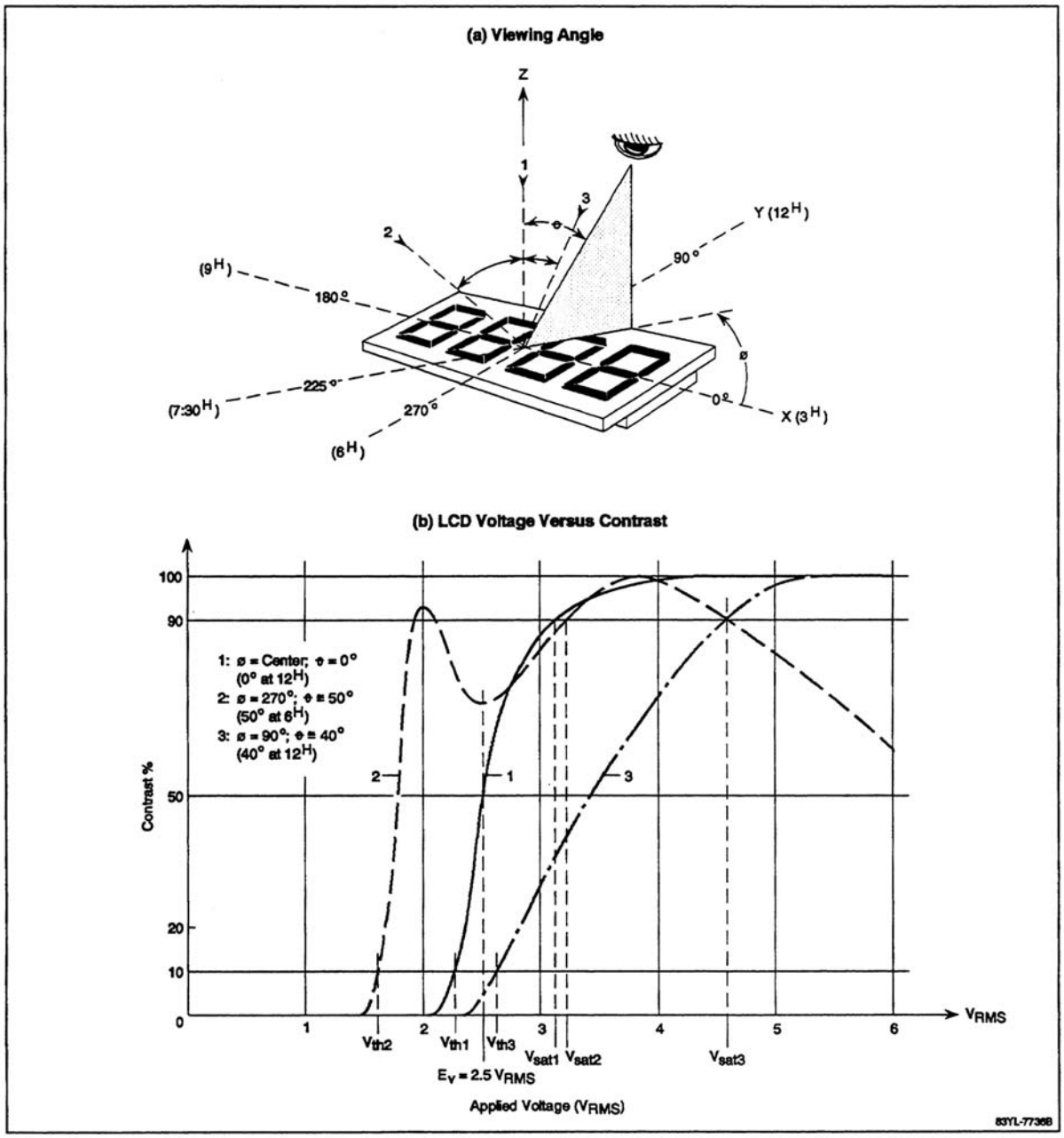
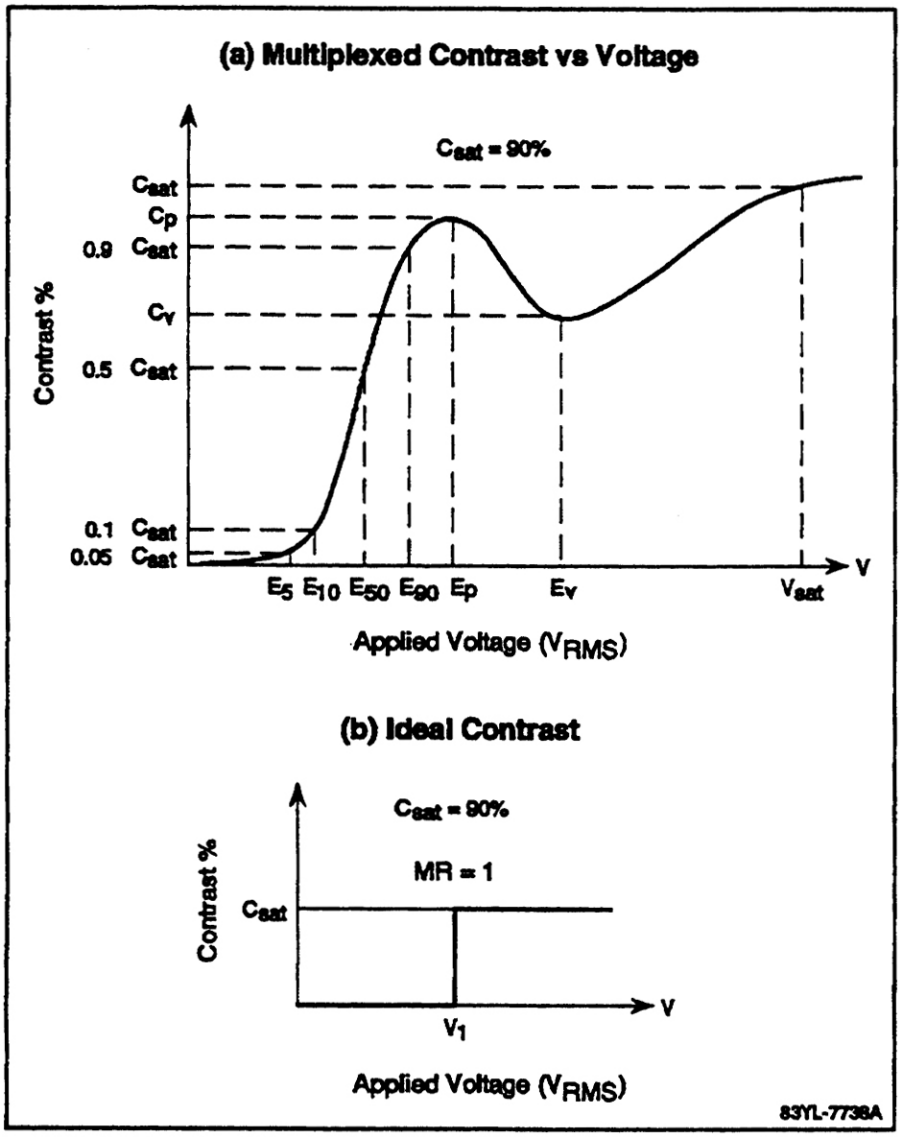


Figure 3-2. Contrast vs. Applied Voltage



The LCD contrast and the angle at which the LCD is viewed are so interrelated that it is virtually impossible to discuss these topics independently. There are multiple coordinate systems used to describe LCD viewing angles. Figure 3-1 shows an LCD and three voltage vs. contrast curves as a function of viewing angle.[39] Figure 3-1a shows a four-digit LCD and the angle at which the LCD is viewed. Figure 3-1b shows voltage vs. contrast curves at specific viewing angles. In Figure 3-1a, the ø angle is known as the *azimuth component* of the viewing angle, where ø is the angle in the

[39] Amperex

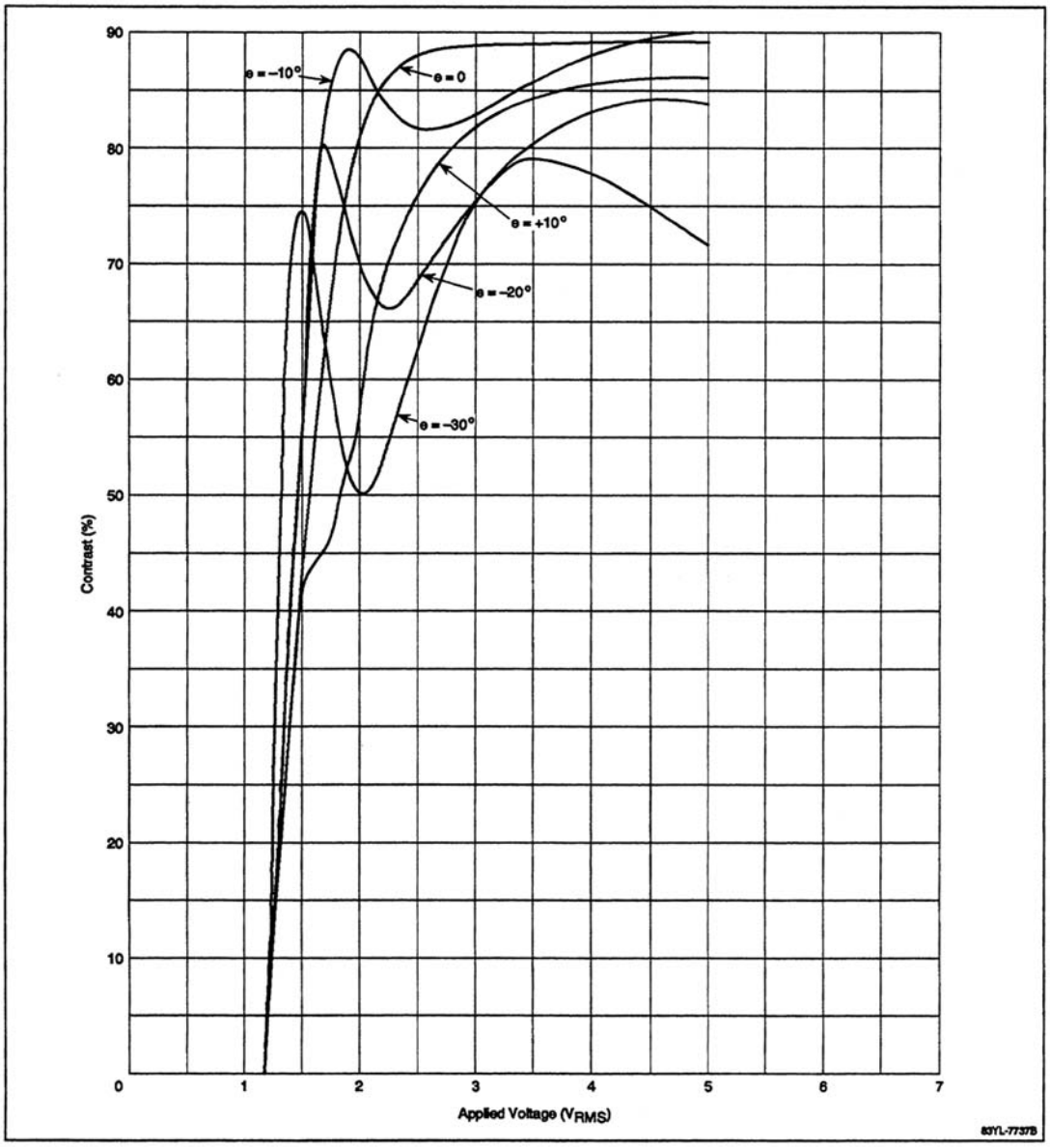
XY plane whose value ranges from 0 to 360 degrees. Many specifications refer to ø = 0 degrees at 3 o'clock (shown in the diagrams and text that follows as 3^H), 270 degrees at 6 o'clock (6^H), 180 degrees at 9 o'clock (9^H), and 90 degrees at 12 o'clock (12^H). The Z-axis is perpendicular to the XY plane; the viewing angle (θ) is the angular deviation from the Z-axis.[40]

This manual will use the system shown in Figure 3-1. Angles in the XY plane will be hours (that is, ø = 0 degrees = 3^H, and so forth), and θ will always be a positive value. Using this system,[41] curve 2's viewing angle is 50 degrees at 6_H. Some systems have negative values of θ, as shown in Figure 3-3. Negative values of θ tend to lead to confusion and, for that reason, are not used in this document.

[40] Standish

[41] Standish

Figure 3-3. LCD Contrast Curves with Negative Viewing Angles



LCDs are designed and built with optimal viewing angles. An LCD designed and built with an optimal viewing angle of 10^H will have its best contrast at 10^H. The shape of the contrast vs. voltage curve depends on the optimal viewing angle for which the LCD was designed and also on the angle at which the LCD is viewed. The three curves in Figure 3-1b are graphs of contrast vs. RMS operating voltage at three viewing angles. The LCD was designed with its optimal viewing angle at 6^H. Curve 1 is a graph of voltage vs. contrast with the viewer directly over the LCD (0 degrees at 12^H). At this viewing angle, the shape of the curve is typical of all LCDs independent of

the optimal viewing angle. Curves 2 and 3 are very dependent on the optimal viewing angle. Curve 2 is 50 degrees at 6^H and curve 3 is 40 degrees at 12^H. The shape of curves 1 and 3 is very similar to that shown in Figure 2-11. Curve 3 has a more gradual slope than curve 1 and is characteristic of viewing angles at 12^H; curve 2 is frequently seen at viewing angles at 6^H. If an LCD were built with an optimal viewing angle at 12^H, the shape of curve 2 would be seen at a viewing angle of 12^H and curve 3's shape would be seen at 6^H.

As an example of contrast evaluation, assume that an LCD having the specifications in Table 2-1 will be used. Fluid type 5 would give the best overall contrast of the four types shown because it has 75 degree viewing at 5V. Fluid type 8 would give the worst contrast because it requires 12V for 75 degree viewing. Since static LCD controllers can produce segment ON voltages of 5V, fluid type 5's overall contrast would be the best. Also, the V10 and V90 specifications could be met easily since the ON voltage at 0°C (the worst-case specification) is 3.1V and the LCD controller would generate 5V.

It should be pointed out that most specifications in Table 2-1 are typical values. For a worst-case design, maximum values always should be used. Since the specification does not give maximum values, the designer should try to get additional information from the manufacturer and should also do a lot of laboratory evaluation on multiple samples.

3.3 Matching LCDs with Their Controllers

The general procedure for matching an LCD controller with an LCD is to pick a fluid type to match (or comes close to matching) the LCD drive scheme. It may be necessary to design in an adjustment to V_{LCD} to fine-tune the drive circuit. To be specific, the designer picks a fluid type that matches the operating voltage generated by the LCD controller. If this is not possible, the designer should choose a fluid type that is a close fit and design in an adjustment to V_{LCD} to bring the V_{ON} and V_{OFF} values in a workable range for the fluid. The V_{LCD} adjustment could be a resistor such as R× as described at the end of the section titled "Static Mode LCDs and Controllers" (Figure 2-16).

A 1/2 bias, duplex-mode LCD is used as an example. Section 2 stated that the AC-like waveforms generated by the LCD controller are not the same for every LCD controller. However, the equations in Table 2-5 for V_{ON} and V_{OFF} are the same for all LCD controllers. So, although the waveforms are different, the final RMS voltage for all LCD controllers is the same.

LCD controllers operating in duplex mode at 1/2 bias will generate:

$V_{OFF} = 0.35\ V_{LCD}$ volts RMS

$V_{ON} = 0.79\ V_{LCD}$ volts RMS, where V_{LCD} is the system supply voltage

Using V_{LCD} = 5V (from Table 2-5):

V_{OFF} = 1.75V

V_{ON} = 3.95V

The design is started by choosing a 1/2 duplex LCD whose V_{th} > 1.75V over the full temperature range.

The V_{ON} voltage generated by the LCD controller will be 3.95V. V_{ON} and the LCD characteristics will determine the contrast at all viewing angles. The shape, steepness, and starting point (V_{th}) of display brightness vs. voltage curves change with viewing angle. The curves for contrast vs. voltage illustrate the contrast at different viewing angles. The LCD controller's drive delivers fixed voltages (V_{ON} and V_{OFF}); V_{ON} is proportional to V_{OFF} and the factor of proportionality is a function of the multiplexing level (Table 2-5). Since V_{OFF} and V_{ON} are fixed, the LCD controller cannot always generate voltages that result in good contrast all viewing angles.[42] Using the curves in Figure 3-1, the contrast at V_{ON} = 3.95V is:

0° at 12^H = 100%

50° at 6^H = 95%

40° at 12^H = 70%

These are acceptable contrasts when the LCD is ON, but there is a problem with V_{th} at 50° at 6^H. V_{OFF} = 1.75V at this viewing angle and is only enough voltage to drive the LCD ON at about 50% contrast; this is not acceptable. A better choice would be fluid type 2 or 8, shown in Table 2-4. With both fluids:

V_{OFF} < V10, and V_{ON} > V90

Either of these LCD fluids would be acceptable.

Another example is a multiplexed application using the LCD specification shown in Table 2-4. Selection of the proper LCD would be heavily dependent on the V10 and V90 voltages. If the LCD is used in quadruplex mode with an LCD controller on one of NEC Electronics' 75X, 75XL, 78K0S or 78K0 microcontrollers operating at 5V DC, then V_{OFF} = 1.66V and V_{ON} = 2.90V.

Fluid type 2 could not be used because:

V_{ON} < V90

The contrast would not be adequate. Fluid type 3 could not be used because:

[42] Standish Industries.

$V_{OFF} > V10$

Here the LCD segments would always be ON. Fluid type 3 could be used if R× is used to adjust V_{OFF} such that:

$V_{OFF} < V10$

An evaluation would have to be made to ensure that R× would not reduce V_{ON} to the point where the contrast is too low when the LCD is ON.

Fluid type 8 would be a good choice because:

$V_{OFF} < V10$ and $V_{ON} > V90$

This specification is incomplete because no specifications are given for:

- Angular viewing
- Contrast ratio

Also, a specification for V10 and V90 over the temperature range can be calculated using the dV/dT specification in Table 2-4, which is described in the next section.

3.4 Drive Voltages vs. Contrast and Viewing Angle

The operating mode of the LCD has a large effect on contrast. In Figure 3-1b, V_{sat}, at 40 degrees at 12^H = 4.6V, which requires an LCD controller with a very high drive capability to get 90% contrast. If a static LCD is used, good contrast can be achieved because static-mode LCD controllers can generate operating voltages of 5V. At 5V, 40 degrees at 12^H would 98% contrast and the LCD would look good. Note that at 4V, the contrast at 50 degrees at 6^H starts dropping off sharply. Operating the LCD at V_{sat} (= 4.6V) would give the best overall display contrast. Zero degrees at 12^H would give 100% contrast and 50 degrees at 6^H and 40 degrees at 12^H would give 90%.

If the LCD controller is used in duplex mode at 5V, then the ON segment voltage is 3.95V. At this voltage, 40 degrees at 12^H would give about 70% contrast. As was shown earlier, the RMS operating voltages get lower as the number of segments being multiplexed increases.

If the LCD was operating at 1/3 bias in triplex mode, V_{OFF} = 1.65V and V_{ON} = 3.3V. In this case, 0 degrees at 12^H and 50 degrees at 6^H would be about 90% contrast, and 50 degrees at 12^H would drop to about 40% contrast. In quadruplex mode at 5V, V_{OFF} = 1.66V and V_{ON} = 2.90V. Zero degrees at 12^H = 80% contrast, 50 degrees at 6^H = 75% contrast, and 40 degrees at 12^H = 25%

contrast. Note the deterioration in contrast at 40 degrees at 12^H as the number of multiplexed segments increased.

If a designer cannot find an LCD that meets the design requirements of an application, they should contact the LCD manufacturer's application engineer. The application engineer can specify a fluid type to meet the design requirements, or, work with the design engineer and design a custom LCD. There is a limit here, however, because LCDs do change contrast with angle and LCD drive voltages are fixed for a given V_{LCD} value.

3.5 Viewing Angles

LCDs do not always need to have excellent viewing at all angles. In many applications, it may be necessary to have good viewing angles only for a few specific areas. At times, this relaxed specification allows the manufacturer to build an LCD for a specific application. For example, for a wristwatch. A right-handed person wears a watch on the left hand and will view the LCD at about $7:30^H$ to read the time. Left-handed people typically wear their watches on the right hand and tend to view the LCD at about 6^H. Hence, the requirement for an LCD in this application would be to have very good viewing between 9^H and 6^H (180 to 270 degrees = quadrant 3).

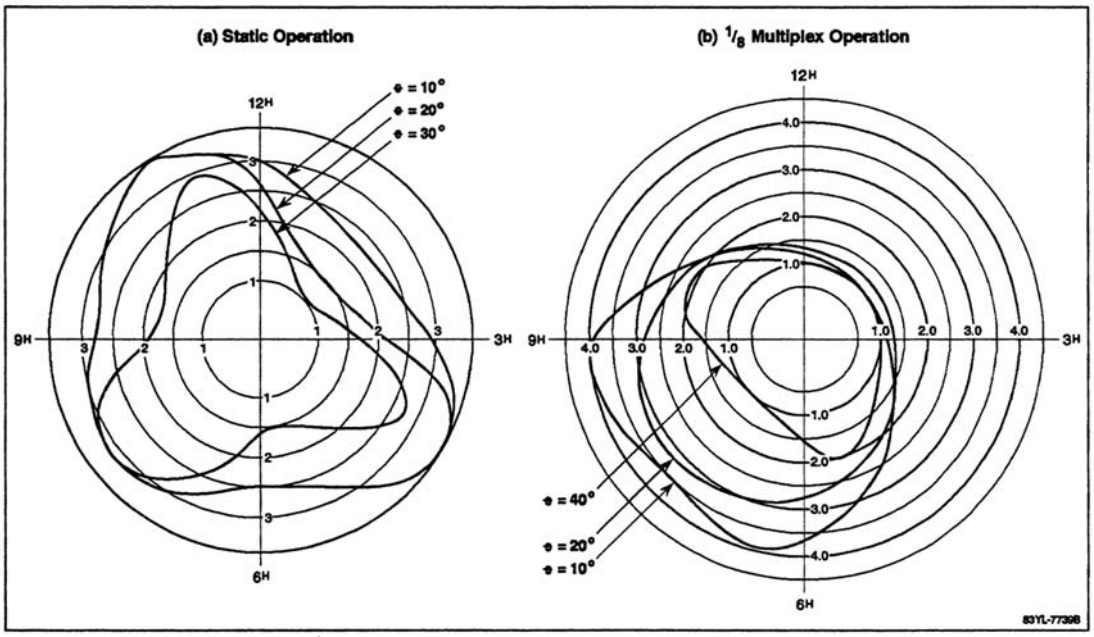
Many LCDs have been designed with the best viewing angle at $7:30^H$ for wristwatch applications.[43] Figure 3-4 shows polar contrast curves for a typical wristwatch LCD. Figure 3-4a is for a static LCD operating at 3V and a frequency of 32 Hz. Figure 3-4b is for a 1/3 bias, 1/8 duty cycle LCD operating at 5.3V and a frequency of 256 Hz.[44]

Figure 3-4a has good general contrast at most azimuth angles. Figure 3-4b illustrates how multiplexed LCDs can be very directional. The contrast is best between 9^H and 6^H (quadrant 3), up to four times as good as the rest of the viewing angles. The LCD that corresponds to this curve is very well suited for wristwatch applications because of its good contrast in quadrant 3. The fact that its contrast is poor in quadrants 1, 2, and 4 is not a serious problem because, as explained earlier, the main viewing angle is in quadrant 3.

[43] Soref, Richard A.

[44] Seiko

Figure 3-4. Contrast Curves for Typical Wristwatch with LCD



One way to determine the slope of the contrast vs. voltage curve is by calculating the threshold ratio using V_{ON} and V_{OFF}. Some publications call this the margin ratio (MR)[45], which is calculated as follows:

Margin ratio = MR = $V_{ON}/V_{OFF} = V_{sat}/V_{th}$.

An ideal curve has an infinite slope and is shown in Figure 3-2b. The voltage $V_1 = V_{th} = V_C = V_{sat}$; hence the MR is:

$MR = V_{sat}/V_{th} = V_1/V_1 = 1$.

With an ideal MR, V_{th} would need to be finitely smaller than V_1 and V_{sat} finitely larger than V_1. Hence, the difference in output voltage from the LCD controller ($V_{ON} - V_{OFF}$) would be finitely small. As shown from equations 2.1–2.5, as the number of segments being multiplexed increases, the difference between V_{ON} and V_{OFF} decreases. When MR = 1, it is easy to multiplex more than four segments because the voltage difference between V_{ON} and V_{OFF} is very small.

Margin ratio depends on the liquid crystal material, device construction, and viewing angle. Normally the MR for multiplexed LCD is 1.1 to 2.4. Low values of MR generally give good contrast, but this is not always true. The absolute values of V_{th} and V_{sat} often are more important than the MR. For the three curves in Figure 3-1b, MR is calculated as follows.

[45] Seiko

$MR_1 = 3.1/2.2 = 1.41$ (0° at 12^H)

$MR_2 = 3.3/1.6 = 2.06$ (50° at 6^H)

$MR_3 = 4.6/2.6 = 1.77$ (40° at 12^H)

An LCD controller operating in quadruplex mode with a 5V power source has a V_{OFF} voltage of 1.66V and a V_{ON} voltage of 2.90V. If V_{OFF} =1.66V is the threshold voltage, this would ensure that the LCD is OFF for all three viewing angles, since the lowest threshold voltage for the three curves is V_{th1}, or about 1.7V. V_{ON} would be 2.90V and would generate different contrast values for the three curves at different viewing angles. Using V_{ON} = 2.90V, the contrast of the three curves in Figure 3-1b would be as follows.

80% contrast for viewing angle 1 (0° at 12^H)

75% contrast for viewing angle 2 (50° at 6^H)

25% contrast for viewing angle 3 (40° at 12^H)

The contrast at 40° at 12^H would be poor. Although MR_3 is less than MR_2, 40° at 12^H has far worse contrast than 50° at 6^H. In this case, the value of the absolute voltage is more important than the MR value. To improve 40° at 12^H to 75% contrast, the LCD controller would have to generate a V_{ON} of 4V. Large voltages like 4V cannot be produced from quadruplex LCD controllers that are limited to V_{LCD} = 5V maximum. There are some LCD controllers, however, that allow V_{LCD} to be as high as 15V.

Some articles are written about LCDs operating at V_C, where V_C is about halfway between V_{th} and V_{sat}. At V_C, the contrast is about 50%, which is adequate in many applications. A great deal of caution should be used if this design approach is taken. The slope of the contrast curve between V_{th} and V_{sat} may be very steep. Small changes in operating voltage can cause large changes in contrast. For example, if the LCD that corresponds to 50° at 6^H in Figure 3-1b were used, then a drop in operating voltage of 0.125V would cause the LCD to drop from 50% to 0% contrast.

3.6 Effect of Temperature on Contrast

This section discusses how temperature affects LCD contrast and contains methods that reduce the undesirable effects. It should be emphasized again that for best results, using empirical data from laboratory evaluations is the best way to analyze the effect of temperature on LCD contrast.

The ambient temperature (T_A) can change the contrast of an LCD. As the temperature changes, the required V_{ON} and V_{OFF} voltages change. If the supplied V_{ON} and V_{OFF} do not change with the temperature, there could be a change in the contrast on the LCD. The phenomenon that changes these voltages is characterized by the LCD temperature coefficient dV/dT, where dV is the voltage

change and dT is the temperature change. The specification describes operating voltage changes with temperature: dV/dT has a negative coefficient and typically is between –3 to –10 mV/°C[2]. Sometimes there are two dV/dT specifications, one for V_{th} and one V_{sat}, which are usually specified by dV_{th}/dT and dVsat/dT.

For example, for fluid type 3 in Table 2-4, the dV/dT specification is:

dV/dT = 8 mV/°C

As a result of dv/dT, as the temperature goes up, the operating voltage goes down by 8 mV/°C. As the temperature goes down, the operating voltage goes up by 8 mV/°C. This phenomenon causes a shift in the contrast vs. voltage curve such that there is a different curve for every temperature. Three curves are shown in Figure 3-5 for T_A = 0°C, 20°C, and 40°C.

V_{th1} is the threshold at 40°C.

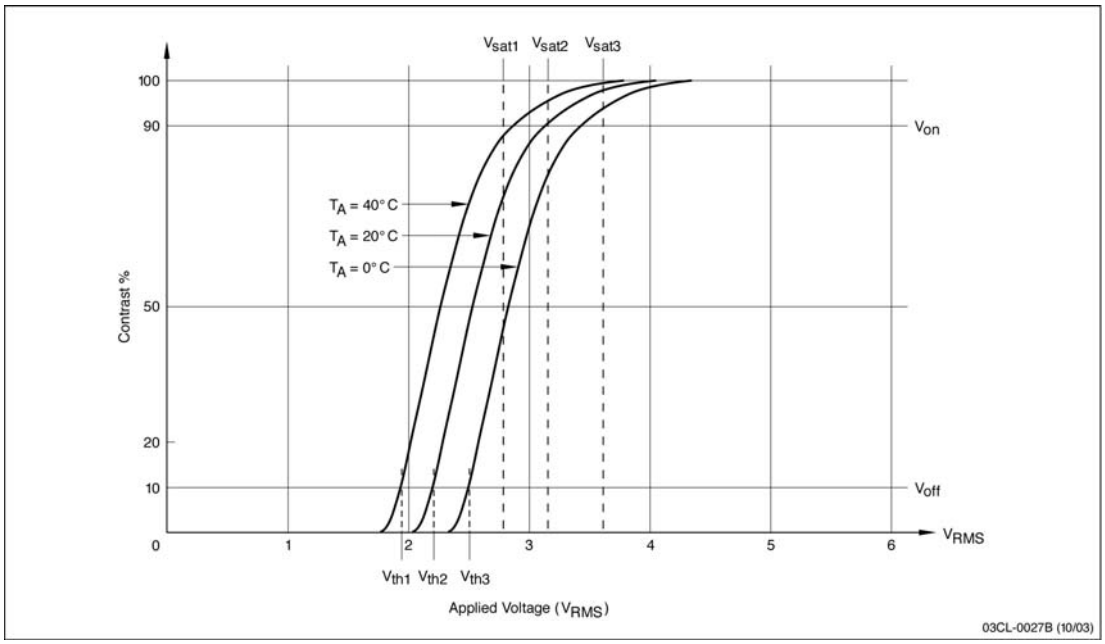
V_{th2} is the threshold at 20°C.

V_{th3} is the threshold at 0°C.

Over a range of 20°C, the operating voltage shifts (changes) by:

dV = –8 × dT = –8 × 20 = ±0.16 volts.[46]

Figure 3-5. Effect of Temperature on Contrast



[46] Rees

The effect is that it requires less voltage to turn ON segments at high temperatures and more voltage to turn the segments ON at lower temperatures. If a fixed voltage is chosen from Figure 3-5, it can be seen that as the temperature increases, the contrast increases, and as the temperature decreases, the contrast decreases. For example, at V = 2.5V RMS:

At T = 0°C, contrast = 10%

At T = 20°C, contrast = 40%

At T = 40°C, contrast = 60%

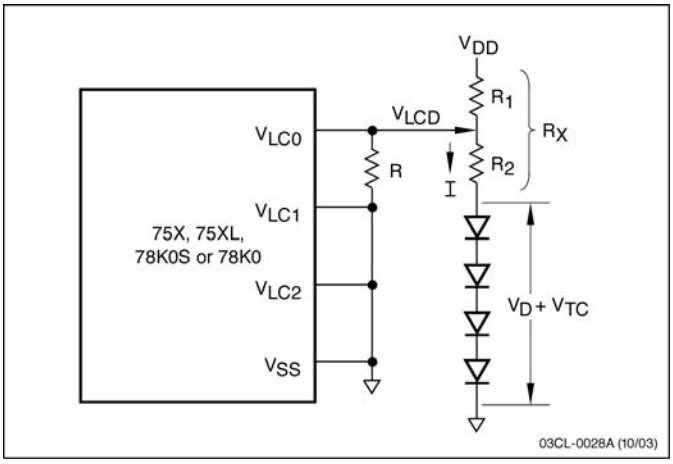
There are a number of ways to design around this problem. One way is to bias the LCD controller such that:

- V_{th} is always ≤ V_{th} at 40°C (the lowest threshold voltage over T_A)
- V_{sat} is always ≥ V_{sat} at 0°C (the highest saturation voltage over T_A)

Another method is to use diodes as termperature-compensating devices. A typical silicon diode has a temperature coefficient of dV/dT = 2 mV/°C. Figure 3-6 shows a temperature-compensation circuit using four diodes for total compensation of:

dV/dt = 2 mV/°C × 4 = –8 mV/°C.

Figure 3-6. Contrast Temperature Compensation Using Diodes



This is equivalent to the typical LCD temperature coefficient of –8 mV/°C. The number of diodes used would depend on the specific LCD being used in the design. The idea is to use the diode's temperature coefficient to track the LCD's temperature coefficient. The tracking coefficient would provide the required voltage changes to the LCD, which the LCD needs as a result of ambient temperature changes. Thus, the LCD would maintain constant contrast. As the temperature goes up, the LCD controller bias voltages would go down by the amount required by the LCD. As the

temperature goes down, the LCD controller bias voltage would go up by the same amount required by the LCD. This method is effective over a temperature range of 0 to 50°C. Beyond this range, the LCD's dV/dT becomes nonlinear and this compensation may not give optimal viewing characteristics.[47]

3.7 Static Mode Operation

Figure 3-6 is the same circuit shown in Figure 2-16b, except that four diodes have been added. Also, R× has been replaced by a potentiometer of value R_X, which is shown as two resistors (R_1 and R_2), where $R\times = R_1 + R_2$. Mathematically (using Figure 3-6), the scheme works if $R >>> R_1$ or R_2. Also, the voltage across the diodes is $V_D + V_{TC}$. V_D is the voltage drop of all the diodes at room temperature and V_{TC} is the total voltage change in the diodes as a result of temperature changes. These temperature changes (V_{TC}) are the temperature compensation of all the diodes. An analysis of the scheme is shown below.

$$I = [V_{DD} - (V_D + V_{TC})] / (R_1 + R_2)$$

$$V_{LCD} = (V_D + V_{TC}) + IR_2$$

$$V_{LCD} = (V_D + V_{TC}) + R_2 [V_{DD} - (V_D + V_{TC})] / (R_1 + R_2)$$

$$V_{LCD} = V_D + V_{TC} + R_2 V_{DD} / (R_1 + R_2) - R_2 V_D / (R_1 + R_2) - R_2 V_{TC} / (R_1 + R_2)$$

$$V_{LCD} = V_D [1 - R_2 / (R_1 + R_2)] + V_{TC} [1 - R_2/(R_1 + R_2)] + R_2 V_{DD} / (R_1 + R_2)$$

If $K_1 = 1 - R_2 / (R_1 + R_2)$ and $K_2 = R_2 / (R_1 + R_2)$, then $V_{LCD} = K_1 V_D + K_2 V_{DD} + K_1 V_{TC}$.

At $T_A = 20°C$, $V_{TC} = 0$ and there is no temperature compensation. Therefore:

$$V_{LCD1} = K1 V_D + K_2 V_{DD}$$

If $T_A \neq 20°C$, then $V_{TC} \neq 0$ and temperature compensation is required. Therefore:

$V_{LCD2} = K_1 V_D + K_2 V_{DD} + K_1 V_{TC}$ then the difference between $V_{TC} = 0$ and $V_{TC} \neq 0$ is:

$$V_{LCD1} - V_{LCD2} = K_1 V_D + K_2 V_{DD} - (K_1 V_D + K_2 V_{DD} + K_1 V_{TC}) = - K_1 V_{TC}$$

$$K_1 = 1 - [R_2 / (R_1 + R_2)] = (R_1 + R_2 - R_2) / (R_1 + R_2) = R_1 / (R_1 + R_2)$$

$$K_1 = R_1 / (R_1 + R_2) = (R_1/R_1) / [(R_2 + R_2) / R_1] = 1/(1 + R_2/R_1)$$

If $R_1 > > R_2$, then $K_1 = 1$ and:

$$-K_1 V_{TC} = 1 \times V_{TC} = V_{TC}.$$

Therefore, if $R_1 >> R_2$, then $K_1 = 1$ and the diodes will fully temperature-compensate the LCD's dV/dT.

[47] Standish

Table 2-1 does not have a dV/dT specification. However, specifications are given at 0° and 25°C. Using this information, some dV/dT specifications can be calculated. For fluid type 2, the formula would be:

$d\ V_{th}/dT = [(V_{th} \text{ at } 0°C) - (V_{th} \text{ at } 25°C)] / 25$

$d\ V_{th}/dT = (2.4 - 2.2)/25 = -8 \text{ mV/°C}$

$d\ V_{sat}/dT = [(V_{sat} \text{ at } 0°C) - (V_{sat} \text{ at } 25°C)] / 25$

$d\ V_{sat}/dT = (3.2 - 3.1) / 25 = -4 \text{ mV/°C}$

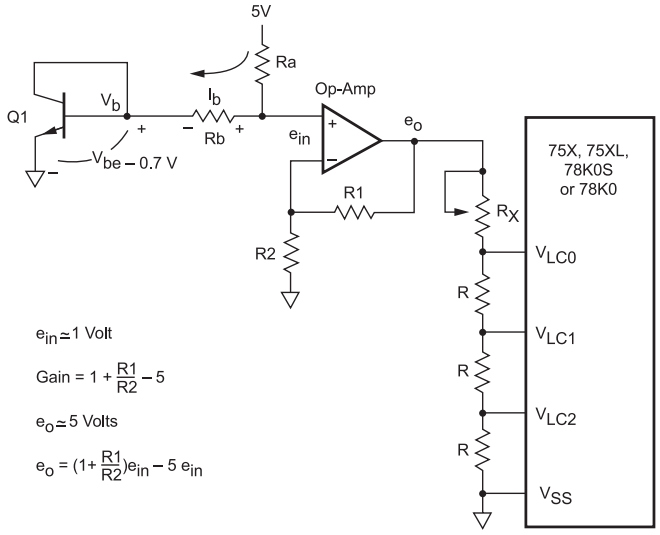
Using this information, the temperature behavior for these static LCDs can be obtained. A reasonable curve could be drawn using the ON/OFF operating voltage threshold specifications. Also, a specification is given for V_{th} and V_{sat} at 0°; the only missing specification is that of V_{th} and V_{sat} at higher temperatures. However, the V_{th} and V_{sat} specifications that are given indicated that the lowest OFF voltage required would be about 0.2V below the specification at 25°C. Because it is relatively easy to get a static LCD controller with an ON voltage close to 5V and an OFF voltage close to 0V, a design can be implemented.

Newer and better temperature-compensating schemes are continually being devised. The scheme shown in Figure 3-7 uses the dV/dT coefficient of the base-emitter junction of a transistor, which is connected to an operational amplifier (op-amp). The op-amp's gain and the transistor's base-emitter dV/dT are set at values that will compensate for the dV/dT specification of the LCD.[48] A scheme like this for static LCDs is shown in the top of Figure 3-7a. An NPN transistor has its collector connected to its base and the base-emitter junction is forward-biased. The voltage at the base of the transistor is V_b. The transistor's base is connected through resistor R_b to the positive input of a non-inverting op-amp. The input voltage to the op-amp is e_{in}, which is multiplied by the gain of the op-amp, resulting in an output voltage from the op-amp of e_o. e_o is connected to the top of the LCD's bias resistor ladder and is used as the supply voltage to the LCD resistor ladder. Because the output impedance of an op-amp is very low (typically ≤ 10 ohms), its output is a very good voltage source.

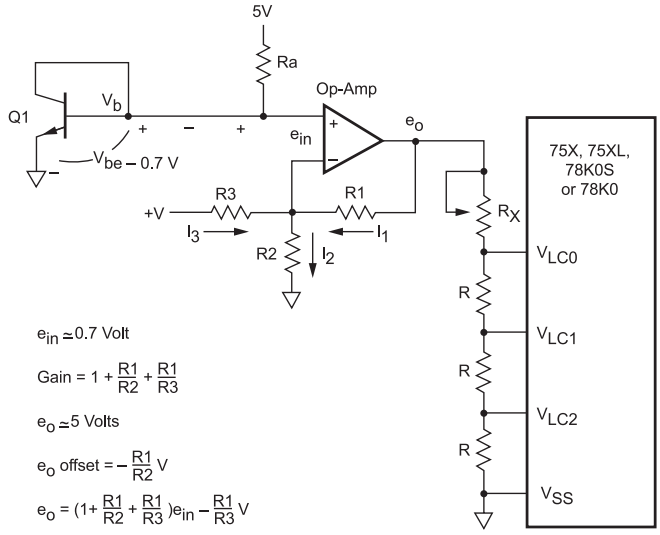
[48] Standish Industries

Figure 3-7. Temperature Compensation Using an Op-Amp

(a) Static Temperature Compensation



(b) Multiplexed LCD Temperature Compensation



The circuit works as follows. When the ambient temperature T_A decreases, V_b increases by the temperature coefficient of the transistor's base-emitter junction, typically a negative value:

$$dV/dT = -2 \text{ mV/°C}.$$

An increase in V_b increases e_{in}, which increases the voltage output of the amplifier by the amplifier's gain, thereby raising the voltage source to the LCD resistor ladder. This results in increased voltage drive from the LCD controller, thereby raising the RMS voltage values from the

LCD controller to the LCD. The increased voltage to the segment and common lines offsets the increased voltage required by the LCD as a result of the temperature decrease. If T_A goes up, V_b and e_O go down. The lower LCD ladder voltage reduces the LCD controller's RMS voltage, thereby compensating for the temperature increase.

The transistor's base-emitter junction has a coefficient of about –2 mV/°C and the voltage at the base of Q_1 (V_b) is about 0.7 volts. The values of resistors R_a and R_b are selected to set the transistor base current (I_b) such that transistor Q_1 is set to the desired dV/dT coefficient. Also, R_a and R_b are selected such that the voltage input e_{in} to the op-amp is at about 1 volt. Setting the gain of the op-amp to 5 (gain = $1 + R_1/R_2$) results in an op-amp output voltage $e_O = 5$ volts. The temperature compensation component of the op-amp's output voltage is $V_{TC} = 5 \times dV/dT$.

Resistor R× may be used optionally for contrast adjustment and may be a fixed resistor or a potentiometer. The LCD resistor ladder drive voltage is temperature-compensated and its value at $T_A = 20°C$ is about 5 volts.

Using this method, if an LCD has a dV/dT specification of 10 mV/°C and the transistor has a base-emitter coefficient of –2 mV/°C, the gain of the op-amp would be set to 5.0. This would provide a temperature coefficient of 10 mV/°C at the output of the op-amp. Also, one diode rather than five diodes would be required to compensate for dV/dT. If the LCD dV/dT specification is 9 mV/°C and the base-emitter temperature coefficient is 2 mV/°C, the op-amp gain is set to 4.5. The overall compensation from the op-amp's output is 9 mV/°C. If the diode scheme just described above is used, an exact compensation is not possible.

This illustrates on of the advantages of this scheme—more accurate compensation is possible. By varying the transistor's base current, the base-emitter temperature coefficient can be varied. Typical values of –1.5mV/°C to –2.5mV/°C can be obtained from many transistors. Having a variable base-emitter dV/dT coefficient and having the flexibility to vary the gain of the op-amp gives a designer more latitude in a design. Also, resistor R× can still be used to adjust the LCD contrast if required.

An analysis of the circuit in Figure 3-7a follows. Transistor Q1 (such as a 2N2222) has a negative temperature coefficient at the base-emitter junction, much the same as the diodes described above. Resistors R_a and R_b are selected such that transistor Q1 is set to the desired dV/dT coefficient.

The gain of the op-amp is set as follows:

$$\text{Gain} = 1 + R_1/R_2 \qquad (3.1)$$

The base current I_b is:

$$I_b = (5 - V_b) / (R_a + R_b) = e_{in} - V_b) / R_a$$

Let e_{in} = 1 volt and V_b = 0.7 volts:

$$I_b = (5 - 0.7) / (R_a + R_b) = (1 - 0.7) / R_b$$

$$I_b = 4.3 / (R_a + R_b) = 0.3 / R_b$$

$$4.3R_b = 0.3(R_a + R_b) = 0.3R_a + 0.3R_b$$

$$4R_b = 0.3R_a$$

$$R_a = 4 \times R_b/0.3 = 13.33 \times R_b$$

$$R_a = 13.33 \times R_b \qquad (3.2)$$

At T_A = 20°C, the input to the op-amp e_{in} is:

$$e_{in} = V_b + I_b \times R_b$$

$$I_b = (5 - V_b) / (R_a + R_b)$$

$$e_{in} = V_b + (5 - V_b) \times R_b / (R_a + R_b)$$

If the temperature increases, the voltage at V_B increases by V_{TC}. The voltage at the input to the op-amp is:

$$e_{in\,1} = (V_b + V_{TC}) + [5 - (V_b + V_{TC})] \times R_b / (R_a + R_b)$$

The change in voltage at the op-amp input is E_{IN1}– E_{IN} is ΔE_{IN}.

$$\Delta e_{in} = (V_b + V_{TC}) + [5 - (V_b + V_{TC})] \times R_b/(R_a + R_b) - [V_b + (5 - V_b)] \times R_b/(R_a + R_b)$$

$$\Delta e_{in} = V_b + V_{TC} - V_b + (5 - V_b) \times R_b/(R_a + R_b) - (5 - V_b) \times R_b/(R_a + R_b - V_{TC} \times R_b/(R_a + R_b)$$

$$\Delta e_{in} = V_{TC} - V_{TC} \times R_b/ (R_a + R_b) = V_{TC} (1 - R_b) / (R_a + R_b)$$

$$\Delta e_{in} = V_{TC} (R_a + R_b - R_b) / (R_a + R_b)$$

$$\Delta e_{in} = V_{TC} \times R_a / (R_a + R_b)$$

By dividing the numerator and denominator by R_a then

$$\Delta e_{in} = V_{TC} / (1 + R_b/R_a)$$

From equation (2-2), $R_a = 13.33 \times R_b$, hence:

$$\Delta e_{in} = V_{TC} / (1 + R_b / 13.33 \times R_b)$$

$$\Delta e_{in} = V_{TC} / (1 + 0.07) = 0.93 \times V_{TC}$$

Using this scheme causes a change in V_B (= V_{TC}) by one part to change E_{IN} by 0.93 parts. The circuit can be adjusted to give full compensation as is shown by the following example:

$e_O = 0.93 \times V_{TC} \times \text{gain}$

If the Gain = 5:

$e_O - 4.65 \times V_{TC}$

If LCD dV/dT = 9, then to fully compensate for dV/dT:

$dV/dT \text{ at } e_O = 4.65 \times V_{TC} = 9$

To achieve dV/dT = 9, then:

$V_{TC} = 9/4.65 = 1.93$

The base current of the transistor should be adjusted to provide a coefficient of 1.93 mV/°C to give full dV/dT compensation.

What has been discussed so far applies to static mode only. In multiplex mode, the problem is more complex. There are two dV/dT values:[49]

dV_{LCD}/dT at the LCD

dV_{PS}/dT from the LCD power supply (e_O from the op-amp—Figure 3-7)

In static mode, $dV_{LCD}/dT = dV_{PS}/dT$, but in multiplexed applications, $dV_{LCD}/dT \neq dV_{PS}/dT$. Using the duplex mode with no temperature compensation as an example, V_{OFF} (from Table 2-10) is the OFF voltage at the LCD and V_{LCD} is the voltage supplied to the resistor ladder:

$$V_{OFF} = 0.35 \times V_{LCD} \tag{3.3}$$

To avoid confusion in the analysis that follows, the term "V_{RL}" is used in place of V_{LCD}, hence:

$V_{OFF} = 0.35 \times V_{RL}$

Using ΔT_A = the ambient temperature change, V_{OFF} with temperature compensation (V_{OFFTC}) is:

$V_{OFF\,TC} = V_{OFF} + (\Delta T_a \times dV_{LCD}/dT)$

V_{RL} with temperature compensation is $V_{LCD\,TC}$:

$V_{LCD\,TC} = V_{RL} + (\Delta T_a \times dV_{PS}/dT)$

With temperature compensation, $V_{OFF\,TC} = 0.35 \times V_{RL\,TC}$ and becomes:

$$V_{OFF\,TC} = [V_{OFF} + (\Delta T_a \times dV_{LCD}/dT)] = [0.35 \times (V_{RL} + (\Delta T_a \times dV_{PS}/dT))] \tag{3.4}$$

The difference between a temperature-compensated and non-temperature-compensated V_{OFF} is:

[49] Standish Industries.

$$[V_{OFF} + (\Delta T_a \times dV_{LCD}/dT)] = [0.35 \times (V_{RL} + (\Delta T_a \times dV_{PS}/dT))] - 0.35 \times V_{RL}$$

$$(\Delta T_a \times dV_{LCD}/dT) = 0.35 \times (\Delta T_a \times dV_{PS}/dT)$$

$$dV_{LCD}/dT = 0.35 \times (dV_{PS}/dT)$$

$$dV_{PS}/dT = (dV_{LCD}/dT) \times 1/0.35$$

$$dV_{PS}/dT = 2.86 \times dV_{LCD}/dT \quad (3.5)$$

Equation (3.5) shows that dV/dT at the power supply must be 2.85 times the dV/dT at the LCD to deliver the correct temperature-compensated voltage at the LCD. Using this same approach, triplex mode with ½ bias is:

$$dV_{PS}/dT = 2.50 \times dV_{LCD}/dT \quad (3.6)$$

Triplex mode with 1/3 bias and quadruplex mode with 1/3 bias have the same values:

$$dV_{PS}/dT = 3.03 \times dV_{LCD}/dT \quad (3.7)$$

Also, for static mode, it can be shown that:

$$dV_{PS}/dT = dV_{LCD}/dT.$$

No extra compensation is required in static mode, and the methods described above work correctly in the static mode. Using the duplex-mode example, if the LCD has a coefficient of $dV_{LCD}/dT = -6$ mV/°C, then from (3.5):

$$dV_{PS}/dT = 2.86 \times dV_{LCD}/dT$$

$$dV_{PS}/dT = 2.86 \times (-6) = -17.16 \text{ mV/°C}$$

The diode approach is not very useful because if dV/dT = 2 mV/°C per diode:

(–17.16 mV/°C) / 2 mV/°C per diode) = 8.5 diodes are required.

For a multiplexed application, the transistor op-amp scheme shown in the top part of Figure 3-7b makes the most sense. To understand this circuit, assume R_3 is not in the circuit. The maximum output of the op-amp is 5V DC and its minimum input is 0.7V DC. R_b will not be used (as it was in Figure 3-7a) and the transistor base will be connected directly to the positive input of the op-amp. Resistor R_a will still be used to set the dV/dT of the transistor's base-emitter junction. For the op-amp's output to be 5 volts maximum, the maximum gain of the op-amp is:

$$\text{Gain} = 5/\ 0.7 = 7.14 \quad (3.8)$$

The dV_{IN}/dT input to the op-amp is:

$$dV_{IN}/dT = (17.16 \text{ mV/°C}) / (7.14) = 2.4 \text{ mV/°C}$$

This requires that the transistor's base-emitter junction be set to 2.4 mV/°C using R_a. It is very difficult (if not impossible) to find a transistor whose base-emitter junction has a dV/dT value of 2.4 mV/°C. This problem becomes worse in 1/3 bias triplex or quadruplex modes because of the high value of $dV_{PS}/dT = 3.03 \times dV_{LCD}/dT$ [see equation (3.7)]. Assume an application uses an LCD with dV/dT = –8 mV/°C. Then:

$$dV_{PS}/dT = 3.03 \times 8 = 24.24 \text{ mV/°C} \quad (3.9)$$

Since the maximum op-amp gain is 7.14, the dV_{in}/dT input to the op-amp is:

$$dV_{in}/dT = (24.24 \text{ mV/°C}) / (7.14) = 3.39 \text{ mV/°C}$$

It would be impossible to find a transistor whose base/emitter junction has a dV/dT = -3.39 mV/°C.

Using resistor R_3 connected to the +V volts solves this problem. R_3 has two functions:[50]

It raises the gain of the op-amp by a factor of R_1/R_3.

It adds in a positive offset voltage at the op-amp's input, which reduces the output voltage from the op-amp to maintain 5 volts at the op-amp's input.

Without R_3, the circuit in Figure 3-7b can described by the equation:

$$e_O = A\, e_{in} \quad (3.10)$$

Where:

A = gain of the op-amp;

e_O = op-amp's output voltage;

e_{in} = op-amp's input voltage

With resistor R_3, the circuit in Figure 3-7b can be described by the linear equation

$$y = mx + b \quad (3.11)$$

Where:

m = gain of the op-amp;

b = an offset voltage introduced by R_3 biased to + V

The gain of an op-amp is usually denoted by A. Using these terms, equation (3.11) can be rewritten as

$$e_O = A\, e_{in} - b \quad (3.12)$$

[50] Standish Industries.

To solve the problem described by equation (3.9):

$e_O = 24.24$ mV/°C

$e_{in} = 2$ mV/°C

The gain must be:

$$\text{Gain} = A = 24.24/2 = 12.12 \tag{3.13}$$

The input is approximately 0.7 volts (V_{BE} of Q_1). Using equation (3.10), the output is:

$e_O = 0.7 \times 12.12 = 8.48$ volts.

In this example, assume that the microcontroller is powered by 5-volts. Latch-up will not occur if the output does not exceed 5 volts. This means that an output offset voltage is required of:

$b = 8.48 - 5.0 = -3.48$ volts.

This offset must be introduced to protect the microcontroller against latch-up. R_1, R_2, and R_3 must be chosen such that the circuit has a gain of 12.12 and an output offset voltage of –3.48 volts. The mathematical procedure to calculate these conditions is as follows. (The mathematical procedure used here can be used to calculate the parameters for gain and offset voltages with different values.)

The equivalent resistance (R_{eq}) of R_2 and R_3 is:

$$R_{eq} = R_2 \times R_3/(R_2 + R_3) \tag{3.14}$$

Using (2-14), the gain (A) of the op-amp is:

$$A = 1 + R_1/R_{eq} = 1 + R_1 \times 1/R_{eq} = 1 + R_1 \times [(R_2 + R_3) / (R_2 \times R_3)]$$

Solving this equation:

$$A = 1 + R_1/R_2 + R_1/R_3 \tag{3.15}$$

Equations (3.1) and (3.15) show that resistor R_3 adds a gain of R_1/R_3 when compared with the gain in Figure 3-7a. From Figure 3-7b, the values of I_1, I_2, and I_3 are:

$$I_1 = (e_O - e_{in}) / R_1;\ I_2 = e_{in} / R_2;\ I_3 = (V - e_{in}) / R_3 \tag{3.16}$$

From Figure 3-7b:

$$I_2 = I_1 + I_3 \tag{3.17}$$

Using (3.16) and (3.17):

$$e_{in} / R_2 = (e_O - e_{in}) / R_1 + (V - e_{in}) / R_3 \tag{3.18}$$

Solving equation (3-18) for e_O:

$$e_O = (1 + R_1/R_2 + R_1/R_3) \times e_{in} - (R_1 \times V) / R_3 \quad (3.19)$$

Equation (3.19) has the form of equation (3.12), where:

$$A = (1 + R_1/R_2 + R_1/R_3) \quad (3.20)$$

$$B = (R_1 \times V)/ R_3$$

$$e_O = A\, e_{in} - (R_1 \times V)/R_3 \quad (3.21)$$

The next step is to determine the values of R_1, R_2, and R_3. The value for $dV_{PS}/dT = dV/dT$ at the op-amp output.

Assuming R_3 is not in the circuit shown in Figure 3-7b, and using equation (3.10), where A = 12.12 and e_{in} = 0.7 volts (V_{BE} of Q_1):

$$e_O = 12.12 \times 0.7 = 8.48 \text{ volts.}$$

Since e_O must = 5 volts, e_O must be reduced by:

$$\Delta E_O = 8.48 - 5 = 3.48 \text{ volts.}$$

The factor $b = (R_1 \times V) / R_3$ must reduce the output voltage by 3.48 volts. Therefore, at V = 5 volts:

$$b = 3.48 = (R_1 \times 5) / R_3 = 5 \times R_1/R_3$$

$$R_3 = 5 \times R_1/3.48 = 1.436 \times R_1$$

$$R_3 = KR_1, \text{ where } K = 1.436 \quad (3.22)$$

Using K = 1.436, A = 12.12, and equations (3.20) and (3.22):

$$A = (1 + R_1/R_2 + R_1/R_3) = 12.12$$

$$12.12 - 1 = R_1/R_2 + R_1/KR_1$$

$$11.12 = R_1/R_2 + 1/K$$

$$R_1 = 10.42 \times R_2 \quad (3.23)$$

Using equations (3.21) and (3.22) with V = 5 volts:

$$e_O = A\, e_{in} - (R_1 \times 5)/R3$$

$$e_O = A\, e_{in} - 5R_1/KR_1 = A\, e_{in} - 5/K = A\, e_{in} - 5/1.436$$

$$e_O = A\, e_{in} - 3.48 \quad (3.24)$$

Equation (3.24) shows that equation (3.22) satisfies the offset voltage requirement of 3.48 volts. Checking the values of R_1, R_2, and R_3 against the desired gain using equations (3.20), 3.22), and (3.23):

$$A = 1 + R_1/R_2 + R_1/R_3$$

$$A = 1 + (10.42 \times R_2)/R_2 + R_1/KR_1$$

$$A = 1 + 10.42 + 1/K$$

$$A = 1 + 10.42 + 0.7$$

$$A = 12.12 \quad (3.25)$$

The results of equation (3.25) shows that R_1, R_2, and R_3 have the correct relationship to achieve gain of 12.12. To get the values of R_1, R_2, and R_3, pick a value for R_1 and use equations (3.23) and (3.22) to calculate the values of R_2 and R_3, respectively.

3.8 LCD ON/OFF Response Times

The best method for evaluating the appearance of an LCD's ON/OFF response time is by using empirical data.

Figure 3-8a[51] shows the LCD response time definitions of T_{ON} (the LCD ON time) and T_{OFF} (the LCD OFF time). T_{ON} consists of a turn-on delay and a rise time; T_{OFF} consists of a turn-off delay and a decay (fall) time. Both delay times are virtually zero.

The turn-on delay time starts when the LCD voltage goes from V_{OFF} to V_{ON} and ends when the contrast rises to 10%. In some LCDs, the turn-on delay time can be as much as 1 ms, but even this time is usually a very small percentage of the T_{ON} rise time. The turn-on rise time is the time it takes the LCD to go from 10% to 90% contrast. The turn-off delay time starts when the LCD voltage goes from V_{ON} to V_{OFF} and ends when the contrast decays from 100% to 90%. The turn-off decay time is the time it takes for the LCD to go from 90% contrast to 10% contrast. Turn-on and turn-off delay times in LCD specifications are used so that they are consistent with JEDEC standards, which explains why these two virtually negligible times are mentioned.

LCD T_{ON} and T_{OFF} response times, shown in Figure 3-8a, are not related to the LCD operating frequency. These times are mainly a function of ambient temperature (T_A), operating voltage, and the distance between the glass plates that enclose the LCD. The viscosity of the liquid crystal (LC) fluid is a function of temperature. As the temperature decreases, the viscosity of the LC increases, thus increasing T_{ON} and T_{OFF}. An increasing temperature has the opposite effect.[52]

In a reflective LCD, T_{ON} is the time it takes to move the LCD molecules from their twisted nematic state (Figure 2-7) to the linear nematic state (Figure 2-8). T_{OFF} is the time it takes to move

[51] Smith, Paul.

[52] Rees, Lynn T.

the LC molecules back to the twisted nematic state from the linear nematic state. The higher the LC viscosity, the longer it takes to move the molecules between the twisted and untwisted states. The applied voltage also affects the response times. At lower temperatures, a higher voltage may be required to move the LC fluid. T_{ON} and T_{OFF} are also directly proportional to the square of the distance between the glass plates that enclose the LCD.[53] As the distance increases, the volume of LC material increases, thereby making more difficult to move the mass of LC material. At very low temperatures, these times could be as long as 1 to 2 seconds. If the temperature decreases too low, the LC may not be able to move without excessive force, which may not be practical to implement in an actual application.

[53] Seiko Instruments USA, Inc.

Figure 3-8. LCD Response Time

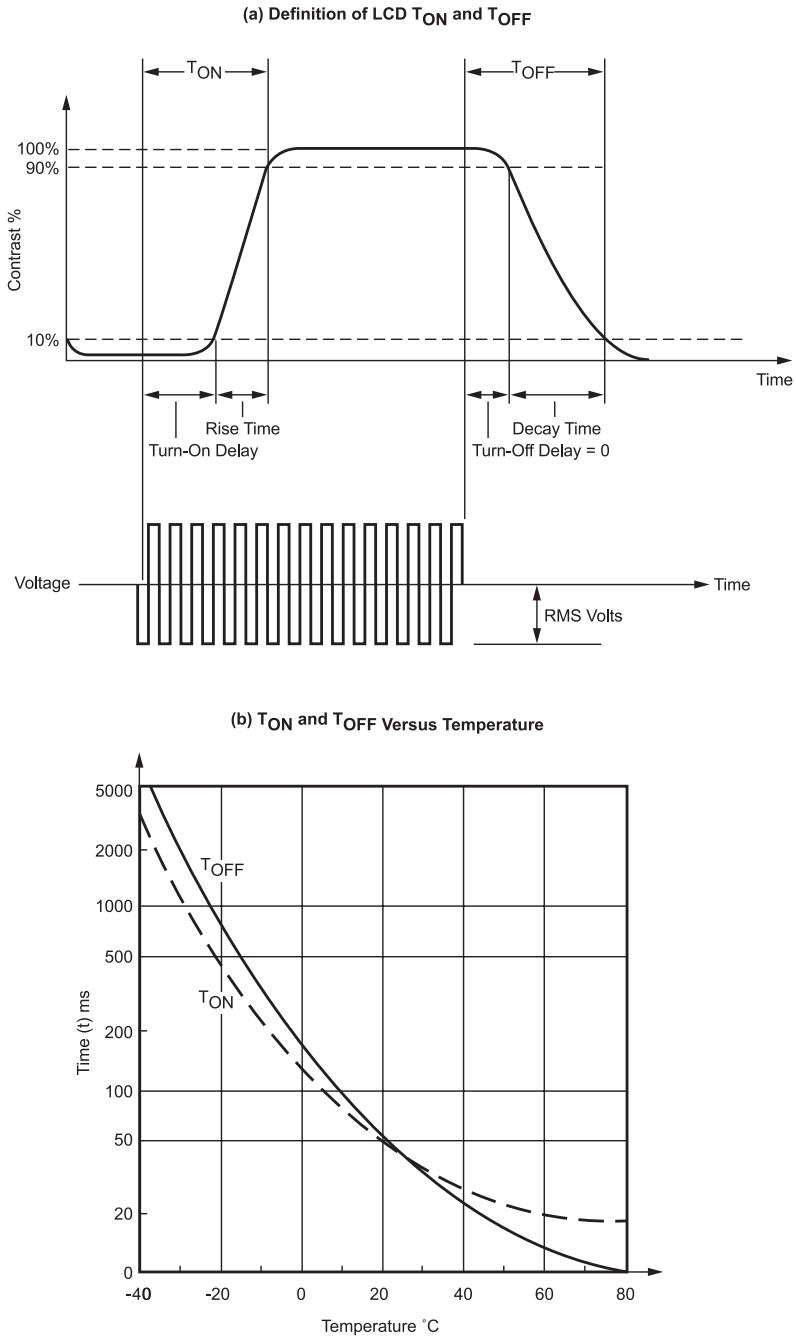


Figure 3-8b[54] shows a typical graph of T_{ON} and T_{OFF} times vs. temperature for an LCD operating at 4.5V and 100 Hz. Note that T_{OFF} is always larger than T_{ON} when $T_A \leq 20°C$. Temperatures above 20°C decrease the viscosity of the fluid to a point where the response time is fast enough to not be

an issue. LCD manufacturers offer multiple LC fluid types, and they can usually recommend a special LC fluid type for very low ambient temperature applications. Table 2-1 will be used as a specific example for a static LCD. T_{ON} can be as small as 15 ms and T_{OFF} as long as 1500 ms. The specifications in Table 2-4 apply to multiplexed LCDs, where T_{ON} is as low as 15 ms and T_{OFF} as long as 350 ms. A combination of the manufacturer's specifications and laboratory experiments should be used for final LCD response evaluations.

3.9 LCD Operating Frequency and Current

This section will show that to reduce the LCD operating current (and therefore power), the LCD must be operated at the lowest acceptable frequency. The lowest operating frequency with no flicker is typically about 50 to 60 Hz. Low operating current is especially important in battery-powered applications and LCDs are frequently used in this application. The frequency being discussed has a period = T_F, as used in the timing diagrams in section 2. For simplicity, *f* will be used in place of T_F. It should be emphasized again that using empirical data from laboratory evaluations is the very best way to evaluate the effect of frequency and current on LCD operation and appearance.

The LCD operating frequency and operating current are not related to the LCD ON/OFF response time. The frequency is a function of the equivalent circuit of the LCD. The lower limit of the operating frequency is the frequency at which there is visible flicker on the LCD. The upper limit is the frequency at which the display becomes dim (loses contrast). Typically, the lower limit is about 30 Hz and the upper limit about 500 to 1000 Hz.

A simplified equivalent circuit for a twisted nematic LCD display element (segment) is shown in Figure 3-9.[55] The voltage V_S across an LCD segment is:

$$V_S = V_{COM} - V_{SEG}$$

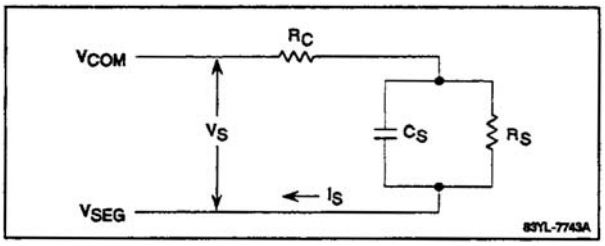
The values of R_C, R_S, and C_S are typical values and may change for different types and designs of LCDs. R_S and C_S are the equivalent resistance and capacitance of one LCD segment. The segment resistor Rs could range from 10 MΩ to 100 MΩ; its value is a function of the fluid type and the cell (containing the LC) thickness. The common resistance, Rc, is a combination of the electrode resistance and any feed-through holes (crossover points) in the LCD layout. R_C can vary from 10 kΩ to 150 kΩ. Narrowing (necking) in the electrode path and long runs of electrode etch increase

[54] Amperex.

[55] Standish Industries.

its resistance. Also, if there are not a lot of crossover points, or if a crossover point is poorly manufactured, the R_C will increase.[56]

Figure 3-9. LCD Segment Equivalent Circuit



Typical values for one LCD segment in a ½ inch, four-digit LCD are:

$R_S = 50$ Mohms

$C_S = 100$ pF.

The total electrode and common resistance for R_C for one LCD common connection is:

$R_C = 10$ kohms.

The segment impedance Z vs. f frequency for this element is:

$$Z = R_C + [(R_S \times 1/C_S S) / (R_S + 1/C_S S)]$$

$$Z = R_C + [(R_S \times 1/ C_S S)] / [(R_S C_S S + 1) / C_S S]$$

$$Z = R_C + [R_S / R_S C_S S + 1)]$$

$$Z = (R_C R_S C_S S + R_C + R_S) / (R_S C_S S + 1)$$

Where $S = j\omega$ and $\omega = 2\pi f$;

$$S = j\omega = j2\pi f.$$

Since $R_S >> R_C$: $R_S + R_C = R_S$:

$$Z = (R_C R_S C_S S + R_S) / (R_C R_S C_S S + 1)$$

$$Z = R_S[R_C C_S S + 1)] / (R_S C_S S + 1) \quad (3.26)$$

From equation (3.26), there is one pole at f_1 that occurs at:

$$R_S C_S S + 1 = j + 1 \quad (3.27)$$

Using this condition and substituting $S = j2\pi f_1$;

[56] Standish Industries.

$$j2\pi f_1 R_S C_S + 1 = j + 1$$

To meet this condition, the following must be true:

$$2\pi f_1 R_S C_S = 1$$

Solving for f_1:

$$f_1 = 1/2\pi R_S C_S \quad (3.28)$$

Using the values R_S = 50 Mohms and C_S = 100 pf, the pole at f_1 occurs at:

$$F_1 = 1/(2 \times 3.14 \times 50 \times 10^6 \times 100 \times 10^{-12}) = 31.8 \text{ Hz} \quad (3.29)$$

From equation (3.26), there is one zero at f_2 that occurs at:

$$R_S C_S S + 1 = j + 1 \quad (3.30)$$

Using this condition and substituting $S = j2\pi f_2$:

$$j2\pi f_2 R_C C_S + 1 = j + 1$$

To meet this condition, the following must be true:

$$2\pi f_2 R_C C_S = 1$$

Solving for f_2:

$$f_2 = 1/2\pi R_C C_S \quad (3.31)$$

Using the values R_C = 10 kohms and C_S = 100 pf, the zero at f_2 occurs at:

$$f_2 = 1/(2 \times 3.14 \times 10 \times 10^3 \times 100 \times 10^{-12}) = 159 \text{ kHz} \quad (3.32)$$

Using equations (3.27) and (3.30), a Bodie plot can be made as shown in Figure 3-10; the Y-axis on the left side of the Bodie plot is for Z. The plot shows the change in Z in decibels (db) vs. frequency f (= f_F). A 6 db change in Z is a change in Z by a factor of 2. A 20 db change in Z results in a change in Z by a factor of 10, and a change in f by a factor of 10. Hence, at 0 db, 50×10^6 ohms and at –20 db, $Z = 5 \times 10^6$ ohms. The Bodie plot can be verified by checking the start and end impedances. At low frequencies (f = 10 Hz), Cs is an open circuit and:

$$Z = R_C + R_S$$

Since $R_S >> R_C$:

$$Z = R_S = 50 \times 10^6 \text{ ohms}$$

At high frequencies (f = 160 kHz), Cs is a short circuit and:

$Z = R_C = 10$ kohms

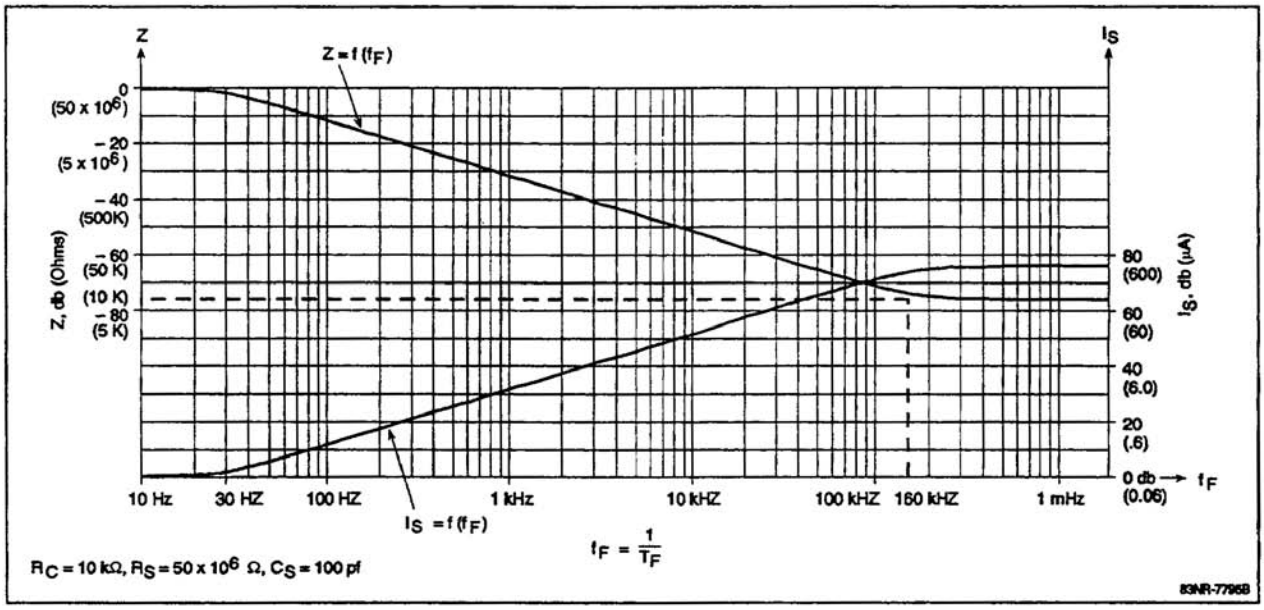
These start and end impedances are the same shown in the Bodie plot in Figure 3-10.

Using equation (3.36) and substituting the values for $R_C = 10 \times 10^3$, $R_S = 50 \times 10^6$, and $C_S = 10 \times 10^{-12}$ gives:

$$Z = 50 \times 10^6 \, (j2\pi f \times 10 \times 10^3 \times 100 \times 10^{-12} + 1) \, / \, (j2\pi f \times 50 \times 10^6 \times 100 \times 10^{-12} + 1)$$

$$Z = 50 \times 10^6 \, (j6.28) \times 10^{-6} f + 1) \, / \, (j31.4 \times 10^{-3} f + 1) \qquad (3.33)$$

Figure 3-10. Bodie Plot of f_F vs. Typical Segment Impedance and Operating Current



Using equation (3.33) at f = 60 Hz,

$$Z = 50 \times 10^6 \, (j6.28) \times 10^6 \times 60 + 1) \, / \, (j31.4 \times 10^{-3} \times 60 + 1)$$

$$Z = 50 \times 10^6 \, (j3.76 \times 10^{-4} + 1) \, / \, (j1.88 + 1) = 50 \times 10^6 / 2.13 = 23.47 \times 10^6 \text{ ohms}$$

Using equation (3.33) at f = 600 Hz,

$$Z = 50 \times 10^6 \, (j6.28 \times 10^{-6} \times 600 + 1) \, / \, (j31.4 \times 10^{-3} \times 600 + 1)$$

$$Z = 50 \times 10^6 \, (j3.76 \times 10^{-3} + 1) \, / \, (j18.84 + 1) = 50 \times 10^6 \, / \, 18.86 = 2.6 \times 10^6 \text{ ohms}$$

Going from frequency f – 60 Hz to f = 600 Hz reduces Z from 23.47 MΩ to 2.6 MΩ, a reduction of about 10:1 (–20 db).

A Bodie plot also can be made for the operating current, an important parameter discussed in this section. The current in one LCD segment (I_S) is:

$$I_S = (V_{COM} - V_{SEG})/Z$$

Using $V_{COM} - V_{SEG} = V_S$:

$$I_S = V_S/Z$$

Using equation (3.36)

$$I_S = V_S \times 1/Z = V_S \times [(R_SC_SS + 1) / (R_S \times (R_CC_SS + 1)]$$

$$I_S = V_S/R_S \times [R_SC_SS + 1) / (R_CC_SS + 1)]$$

There is a zero at $R_SC_SS + 1 = j + 1$. From equations (3.27), (3.28), and (3.29), the zero occurs at f_1 = 31.8 Hz. There is a pole at $R_SC_SS + 1 = j + 1$. From equations (3.30), (03.31), and (3.32) above, this occurs at f_2 = 159 kHz. Therefore, if V_S = 3V and $R_S = 50 \times 10^6$

$$I_S = 3/50 \times 10^6 = 0.06\ \mu A.$$

Using this information, a Bodie plot of IS vs. f is shown in Figure 3-9 using the right side of the graph as the Y-axis for IS. The initial value at 10 Hz is 0.06 μA. At f = 60 Hz, the increase is 7 db, which is a change by a factor of 2.23. Hence, $I_S = 2.23 \times 0.06\ \mu A = 0.13\ \mu A$.

The operating current for one LCD segment at V = 3 volts RMS and f + 600 Hz (Z + 2.6×10^6 ohms is:

$$I_S = 3/2.6 \times 106 = 1.15\ \mu A.$$

Hence, the operating current is much less at low frequencies than at high ones. At 60 Hz, one LCD segment uses 0.13 μA. For a four-digit, 0.5-inch-high, seven-segment-plus-decimal-point LCD, the operating current with all segments and one decimal point on would be:

$$I_{LCD} = 0.13\ (4 \text{ digits} \times 7 \text{ segments/digit} + 1 \text{ DP})$$

$$I_{LCD} = 0.13 \times 29 = 3.77\ \mu A$$

If the LCD were being operated at f = 600 Hz, the operating current would be:

$$I_{LCD} = 1.15 \times 29 = 33.4\ \mu A$$

This is an increase in current of about a factor of 10 (= 20 db).

The LCD controller supplies the current (I_{LCD}) for the switching waveforms that turn the LCD on or off. Ambient light supplies the power to make the display visible. Many applications use battery-powered LCDs. In these applications, battery life is very important, and in most, the LCD

is operated when the rest of the circuitry is in sleep (or stop) mode. In this mode at 3V, the operating current (I_{DDS}) of NEC Electronics microntrollers without the LCD operating is about 10 μA. The current with the LCD operating is I_{TOT}.

Without LCD operating,

$I_{TOT} = I_{DD}$ at 3V = 10 μA (max.).

With LCD operating,

$I_{TOT} = I_{DD}$ at 3V = I_{LCD}

With LCD operating at f = 60 Hz,

I_{TOT} = 10 + 3.77 = 13.77 μA

With LCD operating at f = 600 Hz,

I_{TOT} = 10 + 33.4 = 43.4 μA

An LCD operating at f = 60 Hz will extend battery life significantly compared to an LCD operating at f = 600 Hz. For this reason, the LCD should be operated at the lowest possible frequency, where there is good contrast and no flicker. To avoid flicker, an LCD should be operated at about 50 Hz. It is very advantageous to operate the LCD at low frequencies to minimize power and maximize battery life.

3.10 Ghosting and Dimming

Ghosting and dimming are phenomena that may occur in some LCDs. They usually occur if the LCD is poorly designed or poorly manufactured. They also may occur if the LCD is improperly driven. Ghosting may appear when LCDs are operated at high voltages or high frequencies; dimming may appear when LCDs are operated at high frequencies. Ghosting manifests itself when segments that should be off become partially or fully on; dimming manifests itself when an LCD becomes dim.[57]

Figures 3-11 and 3-12 will be used to illustrate ghosting in a static mode LCD. Figure 3-11 shows a 4-digit static LCD with 28 segments (S1–S28) and one common. Figure 3-12 shows the common and segment waveforms. In this example, segments S1–S27 have V_{ON} at their inputs and segment S28 has V_{OFF} at its input.

[57] Standish Industries.

The impedance between a segment input and the common is:

$$Z = R_C + [R_S \times 1/C_S S/(R_S + 1/C_S S)] = R_C + [R_S/(R_S C_S S + 1)].$$

Since $S = j\omega = j2\pi f$:

$$Z = R_C + R_S/(j2\pi f R_S C_S + 1) \quad (3.34)$$

The LCD is being driven at 15 volts RMS, f = 50 Hz and has the following values:

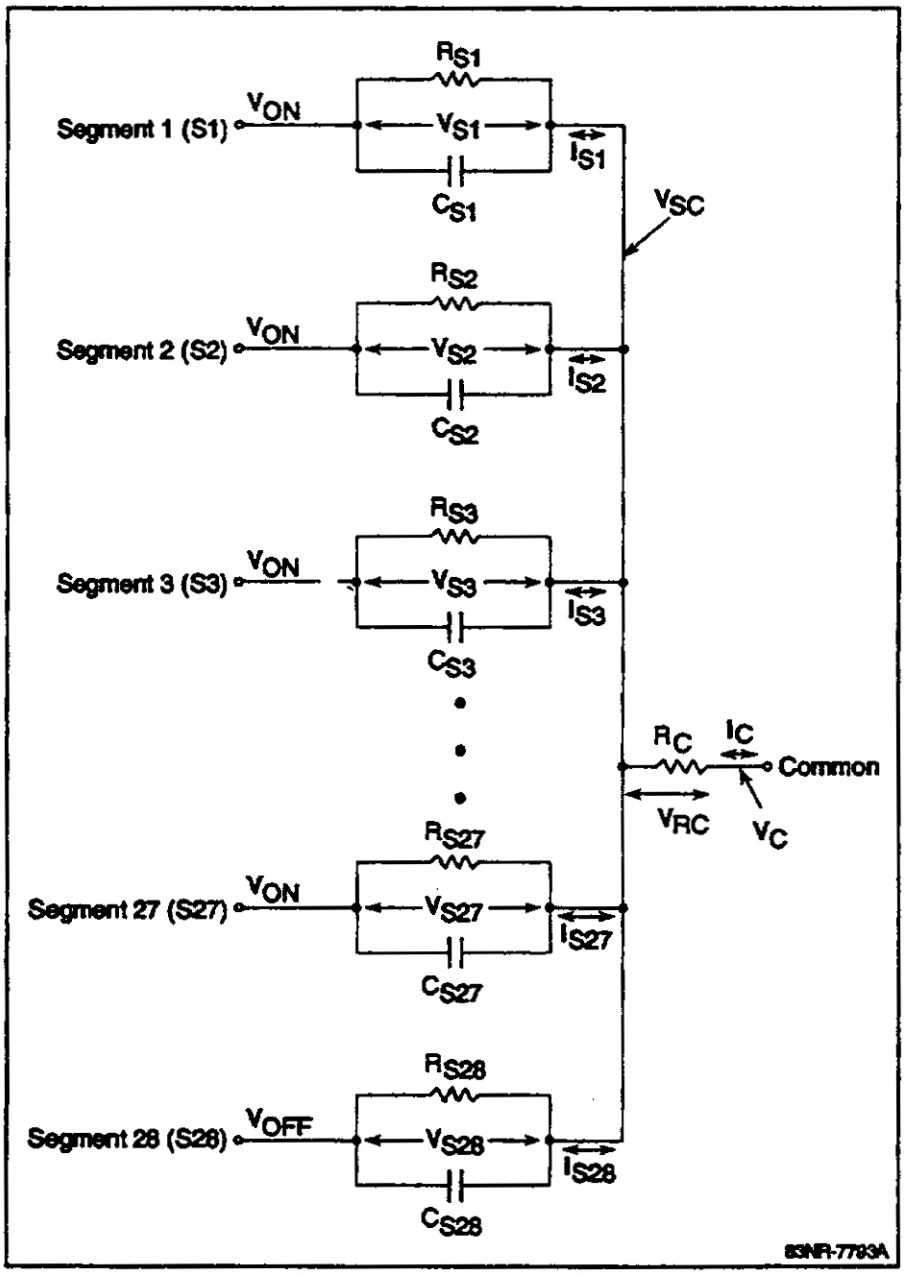
C_S = 100 pF

R_S = 20 Mohms

R_C = 10 kohms (good design)

R_C = 150 kohms (poor design)

Figure 3-11. Multiple LCD Segments



Using the poor design where R_C = 150 kohms:

$$Z = 150 \times 10^3 + [20 \times 10_6 / (j2\pi \times 50 \times 20 \times 10^6 \times 100 \times 10^{-12} + 1)]$$

$$Z = 150 \times 10^3 + 16.9 \times 10^6 = 17 \times 10^6 \text{ ohms}$$

The current thru one ON segment I_S is:

$I_S = (V_{ON} - V_C)/Z = 15/17 \times 10^6 = 0.88\ \mu A/segment$

If all of the segments except one are ON, then 27 segments are ON. The current from all segments flows through resistor R_C. The common current I_C is:

$I_C = 27\ segments \times 0.88\ \mu A/segment = 23.76\ \mu A$

The voltage drop across the common resistor is:

$V_{RC} = I_C \times R_C = 23.76 \times 10^{-6} \times 150 \times 10^3 = 3.5\ volts\ RMS$

In Figure 2-12, the segment ON voltage (V_{ON}) is out of phase with the common voltage (V_C) and the segment OFF voltage (V_{OFF}) is in phase with V_C. Normally the voltage across an OFF segment in the static mode would be 0 volts RMS; this is not true in this example. Using Figure 2-12, the voltage at V_{SC} can be calculated as follows:

During t1, $V_C = 15$ volts, $V_{ON} = 0$ volts, the current I_C flows from the common to the input of the ON segments, and V_{SC} is:

$V_{SC} = V_C - I_C \times R_C = 15 - 3.5 = 11.5\ volts$

During t2, $V_C = 0$ volts, $V_{ON} = 15$ volts, the current I_C flows from the input of the ON segments to the common, and V_{SC} is:

$V_{SC} = V_C + I_C \times R_C = 0 + 3.5 = 3.5\ volts$

The voltage across segment 28 (V_{S28}) is:

$V_{S28} = V_{OFF} - V_{SC} = 3.5\ volts\ squarewave$

Hence;

$V_{S28} = 3.5\ volts\ RMS$

If any static LCD were being used from Figure 1-10 $T_A = 25°C$, $V_{S28} \geq V90$ for all cases and any of the segments for any of these LCDs would be ON. In fact, at $T_A = 0°C$, all LCD segments except fluid type 7 would be fully ON. This is an example of how ghosting occurs.

Assume the LCD is a good design where $R_C = 10$ kohms. Using equation (3.34), the impedance between a segment input and the common is

$Z = R_C + [R_S/(j2\pi \times f\ R_S C_S + 1)]$

$Z = 10 \times 10^3 + [20 \times 10^6/(j2\pi \times 50 \times 20 \times 10^6 \times 100 \times 10^{-12} + 1]$

$Z = 10 \times 10^3 + [20 \times 10^6/(j \times 0.63 + 1)] = 10 \times 10^3 + (20 \times 10^6/1.18)$

$Z = 10 \times 10^3 + 16.9 \times 10^6 = 16.9 \times 10^6\ ohms$

The current thru one ON segment I_S is:

$I_S = (V_{ON} - V_C)/Z = 15/16.9 \times 10^6 = 0.89\ \mu A/segment$

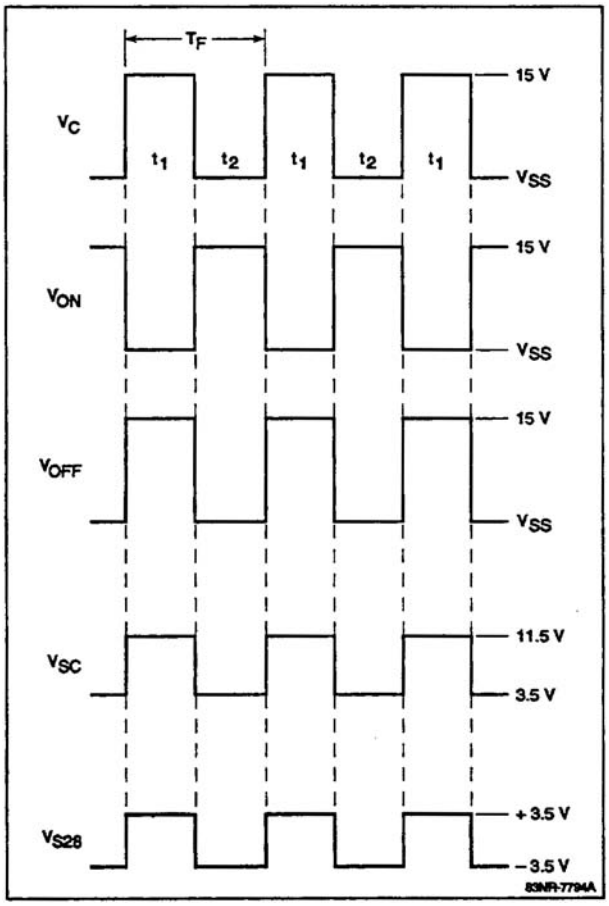
The common current I_C is:

I_C = 27 segments × 0.89 μA/segment = 23.96 μA

The voltage drop across the common resistor is:

$V_{RC} = I_C \times R_C = 23.96 \times 10^{-6} \times 10 \times 10^3 = 0.24$ volts

Figure 3-12. Voltages Resulting in Ghosting



Using the above technique, the voltage across segment S28 is:

V_{S28} = 0.24 volts RMS

Using this value, none of the LCD segments in Table 2-1 would be ON.

Ghosting can also be produced by operating the LCD at higher frequencies.[58] Using equation (3.34), with $R_C = 150 \times 10^3$ ohms, operating at 5 volts RMS, and f = 500 Hz:

$Z = R_C + [R_S/(j2\pi \times R_S C_S + 1)]$

[58] Standish Industries.

$Z = 150 \times 10^3 + [20 \times 10^{-6}/j2\pi \times 500 \times 20 \times 10^6 \times 100 \times 10^{-12} + 1)]$

$Z = 150 \times 10^3 + [20 \times 10^6/j6.28+1)] = 150 \times 10^3 + (20 \times 10^6/6.36) = 150 \times 10^3 + 3.14 \times 10^6$

$Z = 3.29 \times 10^6$ ohms

$I_S = 5/3.29 \times 10^6 = 1.52$ μA/segment

$I_C = 27$ segments × 1.52 μA/segment = 41.0 μA

The voltage drop across the resistor is:

$V_{RC} = I_C \times R_C = 41 \times 10^{-6} \times 150 \times 10^3 = 6.15$ volts

Using the above technique, the voltage across S28 is:

$V_{S28} = 6.15$ volts RMS

In this example, by increasing the frequency to 500 Hz, the voltage across segment S28 is increased by 6.15 volts RMS, which would turn ON any of the segments in Table 2-1. This is a second example of ghosting.

There are multiple ways to avoid ghosting. In the first example above, ghosting could be avoided by driving the LCD at voltages much lower than 15 volts. In the second example, ghosting would be avoided by operating the LCD at 50 Hz. In both examples, the LCD operating current would be reduced. Another way to minimize ghosting in static LCDs is to connect all unused segments to the backplane commons. Section 2 showed that a static LCD's V_{OFF} voltage is zero volts; connecting unused segments to the backplane common and selecting an LCD with a V_{th}=2V would probably eliminate static mode ghosting in applications with a 5-volt power source.

In multiplexed LCDs, ghosting can be eliminated by driving unused inputs to the OFF state and by not overdriving the LCD. Also ghosting in multiplexed LCDs can be eliminated by adjusting potentiometer R× so as to reduce the voltage at the resistor ladder and by keeping a tight control on the power supply voltage. A final way to reduce the chance of ghosting is to not add any external resistor in series with the common. Adding an external resistor is equivalent to increasing the value of R_C. As shown above, this can change a good design into a poor design or make a poor design worse, and it always increases the chance of ghosting.[59]

When LCDs are operating at high frequencies, they become dim. At 2 kHz, the impedances across the LCD (C_S and R_S) are at a lower value. The result is that about 1/3 of the applied voltage is across the electrode resistor R_C and 2/3 is across the LCD segment. Since the voltage across the LCD segment is reduced, the contrast of the LCD may be reduced, as proven in the following example.

Assume the LCD in Figure 3-11 is operating at V = 5 volts RMS and f = 2 kHz. Also assume that it has the following values:

C_S = 200 pF

R_S = 20 Mohms

R_C = 150 kohms

Using equation (3.34) and the values given above, the impedance of C_S is:

$$Z_S = R_S/(j2\pi f \times R_S C_S + 1)$$

$$Z_S = [20 \times 10^6/(j2\pi \times 2 \times 10^3 \times 20 \times 10^6 \times 200 \times 10^{-12} + 1)]$$

$$Z_S = [20 \times 10^6/(J50.3 + 1)] = 2 \times 10^6 / 50.3$$

$$Z_S = 397 \times 10^3 \text{ ohms}$$

The voltage across the segment (across C_S) would be reduced from 5 volts RMS to:

$$V_S = [397 \times 10^3/(397 \times 10^3 + 150 \times 10^3)] \times 5 = 3.62 \text{ volts RMS.}$$

In some LCDs, this voltage may be reduced enough so that contrast would be reduced. If the LCD were operated at 50 Hz, then:

$$Z_S = 20 \times 10^6/j2\pi\ 50 \times 20 \times 10^6 \times 10^{-12} + 1)]$$

$$Z_S = 20 \times 10^6 / (j1.25 + 1) = 20 \times 10^6/1.6$$

$$Z_S = 12.5 \times 10^6\ \Omega$$

$$V_S = [12.5 \times 10^6/(12.5 \times 10^6 + 150 \times 10^3)] \times 5 = 4.94 \text{ volts RMS}$$

Hence, when operating the LCD at 50 Hz, virtually the full voltage is across the segment and there is no dimming.

[59]Standish Industries

4. PROGRAMMING OF NEC ELECTRONICS LCD CONTROLLERS

Sections 2 and 3 of this manual discussed all factors relevant to the hardware design application of an LCD. This section discusses how to program NEC Electronics LCD controllers to achieve the desired results on an LCD. The LCD used will be a seven-segment display. The exact LCD type will be different in each programming example, and each is described in its respective subsection. This section also includes information about the use and programming of icons. Four program examples using the 75X microcontrollers are included: one for static mode, duplex mode, triplex mode, and quadruplex mode. There is also a programming example for the NEC Electronics 78K0 and one for the 78K0S microcontroller.

- 75X Programming Examples
  - » Structure of 75X Programming Examples
  - » ASCII to Seven-Segment Data Conversion Icons
  - » 75X Static Mode Programming Example
  - » 75X Duplex Mode Programming Example
  - » 75X Triplex Mode Programming Example
  - » 75X Quadruplex Mode Programming Example
- Programming NEC's 78K0 and 78K0S Microcontrollers
  - » 78K0S Programming Example
  - » 78K0 Programming Example

4.1 75X Programming Examples

The four 75X programming examples are very similar. This section is written so that similar programming examples are described in a common place. The subsections for each mode describe the areas where are differences among examples. The initialization routines and the handling of icons are so similar that they both are described in "Structure of Programming Examples (Section 4.1.1)." The area where there is the most difference among the modes is the storing of the seven-segment date in the LCD data display area. These routines will be described in detail in their respective subsection. Also, minor differences such as operating frequency or differences in the LCD being used are described in the individual sections.

One critical area is how the LCD display is connected to the LCD controller segment lines. Being careful about how the LCD is connected can make it easy to move data into the LCD data display area, resulting in compact assembly code. A seven-segment data table must be made for each mode. The software uses the table to convert ASCII numbers from ASCII format to seven-

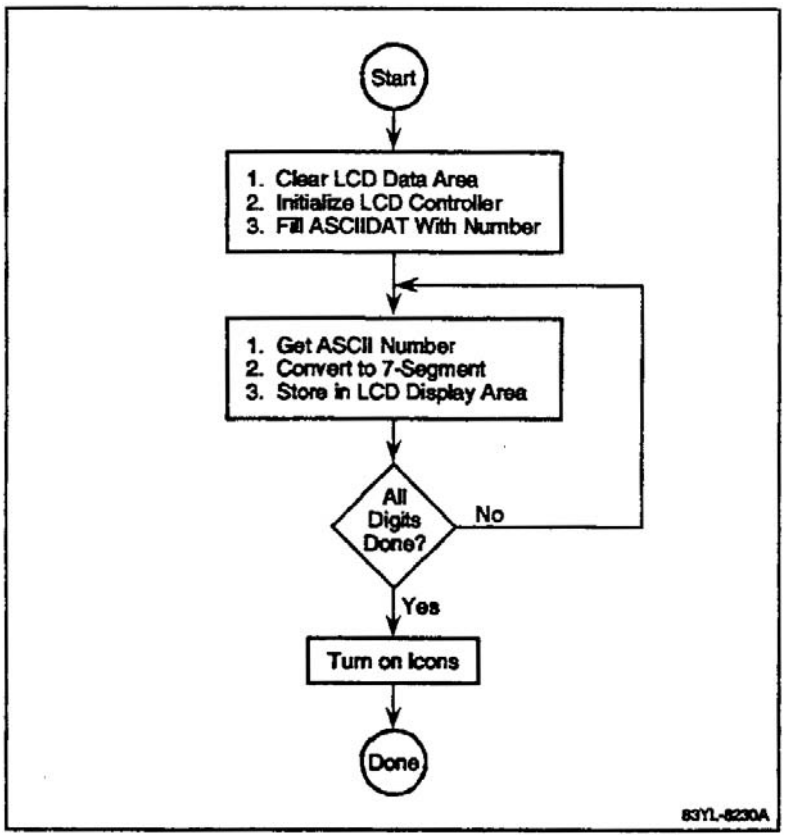
segment format. The way in which the LCD is wired will affect how the bits in the data conversion table are packed.

4.1.1 Structure of 75X Programming Examples

With the exception of the example for static mode, all programming examples have the same structure consisting of a main routine and three subroutines. The example for static mode has four subroutines.

All four 75X examples basically perform the same functions, as shown in Figure 4-1. An area in memory called ASCIIDAT is filled with numbers in ASCII format. The ASCII numbers are converted into seven-segment format and stored in the LCD data display area. After that, a subroutine is called that turns ON one or more icons. All four programs were checked by connecting an LCD and verifying that the correct data was displayed. A “1” in any bit location in the LCD display area causes an LCD segment to be ON and a “0” causes it to be OFF.

Figure 4-1. Flowchart for All Four Modes



If you want to describe the program flow in a list format, it might be good to explain it here.

1. The MAIN routine calls the CLRLCDDA (clear LCD data) subroutine (Figure 4-2), which initializes the 75X LCD data display area locations (1E0H–IFFH) to all zeros. All four programs use this subroutine.
2. MAIN programs the Watch Mode (WM) Register to select the subsystem clock.
3. MAIN programs the LCD mode (LCDM) register to use S24–S31 as LCD segment outputs, the common and segment outputs are enabled, and the LCD mode and frequency are programmed as described in the individual subsections.
4. MAIN then puts four to eight ASCII-formatted numbers into the ASCIIDAT RAM area. The number and value of the digits stored depends upon which of the four modes is used. The numbers are always stored in consecutive ascending order. For example, in the programming example for static mode, the LCD would display 123.4. Storage of the ASCII number would normally be done in a subroutine, but it is done in MAIN to simplify the program.
5. MAIN then calls the CONVERT subroutine to retrieve the ASCII digits from the ASCIIDAT area, convert them to seven-segment format, and store them in the LCD data display area (RAM locations 1E0H–1FFH in memory bank 1). Register pair HL can be used to point to any location in any RAM bank. Register pairs DE and DL can only be used to point to locations in memory bank 0. Getting the ASCII data, converting it to seven-segment data, and storing it in the LCD data display area is a memory bank-to-memory bank move with a conversion. In the four programming examples, register DE always points to locations in memory bank 0. When memory bank 1 is accessed, register HL points to locations in memory bank 1.
6. When the CONVERT subroutine is finished, it returns to MAIN, which then calls the ICON ADJUST subroutine to turn ON one or more icons.
7. When ICON ADJUST is completed, it returns to MAIN and goes into an infinite loop.

In the programming example for static mode, two subroutines are used in place of the one CONVERT subroutine described above. In the Static Mode, after ASCII data is stored in ASCIIDAT, the MAIN routine calls the STACONV subroutine. STACONV gets the ASCII data from ASIIDAT, converts it to seven-segment format, and stores the result in SEVENSEG. When STACONV is finished, it returns to MAIN, which calls STASTORE to take the data from SEVENSEG and store it in the LCD data display area.

4.1.2 ASCII to Seven-Segment Data Conversion

The ASCII input data must be converted from ASCII format to seven-segment format, which is defined here as the data format necessary for information to be displayed on a seven-segment LCD. The conversion is achieved using a conversion table in ROM and the MOVT instruction. This section discusses how to pack the seven-segment data, the use of the MOVT instruction, how to convert ASCII data to seven-segment data, and presents an example of how to build the ROM conversion table.

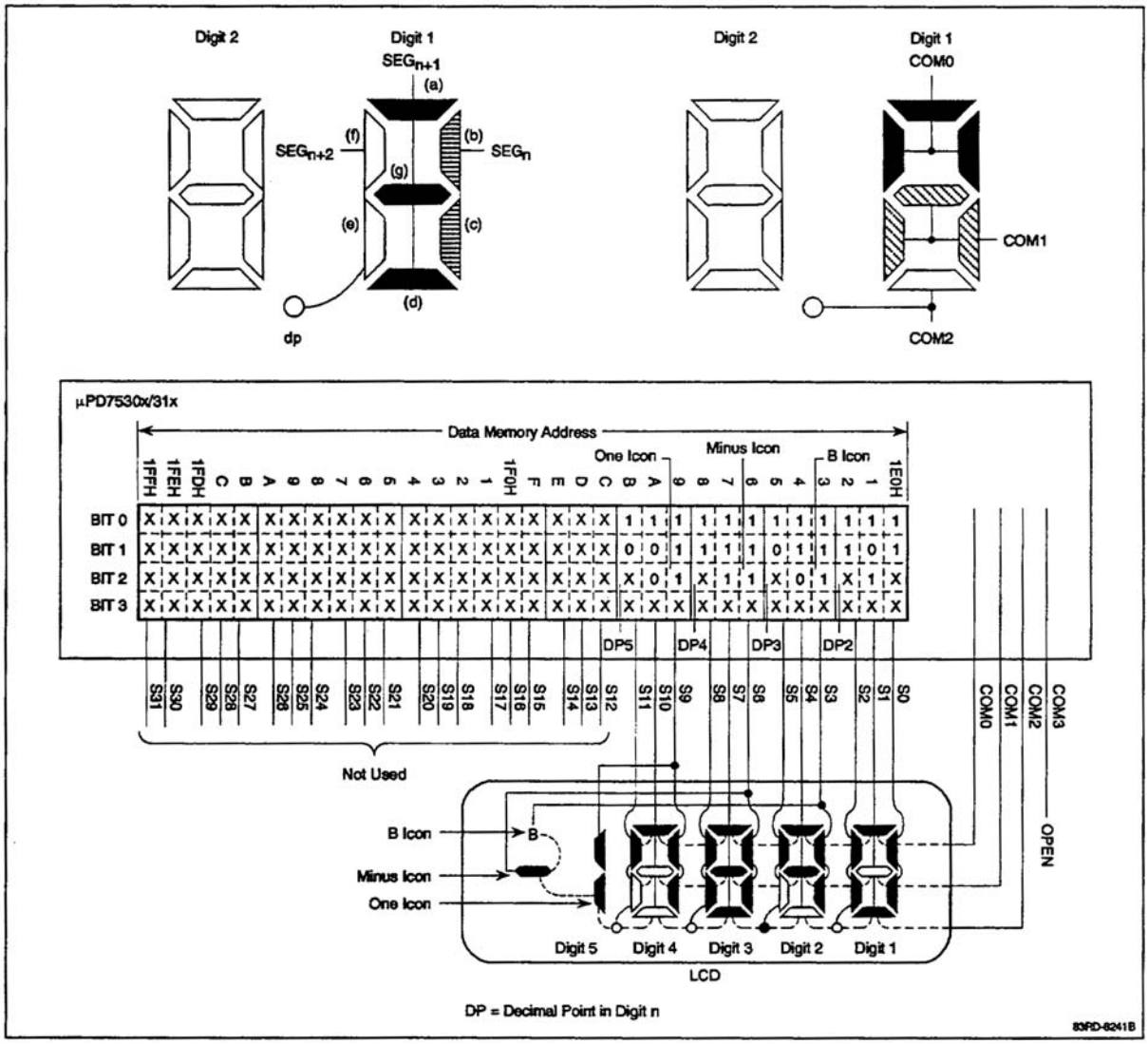
Figure 4-2. CLRLCD Subroutine

```
75X SERIES ASSEMBLER V4.00

   ** CLEAR LCD DATA AREA                                    **

 COMMAND      : -C308 CLRLCD.ASM

  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1
    2                          $              TITLE = 'CLEAR LCD DATA AREA'
    3                          $              PL = 66
    4                          $              PW = 120
    5
    6                                         NAME    CLRLCD
    7                                         PUBLIC  CLRLCDDA
    8
    9                                 ;*********************************************
   10                                 ;  This subroutine clears the LCD Data Display  *
   11                                 ;  area 01E0H TO 01FFH to all 0H.               *
   12                                 ;*********************************************
   13
   14 ----                    CLEAR           CSEG    INBLOCK
   15 0000   9907             CLRLCDDA:       PUSH    BS              ;Save Bank Select REG.
   16 0002   9D90                             SET1    MBE             ;Select memory
   17 0004   9911                             SEL     MB1             ;          Bank 1.
   18 0006   8BE0                             MOV     HL,#0E0H
   19 0008   70                               MOV     A,#0
   20 0009   E8               DISPLINT:       MOV     @HL,A           ;Initialize
   21 000A   C2                               INCS    L               ;   LCD Display
   22 000B   FD                               BR      DISPLINT        ;     Area in Bank 1
   23 000C   C3                               INCS    H               ;         with all zeros.
   24 000D   FB                               BR      DISPLINT
   25 000E   9906                             POP     BS              ;Restore BS REG.
   26 0010   EE                               RET
   27                                         END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Triplex mode will be used as an example, because it is the most complicated mode. Figure 4-3 shows the LCD that will be used in the programming example for the triplex mode. The way in which the LCD is constructed dictates how the LCD is connected to the microcontroller. In triplex mode, three segment lines are required to control one digit, and each LCD controller segment line controls up to three LCD segments.

Figure 4-3. Programming Example for Four-Digit Triplex Mode

Note: Please note in the table representing the RAM data display area, the least significant bit (bit 0) is on the left and the most significant bit (bit 3) is on the right. This is opposite the normal convention.



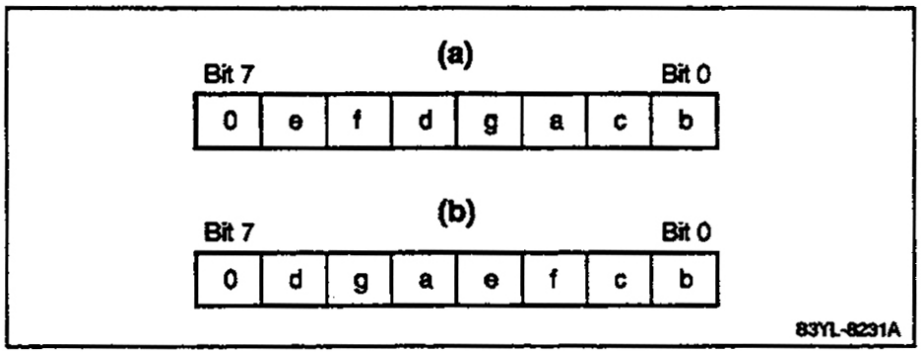
Each digit uses three nibbles of RAM in the LCD display area. There are two segment lines that control three LCD segments and one that controls two LCD segments (Figure 4-3). The first digit in Figure 4-3 is controlled by S0, S1, and S2 from the LCD controller. As a result of the LCD configuration, LCD segments b and c are stored in the first nibble of a digit. Segments a, g, and d are stored in the second. Segments f, e, and the dp (the decimal point of the next digit) are stored in the third.

The way in which the data is packed will have an effect on the complexity of the subroutine that stores the data in the LCD data display area. Before making the conversion table, it is necessary to decide how the data will be packed. Since the ROM conversion table will not control the decimal points, only seven of the eight bits in the ROM table will be used. The unused ROM bit will always be 0. In this example, the right-most digit's segments b and c will be stored in the first

nibble of RAM (location 1E0H, bits 0 and 1), segments a, g, and d in the second (1E1H, bits, 0, 1 and 2), and f and e in the third (1E2H, bits 0 and 1).

In this example, Figure 4-4 shows two ways to pack the seven-segment data in the ROM and assumes that register HL was previously set up and points to RAM location 1E0H. The data is obtained from the ROM table and put into register XA using the MOVT instruction (explained later in this document).

Figure 4-4. Packing of Seven-Segment Data in Triplex Mode



If register XA is filled with ROM packed as shown in Figure 4-4a, then the sequence of the subroutine that stores one LCD digit into the LCD data display area would be as shown in Figure 4-5.

Figure 4-5. Sequence of Subroutine Instructions for Data Packed as Shown in Figure 4-4a

```
MOV     @HL, A      ;Store  segments b and c in 1E0H.
INCS    L           ;Point to second location.
MOV     TEMP,A      ;Store segments b, c, a, and g in TEMP.
MOV     A,X         ;Get segment d and put it
RORC    A           ;     into the carry bit.
MOV     A,TEMP      ;Get back segments b, c, a and g.
RORC    A           ;Get segments a, g and d
RORC    A           ;     into position and store
MOV     @HL,A       ;          segments a, g and d in 1E1H
INCS    L           ;Point to third location.
MOV     A,X         ;Get segments d, f and e into Reg A.
RORC    A           ;Get f and e into position
MOV     @HL,A       ;     and store in 1E2H.
```

If the data is packed as shown in Figure 4-3b, then the sequence of the subroutine would be as listed in Figure 4-6.

Figure 4-6. Sequence of Subroutine Instructions for Data Packed as Shown in Figure 4-4b

```
MOV     @HL,A       ;Store segments b and c in 1E0H.
INCS    L           ;Point to second location.
RORC    A           ;Get f and e
RORC    A           ;    into position
MOV     TEMP,A      ;        and store in TEMP.
MOV     A,X         ;Get a, g and d into Reg A
MOV     @HL,A       ;    and store in 1E1H.
INCS    L           ;Point to third location.
MOV     A,TEMP      ;Get back f and e.
MOV     @HL,A       ;    and store in 1E2H.
```

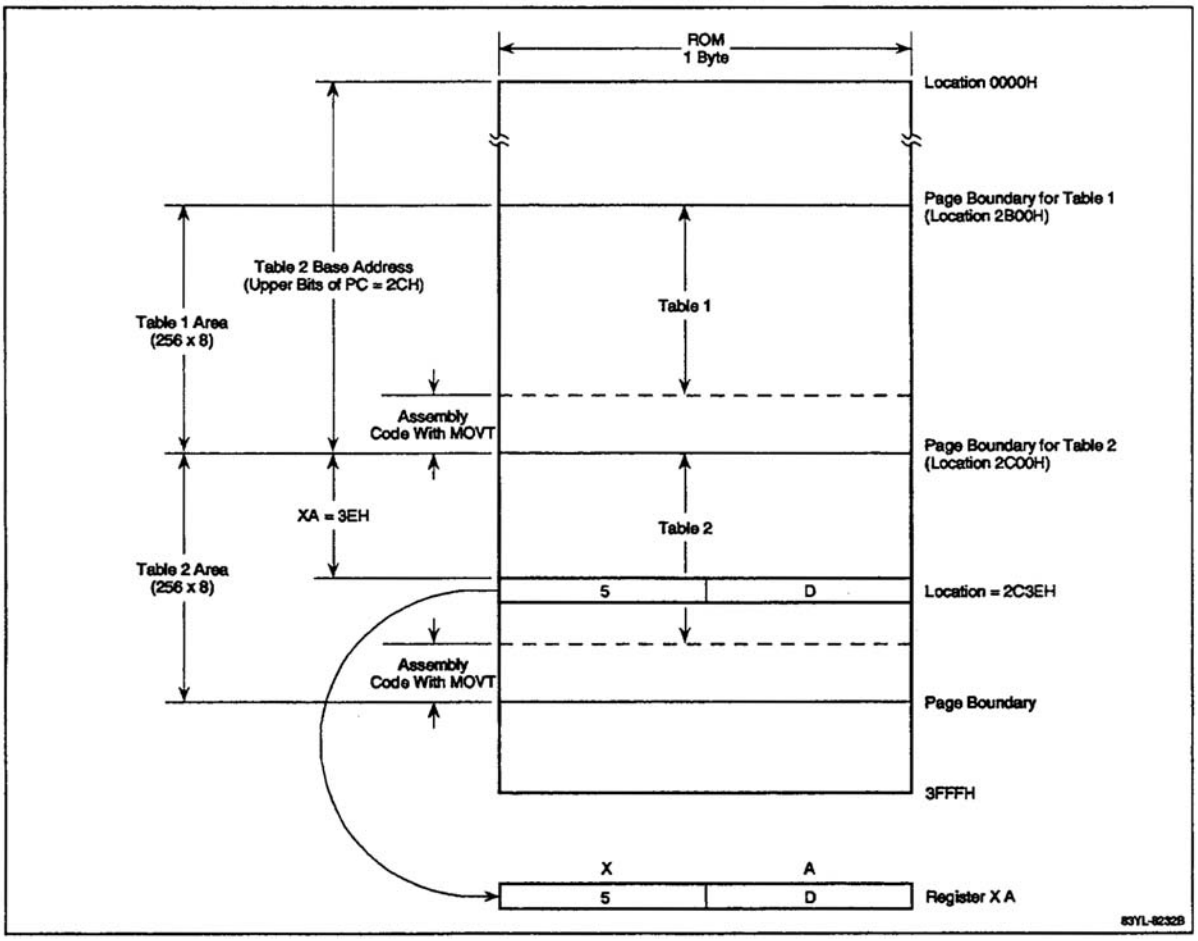
The two routines illustrate that it takes fewer instruction to move ROM data into the LCD display area by packing it as shown in Figure 4-3b.

The MOVT instruction gets data by indexing into a table in ROM using register XA as an offset into the ROM table. The program counter's (PC) upper bits are the base address of the beginning of the ROM table. The MOVT instruction has two formats, the first of which will be used in this application:

- MOVT XA,@PCXA
- MOVT XA,@PCDE

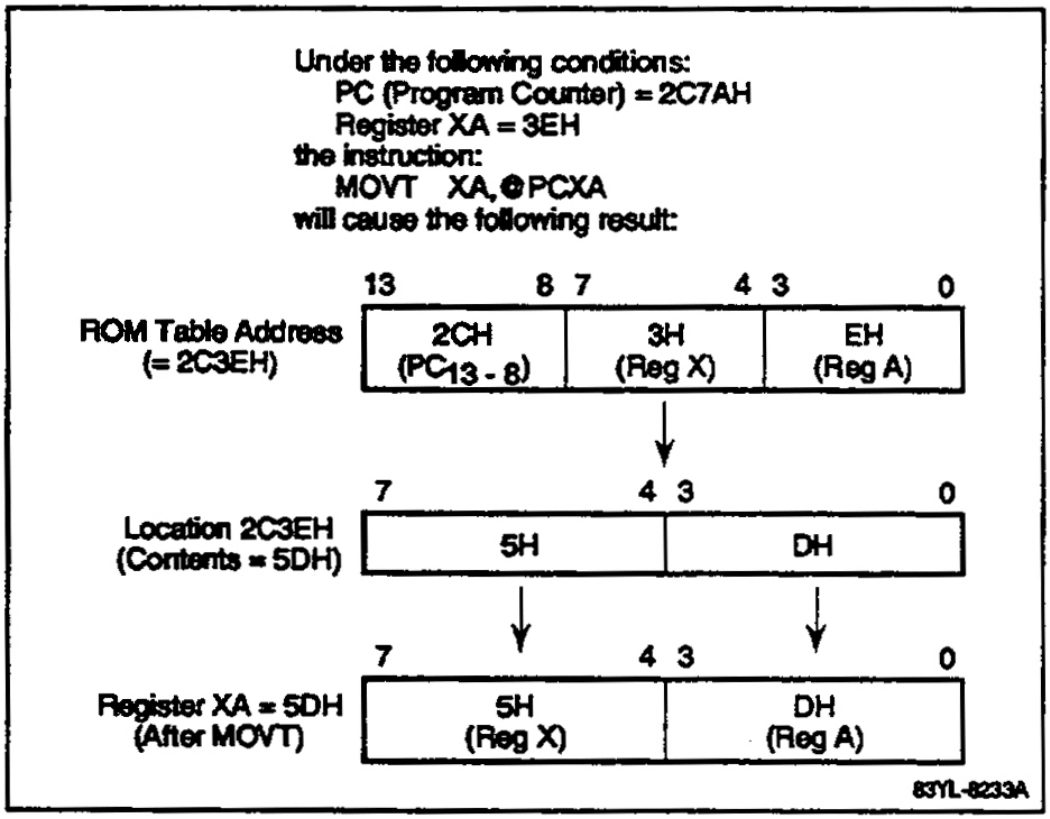
Figures 4-7 and 4-8 show how the MOVT instruction works. Figure 4-7 shows a 16 KB block of ROM. Addresses to access locations in Table 2 of Figure 4-7 are formed using the PC and the contents of XA. The upper bits of the PC are used for the upper address bits of the location, and the contents of XA are used for the lower eight address bits. The upper PC bits are the ROM table's base address and contents of XA are the displacement into the table. Register XA contains 03EH (the lower eight address bits); the upper bits of the address are taken from the PC and are 02CH.

Figure 4-7. ROM Addressing for MOVT Instruction



The ROM location is formed by concatenating XA with the upper PC bits. The address formed is 02C3EH and this location contains 05DH. When the MOVT instruction is executed, the contents of location 02C3EH are stored in register XA. In this example, XA will contain a 5DH after the instruction is executed. The ROM tables do not have to be located on 'Page Boundaries,' but putting the table there results in simpler code. Because the range of the MOVT instruction is eight bits, the MOVT instruction must be contained within the page defined by the upper bits of the PC.

Figure 4-8. MOVT Instruction Example



The conversion of an ASCII value requires a ROM table with seven-segment data and the use of the MOVT instruction. The ASCII equivalent of the numbers 0–9 are shown in Table 4-1. Table 4-2 shows the number and the ROM location that contains the seven-segment data for that number.

1. The first step of the conversion process is to get the ASCII number and put it into the XA register.
2. Next replace the 3 in the ASCII number with a 0. For example, if the number 4 is stored in ASCII format, the ASCII code is 34H. Replacing the 3 with a 0 changes the number to 04H.
3. Table 4-2 shows that the ROM location of the number 4 is XX04H. Executing a 'MOVT XA, PCXA' instruction with the number 04H in XA will cause the value in ROM location xx04H to be stored in XA. Putting the seven-segment number corresponding to04H in location xx04H of the ROM will causes the XA register to be loaded with the seven-segment equivalent of the number 4.

In this manual, a ROM table contains the seven-segment data that will be stored in the LCD display data area. The table always starts on a page boundary. The numbers 0–9 are displayed on

a seven-segment LCD display as shown in Figure 2-24. A ROM table with seven-segment data can be constructed using Figures 2-24 and Figure 4-4b. Figure 2-24 shows the OFF/ON segments for the numbers 0–9, and Figure 4-4b shows the bit positions for segments a–g. A 1 in a ROM bit turns ON a segment; a bit with a 0 turns OFF a segment. In this example, Figure 2-24 shows that to display the number '4', segments b, c, f, and g must be ON and segments a, d, and e must be OFF. Using Figure 4-4b, the bit pattern that will be put in location XX04 for the number 4 is:

Bit 7	Bit 0
00100111	

To display a 4 on the LCD, a 27H must be put into location XX04H of the ROM table. The value 27H is the seven-segment data for the number 4. This technique applies to seven-segment data for the remaining numbers.

Table 4-1. ASCII Code for Numbers 0–9

Number	ASCII Code
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39

Table 4-2. Number vs. ROM Location with Seven-Segment Data

Number Being Converted	ROM Location of Seven-Segment Data
0	xx00H
1	xx01H
2	xx02H
3	xx03H
4	xx04H
5	xx05H
6	xx06H
7	xx07H
8	xx08H
9	xx09H

4.1.3 Icons

Icons are special characters on an LCD. An icon could be a PLUS SIGN (+), the letter "B", a decimal point, or any other special symbol. All icons are programmed using bit manipulation instructions. When a display is to be updated, the memory location is read to get the current state of the icon. The icon state is combined with the new data and written to the memory location just read; this way the state of the icon is unchanged.

The program in Figure 4-9 is taken from the programming example for the triplex mode and shows how icons are handled.

Figure 4-9. List of Locations for the Icons

```
DP2     EQU     0E2H.2      ;Digit 2 DP at location 1E2 bit 2.
DP3     EQU     0E5H.2      ;Digit 3 DP at location 1E5 bit 2.
DP4     EQU     0E8H.2      ;Digit 4 DP at location 1E8 bit 2.
DP5     EQU     0EBH.2      ;Digit 5 DP at location 1EB bit 2.
BICON   EQU     0E3H.2      ;BICON at location 1E3 bit 2.
DASH    EQU     0E6H.2      ;DASH ICON at location 1E6 bit 2.
ONE     EQU     0E9H.2      ;ONE ICON at location 1E9 bit 2.
```

The program in Figure 4-10 turns on the DP3, BICON, DASH, and ONE icons using bit manipulation instructions.

Figure 4-10. Bit Manipulation Instructions That Turn ON DP3 and the BICON, DASH, and ONE Icons

```
ICON      CSEG    INBLOCK
TRIICON   PUSH    BS              ;Save BS register.
          SET1    MBE             ;
          SEL     MB1             ;Select memory Bank 1.
          MOV     H,#BICON SHR 6  ;Loads E into reg H.
          SET1    @H+DP3          ;Turn ON the DP3 icon.
          SET1    @H+BICON        ;Turn ON the BICON.
          SET1    @H+DASH         ;Turn ON the DASH icon.
          SET1    @H+ONE          ;Turn ON the ONE icon.
          POP     BS
          RET
          END
```

In order to program a RAM bit which corresponds to an icon, the location of the bit must be defined. The equate (EQU) statements shown above (Figure 4-9) are the RAM locations of the icons. For example, BICON is an icon located in RAM bank 1 at location 0E3 bit 2. The BICON is a “B” on the LCD.

In the program shown above, the ‘SET1’ and ‘SEL MB1’ instructions select memory bank 1. The instruction:

```
MOV     H,#BICON     SHR6
```

loads the value E into register H. The assembler treats the EQU values as numbers. The notation ‘#BICON SHR 6’ tells the assembler to get the value of BICON (which is E3.2) and shift it right 6 places (SHR 6). In binary form, BICON looks like:

1110 0011 10 (E3.2)

Shifting this value right 6 places leaves the result:

1110

which is the value EH that is stored into register H. When the program is assembled, the expression BICON (in the SET1 @H+BICON instruction) causes the following address to be generated:

0011 10

This is the lower 6 bits of the address. The expression @H+BICON concatenates 0011 10 onto register H to form the address:

1110 0011 10

This is the address E3 bit 2. The instruction:

SET1 @H+BICON

will cause bit 2 at location E3H in RAM bank 1 to be set. The instruction:

CLR1 @H+BICON

would clear the bit. In the subroutine example shown above, four icons (see Figure 4-2) are set.

DP3:	decimal point for digit 3
BICON:	the 'B' on the LCD
DASH:	the 'dash' on the LCD
ONE:	the '1' on the LCD

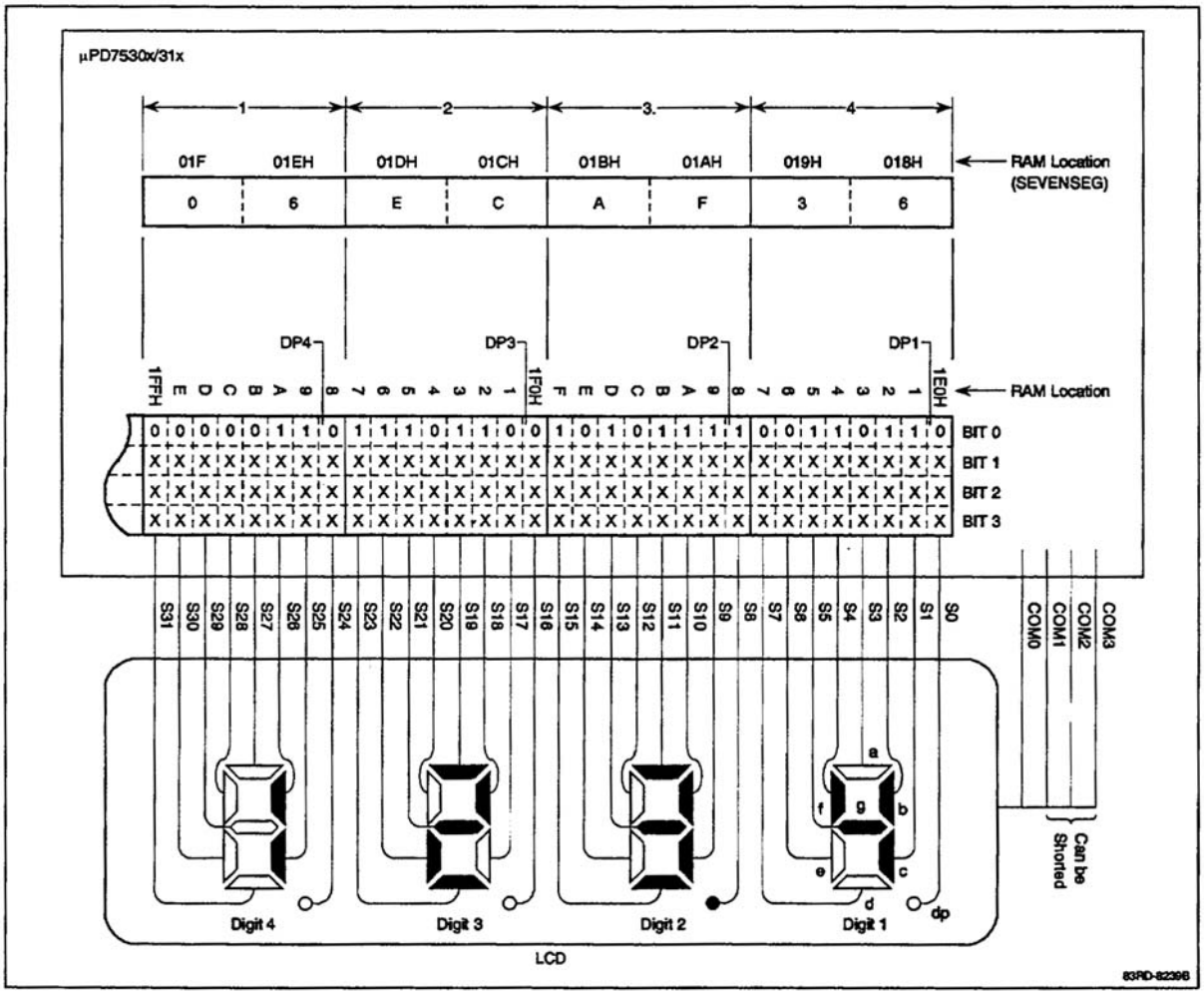
This technique allows bits to be set or reset easily.

4.1.4 Programming Example for a 75X Microcontroller in Static Mode

The LCD that will be used in this example is shown in Figure 4-11 and has four digits. The program causes the LCD to display 123.4.

Figure 4-11. LCD for Static Mode Programming Example

Note: Please note in the table representing the RAM data display area, the least significant bit (bit 0) is on the left and the most significant bit (bit 3) is on the right. This is opposite the normal convention.



The programming example in Figure 4-11 consists of a MAIN routine and four subroutines as shown in Figures 4-12 to 4-15:

- STATMAIN Main routine
- CLRLCD CLRLCDDA subroutine (shown earlier in Figure 4-2)
- STATCONV STATCONV subroutine

- STATMOVE STASORE subroutine
- SICONADJ STATICON subroutine

Figure 4-12. STATMAIN Routine for 75X Microcontroller in Static Mode

```
75X SERIES ASSEMBLER V4.00
   ** STATIC MAIN ROUTINE V1.0                                    **

 COMMAND      : -C308 STATMAIN.ASM

  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                          $              TITLE = 'STATIC MAIN ROUTINE V1.0'
    2                          $              PL = 66
    3                          $              PW = 120
    4
    5
    6                                         NAME    STATMAIN
    7                                 EXTRN   CLRLCDDA, STACONV, STASTORE, STATICON
    8                                         PUBLIC  ASCIIDAT, SEVENSEG
    9
   10 0000 R 0000                             VENT0   MBE = 0, RBE = 0, MAIN
   11
   12                                        ;*************************************************************
   13                                        ;  This is the Static Mode initialization and Main routine. It  *
   14                                        ;  CALLS the CLRLCDDA routine which sets the LCD Data Display   *
   15                                        ;  area to all zeros. It sets the LCD Controller to the Static  *
   16                                        ;  Mode @ 64 Hz. It then stores ASCII digits 1, 2, 3, & 4       *
   17                                        ;  in the ASCIIDAT storage area. It then calls the STACONV      *
   18                                        ;  subroutine, which converts the input ASCII data to seven-segment *
   19                                        ;  data and stores it in the SEVENSEG data area. It CALLS       *
   20                                        ;  routine STASTORE which moves the seven-segment data to the LCD   *
   21                                        ;  Display Area. It calls STATICON which turns ON digit 2 DP.   *
   22                                        ;*************************************************************
   23
   24 ----                    ASCII          DSEG    AT      010H
   25 0010                    ASCIIDAT:      DS      8               ;Reserves 4 bytes for ASCII data.
   26 0018                    SEVENSEG:      DS      8               ;Reserves 4 bytes for seven-segment
                                                                             data.
   27
   28 ----                    STAMAIN        CSEG    INBLOCK
   29 0000   9CB2             MAIN:          DI                      ;Disable interrupts.
   30 0002   9C90                            CLR1    MBE             ;Select lower 0 & upper 15.
   31 0004 E AB4000                          CALL    !CLRLCDDA       ;Clear LCD Data Display area to 0.
   32 0007   8905                            MOV     XA,#05H         ;Set Watch Mode (WM)
   33                                                                ;       Register To Subsystem
   34 0009   9298                            MOV     WM,XA           ;           Clock & Enable Timer.
   35 000B   890C                            MOV     XA,#0CH         ;Enable Commons, Segments
   36 000D   928C                            MOV     LCDM,XA         ;   and set to Static Mode @ 64 Hz.
   37 000F   75                              MOV     A,#05H          ;Enable LCD
   38 0010   938E                            MOV     LCDC,A          ;       commons and segments.
   39 0012   8B10                            MOV     HL,#ASCIIDAT    ;Point to start of ASCII Data.
   40 0014   8934                            MOV     XA,#034H        ;Initialize XA W/an ASCII 34H.
   41 0016   9A3F                            MOV     B,#03H          ;Set data counter.
   42 0018   AA10             STOREDAT:      MOV     @HL,XA          ;Save ASCII Data.
   43 001A   C8                              DECS    A               ;Decrement ASCII Number by -1.
   44 001B   60                              NOP                     ;Prevent skip.
   45 001C   C2                              INCS    L               ;Change pointer to
   46 001D   C2                              INCS    L               ;       new ASCII location.
   47 001E   CF                              DECS    B               ;Change number of digits saved.
   48 001F   F8                              BR      STOREDAT
   49 0020 E AB4000                          CALL    !STACONV        ;Convert data to seven-segment format.
   50 0023 E AB4000                          CALL    !STASTORE       ;Stores seven-segment data in LCD area.
   51 0026 E AB4000                          CALL    !STATICON       ;Turn ON decimal point.
   52 0029   60               DONE:          NOP                     ;Done, loop.
   53 002A   FE                              BR      DONE
   54                                        END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-13. STATCONV Subroutine for 75X Microcontrollers in Static Mode

```
75X SERIES ASSEMBLER V4.00

   ** CONVERT ASCII TO seven-segment STATIC V1.0                    **

 COMMAND      : -C308 STATCONV.ASM

  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT
     1                          $               TITLE = 'CONVERT ASCII TO seven-segment STATIC V1.0'
     2                          $               PL = 66
     3                          $               PW = 120
     4
     5
     6                                          NAME    STATCONV
     7                                          EXTRN   ASCIIDAT, SEVENSEG
     8                                          PUBLIC  STACONV
     9
    10                                  ;***********************************************
    11                                  ;                                             *
    12                                  ; This subroutine gets the ASCII data, converts *
    13                                  ; it to seven-segment data and stores the result    *
    14                                  ; in the SEVENSEG area.                       *
    15                                  ;                                             *
    16                                  ;***********************************************
    17
    18                            ; The following are seven-segment display codes for a Quadruplexed LCD
    19
    20 ----                     DISPSEG                 CSEG    PAGE
    21 0000   DE                                       DB      0DEH            ;Display a 0.
    22 0001   06                                       DB      006H            ;Display a 1.
    23 0002   EC                                       DB      0ECH            ;Display a 2.
    24 0003   AE                                       DB      0AEH            ;Display a 3.
    25 0004   36                                       DB      036H            ;Display a 4.
    26 0005   BA                                       DB      0BAH            ;Display a 5.
    27 0006   FA                                       DB      0FAH            ;Display a 6.
    28 0007   1E                                       DB      01EH            ;Display a 7.
    29 0008   FE                                       DB      0FEH            ;Display a 8.
    30 0009   BE                                       DB      0BEH            ;Display a 9.
    31
    32
    33 000A   9C90               STACONV:        CLR1    MBE             ;Select lower Bank 1 & upper Bank 15.
    34 000C E 8D00                                     MOV     DE,#ASCIIDAT    ;Point DE to address of ASCII data.
    35 000E E 8B00                                     MOV     HL,#SEVENSEG    ;Point HL to seven-segment Date.
    36 0010   9A3E                                     MOV     C,#03H          ;C keeps track of bytes converted.
    37 0012   E4                 CONVDAT:        MOV     A,@DE           ;Get ASCII lower data nibble.
    38 0013   9A09                                     MOV     X,#0H           ;Make upper nibble BCD = 0.
    39 0015   D0                                       MOVT    XA,@PCXA        ;Convert ASCII to seven-segment data.
    40 0016   AA10                                     MOV     @HL,XA          ;Store converted data byte.
    41 0018   C4                                       INCS    E               ;Point to
    42 0019   C4                                       INCS    E               ;       next ASCII byte.
    43 001A   C2                                       INCS    L               ;Point to next
    44 001B   C2                                       INCS    L               ;       seven-segment data area.
    45 001C   60                                       NOP                     ;Do not want a SKIP on second INCS L.
    46 001D   CE                                       DECS    C               ;Change byte count.
    47 001E   F3                                       BR      CONVDAT         ;Not done, convert more data.
    48 001F   EE                                       RET                     ;Conversion done, return.
    49                                                 END

 TARGET CHIP : UPD75308

 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-14. STATMOVE Subroutine for75X Microcontrollers in Static Mode

```
75X SERIES ASSEMBLER V4.00

   ** MOVE seven-segment DATA TO LCD DATA STORAGE                    **

 COMMAND      : -C308 STATMOVE.ASM


  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                          $              TITLE = 'MOVE seven-segment DATA TO LCD DATA STORAGE'
    2                          $              PL = 66
    3                          $              PW = 120
    4
    5
    6                                         NAME    STATMOVE
    7                                         EXTRN   SEVENSEG
    8                                         PUBLIC  STASTORE
    9
   10                                  ;*****************************************************
   11                                  ;                                                    *
   12                                  ; This subroutine gets seven-segment data from SEVENSEG     *
   13                                  ; and moves it to the LCD Data Display area 1E0H-1FFH.  *
   14                                  ;                                                    *
   15                                  ;*****************************************************
   16
   17
   18 ----                      MOVE           CSEG    INBLOCK
   19 0000   9907              STASTORE:      PUSH    BS              ;Save bank select register.
   20 0002 E 8D00                             MOV     DE,#SEVENSEG    ;Point DE to seven-segment data.
   21 0004   9D90                             SET1    MBE             ;Select memory
   22 0006   9911                             SEL     MB1             ;         Bank 1.
   23 0008   8BE0                             MOV     HL,#0E0H        ;Point HL to LCD Display Data.
   24 000A   9A3E              GETNEXT:       MOV     C,#03H          ;C keeps track of bits done.
   25 000C   E4                               MOV     A,@DE           ;Get ASCII data nibble.
   26 000D   E8                MOVEBIT:       MOV     @HL,A           ;Store bit into LCD Data Area.
   27 000E   98                               RORC    A               ;Get next bit into position.
   28 000F   C2                               INCS    L               ;Update LCD Data nibble pointer low.
   29 0010   60                               NOP                     ;Don't want to skip; use NOP.
   30 0011   CE                               DECS    C               ;Count bit.
   31 0012   FA                               BR      MOVEBIT         ;Nibble not done, do another bit.
   32 0013   C4                               INCS    E               ;Update seven-segment nibble pointer.
   33 0014   60                               NOP                     ;Don't want to skip; use NOP.
   34 0015   9A02                             SKE     L,#0H           ;Did Reg L overflow?
   35 0017   F2                               BR      GETNEXT         ;No, get next nibble.
   36 0018   C3                               INCS    H               ;Yes, update LCD nibble pointer
   37                                                                 ;       high. If H overflows, done.
   38 0019   F0                               BR      GETNEXT         ;No Reg H overflow, get next nibble.
   39 001A   9906                             POP     BS              ;Reg h overflow, Restore BS register.
   40 001C   EE                               RET                     ;Finished moving data, return.
   41                                         END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

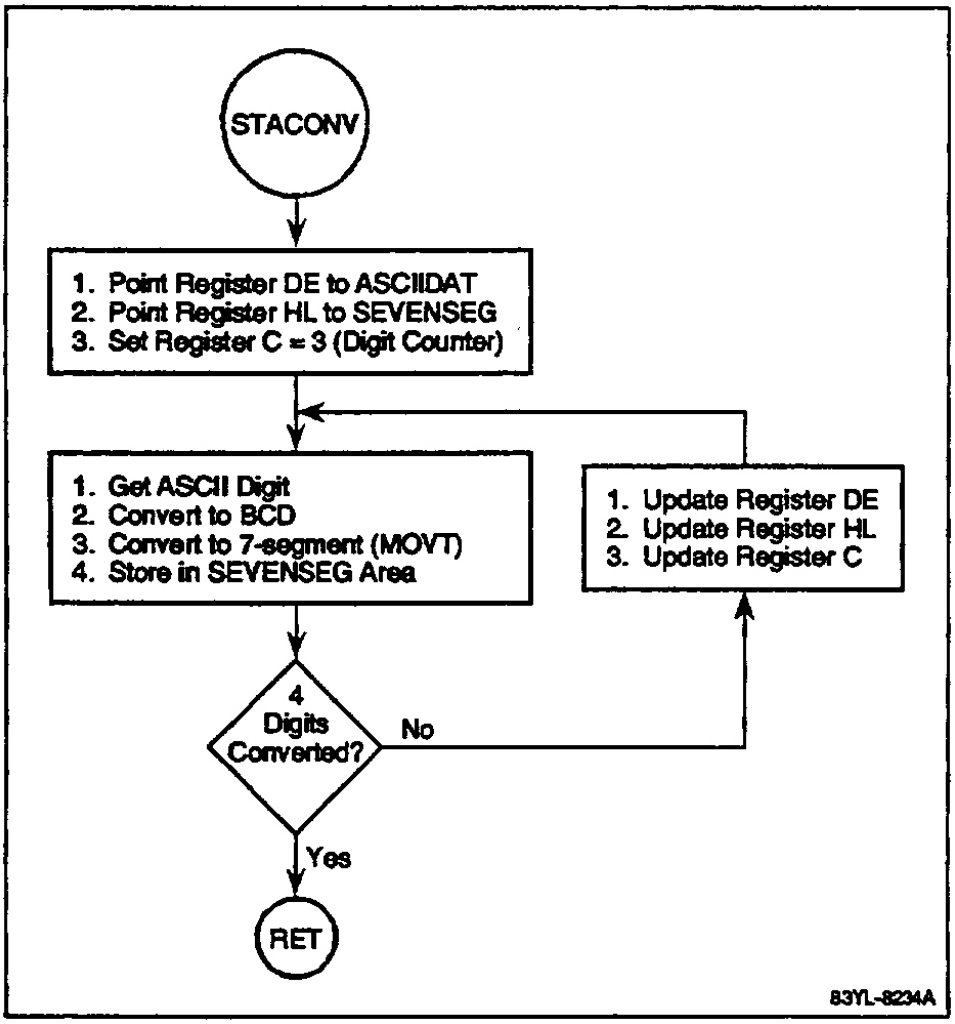
Figure 4-15. SICONADJ Subroutine for 75X Microcontrollers in Static Mode

```
75X SERIES ASSEMBLER V4.00
   ** ADJUST STATIC ICONS                                    **

 COMMAND      : -C308 SICONADJ.ASM


  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                          $              TITLE = 'ADJUST STATIC ICONS'
    2                          $              PL = 66
    3                          $              PW = 120
    4
    5                                         NAME    SICONADJ
    6                                         PUBLIC  STATICON
    7
    8
    9                           ;The following is a list of locations for the ICONs.
   10
   11        (0380)             DP1            EQU     0E0H.0          ;Digit 1 DP at location 1E0 bit 0.
   12        (03A0)             DP2            EQU     0E8H.0          ;Digit 2 DP at location 1E8 bit 0.
   13        (03C0)             DP3            EQU     0F0H.0          ;Digit 3 DP at location 1F0 bit 0.
   14        (03E0)             DP4            EQU     0F8H.0          ;Digit 4 DP at location 1F8 bit 0.
   15
   16                                        ;*********************************************
   17                                        ;  Decimal points are handled as ICONS and are  *
   18                                        ;  turned ON or OFF using bit manipulation      *
   19                                        ;  instructions. This subroutine will turn ON   *
   20                                        ;  the DP2 ICON.                               *
   21                                        ;*********************************************
   22
   23
   24 ----                    ICON           CSEG    INBLOCK
   25 0000   9907             STATICON:      PUSH    BS                      ;Save BS register.
   26 0002   9D90                            SET1    MBE                     ;Select memory
   27 0004   9911                            SEL     MB1                     ;          Bank 1.
   28 0006   9AEB                            MOV     H,#DP2  SHR 6           ;Loads 0EH into Reg H.
   29 0008   9D08                            SET1    @H+DP2                  ;Turn ON the DP2 ICON.
   30 000A   9906                            POP     BS
   31 000C   EE                              RET
   32                                        END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

The STATMAIN routine calls the CLRLCDDA subroutine, which clears the LCD data area to all zeros, programs the LCD controller to operate in Static Mode at 64 Hz by setting the LCDM register to 0CH, and then stores numbers 1234 in ASCII format in ASCIIDAT.

STATMAIN calls the STATCONV subroutine, a flowchart of which is illustrated in Figure in 4-16. This subroutine is a code segment (CSEG) starting on a PAGE boundary. The first ten locations are a ROM table that contains the numbers 0–9 in static seven-segment format. STATCONV gets a digit from ASCIIDAT, converts it to static seven-segment format using the ROM table and MOVT instruction, and stores the seven-segment data in SEVENSEG. STATCONV uses register DE as a pointer to ASCIIDAT and register HL as a pointer to SEVENSEG. Register C keeps track of the number of digits converted and stored.

Figure 4-16. ASCII to Seven-Segment Data Conversion in Static Mode



After STATCONV converts and stores the four ASCII digits, the subroutine returns the program to STATMAIN, which then calls the STASTORE subroutine. A flowchart of the algorithm used to store one digit is shown in Figure 4-17. STASTORE takes the seven-segment data in SEVENSEG and stores it in the LCD data display area 1E0H–1FFH using register DE as a pointer to SEVENSEG and register HL as a pointer to the LCD data area. Figure 4-18 shows the algorithm used for the data transfer. When four digits have been stored, the subroutine returns control to STATMAIN.

STATMAIN then calls the STATICON subroutine to turn ON decimal point DP2 and returns control to STATMAIN, which then goes into an infinite loop.

Figure 4-17. Storing of Seven-Segment Data in LCD Memory in Static Mode

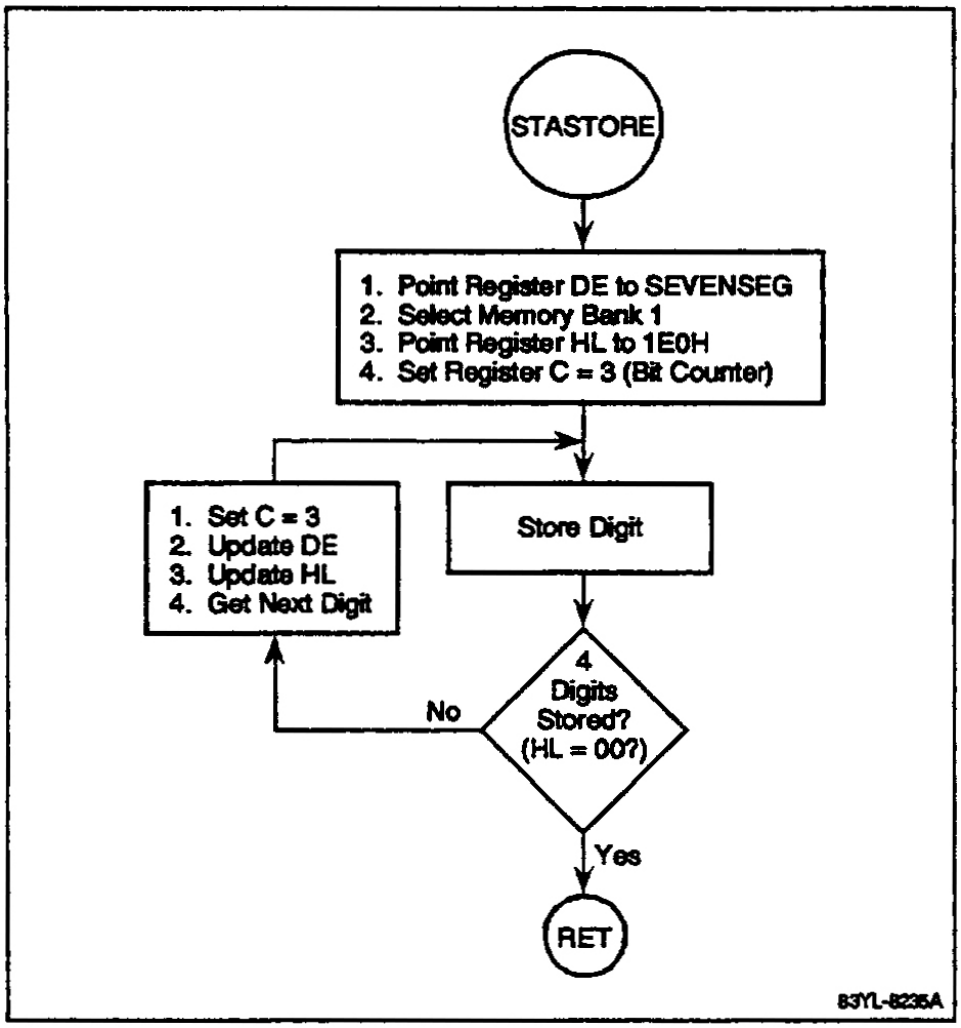
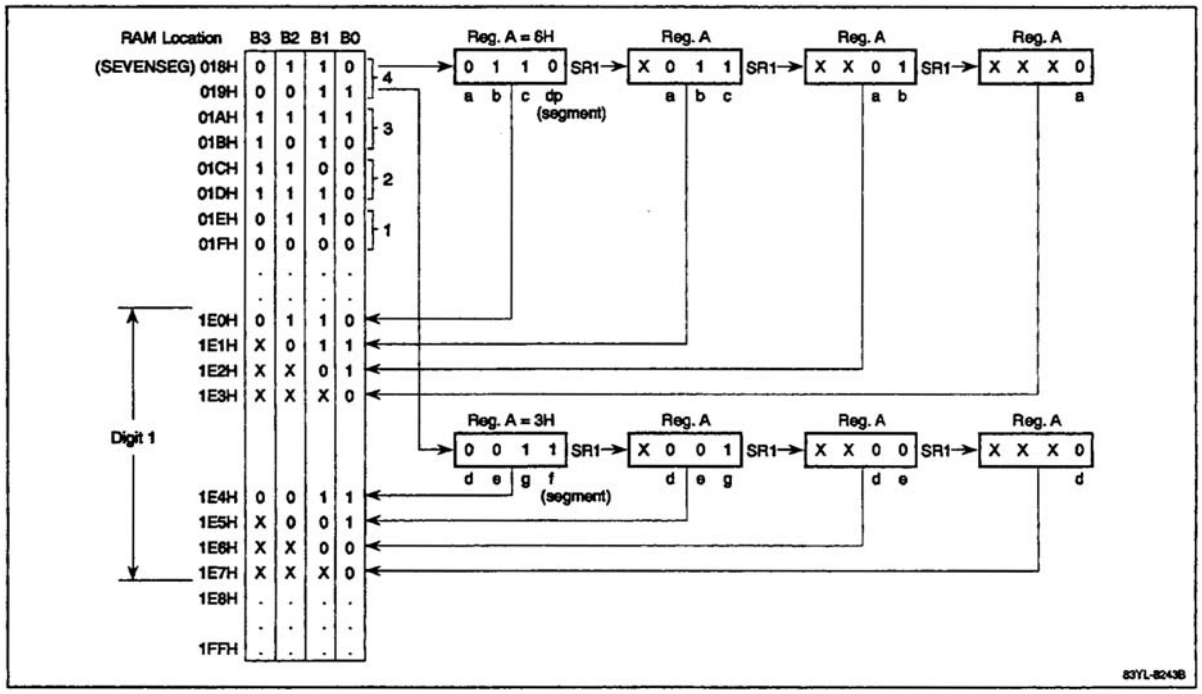


Figure 4-18. **Transfer of Seven-Segment Data to LCD Display Area in Static Mode**

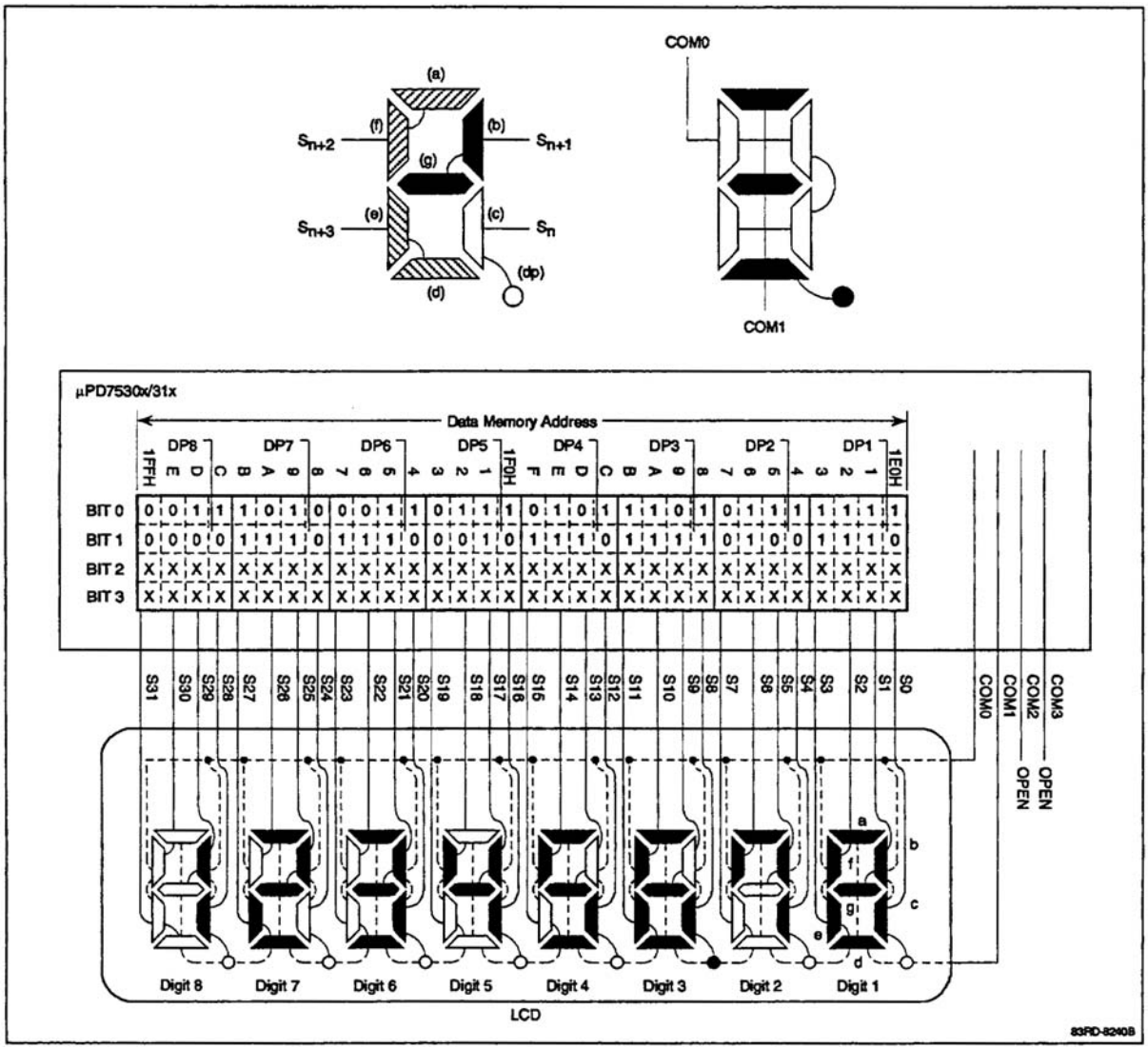


4.1.5 Programming Examples for 75X Microcontrollers in Duplex Mode

The LCD used in these examples has eight digits as shown in Figure 4-19. The program causes the LCD to display 123456.78.

Figure 4-19. LCD for Programming Examples in Duplex Mode

Note: In the table representing the RAM data display area, the least significant bit (bit 0) is on the left and the most significant bit (bit 3) is on the right. This is the opposite of the normal convention.



This programming example consists of a MAIN routine and three subroutines as shown in Figures 4-20 to 4-22.

- DUPLMAIN Main routine
- CLRLCD CLRLCDDA subroutine (shown earlier in Figure 4-2)
- DUPLCONV DCONVERT subroutine
- DICONADJ DUPLICON subroutine

The DUPLMAIN routine calls CLRLCDDA, which clears the LCD data area to all zeros, programs the LCD controller to operate in duplex mode at 64 Hz by setting the LCDM register to 01AH, and then stores the numbers 12345678 in ASCII format in ASCIIDAT.

DUPLMAIN then calls the DCONVERT subroutine. This is a code segment (CSEG) that starts on a PAGE boundary (Figure 4-21). A flowchart for this subroutine is shown in Figure 4-23 and a diagram showing the algorithm to store one digit is shown in Figure 4-24. The first ten locations are a ROM table that contains the numbers 0–9 in seven-segment, duplex-mode format. DCONVERT gets a digit from ASCIIDAT, converts it to seven-segment, duplex-mode format using a ROM table and the MOVT instruction, and then stores the seven-segment data in the LCD data display area 1E0H–1FFH. It uses the DE register as a pointer to ASCIIDAT and register HL as a pointer to the LCD data display area. Register C keeps track of the number of digits converted and stored and register B keeps track of the number of nibbles/digits stored. After DCONVERT converts and stores the eight ASCII digits, the subroutine returns to DUPLMAIN, which calls the DUPLICON subroutine to turn ON decimal point DP3 and then return to DUPLMAIN, which goes into an infinite loop.

Figure 4-20. Main Routine for 75X Microcontrollers in Duplex Mode

```
75X SERIES ASSEMBLER V4.00

   ** DUPLEX MODE V1.0                                          **

 COMMAND      : -C308 DUPLMAIN.ASM


  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                           $               TITLE = 'DUPLEX MODE V1.0'
    2                           $               PL = 66
    3                           $               PW = 120
    4
    5                                           NAME    DUPLMAIN
    6                                           EXTRN   DCONVERT, CLRLCDDA, DUPLICON
    7                                           PUBLIC  MAIN, ASCIIDAT
    8
    9 0000 R 0000                               VENT0   MBE = 0, RBE = 0, MAIN
   10
   11 ----                      SEG0            DSEG    AT   060H
   12 0060                      ASCIIDAT:       DS      10H             ;Reserves 16 nibbles for ASCII data.
   13
   14
   15                                  ;*************************************************************
   16                                  ;  This is the initialization and Main routine for the Duplex   *
   17                                  ;  Mode. It CALLS CLRLCDDA, which clears the LCD Data Display   *
   18                                  ;  area to 0. The main routine sets the LCD Controller to the   *
   19                                  ;  Duplex Mode @ 64 Hz. Main then loads ASCII Data into         *
   20                                  ;  ASCIIDAT. Main then CALLS the DCONVERT subroutine, which     *
   21                                  ;  converts the input ASCII data to seven-segment data and moves    *
   22                                  ;  the seven-segment data to the LCD Display Area. It then CALLS    *
   23                                  ;  DUPLICON which turns ON digit 3 decimal point.              *
   24                                  ;*************************************************************
   25
   26 ----                      MAINPROG        CSEG    INBLOCK
   27 0000   9CB2               MAIN:           DI                      ;Disable interrupts.
   28 0002 E AB4000                            CALL    !CLRLCDDA       ;Clear LCD Data Display area.
   29 0005   9C90                              CLR1    MBE             ;Select lower Bank 0 & upper Bank 15.
   30 0007   8905                              MOV     XA,#05H         ;Set Watch Mode (WM)
   31                                                                  ;       Register To Subsystem
   32 0009   9298                              MOV     WM,XA           ;           Clock & Enable Timer.
   33 000B   891A                              MOV     XA,#01AH        ;Select LCD segments S0-S31 and
   34 000D   928C                              MOV     LCDM,XA         ;       select Duplex Mode @ 64 Hz.
   35 000F   75                                MOV     A,#05H          ;Enable LCD commons
   36 0010   938E                              MOV     LCDC,A          ;               and segments.
   37 0012   8B60                              MOV     HL,#ASCIIDAT    ;Point to start of ASCII Data.
   38 0014   8938                              MOV     XA,#038H        ;Initialize XA W/an ASCII 8.
   39 0016   9A7F                              MOV     B,#07H          ;Set data counter.
   40 0018   AA10               STOREDAT:      MOV     @HL,XA          ;Save ASCII Data.
   41 001A   C8                                DECS    A               ;Decrement ASCII Number by -1.
   42 001B   60                                NOP                     ;Prevent skip.
   43 001C   C2                                INCS    L               ;Change pointer to
   44 001D   60                                NOP                     ;Eliminate skip.
   45 001E   C2                                INCS    L               ;       new ASCII location.
   46 001F   60                                NOP                     ;Don't allow a skip.
   47 0020   CF                                DECS    B               ;Change number of digits saved.
   48 0021   F6                                BR      STOREDAT
   49 0022 E AB4000                            CALL    !DCONVERT       ;Convert data to seven-segment
   50                                                                  ;       format and store in LCD data.
   51 0025 E AB4000                            CALL    !DUPLICON       ;Manipulate ICONS (decimal points).
   52 0028   60                 LOOP:          NOP                     ;Done, loop.
   53 0029   FE                                BR      LOOP
   54                                          END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-21. DUPLCONV Subroutine for 75X Microcontrollers in Duplex Mode

```
75X SERIES ASSEMBLER V4.00
    ** CONVERT DUPLEX & STORE V1.0                                    **
 COMMAND      : -C308 DUPLCONV.ASM
  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT
     1                          $               TITLE = 'CONVERT DUPLEX & STORE V1.0'
     2                          $               PL = 66
     3                          $               PW = 120
     4
     5
     6                                          NAME    DUPLCONV
     7                                          EXTRN   ASCIIDAT
     8                                          PUBLIC  DCONVERT
     9
    10                                  ;*****************************************************
    11                                  ; This subroutine gets the ASCII data, converts it to   *
    12                                  ; seven-segment data and stores the result in the LCD Data  *
    13                                  ; Display area 1E0H-1EBH.                                *
    14                                  ;*****************************************************
    15
    16                          ; The following are seven-segment display codes for a Quadruplexed LCD
    17
    18 ----                     DISPSEG         CSEG    PAGE
    19 0000   F5                                DB      0F5H            ;Display a 0.
    20 0001   05                                DB      005H            ;Display a 1.
    21 0002   EC                                DB      0ECH            ;Display a 2.
    22 0003   AD                                DB      0ADH            ;Display a 3.
    23 0004   1D                                DB      01DH            ;Display a 4.
    24 0005   B9                                DB      0B9H            ;Display a 5.
    25 0006   F9                                DB      0F9H            ;Display a 6.
    26 0007   25                                DB      025H            ;Display a 7.
    27 0008   FD                                DB      0FDH            ;Display a 8.
    28 0009   BD                                DB      0BDH            ;Display a 9.
    29
    30                          $EJECT
    31
    32 000A   9907              DCONVERT:       PUSH    BS              ;Save BS.
    33 000C E 8D00                              MOV     DE,#ASCIIDAT    ;Point DE to address of ASCII data.
    34 000E   9D90                              SET1    MBE             ;Select memory
    35 0010   9911                              SEL     MB1             ;         Bank 1.
    36 0012   8BE0                              MOV     HL,#0E0H        ;Point HL to LCD Display Data.
    37 0014   9A7E                              MOV     C,#07H          ;C keeps track of digits done.
    38 0016   9A3F              CONVSTOR:       MOV     B,#03H          ;B = nibbles stored/digit.
    39 0018   E4                                MOV     A,@DE           ;Get ASCII data.
    40 0019   9A09                              MOV     X,#0            ;Convert upper nibble = 0 BCD.
    41 001B   D0                                MOVT    XA,@PCXA        ;Convert BCD to seven-segment data.
    42 001C   E8                STNIBBLE:       MOV     @HL,A           ;Store first 1/2 nibble of 7-Seg. data.
    43 001D   CF                                DECS    B               ;Update nibbles done.
    44 001E   C2                                INCS    L               ;Update L pointer.
    45 001F   98                                RORC    A               ;Get second 1/2 of
    46 0020   98                                RORC    A               ;       nibble into position.
    47 0021   E8                                MOV     @HL,A           ;Store second 1/2 nibble.
    48 0022   C2                                INCS    L               ;Update L pointer.
    49 0023   02                                BR      PTROK           ;       No L overflow, continue.
    50 0024   C3                                INCS    H               ;       Reg L overflow, update Reg H.
    51 0025   60                                NOP                     ;       Stops skip.
    52 0026   CF                PTROK:          DECS    B               ;Update nibbles done.
    53 0027   09                                BR      LASTHALF        ;Digit not done, get last 1/2 of digit.
    54 0028   C4                                INCS    E               ;Digit done. Point to next
    55 0029   C4                                INCS    E               ;               ASCII data byte.
    56 002A   60                                NOP                     ;Prevent skip.
    57 002B   CE                                DECS    C               ;Change digit count.
    58 002C R 5016                              BR      CONVSTOR        ;8 digits not done, convert more data.
    59 002E   9906                              POP     BS              ;8 digits done, restore
    60 0030   EE                                RET                     ;               BS and return.
    61 0031   9979              LASTHALF:       MOV     A,X             ;Get second half of digit.
    62 0033 R 501C                              BR      STNIBBLE        ;Finish second 1/2 of digit.
    63                                          END
TARGET CHIP : UPD75308
 STACK SIZE = 0000H
 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-22. DICONADJ Subroutine for 75X Microcontrollers in Duplex Mode

```
75X SERIES ASSEMBLER V4.00

   ** ADJUST DUPLEX ICONS                                        **

 COMMAND      : -C308 DICONADJ.ASM


  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                          $              TITLE = 'ADJUST DUPLEX ICONS'
    2                          $              PL = 66
    3                          $              PW = 120
    4
    5                                         NAME    DICONADJ
    6                                         PUBLIC  DUPLICON
    7
    8
    9                           ;The following is a list of locations for the ICONs.
   10
   11        (0381)             DP1            EQU     0E0H.1          ;Digit 1 DP at location 1E0 bit 1.
   12        (0391)             DP2            EQU     0E4H.1          ;Digit 2 DP at location 1E4 bit 1.
   13        (03A1)             DP3            EQU     0E8H.1          ;Digit 3 DP at location 1E8 bit 1.
   14        (03B1)             DP4            EQU     0ECH.1          ;Digit 4 DP at location 1EC bit 1.
   15        (03C1)             DP5            EQU     0F0H.1          ;Digit 5 DP at location 1F0 bit 1.
   16        (03D1)             DP6            EQU     0F4H.1          ;Digit 6 DP at location 1F4 bit 1.
   17        (03E1)             DP7            EQU     0F8H.1          ;Digit 7 DP at location 1F8 bit 1.
   18        (03F1)             DP8            EQU     0FCH.1          ;Digit 8 DP at location 1FC bit 1.
   19
   20
   21                                        ;*********************************************
   22                                        ;  Decimal points are handled as ICONS and are  *
   23                                        ;  turned ON or OFF using bit manipulation      *
   24                                        ;  instructions. This subroutine will turn ON   *
   25                                        ;  the DP3 ICON.                                *
   26                                        ;*********************************************
   27
   28
   29 ----                    ICON           CSEG    INBLOCK
   30 0000   9907             DUPLICON:      PUSH    BS                      ;Save BS register.
   31 0002   9D90                            SET1    MBE                     ;Select memory
   32 0004   9911                            SEL     MB1                     ;          Bank 1.
   33 0006   9AEB                            MOV     H,#DP3  SHR 6           ;Loads 0EH into Reg H.
   34 0008   9D18                            SET1    @H+DP3                  ;Turn ON the DP3 ICON.
   35 000A   9906                            POP     BS
   36 000C   EE                              RET
   37                                        END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-23. **Convert and Store Operations in Duplex Mode**

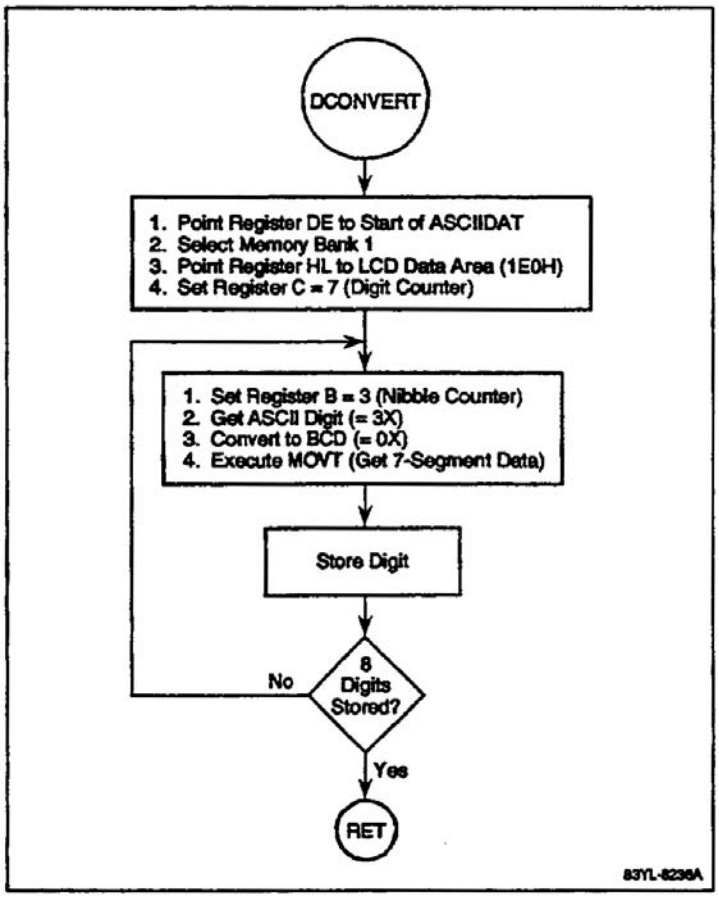
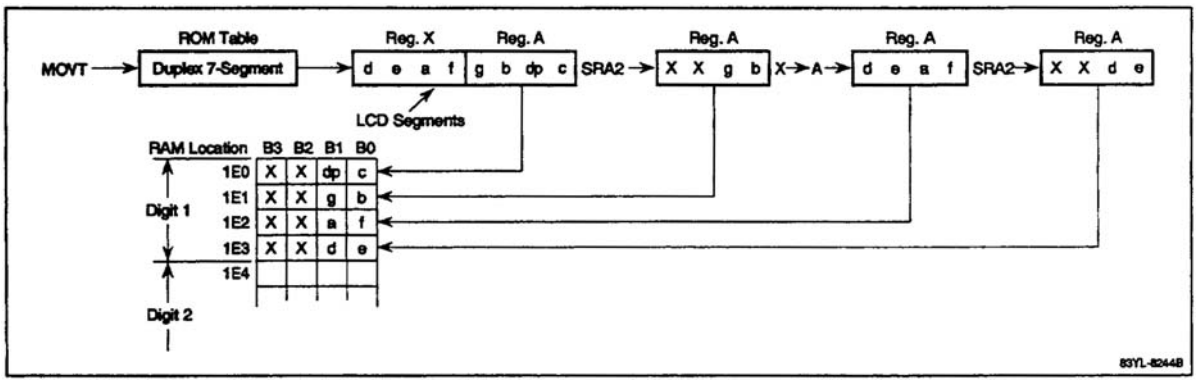


Figure 4-24. **Storing of One-Digit, Seven-Segment Data in LCD Display Area in Duplex Mode**



4.1.6 Programming Examples for 75X Microcontrollers In Triplex Mode

The LCD used in this example has four digits and three icons, as shown in Figure 4-3. The program causes the LCD to display:

B -1 78.90

This example consists of a main routine and three subroutines as shown in Figures 4-25 to 4-27.

- TRIMAIN Main routine
- CLRLCD CLRLCDDA subroutine (shown earlier in Figure 4-2)
- TRICONV TRICONV subroutine
- TICONADJ TICONADJ subroutine

The TRIMAIN routine calls CLRLCDDA subroutine, which clears the LCD data area to all zeros, programs the LCD controller to operate in triplex mode with 1/3 bias at 43 Hz by setting register LCDM to 019H, and then stores numbers 7890 in ASCII format in ASCIIDAT. Decrementing register A from 0H causes an underflow to 0FH, which is not the ASCII code for the number 9. TRIMAIN makes an adjustment by substituting 9H into register A. Now register XA contains 39H, which is the ASCII value for 9.

A flowchart for this subroutine is shown in Figure 4-28, and a diagram showing the algorithm to store one digit is shown in Figure 4-29. TRIMAIN then calls the TRICONV subroutine. This is a code segment (CSEG) that starts on a PAGE boundary (Figure 4-26). The first ten locations are a ROM table that contains the numbers 0–9 in seven-segment, triplex-mode format. TRICONV gets a digit from ASCIIDAT, converts it to seven-segment, triplex-mode format using a ROM table and the MOVT instruction, and stores the seven-segment data in the LCD data display area 1E0H–1FFH. It uses the DE Register as a pointer to ASCIIDAT and the HL register as a pointer to the LCD data display area. Register B keeps track of the number of digits converted and stored. After TRICONV converts and stores the four ASCII digits, the subroutine returns the program to TRIMAIN, which calls the TRIICON subroutine to turn ON decimal point DP3, the B, -, and 1 icons and then returns to TRIMAIN, which goes into an infinite loop.

Figure 4-25. MAIN Routine for 75X Microcontrollers in Triplex Mode

```
75X SERIES ASSEMBLER V4.00

   ** TRIPLEX MAIN ROUTINE V1.0                                    **

 COMMAND      : -C308 TRIMAIN.ASM


  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                           $              TITLE = 'TRIPLEX MAIN ROUTINE V1.0'
    2                           $              PL = 66
    3                           $              PW = 120
    4
    5
    6                                          NAME    TRIMAIN
    7                                          EXTRN   CLRLCDDA, TRICONV, TRIICON
    8                                          PUBLIC  ASCIIDAT
    9
   10 0000 R 0000                              VENT0   MBE = 0, RBE = 0, MAIN
   11
   12 ----                     ASCII           DSEG    AT      010H
   13 0010                     ASCIIDAT:       DS      8               ;Reserves 4 bytes for ASCII data.
   14
   15
   16                                 ;*********************************************
   17                                 ;  This is the initialization and Main routine. *
   18                                 ;  It sets the LCD Controller to the Triplex    *
   19                                 ;  Mode @ 64 Hz. It then calls the routine      *
   20                                 ;  TRICONST subroutine, which converts the      *
   21                                 ;  input ASCII data to seven-segment data and moves *
   22                                 ;  the seven-segment data to the LCD Display Area.  *
   23                                 ;  The TRIICON routine is called; it updates    *
   24                                 ;  the ICONS. When done, the LCD will display   *
   25                                 ;  Icons `B', `-', `1', and `7890'              *
   26                                 ;*********************************************
   27
   28 ----                     MAINROUT        CSEG    INBLOCK
   29 0000   9CB2              MAIN:           DI                      ;Disable interrupts.
   30 0002   9C90                              CLR1    MBE             ;Select lower 0 & upper 15.
   31 0004 E AB4000                            CALL    !CLRLCDDA       ;Clear LCD Data Display area to 0.
   32 0007   8905                              MOV     XA,#05H         ;Set Watch Mode (WM)
   33                                                                  ;       Register To Subsystem
   34 0009   9298                              MOV     WM,XA           ;           Clock & Enable Timer.
   35 000B   8919                              MOV     XA,#019H        ;Enable Commons, Segments and set
   36 000D   928C                              MOV     LCDM,XA         ;   to Triplex 1/3 Bias Mode @ 43 Hz.
   37 000F   75                                MOV     A,#05H          ;Enable LCD
   38 0010   938E                              MOV     LCDC,A          ;       commons and segments.
   39 0012   8B10                              MOV     HL,#ASCIIDAT    ;Point to start of ASCII Data.
   40 0014   8930                              MOV     XA,#030H        ;Initialize XA W/an ASCII 30H.
   41 0016   9A3F                              MOV     B,#03H          ;Set data counter.
   42 0018   AA10              STOREDAT:       MOV     @HL,XA          ;Save ASCII Data.
   43 001A   C8                                DECS    A               ;Decrement ASCII Number by -1.
   44 001B   02                                BR      ASCIIOK         ;ASCII OK, BRANCH and continue.
   45 001C   8927                              MOV     XA,#39          ;Underflow correction.
   46 001E   C2                ASCIIOK:        INCS    L               ;Change pointer to
   47 001F   C2                                INCS    L               ;       new ASCII location.
   48 0020   CF                                DECS    B               ;Change number of digits saved.
   49 0021   F6                                BR      STOREDAT
   50 0022 E AB4000                            CALL    !TRICONV        ;Convert data to seven-segment
   51                                                                  ;       format and store in LCD area.
   52 0025 E AB4000                            CALL    !TRIICON        ;Update Icon into seven-segment data.
   53 0028   60                DONE:           NOP                     ;Done, loop.
   54 0029   FE                                BR      DONE
   55                                          END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-26. TRICONV Subroutine for 75X Microcontrollers in Triplex Mode

```
75X SERIES ASSEMBLER V4.00

   ** CONVERT TRIPLEX & STORE V1.0                                  **

 COMMAND      : -C308 TRICONV.ASM

  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                           $               TITLE = 'CONVERT TRIPLEX & STORE V1.0'
    2                           $               PL = 66
    3                           $               PW = 120
    4
    5                                           NAME    TRICONV
    6                                           EXTRN   ASCIIDAT
    7                                           PUBLIC  TRICONV, DIGIT1, DIGIT2, DIGIT3
    8
    9                                   ;*****************************************************
   10                                   ; This subroutine gets the ASCII data, converts         *
   11                                   ; it to seven-segment data and stores the result           *
   12                                   ; in the LCD Data Display area 1E0H-1EBH.               *
   13                                   ;*****************************************************
   14
   15 ----                      BANK1DAT        DSEG    1       AT 0
   16 0100                      TEMP:           DS      1               ;Current seven-segment data.
   17
   18 ----                      DIGITDAT        DSEG    1       AT 0E0H ;LCD Data area.
   19 01E0                      DIGIT1:         DS      3               ;Digit 1 data.
   20 01E3                      DIGIT2:         DS      3               ;Digit 2 data.
   21 01E6                      DIGIT3:         DS      3               ;Digit 3 data.
   22 01E9                      DIGIT4:         DS      3               ;Digit 4 data.
   23
   24
   25                           ; The layout for the Triplex seven-segment LCD is shown below.
   26                           ; The layout of the segment bits a-g and their data bits (Dn) is also shown.
   27                           ;
   28                           ;
   29                           ;                 a
   30                           ;                ---
   31                           ;             f | g | b                 |---|---|---|---|---|---|---|---|
   32                           ;                ---                  | 0 | d | g | a | e | f | c | b |
   33                           ;             e |   | c                 |---|---|---|---|---|---|---|---|
   34                           ;                ---
   35                           ;                 d
   36
   37                           ; The following are seven-segment display codes for a Triplexed LCD.
   38
   39 ----                      DISPSEG         CSEG    PAGE
   40
   41 0000   5F                                DB      05FH            ;Display a 0.
   42 0001   03                                DB      003H            ;Display a 1.
   43 0002   79                                DB      079H            ;Display a 2.
   44 0003   73                                DB      073H            ;Display a 3.
   45 0004   27                                DB      027H            ;Display a 4.
   46 0005   76                                DB      076H            ;Display a 5.
   47 0006   7E                                DB      07EH            ;Display a 6.
   48 0007   17                                DB      017H            ;Display a 7.
   49 0008   7F                                DB      07FH            ;Display a 8.
   50 0009   37                                DB      037H            ;Display a 9.
   51
   52                           $EJECT
   53
   54 000A   9907               TRICONV:       PUSH    BS              ;Save Bank Select Reg.
   55 000C   9D90                              SET1    MBE             ;Select memory
   56 000E   9911                              SEL     MB1             ;          Bank 1.
   57 0010 E 8D00                              MOV     DE,#ASCIIDAT    ;Point DE to ASCII data in Bank 0.
   58 0012   9A3F                              MOV     B,#03H          ;B keeps track of digits done.
   59 0014   8BE0                              MOV     HL,#DIGIT1      ;Point to first LCD digit.
   60 0016   E4                 GETASCII:      MOV     A,@DE           ;Get ASCII data.
   61 0017   9A09                              MOV     X,#00H          ;Make X = 0 BCD.
   62 0019   D0                                MOVT    XA,@PCXA        ;Get seven-segment data.
```

Figure 4-26 Continued

```
   63 001A   9300                          MOV     TEMP,A          ;Save 7-seg data.
   64 001C   9933                          AND     A,#03H          ;Set upper 2 bits of
   65                                                              ;       first LCD nibbleto 0.
   66 001E   E8                            MOV     @HL,A           ;Save first LCD digit nibble.
   67 001F   9979                          MOV     A,X             ;Get second
   68 0021   C2                            INCS    L               ;       LCD nibble
   69 0022   E8                            MOV     @HL,A           ;               and save it.
   70 0023   A300                          MOV     A,TEMP          ;Get first seven-segment nibble back.
   71 0025   98                            RORC    A               ;Get segments f and e into bit
   72 0026   98                            RORC    A               ;  positions 0 & 1 and set upper
   73 0027   9933                          AND     A,#03H          ;    2 bits of third LCD nibble to 0.
   74 0029   C2                            INCS    L               ;Point to third LCD
   75 002A   E8                            MOV     @HL,A           ;       nibble and save it.
   76 002B   C2                            INCS    L               ;Point to first nibble of next digit.
   77 002C   C4                            INCS    E               ;Point to next
   78 002D   C4                            INCS    E               ;       ASCII char.
   79 002E   CF                            DECS    B               ;Decrement number of digits done.
   80 002F R 5016                          BR      GETASCII        ;Do next digit.
   81 0031   9906                          POP     BS              ;Restore BS register.
   82 0033   EE                            RET                     ;Conversion done, return.
   83                                      END

TARGET CHIP : UPD75308
STACK SIZE = 0000H

ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-27. TICONADJ Subroutine for 75X Microcontrollers in Triplex Mode

```
75X SERIES ASSEMBLER V4.00

   **  ADJUST ICONS TRIPLEX                                    **

 COMMAND      : -C308 TICONADJ.ASM


  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                          $              TITLE = ' ADJUST ICONS TRIPLEX'
    2                          $              PL = 66
    3                          $              PW = 120
    4
    5                                         NAME    TICONADJ
    6                                         PUBLIC  TRIICON
    7
    8
    9                          ;The following is a list of locations for the ICONs.
   10
   11        (038A)            DP2            EQU     0E2H.2          ;Digit 2 DP at location 1E2 bit 2.
   12        (0396)            DP3            EQU     0E5H.2          ;Digit 2 DP at location 1E5 bit 2.
   13        (03A2)            DP4            EQU     0E8H.2          ;Digit 2 DP at location 1E8 bit 2.
   14        (03AE)            DP5            EQU     0EBH.2          ;Digit 2 DP at location 1EB bit 2.
   15        (038E)            BICON          EQU     0E3H.2          ;B ICON at location 1E3 bit 2.
   16        (039A)            DASH           EQU     0E6H.2          ;DASH ICON at location 1E6 bit 2.
   17        (03A6)            ONE            EQU     0E9H.2          ;ONE ICON at location 1E9 bit 2.
   18
   19                                        ;*********************************************
   20                                        ;  This segment will turn ON the DP3, the BICON,*
   21                                        ;  DASH and the ONE ICONs using the bit         *
   22                                        ;  manipulation instructions.                  *
   23                                        ;*********************************************
   24
   25
   26 ----                    ICON           CSEG    INBLOCK
   27 0000   9907             TRIICON:       PUSH    BS                      ;Save BS register.
   28 0002   9D90                            SET1    MBE                     ;Select memory
   29 0004   9911                            SEL     MB1                     ;          Bank 1.
   30 0006   9AEB                            MOV     H,#BICON  SHR 6         ;Loads E into Reg H.
   31 0008   9D25                            SET1    @H+DP3                  ;Turn ON the DP3 ICON.
   32 000A   9D23                            SET1    @H+BICON                ;Turn ON the BICON.
   33 000C   9D26                            SET1    @H+DASH                 ;Turn ON the DASH ICON.
   34 000E   9D29                            SET1    @H+ONE                  ;Turn ON the ONE ICON.
   35 0010   9906                            POP     BS
   36 0012   EE                              RET
   37                                        END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-28. **Convert and Store Operations in Triplex Mode**

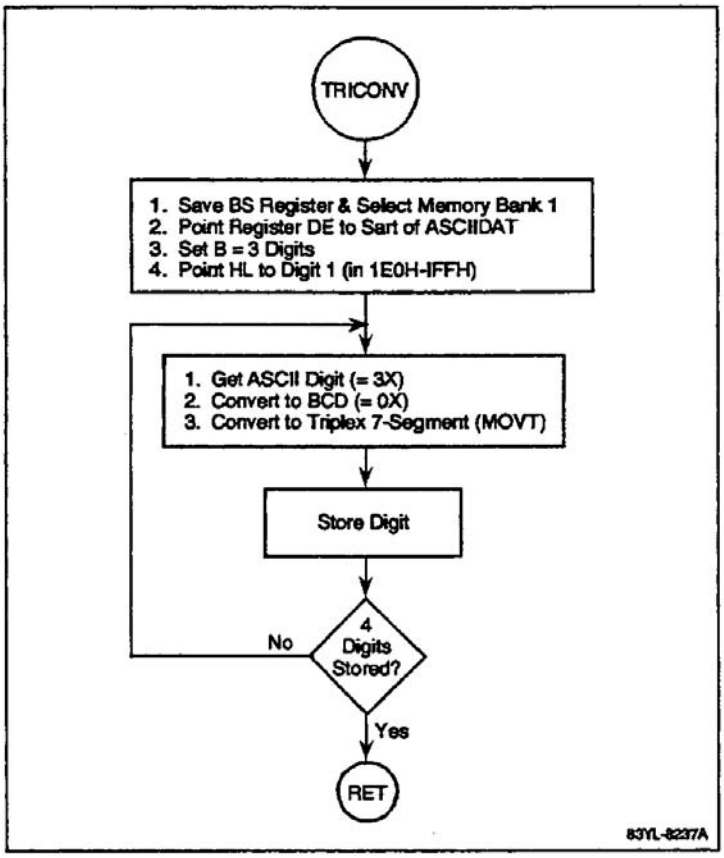
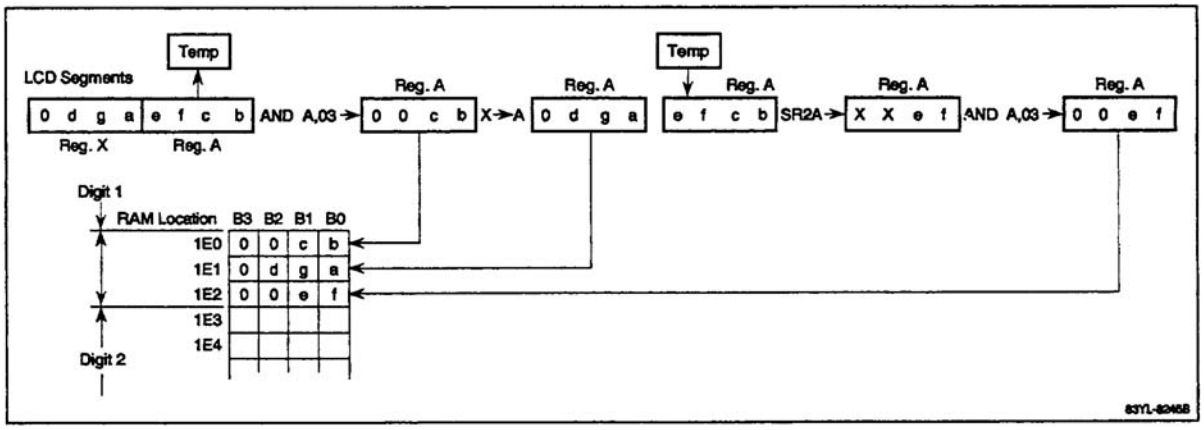


Figure 4-29. **Storing of One-Digit of Seven-Segment Data in LCD Display Area in Triplex Mode**

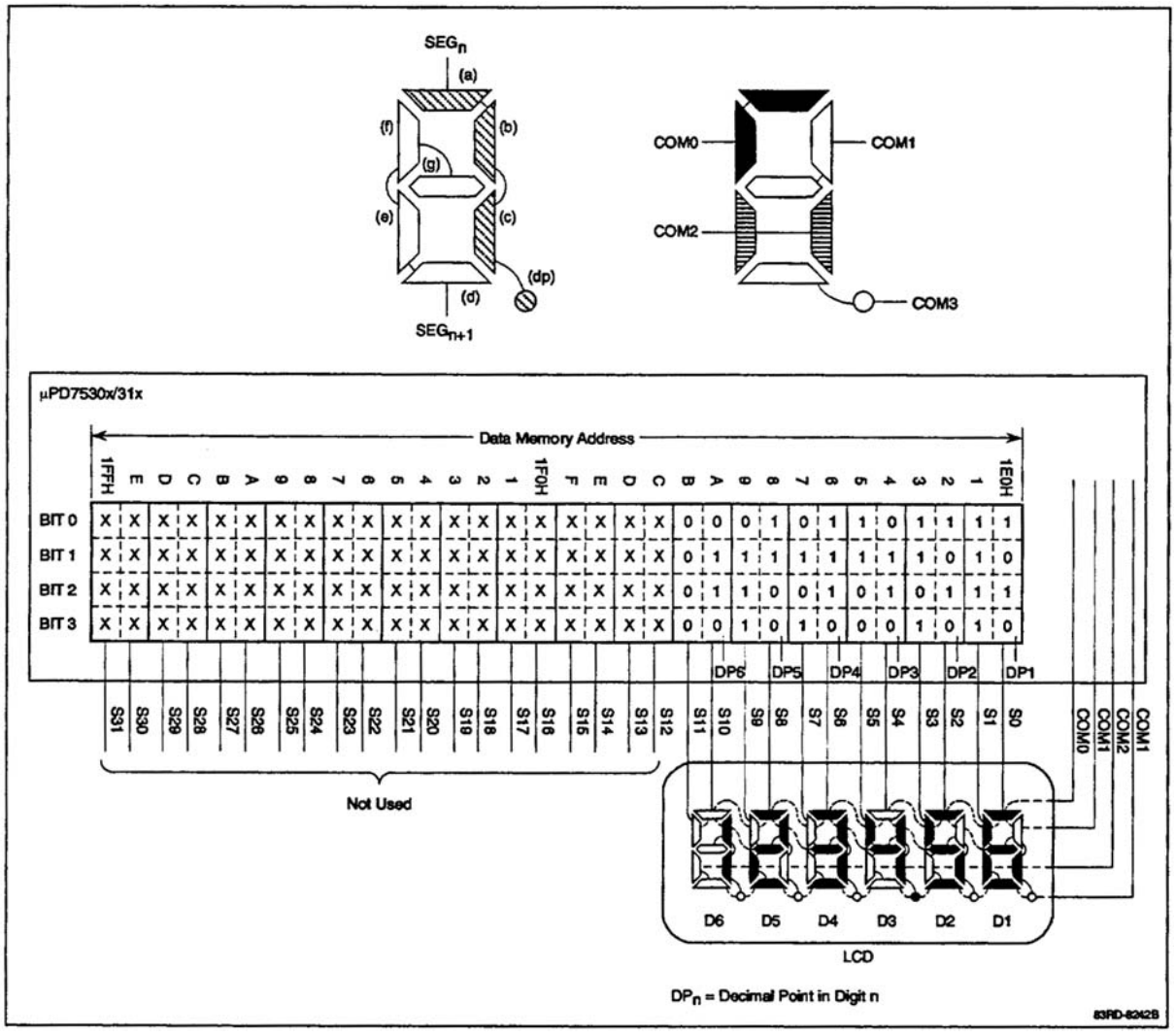


4.1.7 Programming Examples for 75X Microcontrollers in Quadruplex Mode

The LCD used in this example has six digits, as shown in Figure 4-30. The program causes the LCD to display 1234.56.

Figure 4-30. 6-Digit LCD in Quadruplex Mode

NOTE: Please note in the table representation of the RAM data display area, the least significant bit (bit 0) is on the left and the most significant bit (bit 3) is on the right. This is opposite the normal convention.



This programming example consists of a MAIN routine and three subroutines as shown in Figures 4-31 to 4-33.

- QUADMAIN Main routine
- CLRLCD CLRLCDDA subroutine (shown earlier in Figure 4-2)
- QUADCONV CONVERT subroutine

- QICONADJ QUADICON subroutine

Figure 4-31. MAIN Routine for 75X Microcontrollers in Quadruplex Mode

```
75X SERIES ASSEMBLER V4.00

   ** QUADRUPLEX MODE V1.0                                    **

 COMMAND      : -C308 QUADMAIN.ASM


  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1
    2                           $              TITLE = 'QUADRUPLEX MODE V1.0'
    3                           $              PL = 66
    4                           $              PW = 120
    5
    6                                          NAME    QUADMAIN
    7                                          EXTRN   CONVERT, CLRLCDDA, QUADICON
    8                                          PUBLIC  MAIN, ASCIIDAT
    9
   10 0000 R 0000                              VENT0   MBE = 0, RBE = 0, MAIN
   11
   12 ----                      SEG0           DSEG    AT   060H
   13 0060                      ASCIIDAT:      DS      0AH             ;Reserves 6 bytes for ASCII data.
   14
   15
   16                                 ;*************************************************************
   17                                 ;  This is the initialization and Main routine for the          *
   18                                 ;  Quadruplex Mode. It CALLS CLRLCDDA, which clears the LCD     *
   19                                 ;  Data Display area to 0. The main routine sets the LCD        *
   20                                 ;  Controller to the Quadruplex Mode @ 64 Hz. Main then loads   *
   21                                 ;  ASCII Data into ASCIIDAT. Main then CALLS the CONVERT        *
   22                                 ;  subroutine, which converts the input ASCII data to seven-segment *
   23                                 ;  data and moves the seven-segment data to the LCD Display Area.   *
   24                                 ;  It then CALLS QUADICON which turns ON digit 3 decimal point.*
   25                                 ;*************************************************************
   26
   27 ----                      MAINPROG       CSEG    INBLOCK
   28 0000   9CB2               MAIN:          DI                      ;Disable interrupts.
   29 0002 E AB4000                           CALL    !CLRLCDDA       ;Clear LCD Data Display area.
   30 0005   9C90                             CLR1    MBE             ;Select lower Bank 0 & upper Bank 15.
   31 0007   8905                             MOV     XA,#05H         ;Set Watch Mode (WM)
   32                                                                 ;       Register To Subsystem
   33 0009   9298                             MOV     WM,XA           ;           Clock & Enable Timer.
   34 000B   8928                             MOV     XA,#028H        ;Select LCD segments S0-S31 and
   35 000D   928C                             MOV     LCDM,XA         ;       select Quadruplex Mode @ 64 Hz.
   36 000F   75                               MOV     A,#05H          ;Enable LCD commons
   37 0010   938E                             MOV     LCDC,A          ;               and segments.
   38 0012   8B60                             MOV     HL,#ASCIIDAT    ;Point to start of ASCII Data.
   39 0014   8936                             MOV     XA,#036H        ;Initialize XA W/an ASCII 6.
   40 0016   9A5F                             MOV     B,#05H          ;Set data counter.
   41 0018   AA10               STOREDAT:      MOV     @HL,XA          ;Save ASCII Data.
   42 001A   C8                               DECS    A               ;Decrement ASCII Number by -1.
   43 001B   60                               NOP                     ;Prevent skip.
   44 001C   C2                               INCS    L               ;Change pointer to
   45 001D   C2                               INCS    L               ;       new ASCII location.
   46 001E   CF                               DECS    B               ;Change number of digits saved.
   47 001F   F8                               BR      STOREDAT
   48 0020 E AB4000                           CALL    !CONVERT        ;Convert data to seven-segment
   49                                                                 ;       format and store in LCD data.
   50 0023 E AB4000                           CALL    !QUADICON       ;Manipulate ICONS (decimal points).
   51 0026   60                 LOOP:         NOP                     ;Done, loop.
   52 0027   FE                               BR      LOOP
   53                                         END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H
 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-32. QUADCONV Subroutine for 75X Microcontrollers in Quadruplex Mode

```
75X SERIES ASSEMBLER V4.00

   ** CONVERT QUAD & STORE V1.0                                    **

 COMMAND      : -C308 QUADCONV.ASM


  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                          $              TITLE = 'CONVERT QUAD & STORE V1.0'
    2                          $              PL = 66
    3                          $              PW = 120
    4
    5
    6                                         NAME    QUADCONV
    7                                         EXTRN   ASCIIDAT
    8                                         PUBLIC  CONVERT
    9
   10                                 ;*******************************************************
   11                                 ; This subroutine gets the ASCII data, converts it to   *
   12                                 ; seven-segment data and stores the result in the LCD Data  *
   13                                 ; Display area 1E0H-1EBH.                                *
   14                                 ;*******************************************************
   15
   16                          ; The following are seven-segment display codes for a Quadruplexed LCD
   17 ----                     DISPSEG        CSEG    PAGE
   18 0000   D7                              DB      0D7H            ;Display a 0.
   19 0001   06                              DB      006H            ;Display a 1.
   20 0002   E3                              DB      0E3H            ;Display a 2.
   21 0003   A7                              DB      0A7H            ;Display a 3.
   22 0004   36                              DB      036H            ;Display a 4.
   23 0005   B5                              DB      0B5H            ;Display a 5.
   24 0006   F5                              DB      0F5H            ;Display a 6.
   25 0007   17                              DB      017H            ;Display a 7.
   26 0008   F7                              DB      0F7H            ;Display a 8.
   27 0009   37                              DB      037H            ;Display a 9.
   28
   29
   30 000A   9907              CONVERT:      PUSH    BS              ;Save BS register.
   31 000C E 8D00                            MOV     DE,#ASCIIDAT    ;Point DE to address of ASCII data.
   32 000E   9D90                            SET1    MBE             ;Select memory
   33 0010   9911                            SEL     MB1             ;         Bank 1.
   34 0012   8BE0                            MOV     HL,#0E0H        ;Point HL to LCD Display Data.
   35 0014   9A5E                            MOV     C,#05H          ;C keeps track of bytes done.
   36 0016   9A09              CONVSTOR:     MOV     X,#0            ;Convert upper nibble = 0 BCD.
   37 0018   E4                              MOV     A,@DE           ;Get ASCII data.
   38 0019   D0                              MOVT    XA,@PCXA        ;Convert BCD to seven-segment data.
   39 001A   E8                              MOV     @HL,A           ;Store first nibble of converted data.
   40 001B   9979                            MOV     A,X             ;Get next
   41 001D   C2                              INCS    L               ;       nibble
   42 001E   E8                              MOV     @HL,A           ;          and store it.
   43 001F   C4                              INCS    E               ;Point to next
   44 0020   C4                              INCS    E               ;       ASCII data byte.
   45 0021   C2                              INCS    L               ;Point to next LCD data address.
   46 0022   CE                              DECS    C               ;Change byte count.
   47 0023   F2                              BR      CONVSTOR        ;Not done, convert more data.
   48 0024   9906                            POP     BS              ;Restore bank select register.
   49 0026   EE                              RET                     ;Conversion done, return.
   50                                        END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

Figure 4-33. QUICONADJ Subroutine for 75X Microcontrollers in Quadruplex Mode

```
75X SERIES ASSEMBLER V4.00

   ** ADJUST QUADRUPLEX ICONS                                    **

 COMMAND      : -C308 QICONADJ.ASM


  STNO ADRS R OBJECT   IC MAC   SOURCE STATEMENT

    1                          $              TITLE = 'ADJUST QUADRUPLEX ICONS'
    2                          $              PL = 66
    3                          $              PW = 120
    4
    5                                         NAME    QICONADJ
    6                                         PUBLIC  QUADICON
    7
    8
    9                           ;The following is a list of locations for the ICONs.
   10
   11        (0383)             DP1            EQU     0E0H.3          ;Digit 1 DP at location 1E0 bit 3.
   12        (038B)             DP2            EQU     0E2H.3          ;Digit 2 DP at location 1E2 bit 3.
   13        (0393)             DP3            EQU     0E4H.3          ;Digit 3 DP at location 1E4 bit 3.
   14        (039B)             DP4            EQU     0E6H.3          ;Digit 4 DP at location 1E6 bit 3.
   15        (03A3)             DP5            EQU     0E8H.3          ;Digit 5 DP at location 1E8 bit 3.
   16        (03AB)             DP6            EQU     0EAH.3          ;Digit 6 DP at location 1EA bit 3.
   17
   18                                        ;*********************************************
   19                                        ;  Decimal points are handled as ICONS and are  *
   20                                        ;  turned ON or OFF using bit manipulation      *
   21                                        ;  instructions. This subroutine will turn ON   *
   22                                        ;  the DP3 ICON.                                *
   23                                        ;*********************************************
   24
   25
   26 ----                     ICON           CSEG    INBLOCK
   27 0000   9907              QUADICON:      PUSH    BS                      ;Save BS register.
   28 0002   9D90                             SET1    MBE                     ;Select memory
   29 0004   9911                             SEL     MB1                     ;          Bank 1.
   30 0006   9AEB                             MOV     H,#DP3  SHR 6           ;Loads 0EH into Reg H.
   31 0008   9D34                             SET1    @H+DP3                  ;Turn ON the DP3 ICON.
   32 000A   9906                             POP     BS
   33 000C   EE                               RET
   34                                         END

 TARGET CHIP : UPD75308
 STACK SIZE = 0000H

 ASSEMBLY COMPLETE,  NO ERROR FOUND
```

The QUADMAIN routine calls the CLRLCDDA subroutine, which clears the LCD data area to all zeros, programs the LCD controller to operate in quadruplex mode with 1/3 bias at 64 Hz by setting register LCDM to 028H, and then stores the numbers 123456 in ASCII format in ASCIIDAT.

QUADMAIN then calls the QUADCONV subroutine. This is a code segment (CSEG) that starts on a PAGE boundary (Figure 4-32). A flowchart for this subroutine is shown in Figure 4-34 and a diagram showing the algorithm to store one digit is shown I Figure 4-35. The first ten locations are a ROM table that contains the numbers 0–9 in seven-segment, quadruplex-mode format. QUADCONV gets a digit from ASCIIDAT, converts it to seven-segment, quadruplex format using a ROM table and MOVT instruction, and stores the seven-segment data in the LCD data display area 1E0H–1FFH. It uses the DE register as a pointer to ASCIIDAT and the HL register as a pointer to the LCD data display area. Register C keeps track of the number of digits converted and stored. After QUADCONV converts and stores the six ASCII digits, the program returns to QUADMAIN, which calls the QUADICON subroutine to turn ON decimal point DP3 and then returns to QUADMAIN again, which goes into an infinite loop.

Figure 4-34. Convert and Store Operations in Quadruplex Mode

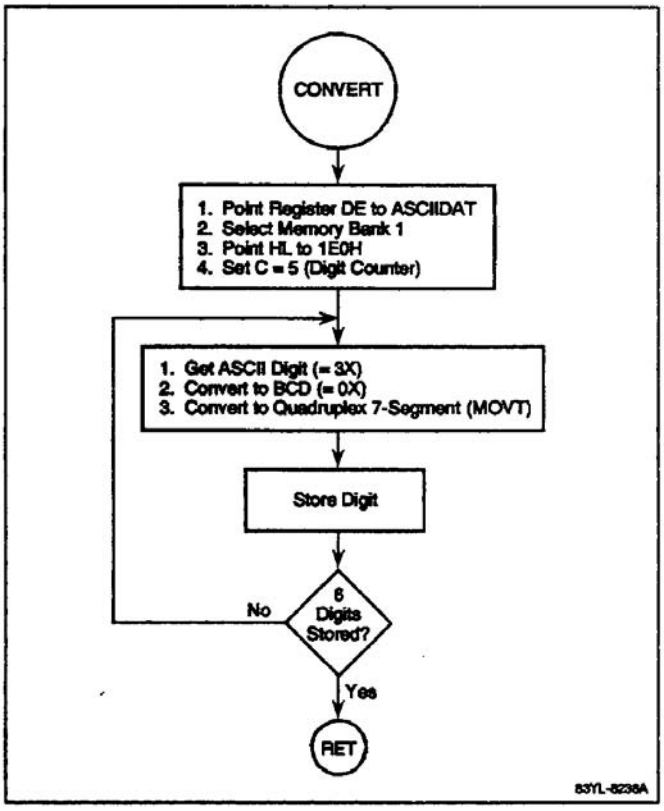
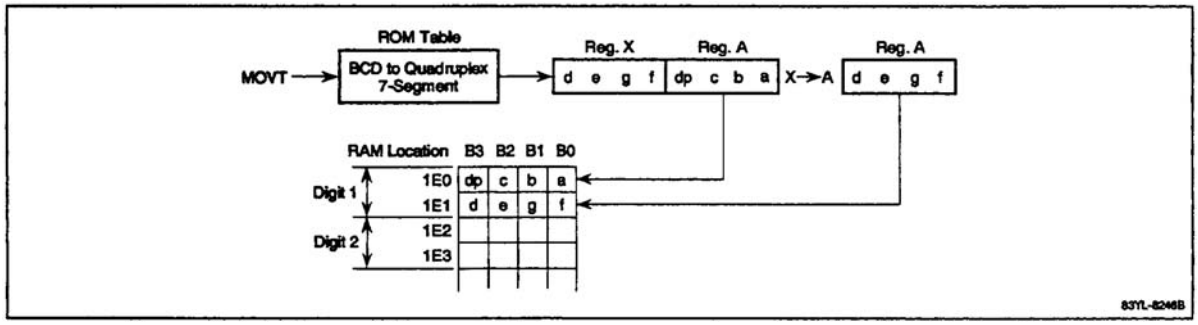


Figure 4-35. **Storing of One-Digit, Seven-Segment Data in Quadruplex Mode**



4.2 78K0 and 78K0S Microcontrollers

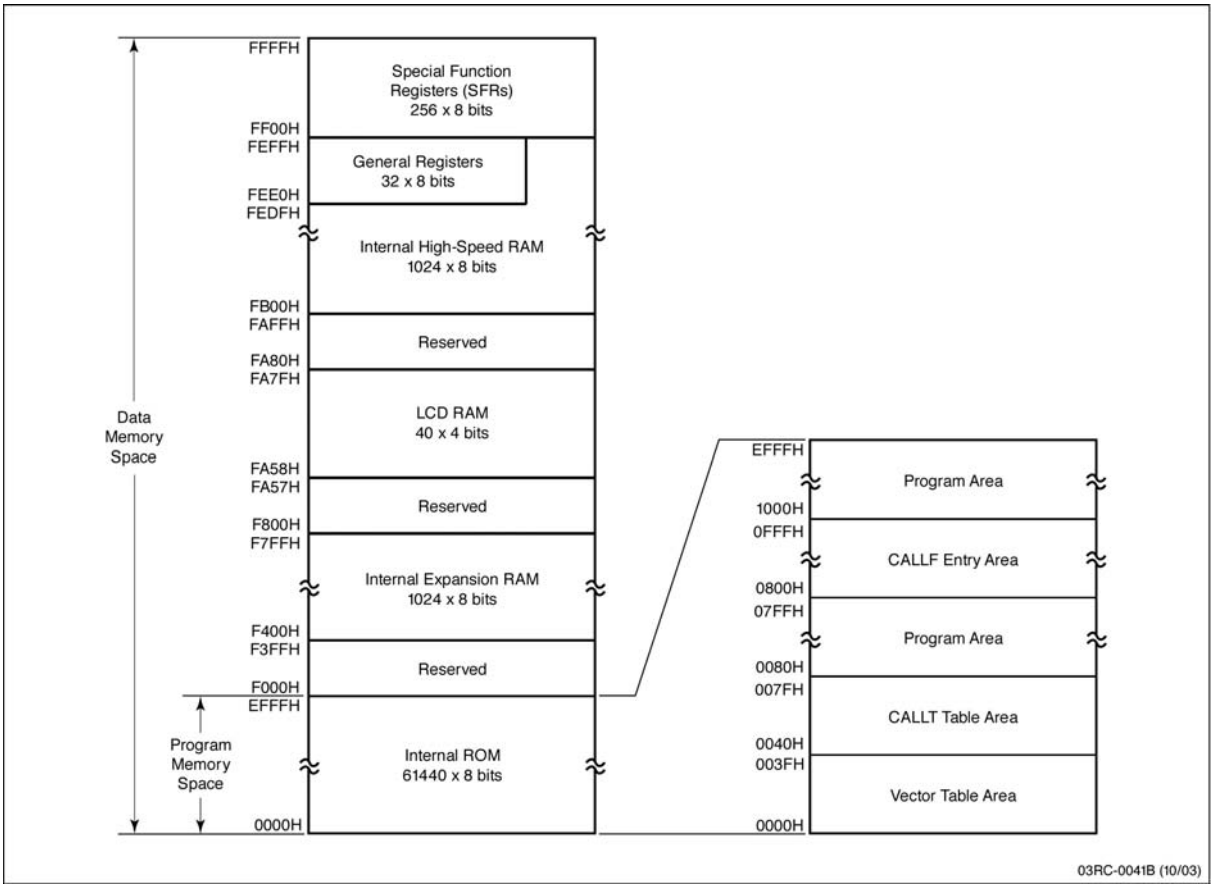
NEC Electronics has a broad line of 8-bit microcontrollers with LCD controllers in the configurations shown in Table 4-3.

Table 4-3. LCD Controller Configurations

Configuration	Segments
40 × 4	160
32 × 4	128
30 × 3	90
28 × 4	112
26 × 4	104
24 × 4	96
23 × 4	92
20 × 4	80
19 × 1	19
15 × 4	60
5 × 4	20

The 78K0S microcontrollers are low-end 8-bit devices, while the 78K0 microcontrollers are higher end 8-bit devices. Unlike the 4-bit microcontrollers, these have one 64 KB address space that contains the ROM, RAM, LCD display area, interrupts, and all special function registers (as shown in Figure 4-36).

Figure 4-36. Memory Address Space in μPD780308 78K0 Microcontroller



Programming examples for the μPD789407 78K0S microcontroller and the μPD780308 78K0 device are shown on the following pages. The format of these programs differs from that of the 75X product. These examples contain a program description, program specification, flowchart, and program examples written in C and assembly languages.

4.2.1 Programming Examples for 78K0S LCD Controllers

Program Description

The LCD Controller/Driver in the μPD78940X/78941X subseries has the following functions.

- Automatic output of segment and common signals with data from the LCD RAM display memory
- Five different display modes
  - » Static
  - » 1/2 duty (1/2 bias)
  - » 1/3 duty (1/2 bias)
  - » 1/3 duty (1/3 bias)
  - » 1/4 duty (1/3 bias)
- Four different frame frequencies (selectable in each display mode)

- 28 segment signal outputs (S0–S27); 12 of the segment outputs (P80/S27–P87/S20 and P90/S19–P93/S16) can be switched to I/O ports.
- Voltage divider resistors (for LCD drive voltage operation) can be specified with a mask option.
- Main clock or subsystem clock operation

This program demonstrates how to use the LCD controller/driver to write the values 0–7 to the eight digits of the 78K0S LCD on NEC Electronics' 78K0/78K0S LCD multi-use board (MUB).

The 78K0S LCD on the MUB is an eight-digit LCD configured for operation at 1/3 duty, 1/3 bias. Each seven-segment digit has an "up" icon on the top left and a decimal point icon on the bottom right, providing a total of 72 segments [(seven-segment digit + two icons) × eight digits]. With three common lines (COM0, COM1, and COM2), 72 ÷ 3 = 24 segment lines are required.

The μPD78940x/μPD78941x microcontrollers have 16 dedicated segment lines, and an additional eight are reassigned from port lines to segment lines by writing to the LCD Port Selector 0 register (LPS0). The LCD operating clock is programmed to be the 32.768 kHz subsystem clock. The LCD clock frequency is programmed to 256 Hz (32.768 kHz ÷ 2^7) by writing to the LCD Clock Control Register LCDC0. Writing to the LCD display mode register 0 (LCDM0) enables the LCD and specifies the power and display mode (1/3 duty and 1/3 bias).

The program initializes the display memory with all ones. The is a NOP inserted in the program to provide a convenient place for a programmer to break and check that all segments on the LCD are working. The program then writes the seven-segment values 0–7 to the LCD digits 0–7 before idling in an endless loop.

Table 4-4. Program Specifications

Feature	Specification
LCD Bias mode	1/3
LCD Duty	1/3
LCD Operating clock	32.768 kHz subsystem clock
LCD Clock frequency	256 Hz (32.768 kHz ÷ 2^7)
LCD Frame frequency	85 Hz (256 Hz ÷ 3)
LCD Common signals	COM0, COM1, COM2
Number of LCD segments	72 (24 segment lines × 3 common)
LCD Pins used in program	BIAS, VLC0, VLC1, VLC2, COM0, COM1, COM2, S0–S15, P93/S16
	P92/S17, P91/S18, P90/S19, P87/S20, P86/S21, P85/S22, P84/S23
Linker options for C program	Check boxes
	» Create Link Map File [-p]
	» Create Stack Symbol [-s]
	» Output Symbol Information [-g]
	Other options: c\nectools\lib78k05\lib\s0s –olcd -plcd

Figure 4-37. LCD Program Flowchart

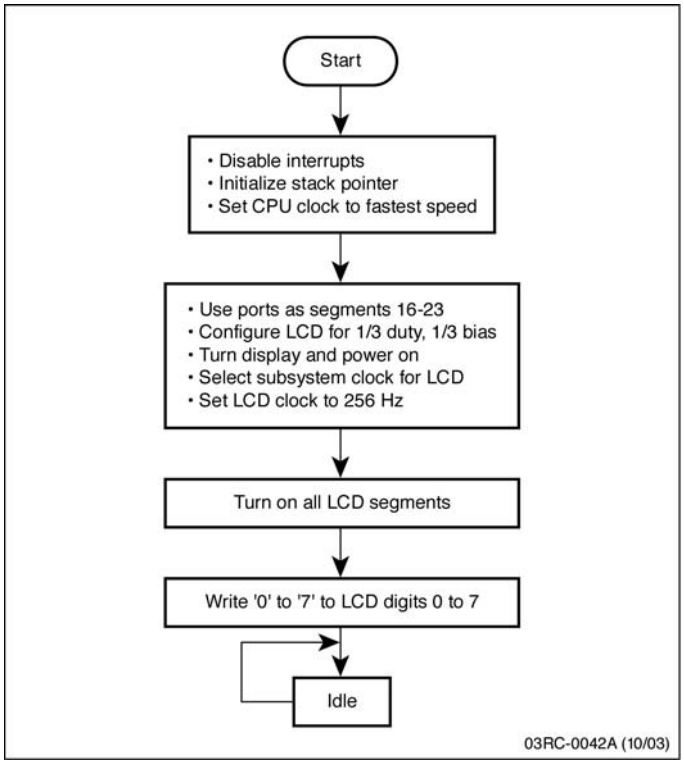


Figure 4-38. Assembly Language Program Example of LCD Controller Using a 78K0S Microcontroller

```
;*********************************************************************
; Date:          5/19/00
;
; Parameters:    - fastest CPU clock (fx=5.00MHz)
;                - Bias mode: 1/3
;                - Duty: 1/3
;                - Common signals used: COM0-COM2
;                - Number of LCD segments: 72 (24 segment lines X 3 common)
;                - LCD operation clock: subsystem clock (32.768kHz)
;                - LCD clock frequency: 256Hz (32.768kHz/2^7)
;                - Frame frequency: 85Hz (256Hz/3)
;                - LCD: K0S LCD on MUB-K0-K0S LCD Multi-Use Board
;
;*********************************************************************


;=======================================
;       Constants
;=======================================

LCDRam          EQU     0FA00h          ; Start of LCD display Ram area
Segment_lines   EQU     24              ; Number of LCD segment lines

Bit2            EQU 4                   ; bit 2
Bit1            EQU 2                   ; bit 1
Bit0            EQU 1                   ; bit 0


;=======================================
;       Specify Interrupt Vectors
;=======================================

RESET_VECTOR    CSEG    AT 0000h ; On reset, go to Start
                DW      Start


;=======================================
;       8-Digit K0S LCD on MUB Board
;=======================================

;_____________________________________________________________________________
;  ^  --     ^  --     ^  --     ^  --     ^  --     ^  --     ^  --     ^  --
;    |  |      |  |      |  |      |  |      |  |      |  |      |  |      |  |
;     --        --        --        --        --        --        --        --
;    |  |      |  |      |  |      |  |      |  |      |  |      |  |      |  |
;     -- o      -- o      -- o      -- o      -- o      -- o      -- o      -- o
;
;  Digit 0   Digit 1   Digit 2   Digit 3   Digit 4   Digit 5   Digit 6   Digit 7
;
;_____________________________________________________________________________
;
;
;=======================================
;       Display Ram for Digit n
;=======================================

;   up
;     ^  --a--
;       |     |                 (COM2)  (COM1)  (COM0)
;       f     b                   Bit2    Bit1    Bit0    Ram Display Address
;       |     |
;        --g--                    d       g       a       FA00 + (3n)
;       |     |                   e       f       up      FA00 + (3n + 1)
;       e     c                   dp      c       b       FA00 + (3n + 2)
;       |     |
;        --d--  o
;               dp
;
;
```

```
;
a_seg   EQU     Bit0            ; seven-segment Assignment
b_seg   EQU     Bit0
c_seg   EQU     Bit1
d_seg   EQU     Bit2
e_seg   EQU     Bit2
f_seg   EQU     Bit1
g_seg   EQU     Bit1
dp_seg  EQU     Bit2
up_seg  EQU     Bit0


;=====================================
;       seven-segment digit table
;=====================================

digits_7_seg:                           ; 3 bytes per digit

                                        ; "seven-segment 0" on segments
zero:   DB      a_seg+d_seg             ; a,d    ---
        DB      e_seg+f_seg             ; e,f   |   |
        DB      b_seg+c_seg             ; b,c
                                        ;       |   |
                                        ;        ---

                                        ; "seven-segment 1" on segments
one:    DB      0                       ;
        DB      0                       ;           |
        DB      b_seg+c_seg             ; b,c
                                        ;           |


                                        ; "seven-segment 2" on segments
two:    DB      a_seg+d_seg+g_seg       ; a,d,g   ---
        DB      e_seg                   ; e          |
        DB      b_seg                   ; b       ---
                                        ;       |
                                        ;        ---

                                        ; "seven-segment 3" on segments
three:  DB      a_seg+g_seg+d_seg       ; a,d,g  ---
        DB      0                       ;           |
        DB      b_seg+c_seg             ; b,c    ---
                                        ;           |
                                        ;        ---

                                        ; "seven-segment 4" on segments
four:   DB      g_seg                   ; a,d,g
        DB      f_seg                   ; f     |   |
        DB      b_seg+c_seg             ; b,c    ---
                                        ;           |
                                        ;


                                        ; "seven-segment 5" on segments
five:   DB      a_seg+g_seg+d_seg;      ;  a,d,g  ---
        DB      f_seg                   ; f     |
        DB      c_seg                   ; c      ---
                                        ;           |
                                        ;        ---

                                        ; "seven-segment 6" on segments
six:    DB      a_seg+g_seg+d_seg       ; a,d,g  ---
        DB      e_seg+f_seg             ; e,f   |
        DB      c_seg                   ; c      ---
                                        ;       |   |
                                        ;        ---

                                        ; "seven-segment 7" on segments
seven:  DB      a_seg                   ; a      ---
        DB      0                       ;           |
        DB      b_seg+c_seg             ; b,c
                                        ;           |
                                        ;
```

```
                                        ; "seven-segment 8" on segments
eight:  DB      a_seg+g_seg+d_seg       ; a,d,g  ---
        DB      e_seg+f_seg             ; e,f    |   |
        DB      b_seg+c_seg             ; b,c     ---
                                        ;        |   |
                                        ;         ---

                                        ; "seven-segment 9" on segments
nine:   DB      a_seg+g_seg+d_seg       ; a,d,g  ---
        DB      f_seg                   ; f      |   |
        DB      b_seg+c_seg             ; b,c     ---
                                        ;            |
                                        ;         ---

;=======================================
;       Main Program
;=======================================

MAIN    CSEG

Start:  DI                              ; Disable interrupts
        MOVW    AX, #0FE20h             ; Set Stack Pointer to top of stack
        MOVW    SP, AX
        MOV     PCC, #00h               ; Set CPU clock to fastest speed

                                        ; Configure LCD
        MOV     LPS0, #3Ch              ; Use ports as segments(S16-S23)
        MOV     LCDM0, #91h             ; Turn display on
                                        ; Normal operation
                                        ; LCD drive power is supplied
                                        ; 3 time slices - 1/3 bias
        MOV     LCDC0, #05h             ; Select subsystem clock for LCD
                                        ; LCD clock = 256Hz


        MOVW    HL, #LCDRam             ; Write 1's to bits 2-0 of display
        MOV     A, #(Bit2+Bit1+Bit0)    ; memory to turn all segments on
        MOV     B, #Segment_lines

All_on: MOV     [HL],A
        INCW    HL
        DBNZ    B,$All_on

        NOP                             ; Break here to check all segments

                                        ; Write value n to digit n
        MOVW    HL, #digits_7_seg       ; Set up source pointer
        MOVW    DE, #LCDRam             ; Set up destination pointer
        MOV     B, #Segment_lines       ; Set up count
Load:   MOV     A,[HL]                  ; Load first 8 values of table (0-7)
        MOV     [DE],A                  ; into display ram for Digit 0 -
        INCW    HL                      ; Digit 7 of MUB LCD
        INCW    DE
        DBNZ    B, $Load

Loop:   NOP                             ; Endless loop
        BR      $Loop                   ; Automatic readout from display ram

        END
;*********************************************************************
```

Figure 4-39. C Language Program Example of LCD Controller Using a 78K0S Microcontroller

```
/*********************************************************************
; Date:          5/19/00
;
; Parameters:    - fastest CPU clock (fx=5.00MHz)
;                - Bias mode: 1/3
;                - Duty: 1/3
;                - Common signals used: COM0-COM2
;                - Number of LCD segments: 72 (24 segment lines X 3 common)
;                - LCD operation clock: subsystem clock (32.768kHz)
;                - LCD clock frequency: 256Hz (32.768kHz/2^7)
;                - Frame frequency: 85Hz (256Hz/3)
;                - LCD: K0S LCD on MUB-K0-K0S LCD Multi-Use Board
;
;*********************************************************************/
/*=====================================
; Linker Options
;=====================================


Check boxes:
                Create Link Map File[-p]
                Create Stack Symbol[-s]
                Output Symbol Information[-g]

Other options:   c:\nectools\lib78k0s\lib\s0s -olcd -plcd

;=====================================
; extension functions in K0/K0S compiler
;=====================================*/

#pragma sfr                             /* allow SFR names in C code */
#pragma DI                              /* key word for DI instruction */
#pragma NOP                             /* key word for NOP instruction */


/*=====================================
; Constants/Variables
;=====================================*/

#define TRUE  1
#define FALSE 0

#define  LCDRam 0xFA00                  /* Start of LCD display Ram area */
#define Segment_lines 24                /* Number of LCD segment lines */

#define Bit2 4                          /* bit 2 */
#define Bit1 2                          /* bit 1 */
#define Bit0 1                          /* bit 0 */

sreg     unsigned char * LCD_Ptr;       /* LCD Display memory pointer */
sreg     unsigned char * Tbl_Ptr;       /* Seven segment table pointer */
unsigned char    i;                     /* General purpose count variable */

/*=====================================
;       8-Digit K0S LCD on MUB Board
;=====================================

;_____________________________________________________________________________

;  ^  --     ^  --     ^  --     ^  --     ^  --     ^  --     ^  --     ^  --
;    |  |      |  |      |  |      |  |      |  |      |  |      |  |      |  |
;     --        --        --        --        --        --        --        --
;    |  |      |  |      |  |      |  |      |  |      |  |      |  |      |  |
;     -- o      -- o      -- o      -- o      -- o      -- o      -- o      -- o
;
;  Digit 0   Digit 1   Digit 2   Digit 3   Digit 4   Digit 5   Digit 6   Digit 7
;
;_____________________________________________________________________________
;
;
```

```
;=======================================
;       Display Ram for Digit n
;=======================================

;   up
;     ^  --a--
;       |     |                (COM2)  (COM1)  (COM0)
;       f     b                 Bit2    Bit1    Bit0    Ram Display Address
;       |     |
;        --g--                  d       g       a       FA00 + (3n)
;       |     |                 e       f       up      FA00 + (3n + 1)
;       e     c                 dp      c       b       FA00 + (3n + 2)
;       |     |
;        --d--  o
;               dp
;
;
*/
#define  a_seg    Bit0                    /* seven-segment Assignment */
#define  b_seg    Bit0
#define  c_seg    Bit1
#define  d_seg   Bit2
#define  e_seg   Bit2
#define  f_seg   Bit1
#define  g_seg   Bit1
#define  dp_seg   Bit2
#define  up_seg   Bit0


/*=======================================
;       seven-segment digit table
;=======================================*/

const unsigned char digits_7_seg[] =       /* 3 bytes per digit */
{
                                           /* "seven-segment 0" on segments */
a_seg+d_seg,                               /* a,d   ---     */
e_seg+f_seg,                               /* e,f   |   |    */
b_seg+c_seg,                               /* b,c           */
                                           /*       |   |    */
                                           /*        ---     */

                                           /* "seven-segment 1" on segments */
0,                                         /*               */
0,                                         /*           |    */
b_seg+c_seg,                               /* b,c           */
                                           /*           |    */
                                           /*               */

                                           /* "seven-segment 2" on segments */
a_seg+d_seg+g_seg,                         /* a,d,g ---     */
e_seg,                                     /* e         |    */
b_seg,                                     /* b      ---     */
                                           /*       |       */
                                           /*        ---     */

                                           /* "seven-segment 3" on segments */
a_seg+g_seg+d_seg,                         /* a,d,g ---     */
0,                                         /*           |    */
b_seg+c_seg,                               /* b,c    ---     */
                                           /*           |    */
                                           /*        ---     */

                                           /* "seven-segment 4" on segments */
g_seg,                                     /* a,d,g         */
f_seg,                                     /* f     |   |    */
b_seg+c_seg,                               /* b,c    ---     */
                                           /*           |    */
                                           /*               */

                                           /* "seven-segment 5" on segments */
a_seg+g_seg+d_seg,                         /* a,d,g ---     */
f_seg,                                     /* f     |       */
c_seg,                                     /* c      ---     */
                                           /*           |    */
                                           /*        ---     */
```

```
                                        /* "seven-segment 6" on segments */
a_seg+g_seg+d_seg,                      /* a,d,g ---     */
e_seg+f_seg,                            /* e,f   |       */
c_seg,                                  /* c      ---    */
                                        /*       |   |   */
                                        /*        ---    */

                                        /* "seven-segment 7" on segments */
a_seg,                                  /* a     ---     */
0,                                      /*           |   */
b_seg+c_seg,                            /* b,c           */
                                        /*           |   */
                                        /*               */

                                        /* "seven-segment 8" on segments */
a_seg+g_seg+d_seg,                      /* a,d,g ---     */
e_seg+f_seg,                            /* e,f   |   |   */
b_seg+c_seg,                            /* b,c    ---    */
                                        /*       |   |   */
                                        /*        ---    */

                                        /* "seven-segment 9" on segments */
a_seg+g_seg+d_seg,                      /* a,d,g ---     */
f_seg,                                  /* f     |   |   */
b_seg+c_seg                             /* b,c    ---    */
                                        /*           |   */
                                        /*        ---    */
};


/*=====================================
; Main Program
;=====================================*/

void main(void)
{

DI();                                   /* Disable interrupts */
                                        /* Stack pointer set by compiler */
PCC     = 0x00;                         /* Set CPU clock to fastest speed */


                                        /* Configure LCD */
LPS0    = 0x3C;                         /* Use ports as segments(S16-S23) */
LCDM0   = 0x91;                         /* Turn display on */
                                        /* Normal operation */
                                        /* LCD drive power is supplied */
                                        /* 3 time slices - 1/3 bias */
LCDC0   = 0x05;                         /* Select subsystem clock for LCD */
                                        /* LCD clock = 256Hz */


LCD_Ptr  = (unsigned char *)LCDRam;     /* Write 1's to bits 2-0 of display */
for(i=0;i<Segment_lines;i++)            /* memory to turn all segments on */
{
        *LCD_Ptr++=Bit2+Bit1+Bit0;
}

        NOP();                          /* Break here to check all segments */

                                        /* Write value n to digit n */
Tbl_Ptr  = &digits_7_seg[0];            /* Set up source pointer */
LCD_Ptr  = (unsigned char *)LCDRam;     /* Set up destination pointer */
for(i=0;i<Segment_lines;i++)            /* Load first 8 values of table (0-7) */
{                                       /* into display ram for Digit 0 -  */
        *LCD_Ptr=*Tbl_Ptr;              /* Digit 7 of MUB LCD */
        LCD_Ptr++;
        Tbl_Ptr++;
}

while(TRUE)                             /* Endless loop */
{                                       /* Automatic readout from display ram */
        NOP();
}

}                                       /* End of function main */

/*****************************************************************/
```

4.2.2 Programming Example for 78K0 Microcontrollers

Program Description

The LCD controller/driver in the μPD78030x microcontrollers has the following functions.

- Automatic output of segment and common signals with data from display memory
- Five different display modes:
  - » Static
  - » 1/2 duty (1/2 bias)
  - » 1/3 duty (1/2 bias)
  - » 1/3 duty (1/3 bias)
  - » 1/4 duty (1/3 bias)
- Four different frame frequencies, selectable in each display mode
- 40 segment signal outputs (S0–S39); 16 of these segment outputs can be switched to I/O ports (P80/S39 to P87/S32 and P90/S31 to P97/S24).
- Voltage divider resistors (for LCD drive voltage operation) can be specified with a mask option.
- Operation from main clock or subsystem clock

This program demonstrates how to use the LCD Controller/Driver to write the values 0 to 7 to the 8 digits of the K0 LCD on NEC's MUB-K0/K0S LCD Multi-Use Board.

The K0 LCD on the MUB board is an eight digit LCD configured for operation 1/3 duty, 1/3 bias mode. Each seven-segment digit has an "up" icon on the top left and a decimal point icon on the bottom right. This gives a total of 72 segments [(7 segment digit + 2 icons) x 8 digits]. With three common lines (COM0, COM1, and COM2), 72 ÷ 3 = 24 segment lines are required. The μPD78030x microcontrollers have 24 dedicated segment lines, so no additional segment lines need to be reassigned from port lines.

The subsystem clock is chosen for LCD operation by setting bit TCL24 of the Timer Clock Select 2 (TCL2) register. The prescaler for the clock is enabled by setting bit TMC21 of the Watch Timer Mode Control Register 2 (TMC2). Writing to the LCD display control register (LCDC) specifies that P80–P97 are to be left as port pins and that LCD power is to be supplied from the V_{DD} pin.

The LCD display mode (LCDM) register is set up as follows:

- Display on
- LCD clock = 256 Hz (32.768 kHz ÷ 2^7) giving a frame frequency of 85 Hz (256 Hz/3)
- Normal operation (2.5V–5.5V in 1/3 bias mode)
- 1/3 duty and 1/3 bias

The program initializes the display memory with all 1's. There is a NOP inserted in the program here to give the user a convenient place to break and check that all segments on the LCD are working. The program then writes the seven segment values 0–7 to the LCD digits 0–7 before idling in an endless loop.

Table 4-5. Program Specifications

Feature	Specification
LCD Bias mode	1/3
LCD Duty	1/3
LCD Operating clock	32.768 kHz subsystem clock
LCD Clock frequency	250 Hz (32.768 kHz ÷ 2^7)
LCD Frame frequency	85 Hz (250 Hz ÷ 3)
LCD Common signals used	COM0, COM1, COM2
LCD Pins used in program	BIAS, VLC0, VLC1, VLC2, COM0, COM1, COM2, S0–S23
Number of LCD Segments	72 (24 segment lines x 3 common)
Other options	c:\nectools\lib78k0\lib\s0 -bcl0 -olcd -plcd
Number of LCD Segments	72 (24 segment lines x 3 common)
Linker options for C program	Check boxes
	» Create Link Map File [-p]
	» Create Stack Symbol [-s]
	» Output Symbol Information [-g]
	Other options: c:\nectools\lib78k0\lib\s0 -bc10 –olcd -plcd

Figure 4-40. LCD Program Flowchart

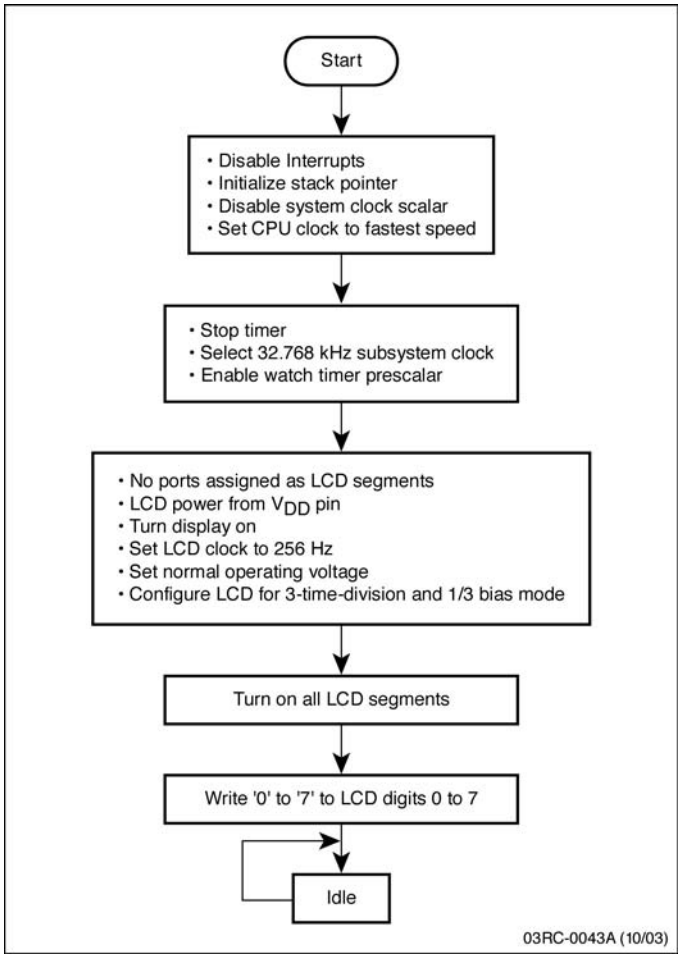


Figure 4-41. Assembly Language Program Example of LCD Controller Using a 78K0 Microcontroller

```
;*********************************************************************
; Date:          10/31/00
;
; Parameters:    - fastest CPU clock (fx=5.00MHz)
;                - Bias mode: 1/3
;                - Time divisions: 3
;                - Common signals used: COM0-COM2
;                - Number of LCD segments: 72 (24 segment lines X 3 common)
;                - LCD operation clock: subsystem clock (32.768kHz)
;                - LCD clock frequency: 256Hz (32.768kHz/2^7)
;                - Frame frequency: 85Hz (256Hz/3)
;                - LCD: K0 LCD on MUB-K0-K0S LCD Multi-Use Board
;
;*********************************************************************


;=======================================
;       Constants
;=======================================

LCDRam          EQU     0FA7Fh          ; Start of LCD display Ram area
Segment_lines   EQU     24              ; Number of LCD segment lines

Bit2            EQU 4                   ; bit 2
Bit1            EQU 2                   ; bit 1
Bit0            EQU 1                   ; bit 0


;=======================================
;       Specify Interrupt Vectors
;=======================================

RESET_VECTOR    CSEG    AT 0000h ; On reset, go to Start
                DW      Start


;=======================================
;       8-Digit K0 LCD on MUB Board
;=======================================

;_____________________________________________________________________________
;  ^  --     ^  --     ^  --     ^  --     ^  --     ^  --     ^  --     ^  --
;    |  |      |  |      |  |      |  |      |  |      |  |      |  |      |  |
;     --        --        --        --        --        --        --        --
;    |  |      |  |      |  |      |  |      |  |      |  |      |  |      |  |
;     -- o      -- o      -- o      -- o      -- o      -- o      -- o      -- o
;
;  Digit 0   Digit 1   Digit 2   Digit 3   Digit 4   Digit 5   Digit 6   Digit 7
;
;_____________________________________________________________________________
;
;
;=======================================
;       Display Ram for Digit n
;=======================================

;   up
;     ^  --a--
;       |     |                 (COM2)  (COM1)  (COM0)
;       f     b                   Bit2    Bit1    Bit0    Ram Display Address
;       |     |
;        --g--                    d       g       a       FA7F - (3n)
;       |     |                   e       f       up      FA7F - (3n + 1)
;       e     c                   dp      c       b       FA7F - (3n + 2)
;       |     |
;        --d--  o
;               dp
;
;
```

```
;
a_seg    EQU     Bit0             ; seven-segment Assignment
b_seg    EQU     Bit0
c_seg    EQU     Bit1
d_seg    EQU     Bit2
e_seg    EQU     Bit2
f_seg    EQU     Bit1
g_seg    EQU     Bit1
dp_seg   EQU     Bit2
up_seg   EQU     Bit0


;=======================================
;       seven-segment digit table
;=======================================

digits_7_seg:                           ; 3 bytes per digit

                                        ; "seven-segment 0" on segments
zero:    DB      a_seg+d_seg            ; a,d    ---
         DB      e_seg+f_seg            ; e,f   |   |
         DB      b_seg+c_seg            ; b,c
                                        ;       |   |
                                        ;        ---

                                        ; "seven-segment 1" on segments
one:     DB      0                      ;
         DB      0                      ;           |
         DB      b_seg+c_seg            ; b,c
                                        ;           |


                                        ; "seven-segment 2" on segments
two:     DB      a_seg+d_seg+g_seg      ; a,d,g   ---
         DB      e_seg                  ; e          |
         DB      b_seg                  ; b       ---
                                        ;       |
                                        ;        ---

                                        ; "seven-segment 3" on segments
three:   DB      a_seg+g_seg+d_seg      ; a,d,g  ---
         DB      0                      ;           |
         DB      b_seg+c_seg            ; b,c     ---
                                        ;           |
                                        ;        ---

                                        ; "seven-segment 4" on segments
four:    DB      g_seg                  ; a,d,g
         DB      f_seg                  ; f      |   |
         DB      b_seg+c_seg            ; b,c     ---
                                        ;           |
                                        ;


                                        ; "seven-segment 5" on segments
five:    DB      a_seg+g_seg+d_seg      ; a,d,g  ---
         DB      f_seg                  ; f      |
         DB      c_seg                  ; c       ---
                                        ;           |
                                        ;        ---

                                        ; "seven-segment 6" on segments
six:     DB      a_seg+g_seg+d_seg      ; a,d,g  ---
         DB      e_seg+f_seg            ; e,f   |
         DB      c_seg                  ; c       ---
                                        ;       |   |
                                        ;        ---

                                        ; "seven-segment 7" on segments
seven:   DB      a_seg                  ; a      ---
         DB      0                      ;           |
         DB      b_seg+c_seg            ; b,c
                                        ;           |
                                        ;
```

```
                                        ; "seven-segment 8" on segments
eight:  DB      a_seg+g_seg+d_seg       ; a,d,g  ---
        DB      e_seg+f_seg             ; e,f    |   |
        DB      b_seg+c_seg             ; b,c     ---
                                        ;       |   |
                                        ;        ---

                                        ; "seven-segment 9" on segments
nine:   DB      a_seg+g_seg+d_seg       ; a,d,g  ---
        DB      f_seg                   ; f      |   |
        DB      b_seg+c_seg             ; b,c     ---
                                        ;           |
                                        ;        ---

;=======================================
;       Main Program
;=======================================

MAIN    CSEG

Start:  DI                              ; Disable interrupts
        MOVW    SP, #0FE20h             ; Set Stack Pointer to top of stack
        MOV     OSMS, #01h              ; Disable system clock scaler
        MOV     PCC, #00h               ; Set CPU clock to fastest speed

        MOV     TMC2, #00h              ; Stop and clear watch timer
        MOV     TCL2, #10h              ; Select 32.768kHz subsystem clock
        MOV     TMC2, #02h              ; Enable watch timer prescaler

                                        ; Configure LCD
        MOV     LCDC, #01h              ; No ports reassigned to segment use
                                        ; LCD power from Vdd pin
        MOV     LCDM, #0A1h             ; Turn display on
                                        ; LCD clock = 256Hz
                                        ; Normal operating voltage
                                        ; 3 time divisions - 1/3 bias


        MOVW    HL, #LCDRam             ; Write 1's to bits 2-0 of display
        MOV     A, #(Bit2+Bit1+Bit0)    ; memory to turn all segments on
        MOV     B, #Segment_lines

All_on: MOV     [HL],A
        DECW    HL
        DBNZ    B,$All_on

        NOP                             ; Break here to check all segments

                                        ; Write value n to digit n
        ;
        MOVW    HL, #digits_7_seg       ; Set up source pointer
        MOVW    DE, #LCDRam             ; Set up destination pointer
        MOV     B, #Segment_lines       ; Set up count
Load:   MOV     A,[HL]                  ; Load first 8 values of table (0-7)
        MOV     [DE],A                  ; into display ram for Digit 0 -
        INCW    HL                      ; Digit 7 of MUB LCD
        DECW    DE
        DBNZ    B, $Load

Loop:   NOP                             ; Endless loop
        BR      $Loop                   ; Automatic readout from display ram

        END
;*********************************************************************
```

C Language Program Example of LCD Controller Using a 78K0 Microcontroller

```
/*****************************************************************
; Date:          10/31/00
;
; Parameters:    - fastest CPU clock (fx=5.00MHz)
;                - Bias mode: 1/3
;                - Time divisions: 3
;                - Common signals used: COM0-COM2
;                - Number of LCD segments: 72 (24 segment lines X 3 common)
;                - LCD operation clock: subsystem clock (32.768kHz)
;                - LCD clock frequency: 256Hz (32.768kHz/2^7)
;                - Frame frequency: 85Hz (256Hz/3)
;                - LCD: K0 LCD on MUB-K0-K0S LCD Multi-Use Board
;
*****************************************************************/
/*=====================================
; Project Manager Linker Options
;=====================================


Check boxes:
                Create Link Map File[-p]
                Create Stack Symbol[-s]
                Output Symbol Information[-g]

Other options:   c:\nectools\lib78k0\lib\s0 -bcl0 -olcd -plcd

;=====================================
; extension functions in K0/K0S compiler
;=====================================*/

#pragma sfr                             /* allow SFR names in C code */
#pragma DI                              /* key word for DI instruction */
#pragma NOP                             /* key word for NOP instruction */


/*=====================================
; Constants/Variables
;=====================================*/

#define TRUE  1
#define FALSE 0

#define  LCDRam 0xFA7F                  /* Start of LCD display Ram area */
#define Segment_lines 24                /* Number of LCD segment lines */

#define Bit2 4                          /* bit 2 */
#define Bit1 2                          /* bit 1 */
#define Bit0 1                          /* bit 0 */

sreg     unsigned char * LCD_Ptr;       /* LCD Display memory pointer */
sreg     unsigned char * Tbl_Ptr;       /* Seven segment table pointer */
unsigned char    i;                     /* General purpose count variable */

/*=====================================
;       8-Digit K0 LCD on MUB Board
;=====================================

;_____________________________________________________________________________

;  ^  --     ^  --     ^  --     ^  --     ^  --     ^  --     ^  --     ^  --
;    |  |      |  |      |  |      |  |      |  |      |  |      |  |      |  |
;     --        --        --       --        --       --        --       --
;    |  |      |  |      |  |      |  |      |  |      |  |      |  |      |  |
;     -- o      -- o      -- o      -- o      -- o      -- o      -- o      -- o
;
;  Digit 0   Digit 1   Digit 2   Digit 3   Digit 4   Digit 5   Digit 6   Digit 7
;
;_____________________________________________________________________________
;
;
;=====================================
;       Display Ram for Digit n
;=====================================
```

```
;   up
;     ^  --a--
;       |     |                 (COM2)  (COM1)  (COM0)
;       f     b                   Bit2    Bit1    Bit0    Ram Display Address
;       |     |
;        --g--                    d       g       a       FA7F - (3n)
;       |     |                   e       f       up      FA7F - (3n + 1)
;       e     c                   dp      c       b       FA7F - (3n + 2)
;       |     |
;        --d--  o
;               dp
;
;
*/
#define  a_seg    Bit0                    /* seven-segment Assignment */
#define  b_seg    Bit0
#define  c_seg    Bit1
#define  d_seg   Bit2
#define  e_seg   Bit2
#define  f_seg   Bit1
#define  g_seg   Bit1
#define  dp_seg   Bit2
#define  up_seg   Bit0


/*=======================================
;       seven-segment digit table
;=======================================*/

const unsigned char digits_7_seg[] =       /* 3 bytes per digit */
{
                                           /* "seven-segment 0" on segments */
a_seg+d_seg,                               /* a,d   ---     */
e_seg+f_seg,                               /* e,f   |   |    */
b_seg+c_seg,                               /* b,c           */
                                           /*       |   |    */
                                           /*        ---     */

                                           /* "seven-segment 1" on segments */
0,                                         /*               */
0,                                         /*           |   */
b_seg+c_seg,                               /* b,c           */
                                           /*           |   */
                                           /*               */

                                           /* "seven-segment 2" on segments */
a_seg+d_seg+g_seg,                         /* a,d,g ---     */
e_seg,                                     /* e         |   */
b_seg,                                     /* b      ---    */
                                           /*       |       */
                                           /*        ---    */

                                           /* "seven-segment 3" on segments */
a_seg+g_seg+d_seg,                         /* a,d,g ---     */
0,                                         /*           |   */
b_seg+c_seg,                               /* b,c    ---    */
                                           /*           |   */
                                           /*        ---    */

                                           /* "seven-segment 4" on segments */
g_seg,                                     /* a,d,g         */
f_seg,                                     /* f     |   |   */
b_seg+c_seg,                               /* b,c    ---    */
                                           /*           |   */
                                           /*               */

                                           /* "seven-segment 5" on segments */
a_seg+g_seg+d_seg,                         /* a,d,g ---     */
f_seg,                                     /* f     |       */
c_seg,                                     /* c      ---    */
                                           /*           |   */
                                           /*        ---    */

                                           /* "seven-segment 6" on segments */
```

```
a_seg+g_seg+d_seg,                      /* a,d,g ---     */
e_seg+f_seg,                            /* e,f   |       */
c_seg,                                  /* c      ---    */
                                        /*       |   |   */
                                        /*        ---    */

                                        /* "seven-segment 7" on segments */
a_seg,                                  /* a     ---     */
0,                                      /*           |   */
b_seg+c_seg,                            /* b,c           */
                                        /*           |   */
                                        /*               */

                                        /* "seven-segment 8" on segments */
a_seg+g_seg+d_seg,                      /* a,d,g ---     */
e_seg+f_seg,                            /* e,f   |   |   */
b_seg+c_seg,                            /* b,c    ---    */
                                        /*       |   |   */
                                        /*        ---    */

                                        /* "seven-segment 9" on segments */
a_seg+g_seg+d_seg,                      /* a,d,g ---     */
f_seg,                                  /* f     |   |   */
b_seg+c_seg                             /* b,c    ---    */
                                        /*           |   */
                                        /*        ---    */
};

/*====================================
; Main Program
;====================================*/

void main(void)
{

DI();                                   /* Disable interrupts */
                                        /* Stack pointer set by compiler */
OSMS     = 0x01;                        /* Disable system clock scaler */
PCC      = 0x00;                        /* Set CPU clock to fastest speed */

TMC2     = 0x00;                        /* Stop and clear watch timer */
TCL2     = 0x10;                        /* Select 32.768kHz subsystem clock */
TMC2    = 0x02;                         /* Enable watch timer prescaler */

                                        /* Configure LCD */
LCDC     = 0x01;                        /* No ports reassigned to segment use */
                                        /* LCD power from Vdd pin */
LCDM     = 0xA1;                        /* Turn display on */
                                        /* LCD clock = 256Hz
                                        /* Normal operating voltage */
                                        /* 3 time divisions - 1/3 bias */

LCD_Ptr  = (unsigned char *)LCDRam;     /* Write 1's to bits 2-0 of display */
for(i=0;i<Segment_lines;i++)            /* memory to turn all segments on */
{
        *LCD_Ptr-- = Bit2+Bit1+Bit0;
}

        NOP();                          /* Break here to check all segments */

                                        /* Write value n to digit n */
Tbl_Ptr  = &digits_7_seg[0];            /* Set up source pointer */
LCD_Ptr  = (unsigned char *)LCDRam;     /* Set up destination pointer */
for(i=0;i<Segment_lines;i++)            /* Load first 8 values of table (0-7) */
{                                       /* into display ram for Digit 0 -  */
        *LCD_Ptr=*Tbl_Ptr;              /* Digit 7 of MUB LCD */
        LCD_Ptr--;
        Tbl_Ptr++;
}

while(TRUE)                             /* Endless loop */
{                                       /* Automatic readout from display ram */
        NOP();
}

}                                       /* End of function main */
/*****************************************************************/
```

These commodities, technology or software, must be exported from the U.S. in accordance with the export administration regulations. Diversion contrary to U.S. law prohibited.

The information in this document is current as of August 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.

Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC Electronics no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

NEC Electronics products are classified into the following three quality grades: “Standard”, “Special” and “Specific”.

The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated “quality assurance program” for a specific application. The recommended applications of NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

"Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

"Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

"Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is “Standard” unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact NEC Electronics sales representative in advance to determine NEC Electronics 's willingness to support a given application.

(Notes)

(1) " NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) " NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).