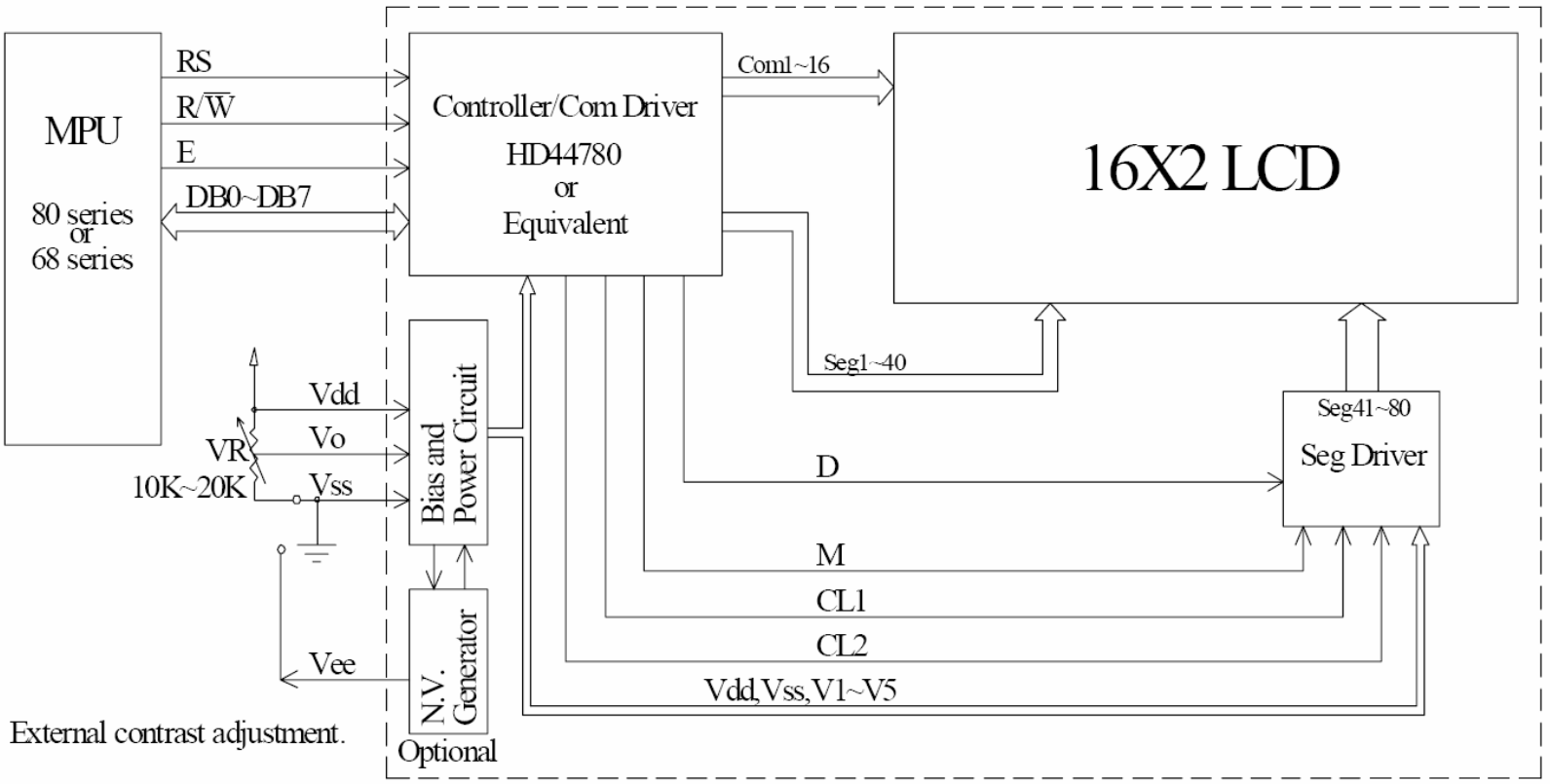
Micro Processor & Controller

Parallel Bus – LCD Display

LCD Display Pin Assignment

Pin No.	Symbol	Level	Description
1	V_{SS}	0V	Ground
2	V_{DD}	5.0V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→Module) L: Write(MPU→Module)
6	E	H,H→L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	A	—	LED +
16	K	—	LED—

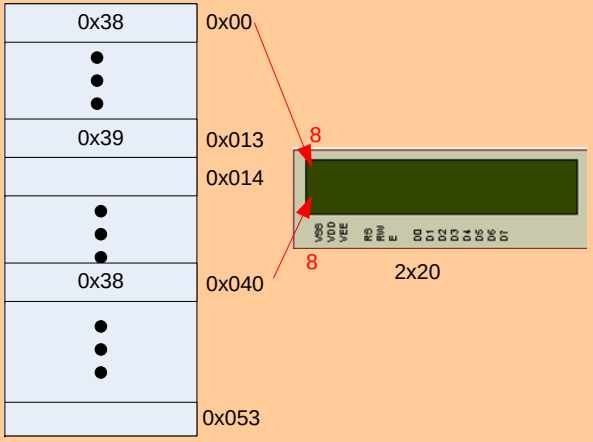
LCD Display Architecture



Character located	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DDRAM address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

LCD Display Hardware

- **Hardware**
  - **16 x 2-line LCD displays (two lines with 16 characters per line)**
  - **LCD has a display Data RAM (registers) that stores data in 8-bit character code.**
  - **Each register in Data RAM has its own address that corresponds to its position on the line.**
  - **The address range for Line 1 is 00 to 0FH and Line 2 is 40H to 4FH.**



Instructions Table

Instruction	Instruction Code										Description	Execution time (fosc=270Khz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "00H" to DDRAM and set DDRAM address to "00H" from AC	1.53ms
Return Home	0	0	0	0	0	0	0	0	1	—	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor move direction and enable the shift of entire display.	39 μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display (D), cursor (C), and blinking of cursor (B) on/off control bit.	39 μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	—	—	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	39 μs
Function Set	0	0	0	0	1	DL	N	F	—	—	Set interface data length (DL:8-bit/4-bit), numbers of display line (N:2-line/1-line)and, display font type (F:5×11 dots/5×8 dots)	39 μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 μs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 μs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 μs

＊ "—" : don't care

Instructions Table

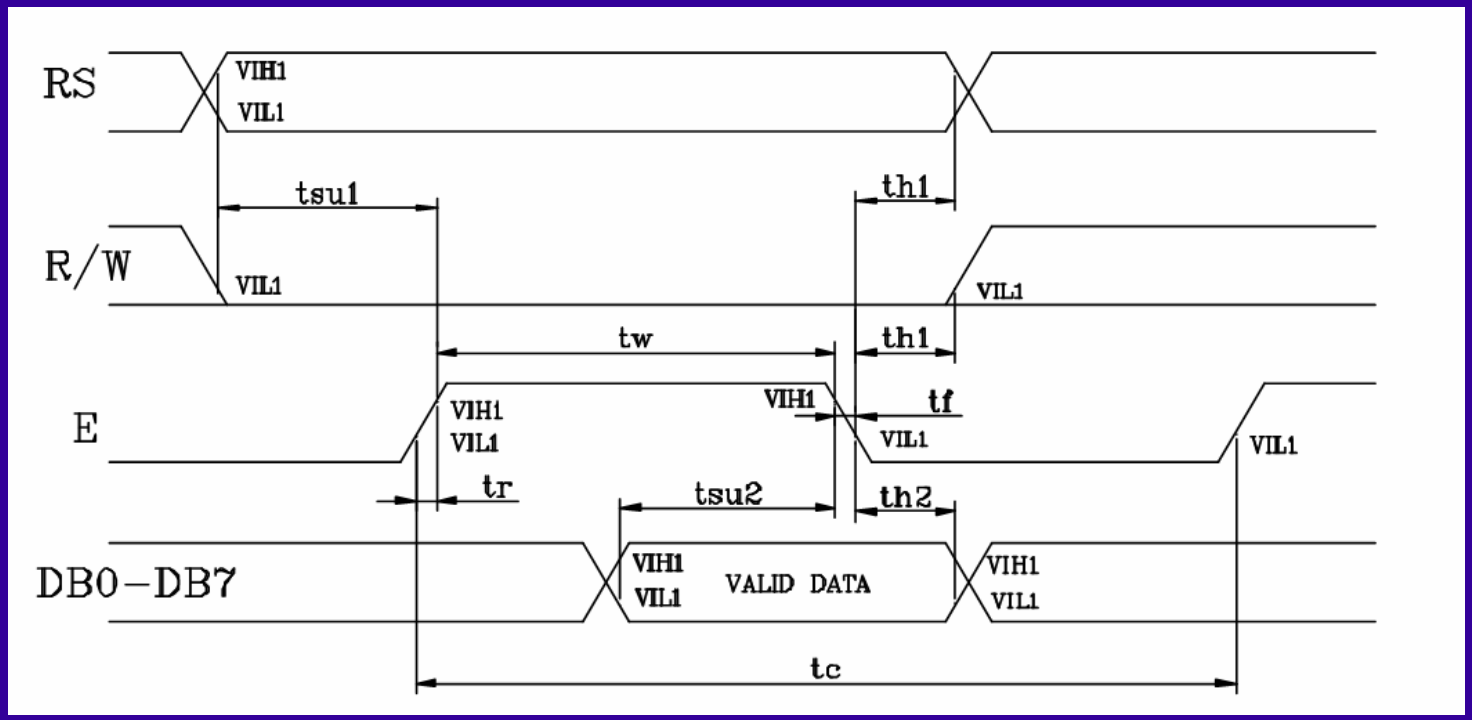
RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0	Description
0	0	0	0	0	0	0	0	0	1	Clears all display, return cursor home
0	0	0	0	0	0	0	0	1	*	Returns cursor home
0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and/or specifies not to shift display
0	0	0	0	0	0	1	D	C	B	ON/OFF of all display(D), cursor ON/OFF (C), and blink position (B)
0	0	0	0	0	1	S/C	R/L	*	*	Move cursor and shifts display
0	0	0	0	1	DL	N	F	*	*	Sets interface data length, number of display lines, and character font
1	0	WRITE DATA								Writes Data

Control & Data Format

CODES	
I/D = 1 cursor moves left	DL = 1 8-bit
I/D = 0 cursor moves right	DL = 0 4-bit
S = 1 with display shift	N = 1 2 rows
S/C =1 display shift	N = 0 1 row
S/C = 0 cursor movement	F = 1 5x10 dots
R/L = 1 shift to right	F = 0 5x7 dots
R/L = 0 shift to left	

LCD Controller Timing

Parameter	Symbol	Test pin	Min.	Typ.	Max.	Unit
Enable cycle time	t_c	E	500	-	-	ns
Enable pulse width	t_w		300	-	-	
Enable rise/fall time	t_r, t_f		-	-	25	
RS; R/W setup time	t_{su1}	RS; R/W	100	-	-	
RS; R/W address hold time	t_{h1}	RS; R/W	10	-	-	
Read data output delay	t_{su2}	DB0~DB7	60	-	-	
Read data hold time	t_{h2}	DB0~DB7	10	-	-	

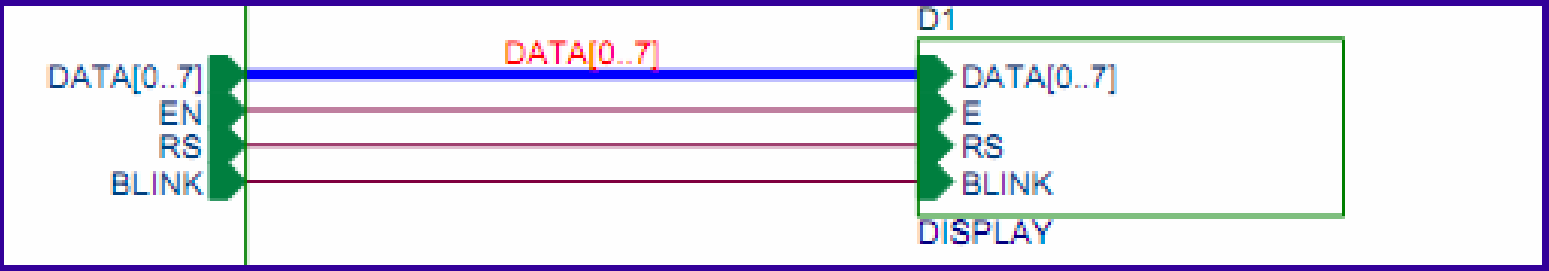


Interfacing LCD

Writing to LCD.

- The MPU:
  - Asserts RS (low to select IR, high to select DR).
  - Writes into LCD by asserting the R/W signal low.
  - Write data (char or instruction) to Data Bus.
  - Asserts the E signal high and then low (toggles) to latch a data byte or an instruction (Delay of 1 us is needed).

EVB with LCD controller



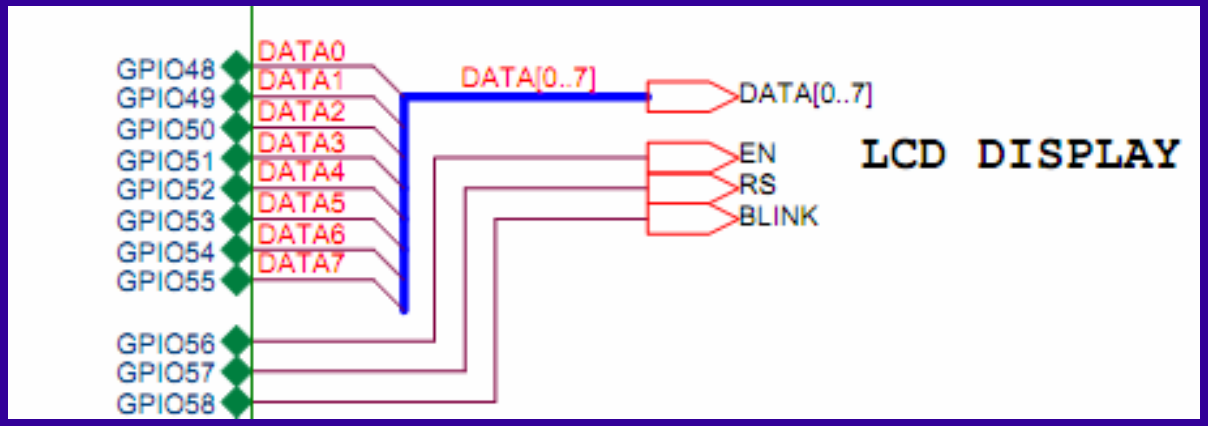
RS (function select) RS=0 command mode, RS=1 data mode **– GPIO57**

W/R (write or read) Always 0 **– write only mode**

E (Latch enable) falling edge **– GPIO56**

Blink (display background light) **– GPIO58**

Data bus (8 bit) **– GPIO 48-55**



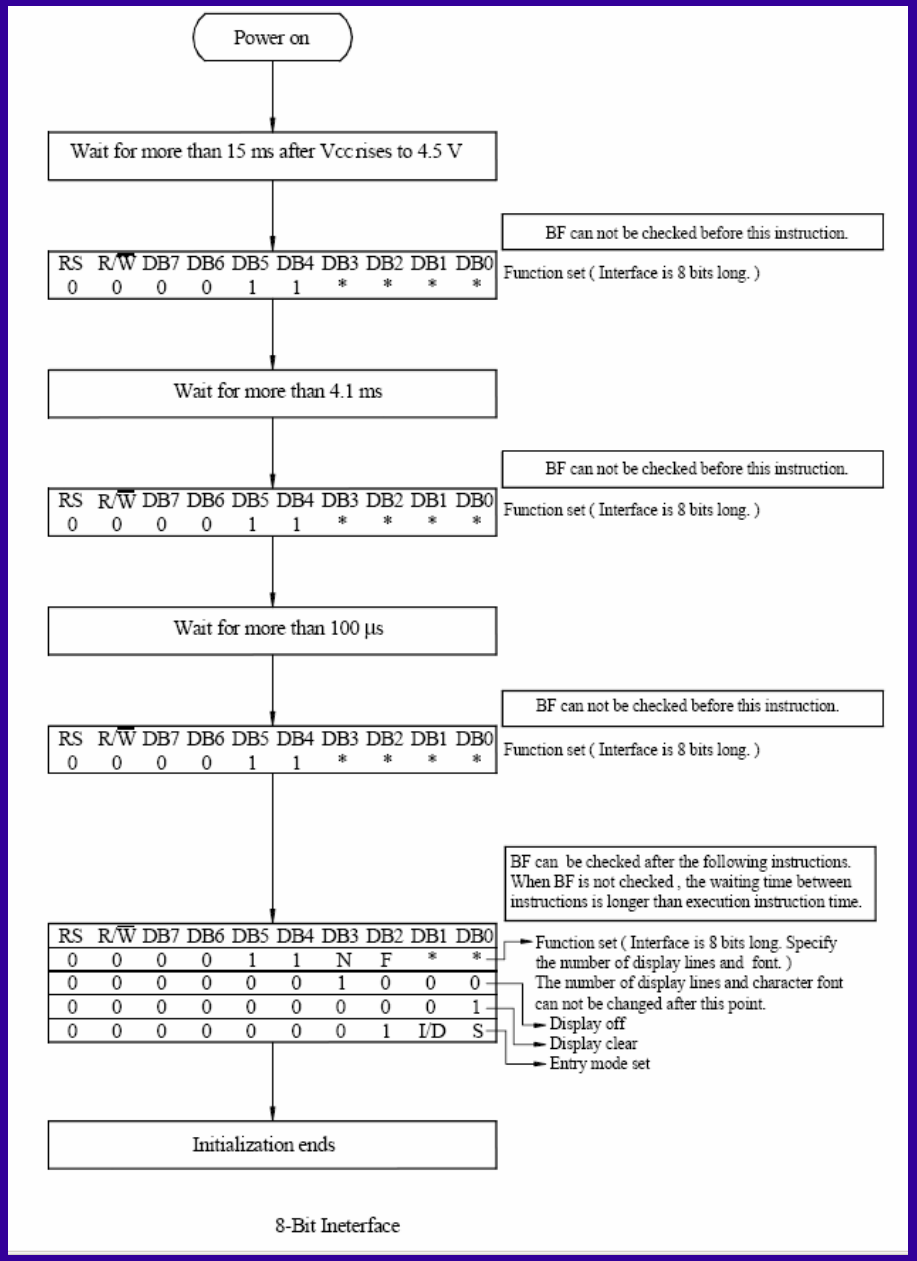
Initialize LCD

- Software

  To write into the LCD, the program should:

  - Send the initial instructions (commands) to set up the LCD in the 4-bit or the 8-bit mode.
  - Write instructions to IR to set up the LCD parameters such as the number of display lines and cursor status.
  - Write data to display a message.

Initialization Procedure




```
void InItLCD(void)
  {
  static const char LcdInitStr[] = {0x38, 0x0E, 0x06, 0x01};
  int i;

  LcdControlBit(0);      // Control

  for (i=0; i<4; i++)
    {
    LcdEnablelBit(1);
    LcdWriteData(LcdInitStr[i]);
    DELAY_US(10000);
    LcdEnablelBit(0);
    DELAY_US(5000);
    }
  }
```

LCD Library

```
/*********************************************************************
 *   Tittle:            Delfino Evaluation Board LCD 2x16 Display     *
 *   Filename:          LCD2x16Display.c                              *
 *   Date:              20-11-2014                                    *
 *   Last Modify:       20-11-2014                                    *
 *   File Version:      1.0                                           *
 *                                                                    *
 *   Author:            Flaxer Eli                                    *
 *   Company:           Flaxer.net                                    *
 *                                                                    *
 *                                                                    */


#include "DSP28x_Project.h"     // Device Header file and Examples Include File

/*****************************************************************************/
static inline void LcdControlBit(int bit)
  {
  if (bit)
      GpioDataRegs.GPBSET.bit.GPIO57 = 1;
  else
      GpioDataRegs.GPBCLEAR.bit.GPIO57 = 1;
  }
/*****************************************************************************/
static inline void LcdEnablelBit(int bit)
  {
  if (bit)
      GpioDataRegs.GPBSET.bit.GPIO56 = 1;
  else
      GpioDataRegs.GPBCLEAR.bit.GPIO56 = 1;
  }
/*****************************************************************************/
static inline void LcdBlinklBit(int bit)
  {
  if (bit)
      GpioDataRegs.GPBSET.bit.GPIO58 = 1;
  else
      GpioDataRegs.GPBCLEAR.bit.GPIO58 = 1;
  }
/*****************************************************************************/
static inline void LcdWriteData(char data)
  {
  GpioDataRegs.GPBCLEAR.all = (0xFFL << 16);    // Clear all data bits GPIO48-GPIO55
  GpioDataRegs.GPBSET.all = ((long)data << 16); // Set the relevant data bits GPIO48-GPIO55
  }
/*****************************************************************************/
void BackLightLCD(int x)
  {
  LcdBlinklBit(x);
  }
/*****************************************************************************/
void PutcLCD(const char c)
  {
  LcdControlBit(1);     // Data
  DELAY_US(1);
  LcdEnablelBit(1);
  LcdWriteData(c);
```