



## Module Classification Information

---

W M - C 1 6 0 2 A - 1 T L Y a

1 2 - 3 4 5 6 7 8 - 9 10 11 12 13

1 ----- Brand:wintek Corp.

2 ----- LCM Type:M:Standard LCM. D:Customer's Design LCM.

3 ----- Display Type:S:Segment. C:Character. G:Graphic.

4-7 --- Display Function:Column and Row of segments/Column and Row of Characters/Column and Row of Dots.

8 ----- Package of Driving Circuit:A-L:SMT. M-U:COB(except"0")  
V-Z:COG. 1-9:TAB. 0:other.

9 ----- Temperature Range & View Direction:  
General Purpose: 1: 6:00, 2: 12:00, 3: 3:00, 4: 9:00  
High Performance:6: 6:00, 7: 12:00, 8: 3:00,9: 9:00.

10 ----- LCD model & background color:  
T: TN Natural Color , H:HTN, G:STN Gray, B:STN Blue  
Y:STN Yellow, W:FSTN Black/White , 0:OTHER.

11 ----- Backlight Type : N:Without Backlight , L:LED , E:EL  
F:CCFL.

12 ----- Backlight Color:N:Without Backlight , Y:Yellow Green,  
G:Green , B:Blue, A:Amber, W:White.  
R:Red , O: Orange , X:other.

13 ----- Serial Code.

**Optical Characteristics**

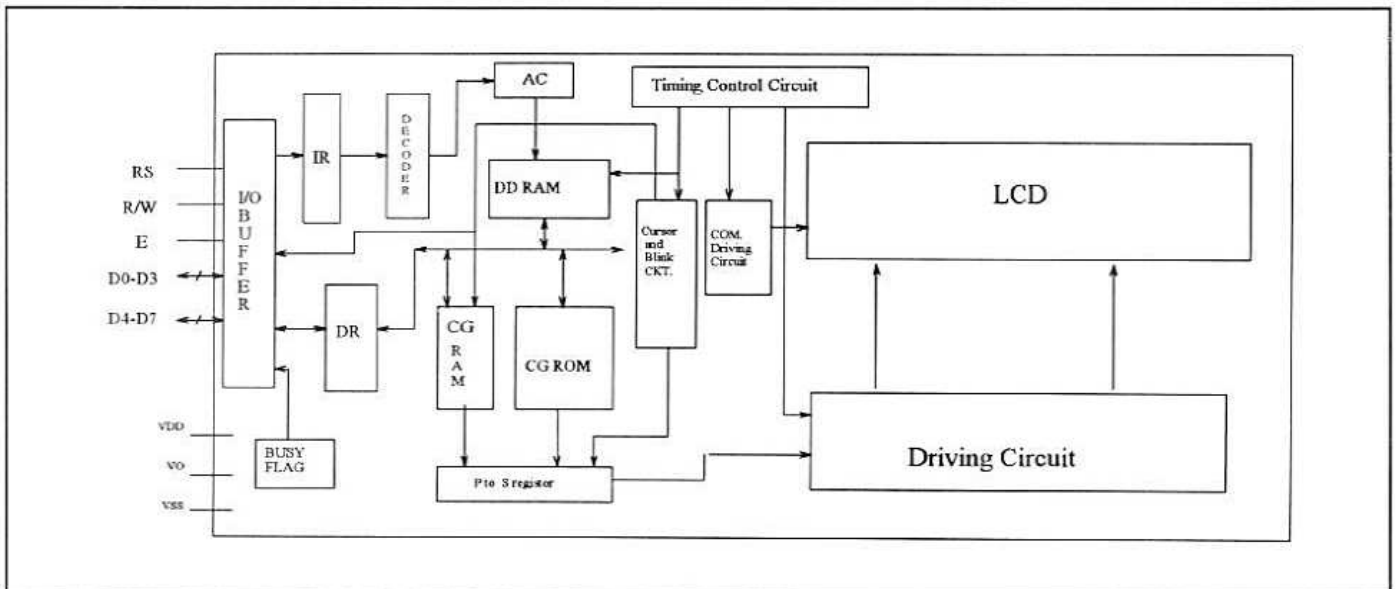
**TN TYPE**

Item	Sym.	Condition	Min.	Type	Max.	Unit
View Angle(V)	$\theta$	$C_R \geq 2.0$	10	-	30	deg.
View Angle(H)	$\phi$	$C_R \geq 2.0$	-30	-	30	deg.
Contrast Ratio	$C_R$	-	-	3	-	-
Response Time	$T_{ON}$	-	-	100	150	ms
Response Time	$T_{OFF}$	-	-	100	150	ms

**STN TYPE:**

Item	Sym.	Condition	Min.	Type	Max.	Unit
View Angle(V)	$\theta$	$C_R \geq 2.0$	10	-	40	deg.
View Angle(H)	$\phi$	$C_R \geq 2.0$	-30	-	30	deg.
Contrast Ratio	$C_R$	-	-	5	-	-
Response Time	$T_{ON}$	-	-	200	300	ms
Response Time	$T_{OFF}$	-	-	200	300	ms

**Block Diagram**

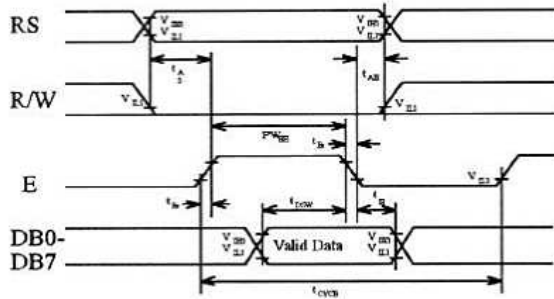


- a. Data Register (DR): DR is a register used for temporary Storage of the data read/write from/into DD RAM and CG RAM.
- b. Instruction Register (IR): IR is a register available for storing the instruction codes and address information of display data (DD) RAM and character generator (CG) RAM.
- c. BUSY FLAG (BF): When the Busy Flag is "1", it shows that LCM is in internal operation and it can not accept the next instruction.
- d. Character Generator (CG) ROM: This ROM generates character pattern from 8-bit character code and provides 192 character patterns.

- e. Character Generator (CG) RAM: This RAM allows the user to rewrite the character patterns freely according to the program.
- f. Address Counter (AC): This address counter is used to give the address information of DD RAM and CG RAM.
- g. Display Data (DD) RAM: This display data RAM is used to store the display data expressed by 8-bit character code. The capacity is 80x8 bits and data for 80 characters can be storage.
- h. Cursor and Blink Control Circuit: This circuit generates the cursor and blink.

# Timing Control

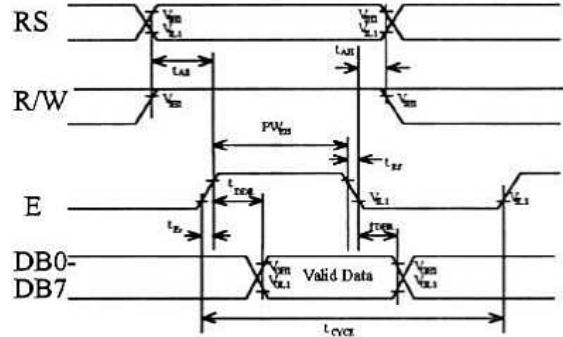
## 1. Write Operation



(Writing data from MPU to LCM)

Item	Symbol	Limit (Min.)	Limit (Max.)	Unit
Enable Cycle Time	$t_{CYCE}$	666	-	nS
Enable Pulse Width (High level)	$PW_{EH}$	300	-	nS
Enable Rise/Fall Time	$t_{ER}, t_{EF}$	-	25	nS
Address Set-Up Time (RS,R/W,E)	$t_{AS}$	100	-	nS
Address Hole Time	$t_{AH}$	10	-	nS
Data Set-Up Time	$t_{DSW}$	100	-	nS
Data Hold Time	$t_H$	10	-	nS

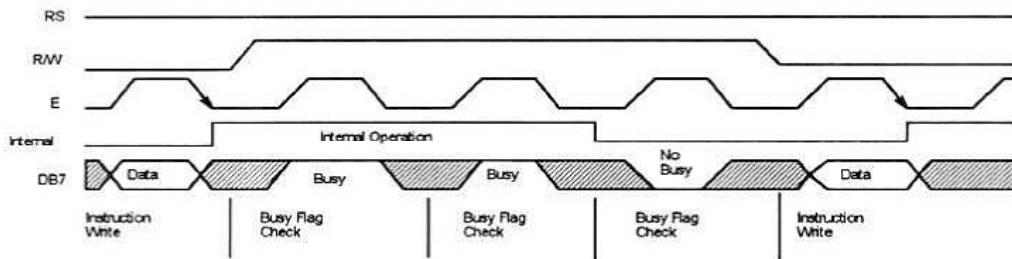
## 2. Read Operation



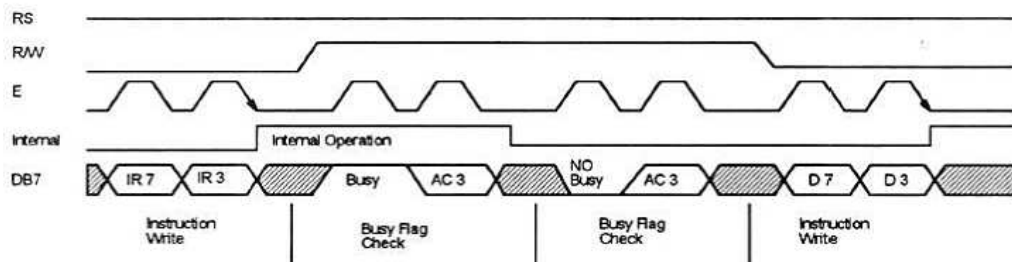
(Reading data from LCM to MPU)

Item	Symbol	Limit (Min.)	Limit (Max.)	Unit
Enable Cycle Time	$t_{CYCE}$	666	-	nS
Enable Pulse Width (High level)	$PW_{EH}$	300	-	nS
Enable Rise/Fall Time	$t_{ER}, t_{EF}$	-	25	nS
Address Set-Up Time (RS,R/W,E)	$t_{AS}$	100	-	nS
Address Hole Time	$t_{AH}$	10	-	nS
Data Delay Time	$t_{DDR}$	-	190	nS
Data Hold Time	$t_{DHR}$	20	-	nS

## 3. 8-bit busy flag check timing



## 4. 4-bit busy check timing



(Note) IR 7, IR 3: Instruction 7th bit, 3rd bit; AC3: Address Counter 3 rd bit

# Character Generator ROM Map (OB)

Upper 4bit Lower 4bit	Lower 4bit															
	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)	±		0	1	2	3	4	5	6	7	8	9	A	B	C
LLLH	(2)	≡	!	1	A	Q	a	9	0	æ	i	"	J	+	Y	0
LLHL	(3)	7	"	2	B	R	b	r	e	æ	ø	°	ø	3	8	Y
LLHH	(4)	△	#	3	C	S	c	s	æ	ø	ü	'	P	7	E	ψ
LHLL	(5)	Y	*	4	D	T	d	t	æ	ø	¢	'	4	7	Z	0
LHLH	(6)	l	%	5	E	U	e	u	æ	ø	£	¢	†	△	n	¶
LHHL	(7)	Y	&	6	F	V	f	v	æ	ø	¥	¢	↓	8	8	¶
LHHH	(8)	J	'	7	G	W	g	w	æ	ø	¥	×	*	△	l	¶
HLLL	(1)	Y	0	8	H	X	h	x	æ	ø	*	÷	*	E	K	¶
HLLH	(2)	Y	0	9	I	Y	i	y	æ	ø	l	£	∏	∏	∏	∏
HLHL	(3)	※	※	∏	J	Z	j	z	æ	ø	∏	∏	∏	∏	∏	∏
HLHH	(4)	∏	+	∏	K	C	k	c	∏	∏	∏	∏	∏	∏	∏	∏
HHLL	(5)	∏	,	∏	L	X	l	x	∏	∏	∏	∏	∏	∏	∏	∏
HHLH	(6)	∏	-	∏	M	D	m	d	∏	∏	∏	∏	∏	∏	∏	∏
HHHL	(7)	∏	.	∏	N	O	n	o	∏	∏	∏	∏	∏	∏	∏	∏
HHHH	(8)	∏	/	∏	O		o		∏	∏	∏	∏	∏	∏	∏	∏

### User Font Patterns (CG RAM Character)

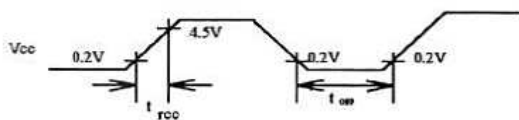
Character Code (DD RAM Data)	CG RAM Address	Character Pattern (CG RAM Data)																																																												
7 6 5 4 3 2 1 0 ← Hi LO →	5 4 3 2 1 0	7 6 5 4 3 2 1 0 ← Hi LO →																																																												
0000x000	000	<table border="1"> <tr><td>xxx</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>↑</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>↑</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>↑</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>↑</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>↑</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>↑</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>↑</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>xxx</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	xxx	1	1	1	1	0	↑	1	0	0	0	1	↑	1	0	0	0	1	↑	1	0	0	0	1	↑	1	1	1	1	0	↑	1	0	1	0	0	↑	1	0	0	1	0	↑	1	0	0	0	1	xxx	1	0	0	0	1		0	0	0	0	0
xxx	1	1	1	1	0																																																									
↑	1	0	0	0	1																																																									
↑	1	0	0	0	1																																																									
↑	1	0	0	0	1																																																									
↑	1	1	1	1	0																																																									
↑	1	0	1	0	0																																																									
↑	1	0	0	1	0																																																									
↑	1	0	0	0	1																																																									
xxx	1	0	0	0	1																																																									
	0	0	0	0	0																																																									
0000x001	001	<table border="1"> <tr><td>xxx</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>↑</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>↑</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>↑</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>↑</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>↑</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>↑</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>↑</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>xxx</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	xxx	1	0	0	0	1	↑	1	0	1	0	0	↑	1	1	1	1	1	↑	0	0	1	0	0	↑	1	1	1	1	1	↑	0	0	1	0	0	↑	0	0	1	0	0	↑	0	0	1	0	0	xxx	0	0	1	0	0		0	0	0	0	0
xxx	1	0	0	0	1																																																									
↑	1	0	1	0	0																																																									
↑	1	1	1	1	1																																																									
↑	0	0	1	0	0																																																									
↑	1	1	1	1	1																																																									
↑	0	0	1	0	0																																																									
↑	0	0	1	0	0																																																									
↑	0	0	1	0	0																																																									
xxx	0	0	1	0	0																																																									
	0	0	0	0	0																																																									
	000 001	xxx ↑																																																												
0000x111	111	↓ xxx																																																												

← Cursor Position

### Initialization of LCM

The LCM automatically initializes (reset) when power is turned on using the internal reset circuit. If the power supply conditions for correctly operating of the internal reset circuit are not met, initialization by instruction is required. Use the procedure in next page for initialization.

Valid Power Supply Condition

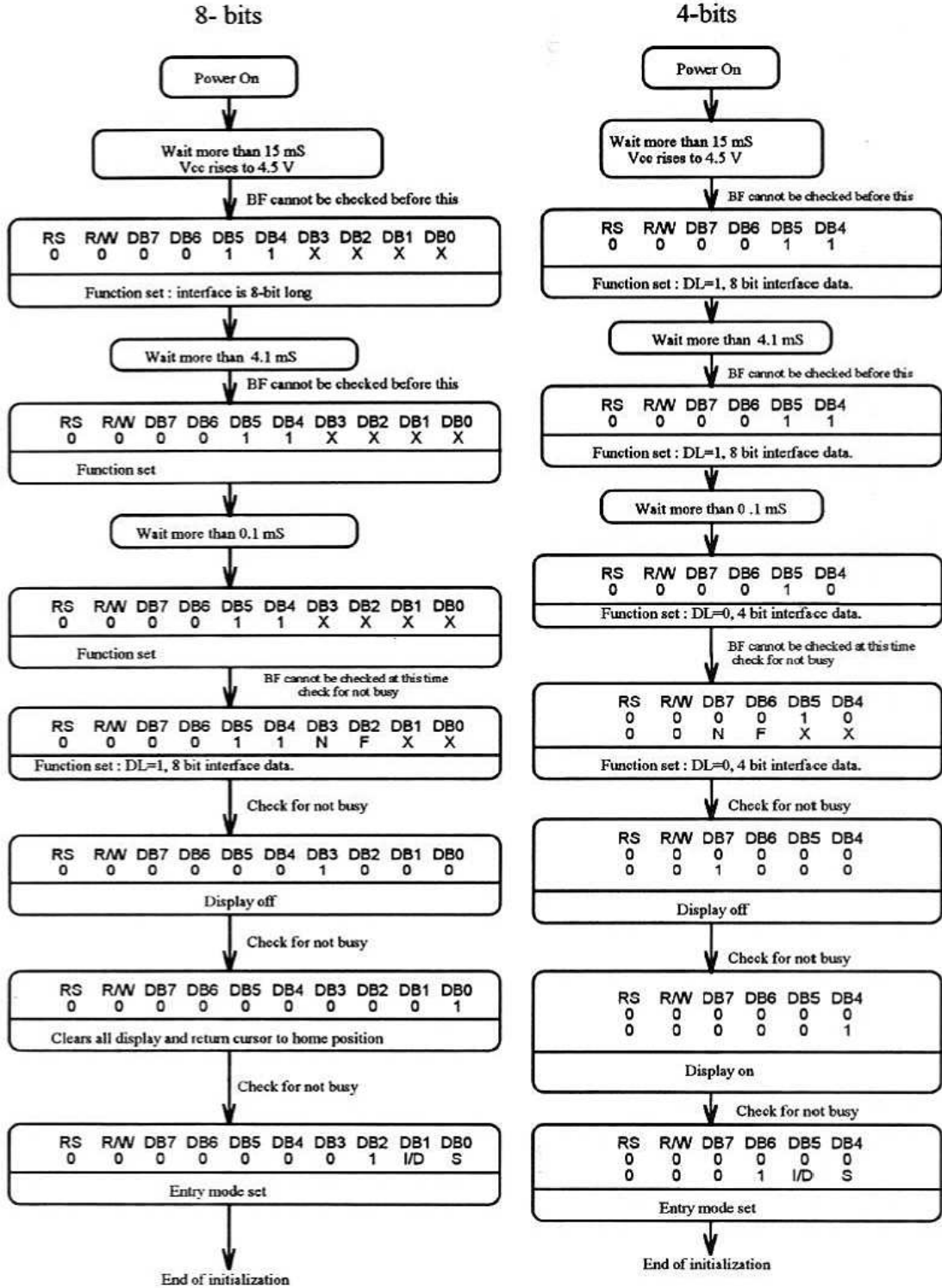


(Not1)  $0.1 \text{ mS} \geq t_{rcc} \geq 10 \text{ mS}$ ,  $t_{off} \geq 1 \text{ mS}$

(Note 2)  $t_{off}$  stipulates the time of power OFF for momentary power supply dip or when power supply cycles ON and OFF.

Item	Sym	Test condition	limit (Min.)	limit (Max.)	Unit
Power supply rise time	$t_{rcc}$	-	0.1	10	ms
Power supply OFF time	$t_{OFF}$	-	1	-	ms

# Initialization by Instruction



### Instruction Set

Instruction operation											Description	Execu. Time* (Max.)
Function	R S	R / W	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears entire display and returns the cursor to home position (address 0).	1.64mS
Return Home	0	0	0	0	0	0	0	0	0	1	Return the cursor to the home position. Also returns the display being shifts to the original position. DD RAM contents remain unchanged. Set DD RAM address to zero.	1.64mS
Entry mode set	0	0	0	0	0	0	0	0	1	I / D	Set cursor move direct and specifies or not to shift the display. These operations are performed during data write/read of DD RAM/CG RAM. For normal operation, sets S to zero. I/D=1; increment; I/D=0; decrement S=1; accompanies display shift when data is written. for normal operation, set to zero.	40μS
Display ON/OFF control	0	0	0	0	0	0	0	1	D	C B	Set ON/OFF all display (D), cursor ON/OFF (C), and blink of cursor position character (B). D=1: ON display, D=0: OFF display. C=1: ON Cursor, C=0: OFF cursor, B=1: ON Blink Cursor, B=0, OFF Blink Cursor	40μS
Cursor or display shift	0	0	0	0	0	0	1	S / C	R / L	x x	Move the cursor and shift the display without changing DD RAM contents. S/C=1: Display Shift, S/C=0: Cursor move, R/L=1: shift to right, R/L=0: shift to left.	40μS
Function Set	0	0	0	0	1	D L	N	F	x	x	Set the interface data length (DL), number of display lines (N) and character font (F). DL=1: 8 bits, DL=0: 4 bits N=1: 2 lines, N=0:1 lines, F=1: 5x10 dots, F=0: 5x7 dots	40μS
Set CG RAM Addr.	0	0	0	1	ACG					Set CG RAM address. CG RAM data is sent and received after this setting.	40μS	
Set DD RAM Addr.	0	0	1	ADD					Set DD RAM address. DD RAM data is sent and received after this setting.	40μS		
Read busy flag & Addr	0	0	B F	AC					Reads BUSY FLAG (BF) indicating internal operation is beginning performed and reads address counter contents. BF=1: internally operating. BF=0: can accept instruction.	0μS		
Write Data to CG-RAM	1	0	WRITE DATA					Write data into DD RAM or CG RAM.			40μS**	
Read Data from CG/DD RAM	1	1	READ DATA					Read data from DD RAM or CG RAM			40μS**	

\* 1. When  $f_{cp}$  or  $f_{osc}$  is 250KHz.

2. Execution time changes when frequency changes. When  $f_{cp}$  or  $f_{osc}$  is 270KHz:  $40\mu S \times \frac{250}{270} = 37\mu S$

\*\* $t_{ADD} = 6\mu S$

### Software Examples

#### 8-bit operation (8-bits 2lines)

Function	RS RW D7 D6 D5 D4 D3 D2 D1 D0	Display	Description
power on delay			Initialization. No display appears.
Function set	0 0 0 0 1 1 0 0 x x		Sets to 8-bit operation and selects 2-line display and 5x7 dots character font. (Note: number of display lines and character fonts cannot be change after this.)
Display OFF	0 0 0 0 0 0 0 1 0 0 0		Turn off display.
Display ON	0 0 0 0 0 0 0 1 1 1 0		Turn on display and cursor.
Entry Mode Set	0 0 0 0 0 0 0 0 1 1 0		Set mode to increment the address by one and to shift the cursor to the right, at the time of write, to the DD/CG RAM Display is not shifted.



Write data to CG/DD RAM	1 0 0 1 0 1 0 1 1 1	W	Write "W". Cursor incremented by one and shift to right
Write data to CG/DD RAM	1 0 0 1 0 0 1 0 0 1	W I	Write "I". Cursor incremented by one and shift to right
Write data to CG/DD RAM	.	W I N T E K	Write "N", "I", "E", and "K".
Set DD address.	0 0 1 1 0 0 0 0 0 0	W I N T E K -	Set RAM address so that the cursor is propositioned at the head of the second line.
Write data to CG/DD RAM	.	W I N T E K C R	Write "C", and "R".
Cursor or display shift	0 0 0 0 0 1 0 0 x x	W I N T E K C R	Shift only the cursor position to the left.
Write data to CG/DD RAM	.	W I N T E K C O R P O R A T	Write "O", "R", "P", "O", "R", "A", and "T".
Entry Mode Set	0 0 0 0 0 0 0 1 1 1	W I N T E K C O R P O R A T	Set display mode shift at the time during writing operation.
Write data to CG/DD RAM	1 0 0 1 0 0 1 0 0 1	I N T E K O R P O R A T I	Write "I". Cursor incremented by one and shift to right. (The display move to left)
Write data to CG/DD RAM	.		Write other characters.
Return Home	0 0 0 0 0 0 0 0 1 0	W I N T E K C O R P O R A T I O	Return both display and cursor to the original position (Set address to zero.

#### 4-bit operation (4-bits 1 line)

Function	RS RW D7 D6 D5	Display	Description
power on delay			Initialization. No display appears.
Function set	0 0 0 0 1 0		Sets to 4-bit operation. In this case, operation is handled as 8-bits by initialization, and only this instruction completes with one write.
Function set	0 0 0 0 1 0 0 0 0 0 x x		Sets 4-bit operation and selects 1-line display and 5x7 dot character font on and resetting is needed. (number of display lines and character fonts cannot be changed hence after.)
Display ON/OFF Control	0 0 0 0 0 0 0 0 1 1 1 0		Turn on display and cursor.
Entry Mode Set	0 0 0 0 0 0 0 0 0 1 1 0		Set mode to increment the address by one and to shift the cursor to the right, at the time of write, to the DD/CG RAM display is not shifted.
Write data to CG/DD RAM	1 0 0 1 0 1 1 0 0 1 1 1	W	Write "W". Cursor incremented by one and shift to right
Same as 8-bit operation			