Input and Output Devices

Minsoo Ryu

Department of Computer Science and Engineering
Hanyang University

msryu@hanyang.ac.kr

Topics Covered

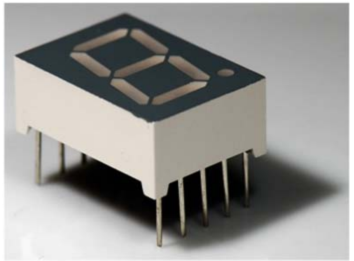
1. Discrete LEDs
2. Seven-Segment LEDs
3. Character LCD
4. Keypads
5. Touch Screens

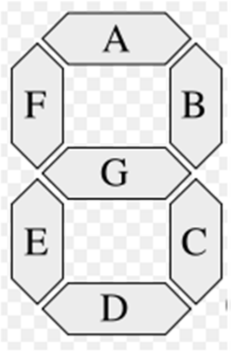
LED (Light Emitting Diode)

❒ A light-emitting diode, usually called an LED is a semiconductor diode that emits incoherent narrow-spectrum light when electrically biased in the forward direction of the p-n junction

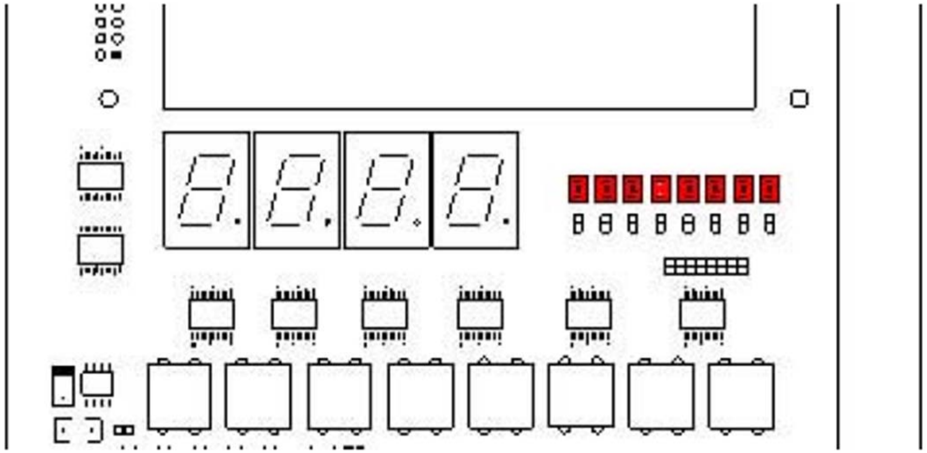
❒ FND (Flexible Numeric Display)

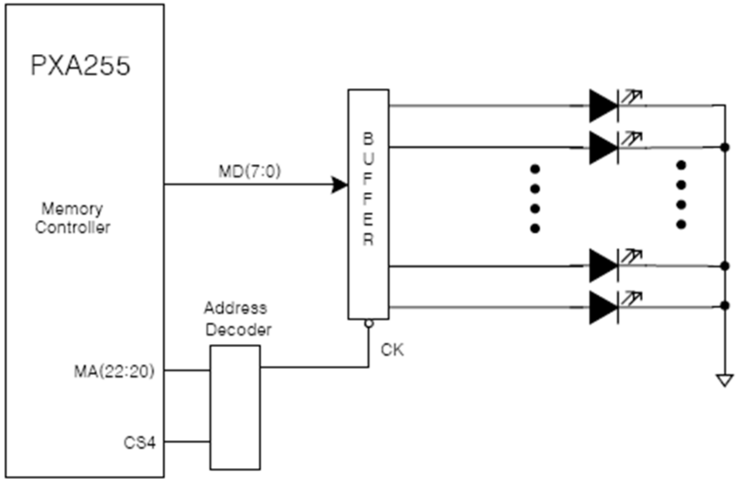
- Seven-segment LED



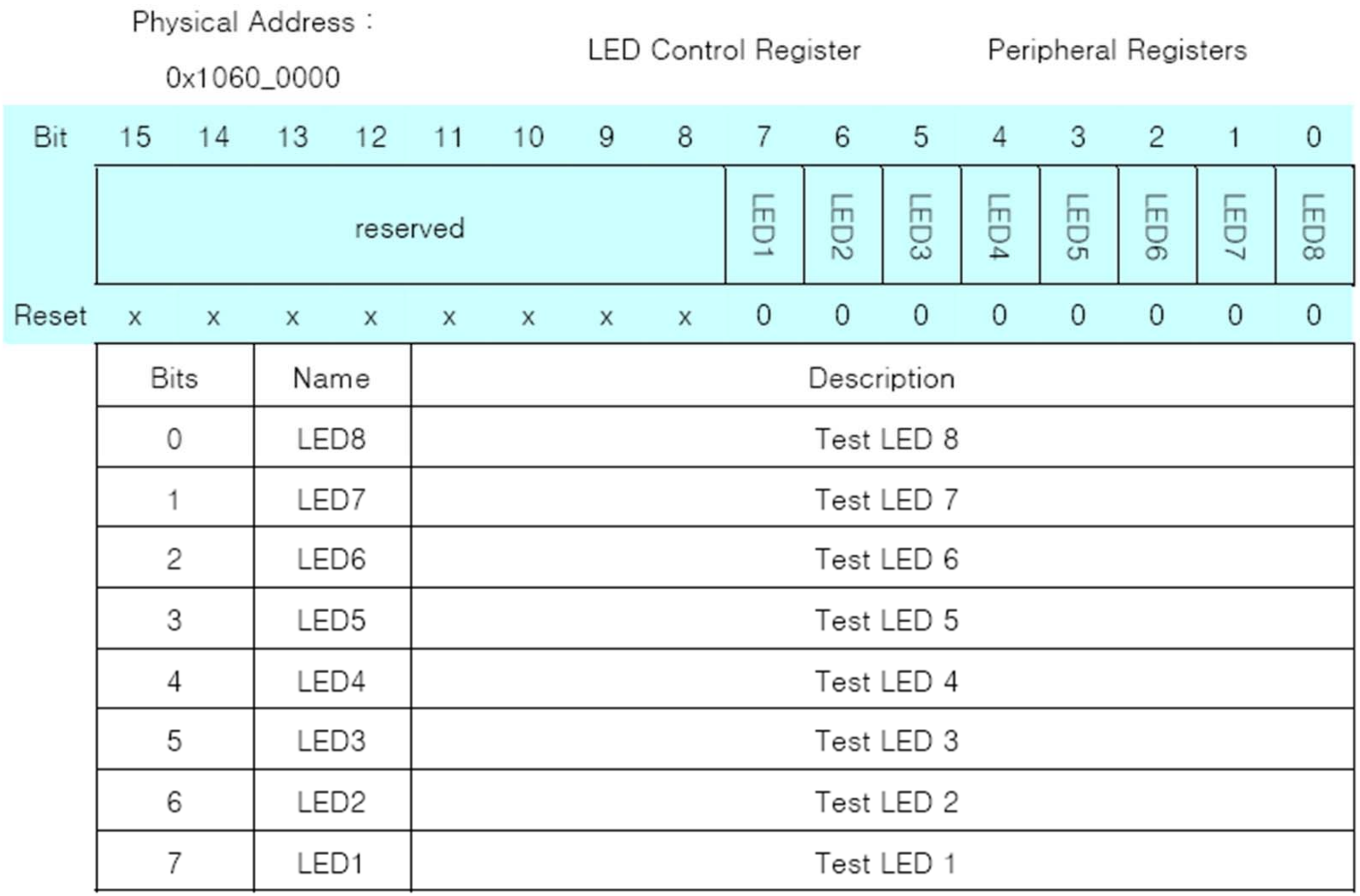


Eight Discrete LEDs





Discrete LED Register



Physical Address :
0x1060_0000

LED Control Register

Peripheral Registers

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED8
Reset	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bits	Name	Description
0	LED8	Test LED 8
1	LED7	Test LED 7
2	LED6	Test LED 6
3	LED5	Test LED 5
4	LED4	Test LED 4
5	LED3	Test LED 3
6	LED2	Test LED 2
7	LED1	Test LED 1

Programming Discrete LEDs

❒ Blink LEDs for 1 second

```
#define   LED_BASE          0x10600000

void blink_led ()
{
            unsigned* led_addr = LED_BASE;
            unsigned led_val = 1;
            int blink_count = 8;
            int blink_delay = 1000000;
            int I;

            for (i=0; i<blink_count; i++) {
                        *led_addr = led_val;
                        led_val << 1;
                        usleep(blink_delay);
            }
}
```

When an MMU-enabled operating system like Linux is used, an appropriate virtual address must be obtained first.

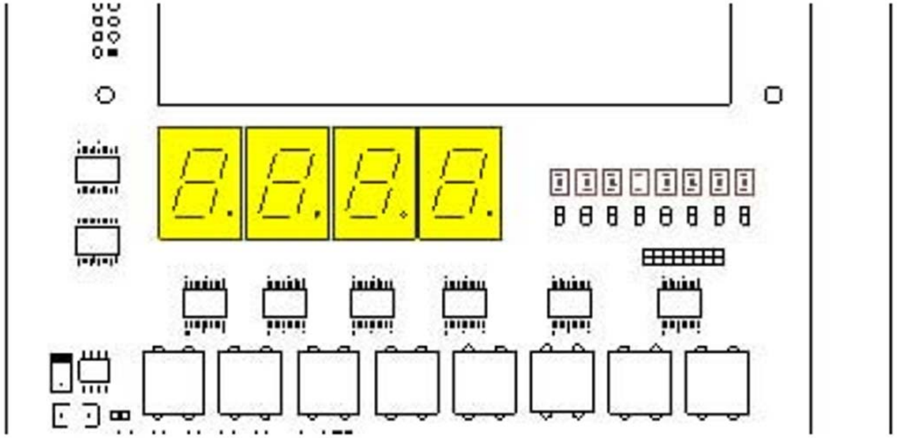
The code should then be changed to aim at the virtual address instead of the physical address.

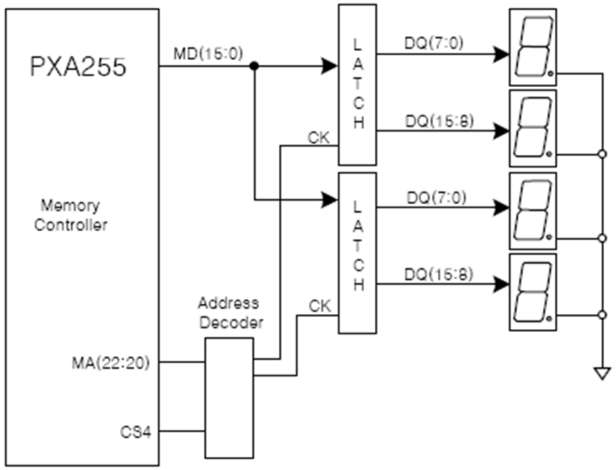
In Linux, the mmap function can return a virtual address from a given physical address.

Topics Covered

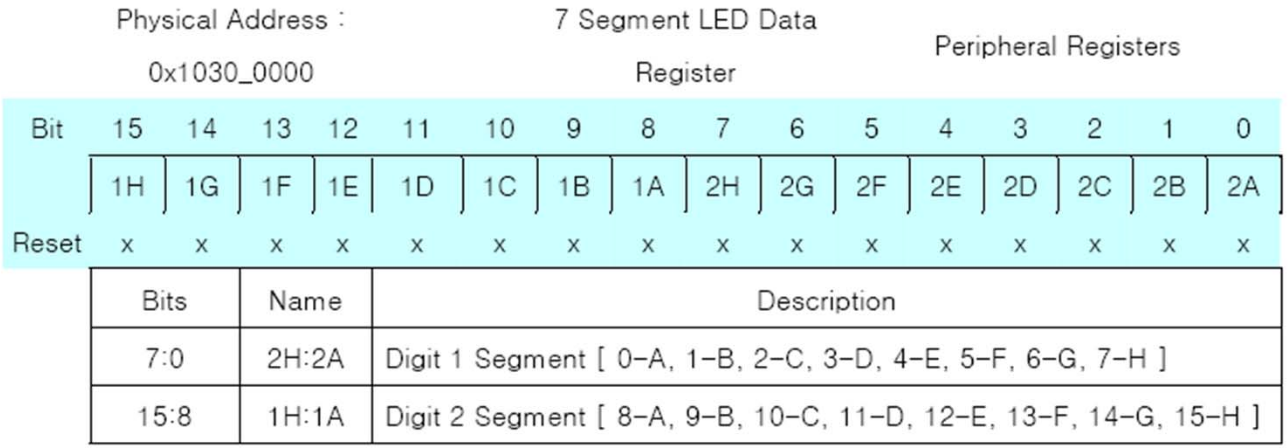
1. Discrete LEDs
2. Seven-Segment LEDs
3. Character LCD
4. Keypads
5. Touch Screens

Four Seven-Segment LEDs





Seven-Segment LED Registers



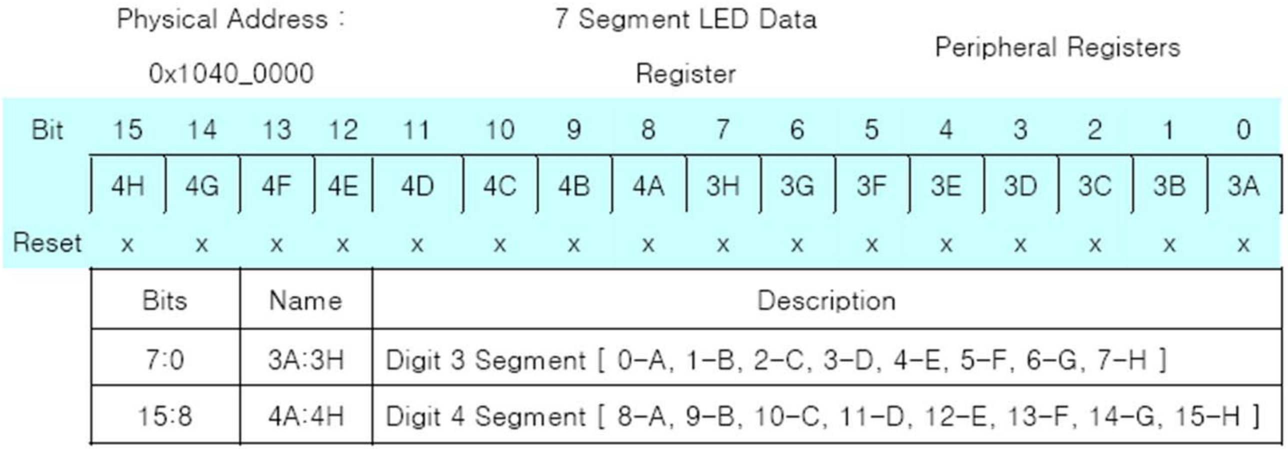
Physical Address : 0x1030_0000

7 Segment LED Data Register

Peripheral Registers

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1H	1G	1F	1E	1D	1C	1B	1A	2H	2G	2F	2E	2D	2C	2B	2A
Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bits	Name	Description
7:0	2H:2A	Digit 1 Segment [0–A, 1–B, 2–C, 3–D, 4–E, 5–F, 6–G, 7–H]
15:8	1H:1A	Digit 2 Segment [8–A, 9–B, 10–C, 11–D, 12–E, 13–F, 14–G, 15–H]



Physical Address : 0x1040_0000

7 Segment LED Data Register

Peripheral Registers

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	4H	4G	4F	4E	4D	4C	4B	4A	3H	3G	3F	3E	3D	3C	3B	3A
Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bits	Name	Description
7:0	3A:3H	Digit 3 Segment [0–A, 1–B, 2–C, 3–D, 4–E, 5–F, 6–G, 7–H]
15:8	4A:4H	Digit 4 Segment [8–A, 9–B, 10–C, 11–D, 12–E, 13–F, 14–G, 15–H]

Programming Seven-Segment LEDs

❒ Display 1, 2, 3, and 4

```
#define   SS_LED_LOW       0x10300000
#define   SS_LED_HIGH      0x10400000

void display_1234 ()
{
          unsigned* led_addr_low = SS_LED_LOW;
          unsigned* led_addr_high = SS_LED_HIGH;
          char val[4];
          unsigned int low_val, high_val;

          val[0] = 0x06; val[1] = 0x5b;          /* 0x06 (1), 0x5b (2) */
          val[2] = 0x4f; val[3] = 0x66;          /* 0x4f (3), 0x66 (4) */
          high_val = (val[0]<<8) | val[1];       /* 0x065b */
          low_val = (val[2]<<8) | val[3];        /* 0x4f66 */
          *led_addr_high = high_val;
          *led_addr_low = low_val;
}
```

Topics Covered

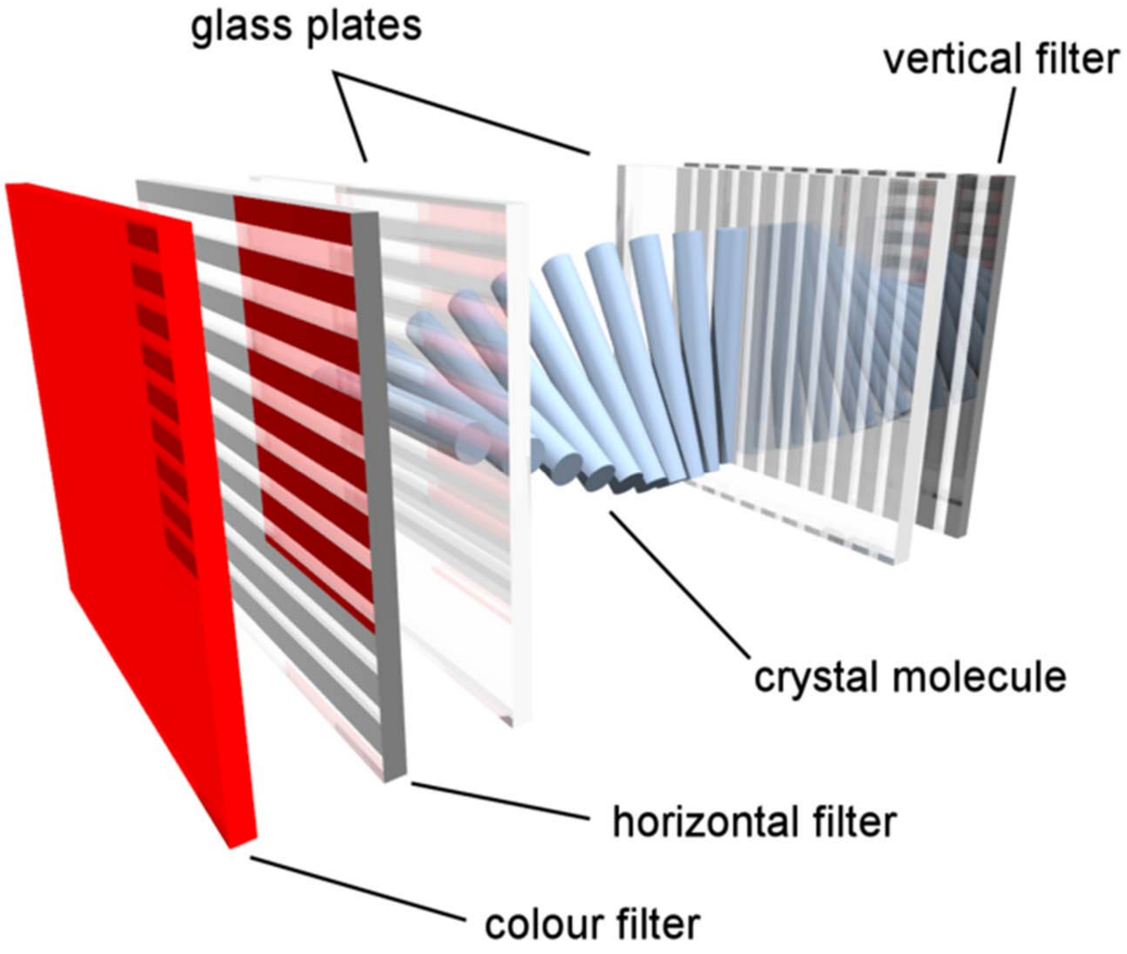
1. Discrete LEDs
2. Seven-Segment LEDs
3. Character LCD
4. Keypads
5. Touch Screens

LCD (Liquid Crystal Display)

❒ A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector

- It is often utilized in battery-powered electronic devices because it uses very small amounts of electric power
- Each pixel of an LCD typically consists of a layer of molecules aligned between two transparent electrodes, and two polarizing filters

LCD (Liquid Crystal Display)



LCD (Liquid Crystal Display)

❐ **Two types**

- **Passive Matrix: STN-LCD (Super Twisted Nematic LCD)**
- **Active Matrix: TFT-LCD (Thin-Film Transistors LCD)**
  - Better response time, brightness, and sharpness

❐ **Two types from a SW perspective**

- **Character LCD**
  - Support for text mode
- **Graphics LCD**
  - Support both for text and graphics mode

Character LCD

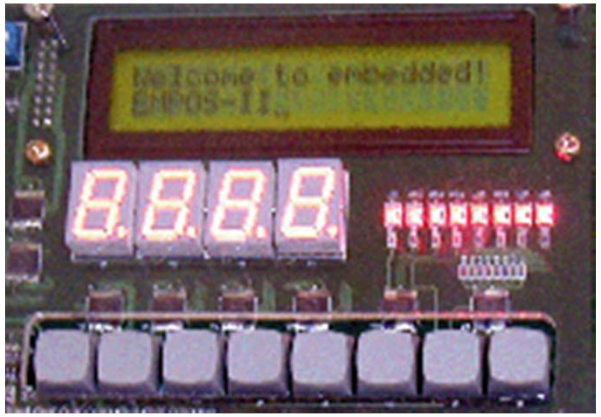
- ❒ Character LCD modules are available from a wide range of manufacturers and should all be compatible with the Hitachi HD44780
  - ▪ HD44780 Character LCD is an industry standard liquid crystal display (LCD) display device designed for interfacing with embedded systems

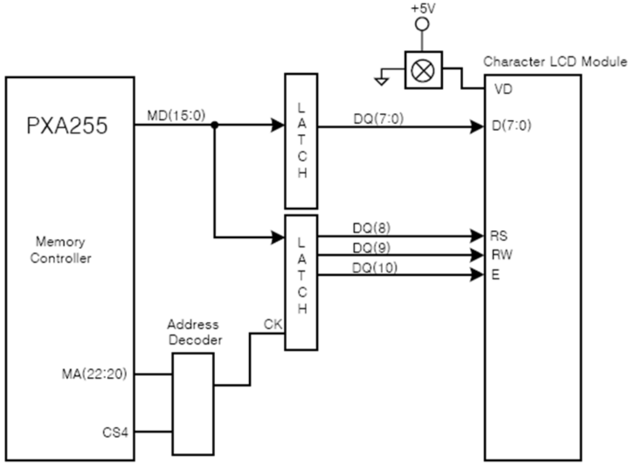
- ❒ Thus, it suffices to consider a typical example
  - ▪ Here we will investigate CM2020S1LY-K from Data Image, which is equipped in HBE MPOS-II

CM2020S1LY-K from Data Image Corporation (HBE MPOS-II)

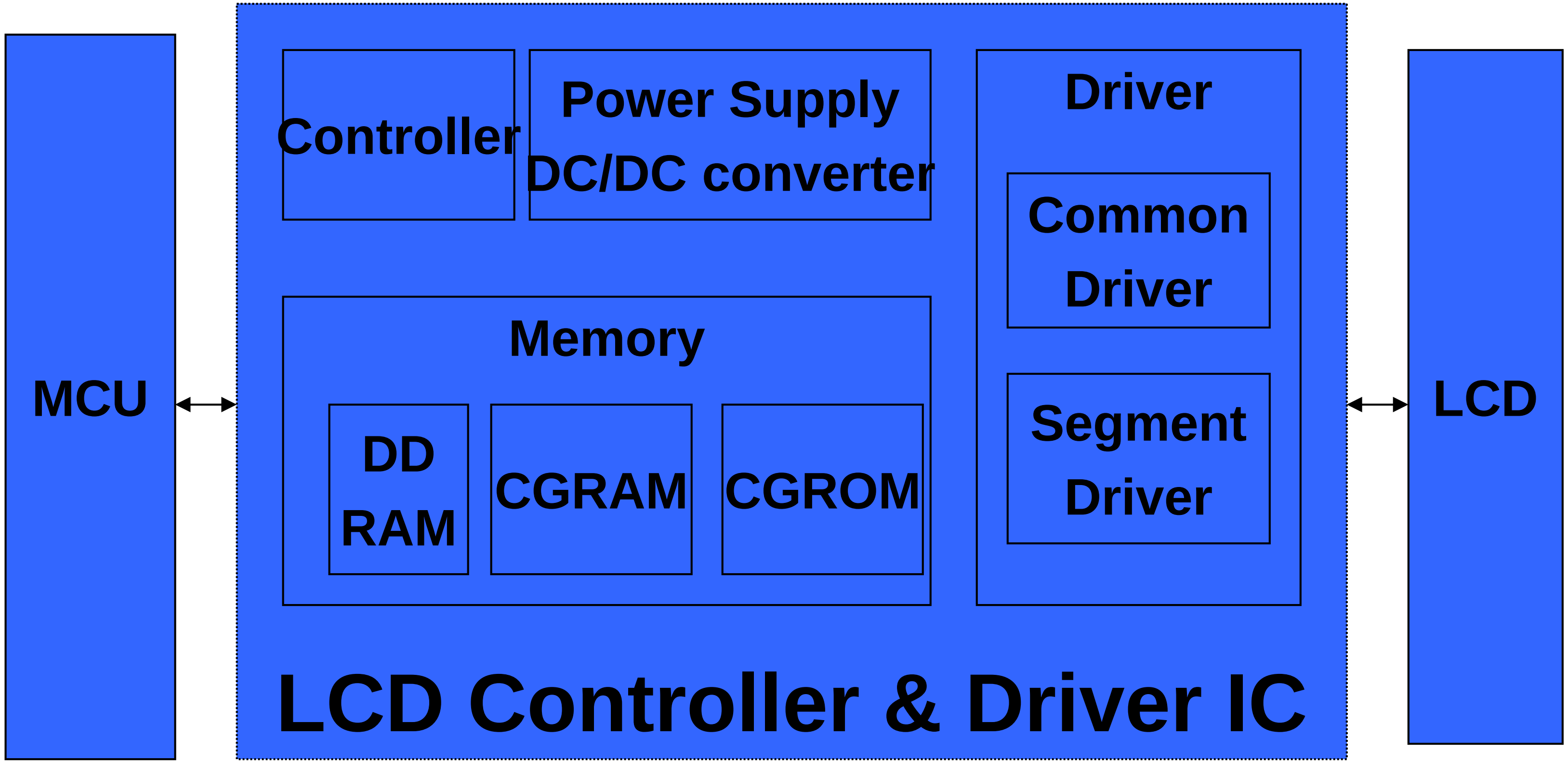
❒ CM2020S1LY-K from Data Image Corporation is equipped in HBE MPOS-II

- 20 Characters x 2 Lines / Backlight Type
- 5 x 7 dots per character or 5 x 10 dots per character
- 8 bits per character

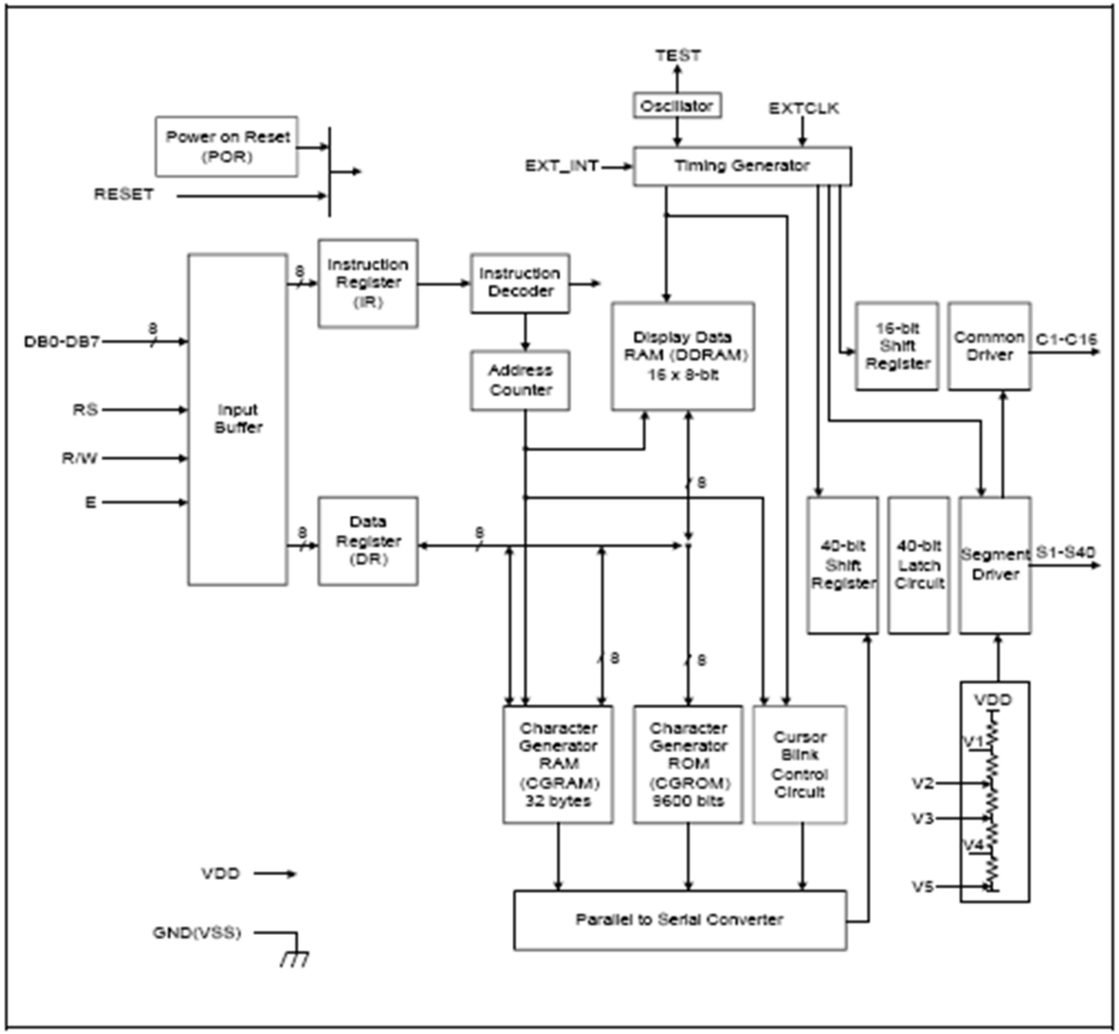




Block Diagram (CM2020S1LY-K)



Another Block Diagram (Samsung 16COM/40SEG)



Memory Components in LCD Module

❒ DDRAM (Display Data RAM)

- **Stores 8 bit character code that is displayed on LCD**

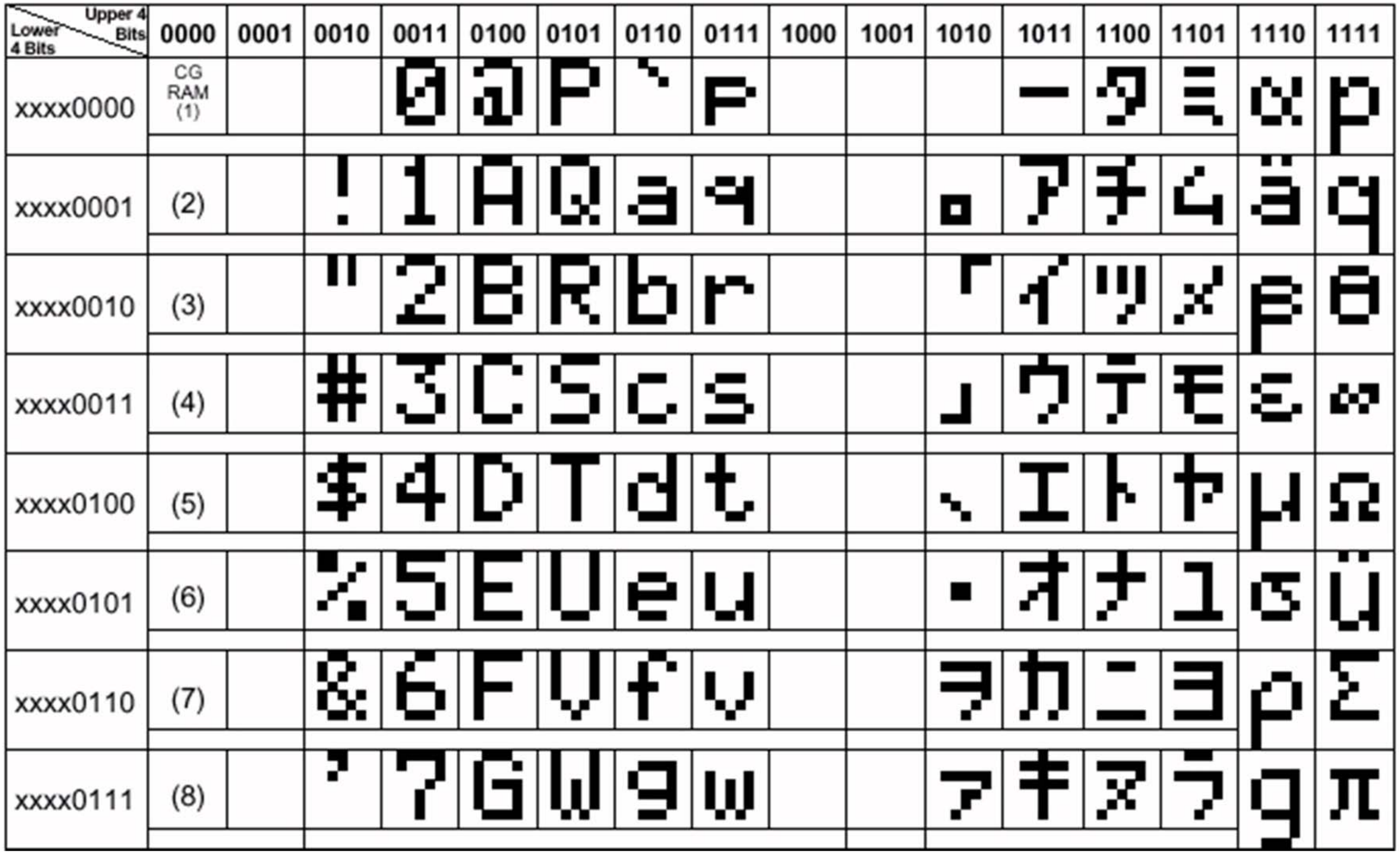
❒ CGROM (Character Generator ROM)

- **Stores character patterns (font)**
  - Less than 256 – 16 characters (the same as ASCII table)
- **Generates a character from the character code of DDRAM onto LCD**

❒ CGRAM (Character Generator RAM)

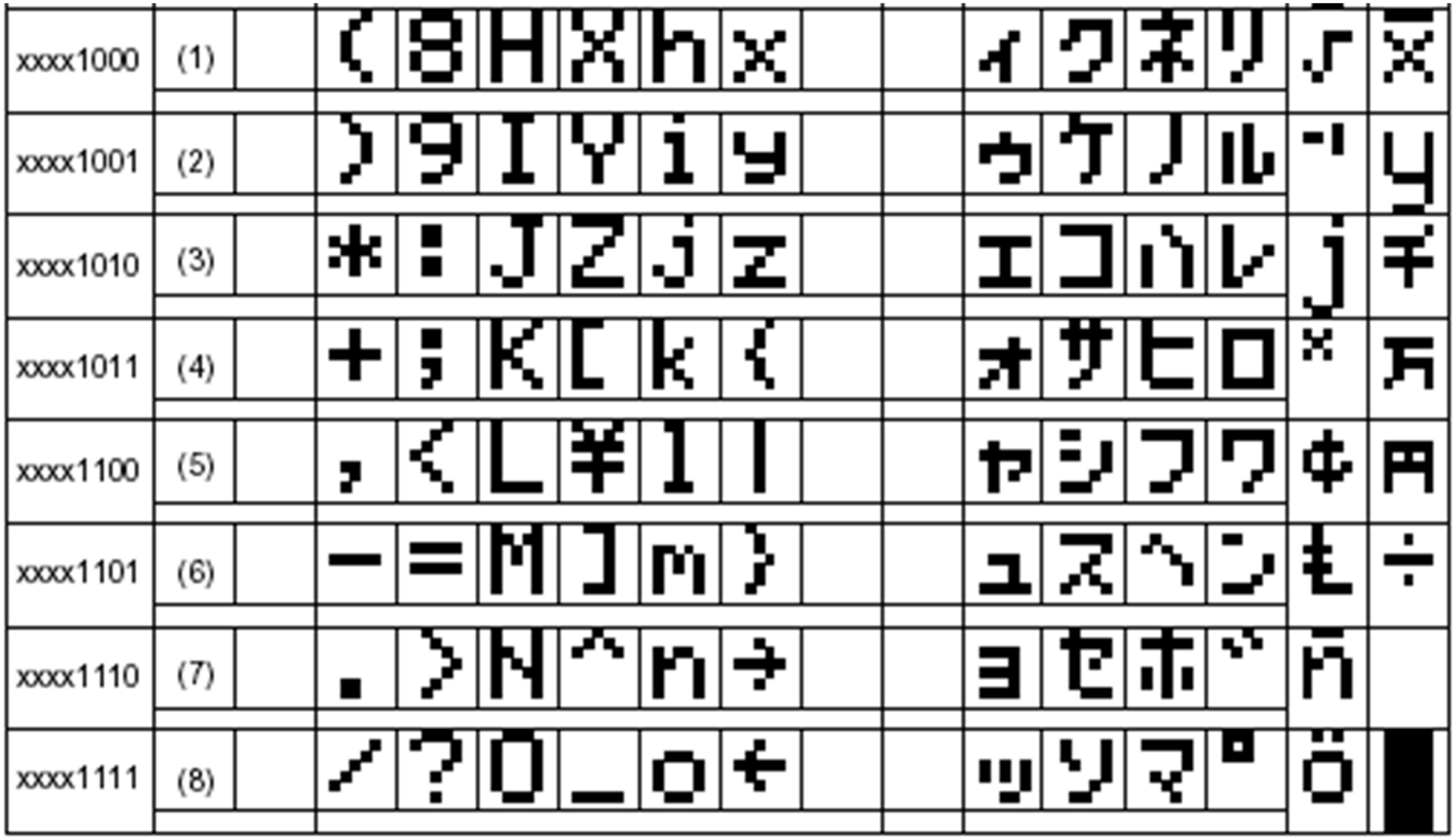
- **Stores user-defined character patterns**
  - 8 characters (5x7 dots) or 4 characters (5x10 dots)
- **Then can be used like CGROM**

Character Patterns Stored in CGROM



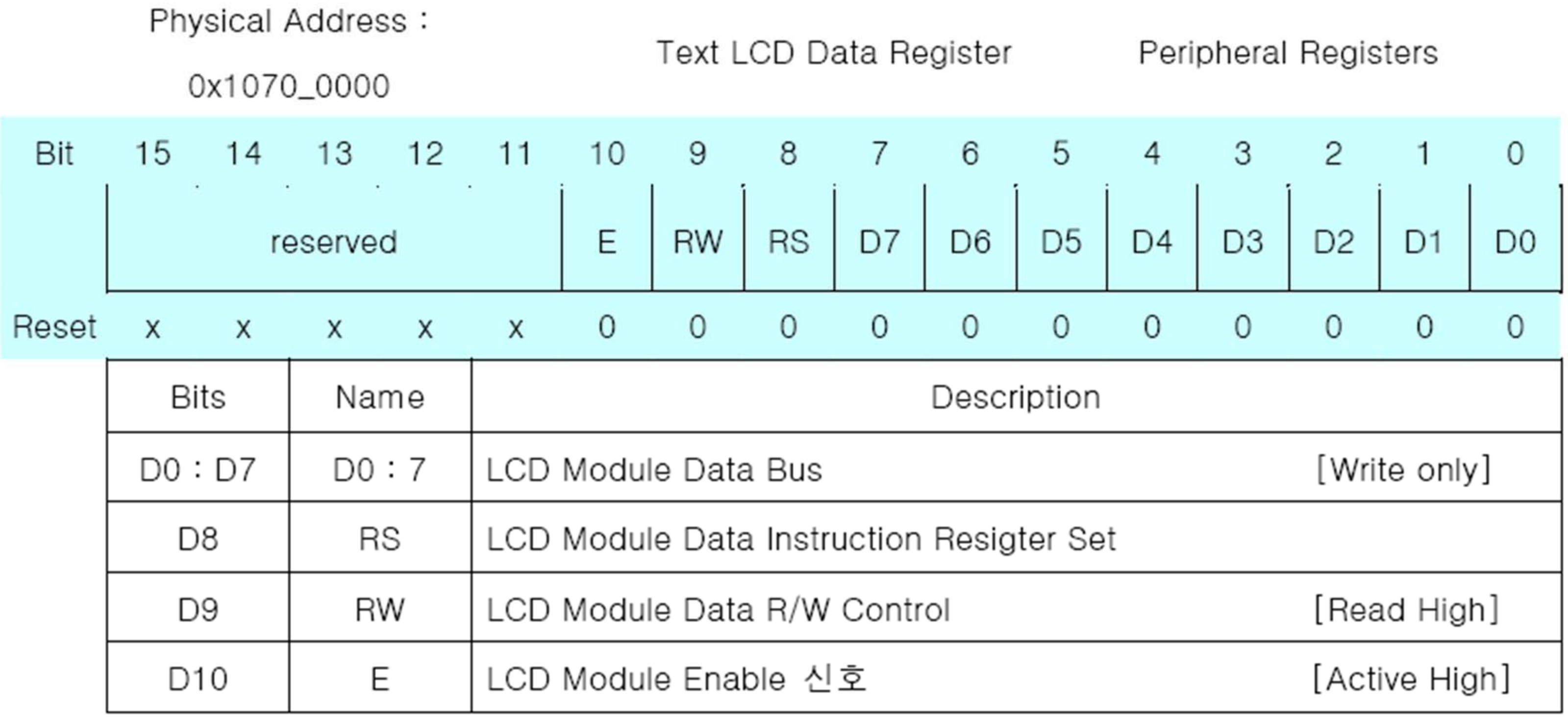
Upper 4 Bits Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	メ	β	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π

Character Patterns Stored in CGROM



xxxx1000	(1)		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x̄
xxxx1001	(2)		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010	(3)		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
xxxx1011	(4)		+	;	K	[	k	{			ォ	サ	ヒ	ロ	ˣ	万
xxxx1100	(5)		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
xxxx1101	(6)		-	=	M	]	m	}			ュ	ス	ヘ	ン	£	÷
xxxx1110	(7)		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
xxxx1111	(8)		/	?	O	_	o	←			ッ	ソ	マ	゜	ö	█

I/O Register Format



Physical Address :

0x1070_0000

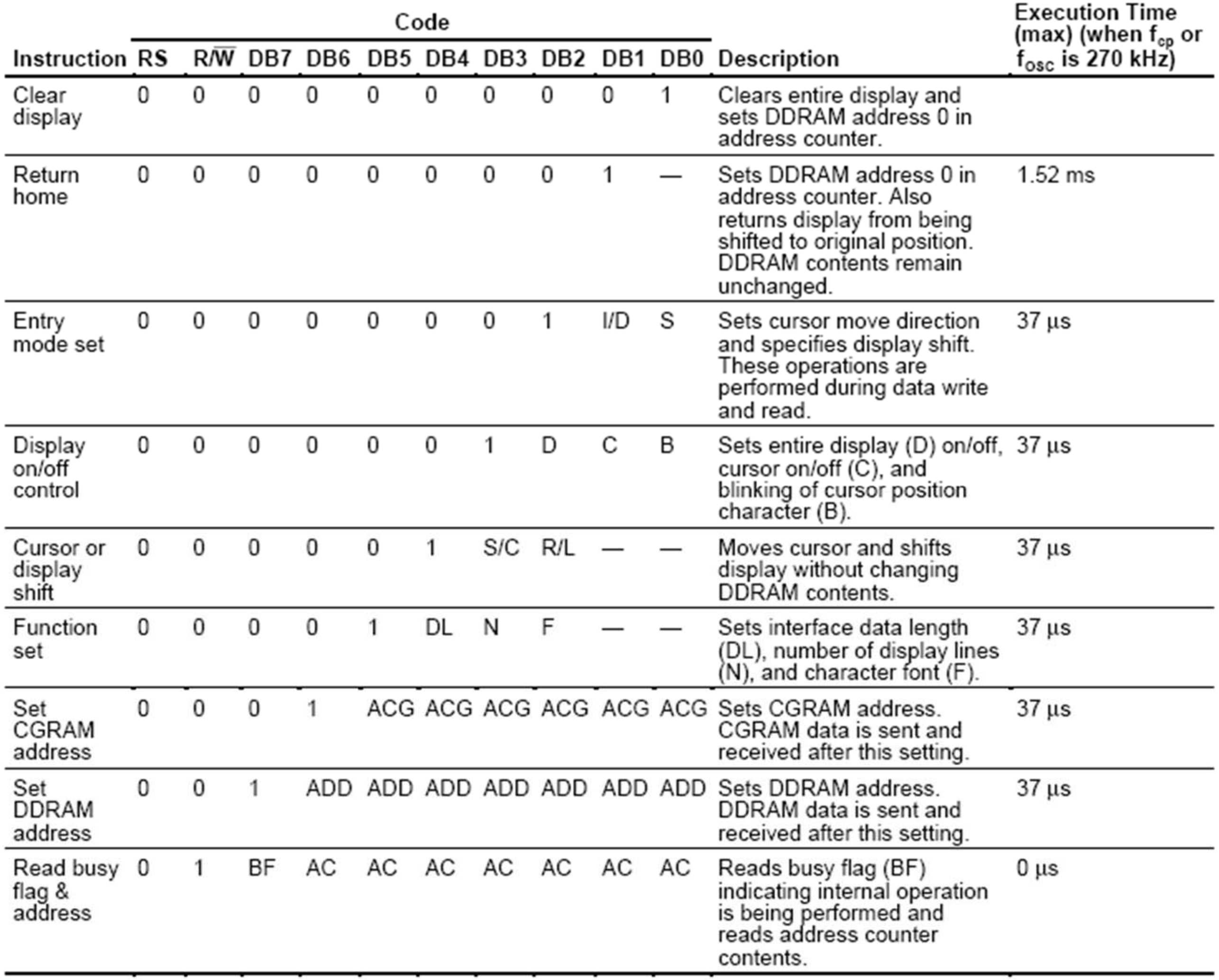
Text LCD Data Register

Peripheral Registers

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved					E	RW	RS	D7	D6	D5	D4	D3	D2	D1	D0
Reset	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

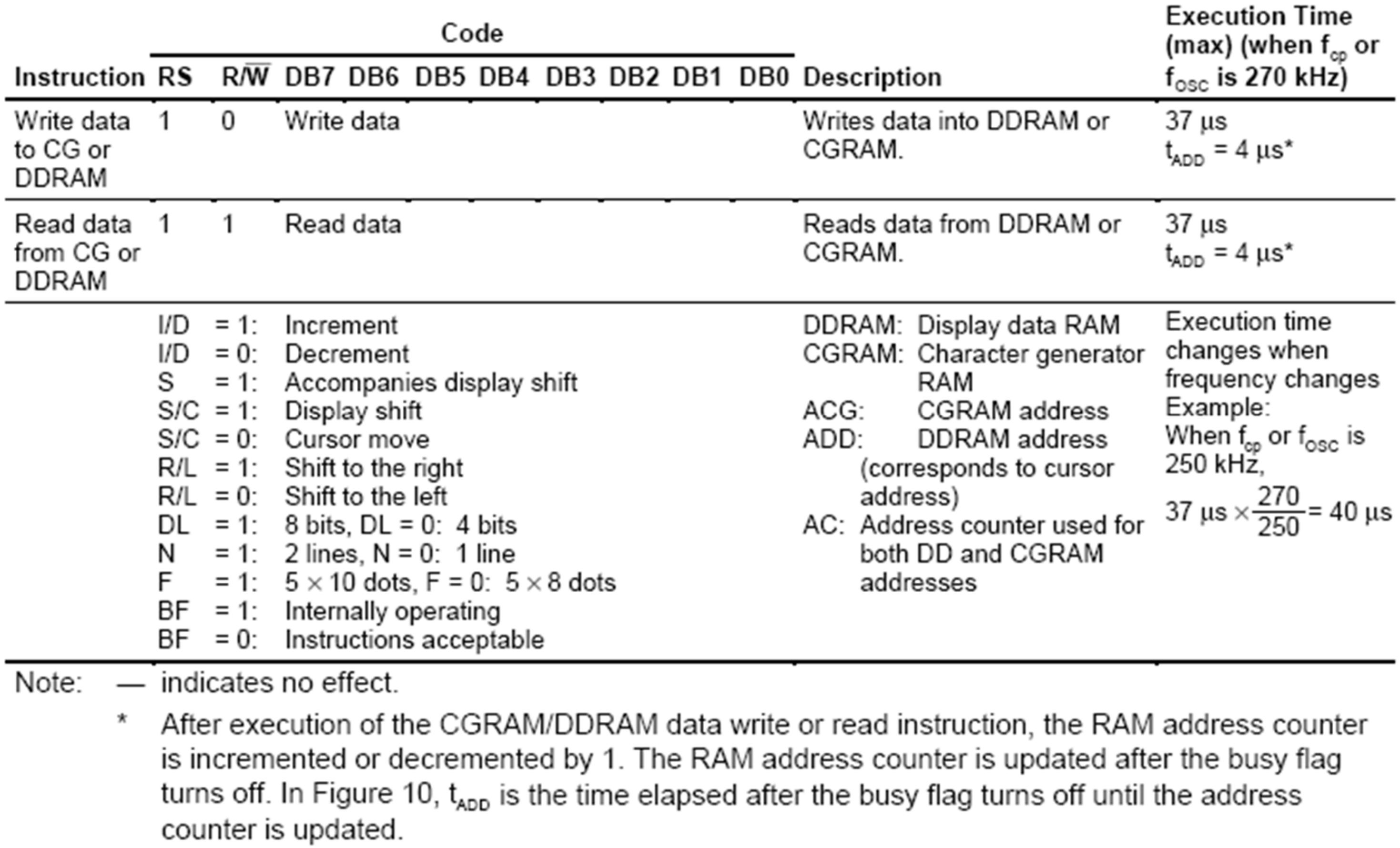
Bits	Name	Description	
D0 : D7	D0 : 7	LCD Module Data Bus	[Write only]
D8	RS	LCD Module Data Instruction Resigter Set	
D9	RW	LCD Module Data R/W Control	[Read High]
D10	E	LCD Module Enable 신호	[Active High]

HD44780 Instructions



	Code											Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
Instruction	RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 μs
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 μs
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 μs
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 μs
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 μs
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 μs
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μs

HD44780 Instructions



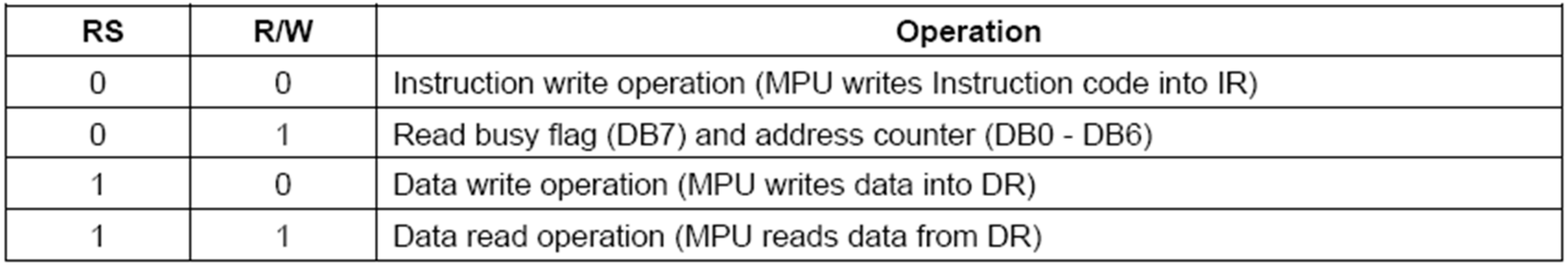
	Code										Execution Time (max) (when f_{cp} or f_{OSC} is 270 kHz)
Instruction	**RS**	**$R/\overline{W}$**	**DB7**	**DB6**	**DB5**	**DB4**	**DB3**	**DB2**	**DB1**	**DB0**	**Description**	
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	37 μs t_{ADD} = 4 μs*
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.	37 μs t_{ADD} = 4 μs*
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8 bits, DL = 0: 4 bits N = 1: 2 lines, N = 0: 1 line F = 1: 5 × 10 dots, F = 0: 5 × 8 dots BF = 1: Internally operating BF = 0: Instructions acceptable										DDRAM: Display data RAM CGRAM: Character generator RAM ACG: CGRAM address ADD: DDRAM address (corresponds to cursor address) AC: Address counter used for both DD and CGRAM addresses	Execution time changes when frequency changes Example: When f_{cp} or f_{OSC} is 250 kHz, $37\ \mu s \times \frac{270}{250} = 40\ \mu s$

Note: — indicates no effect.

* After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.

Instructions

- Four types of instructions determined by RS (register select) and R/W (read/write) bits
  - RS = 0: instruction
  - RS = 1: data
  - R/W = 0: write
  - R/W = 1: read

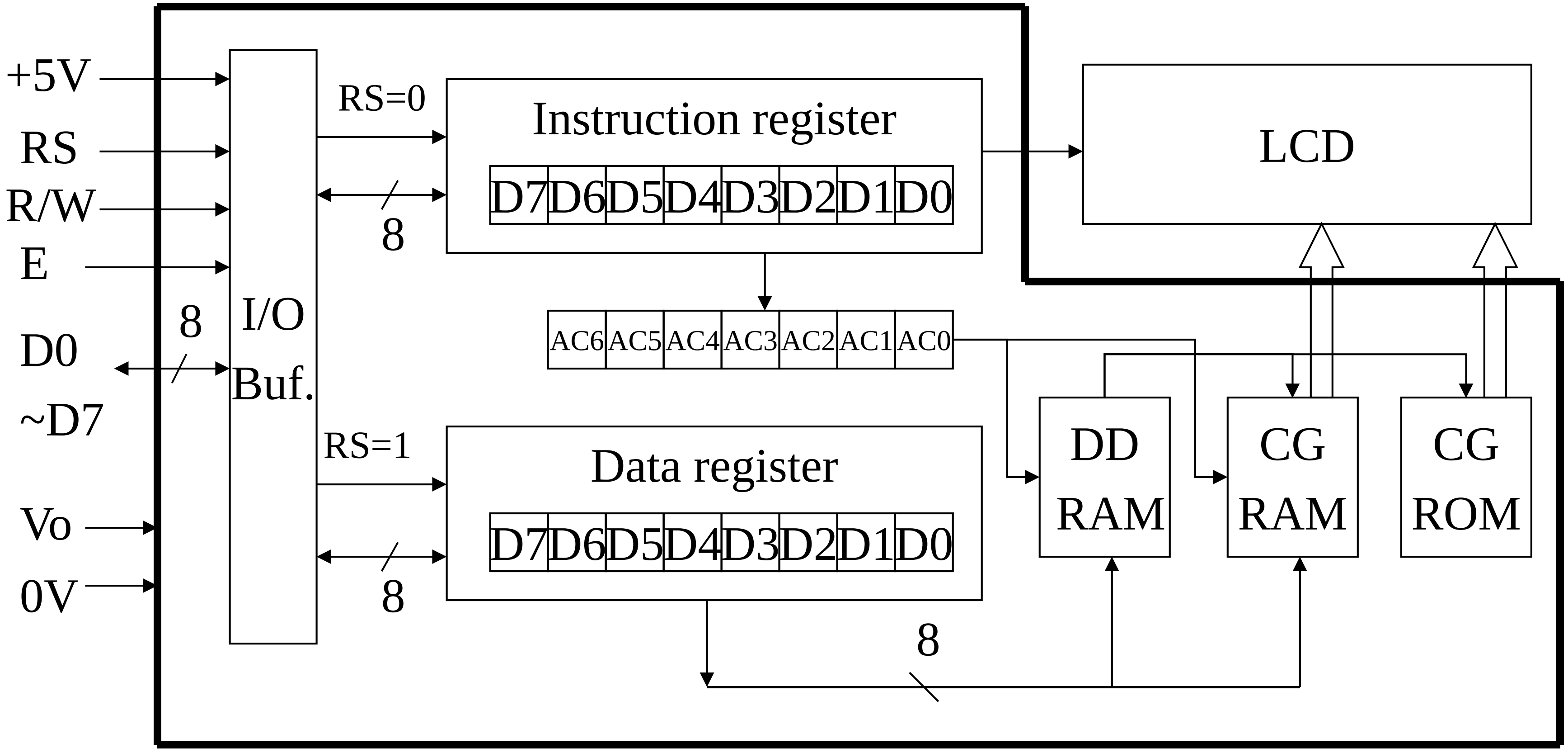


RS	R/W	Operation
0	0	Instruction write operation (MPU writes Instruction code into IR)
0	1	Read busy flag (DB7) and address counter (DB0 - DB6)
1	0	Data write operation (MPU writes data into DR)
1	1	Data read operation (MPU reads data from DR)

Descriptions

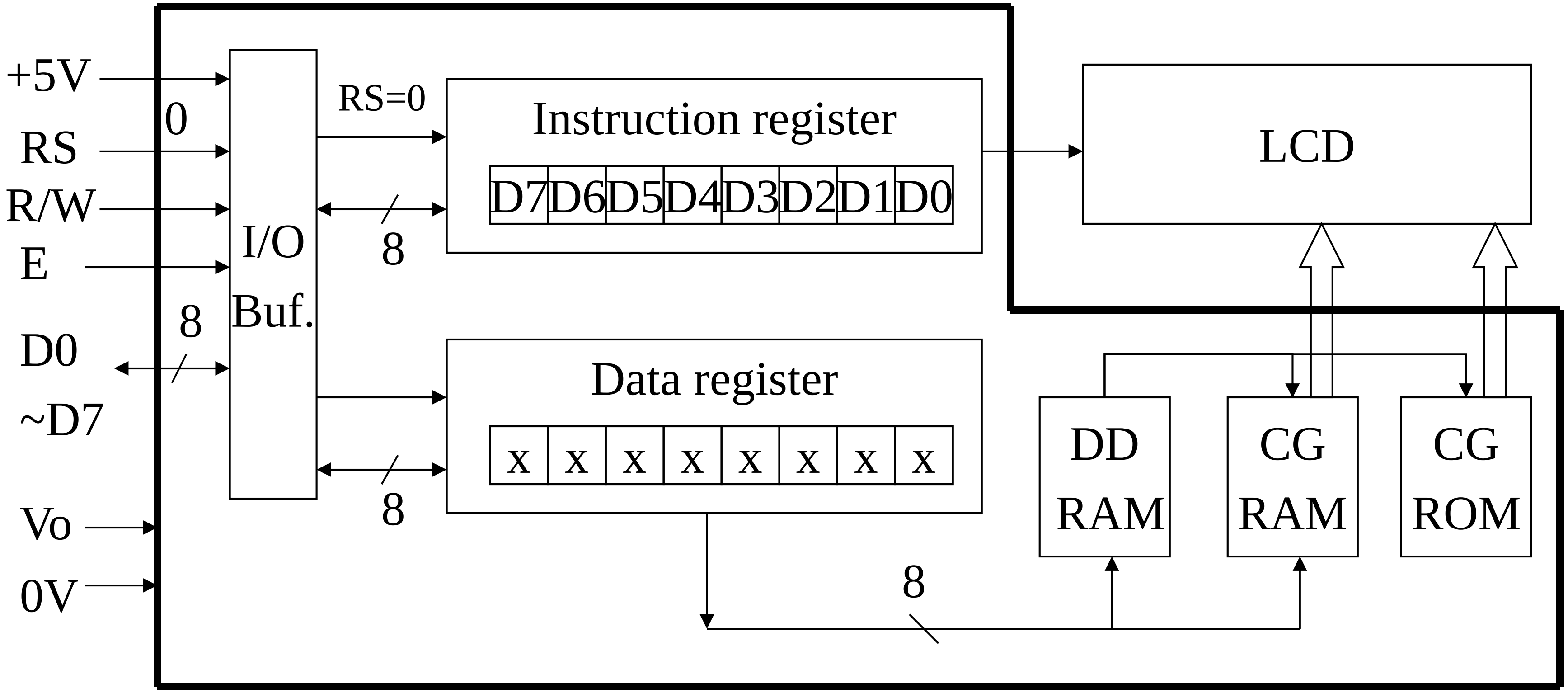
- ❒ DR(Data Register)
  - 8 bit register is used as a temporary data storage place for being written into or read from DDRAM or CGRAM
- ❒ IR(Instruction Register)
  - 8 bit register is used as a temporary data storage place
- ❒ BF (Busy Flag)
  - When RS=0, R/W=1, BF is output to DB7
  - BF=1: busy, the next instruction cannot be accepted
  - BF=0: not busy , the next instruction can be accepted
- ❒ AC (Address Counter)
  - When RS=0, R/W=1, AC value can be read through DB0 - DB6 ports
  - Address Counter (AC) stores the address of DDRAM/CGRAM that are transferred from IR
  - After writing into (reading from) DDRAM/CGRAM data, AC is increased (decreased) by 1 automatically

IR and DR


+5V
RS
R/W
E
D0
~D7
8
Vo
0V
I/O
Buf.
RS=0
8
RS=1
8
Instruction register
D7 D6 D5 D4 D3 D2 D1 D0
AC6 AC5 AC4 AC3 AC2 AC1 AC0
Data register
D7 D6 D5 D4 D3 D2 D1 D0
8
LCD
DD RAM
CG RAM
CG ROM

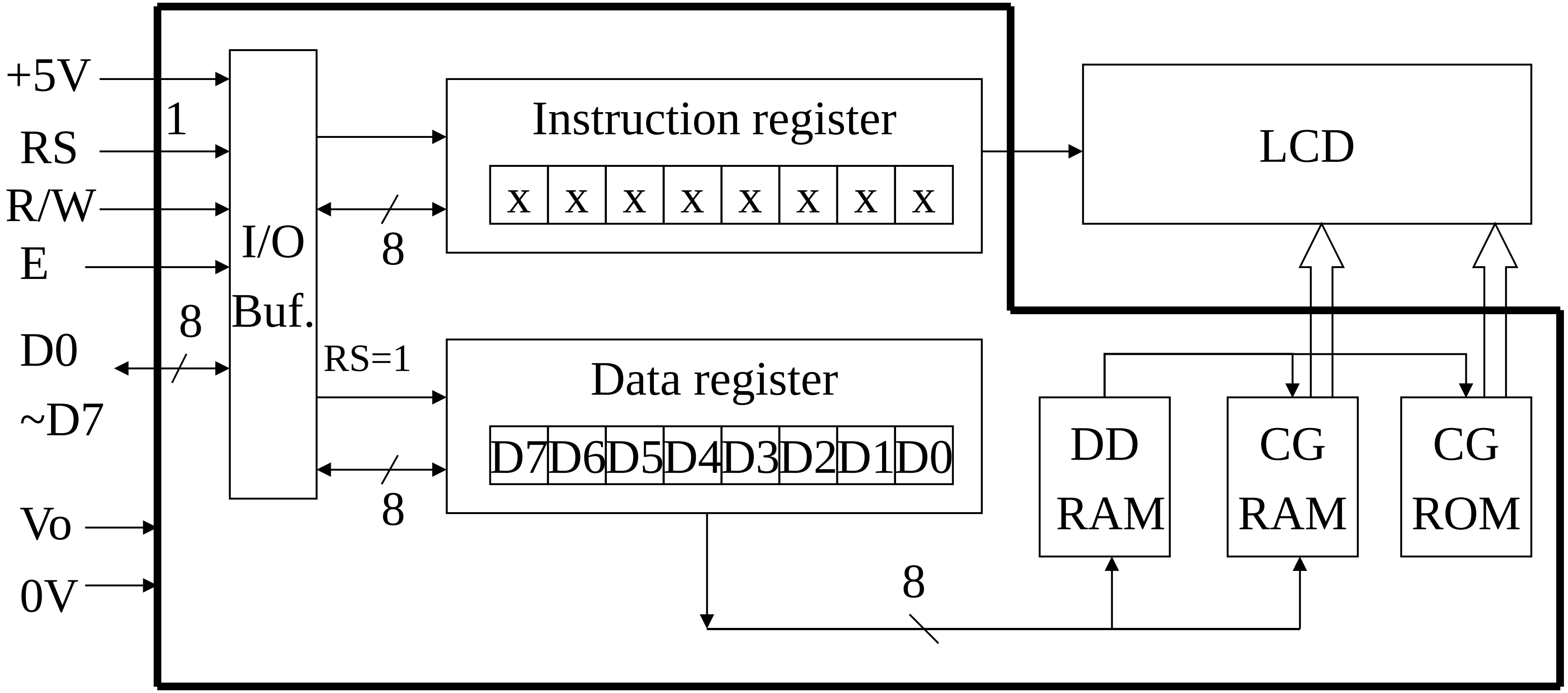
IR and DR

❒ **When RS = 0,**



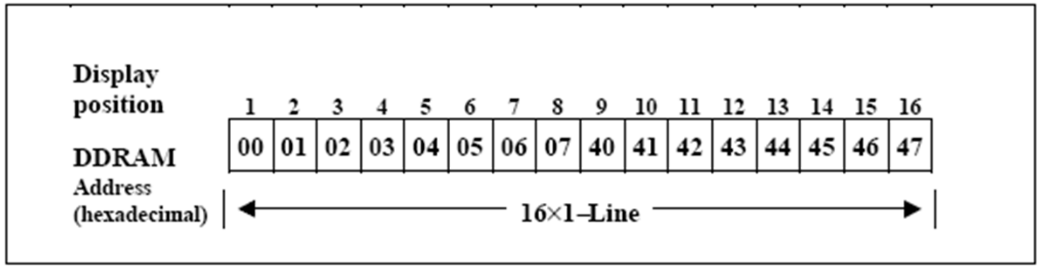
IR and DR

❒ When RS = 1,



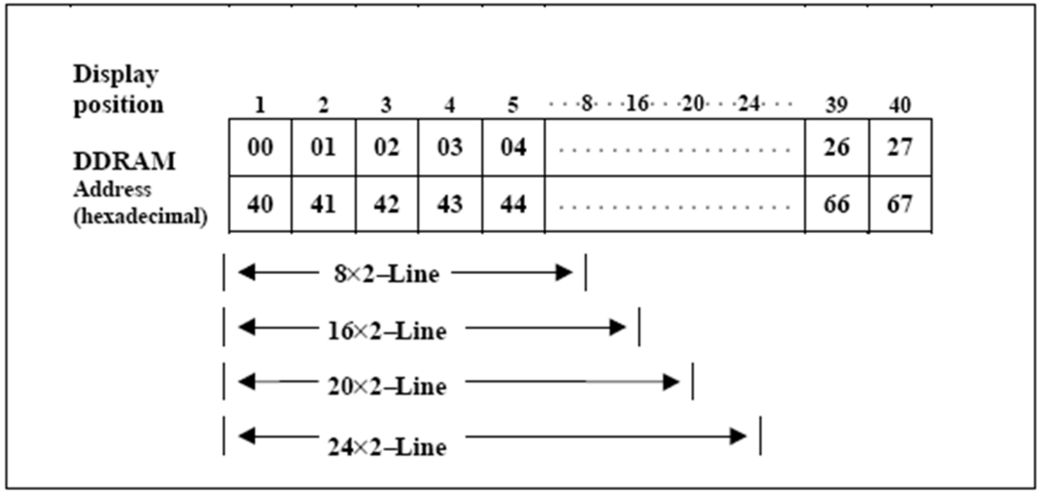
DDRAM Addresses

10.4 1-Line Display



Display position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM Address (hexadecimal)	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

2-Line Display



Display position	1	2	3	4	5	· · ·8· · ·16· · ·20· · ·24· · ·	39	40
DDRAM Address (hexadecimal)	00	01	02	03	04	. . . .	26	27
	40	41	42	43	44	. . . .	66	67

Function Set Instruction

❒ Sets the following

- **Interface data length (DL)**
    - DL=1 (8 bit data line), DL=0 (4 bit data line)
- **Number of display lines**
    - N=0 (1 line), N=1 (2 lines)
- **Character font**
    - F=0: 5 x 8 dots per character
    - F=1: 5 x 10 dots per character

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

Other Instructions

❒ Display On/Off

- ON/OFF of all display (D=0: OFF)
- ON/OFF of cursor (C=0: OFF)
- Blink of cursor position (B=0: OFF)

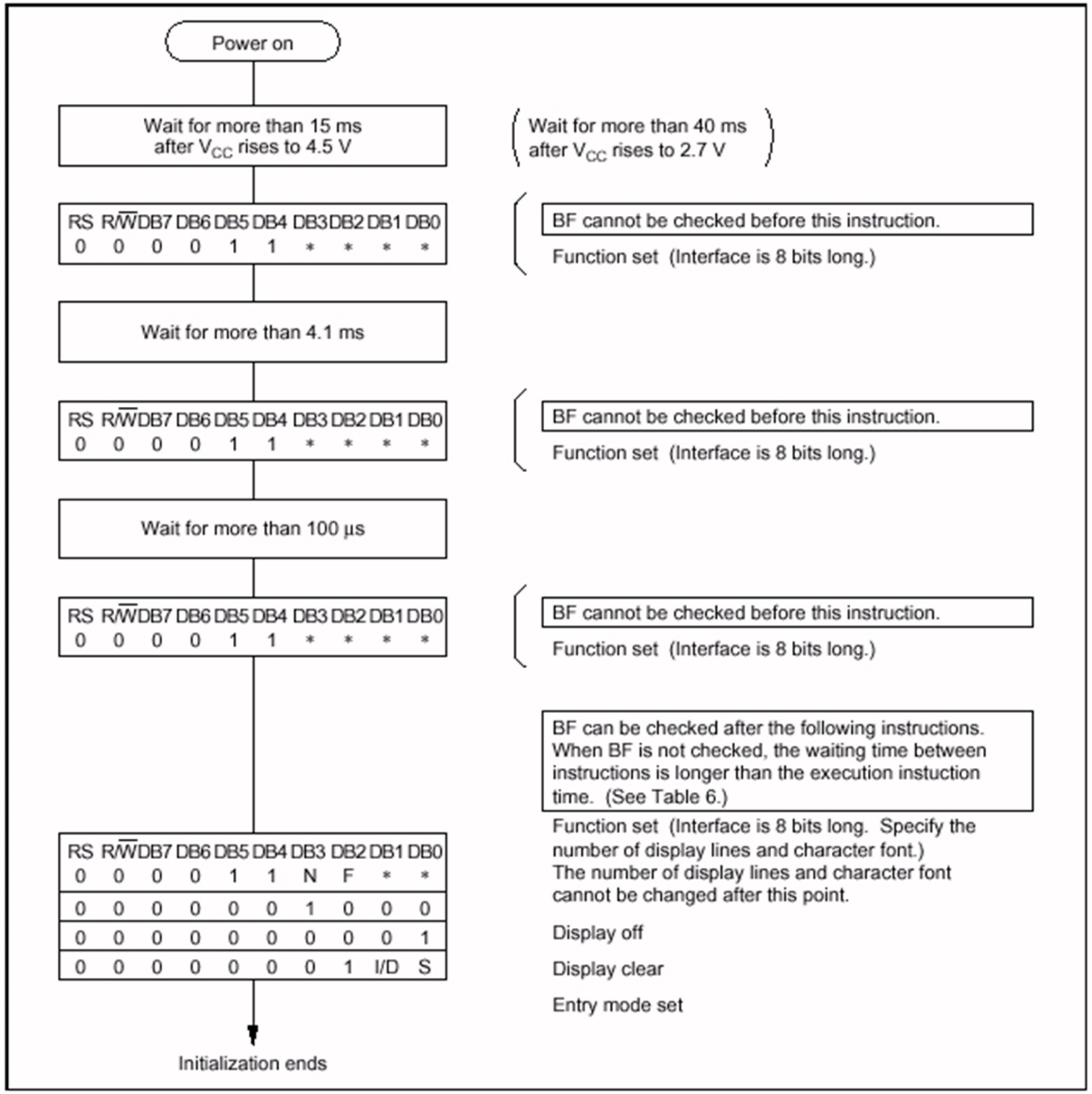
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	D	C	B

❒ Entry Mode Set

- Cursor move direction (I/D=1: increment, I/D=0: decrement)
- Display shift (S=1: shift)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

Initialization



Power on

Wait for more than 15 ms after V_{CC} rises to 4.5 V

(Wait for more than 40 ms after V_{CC} rises to 2.7 V)

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

BF cannot be checked before this instruction.

Function set (Interface is 8 bits long.)

Wait for more than 4.1 ms

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

BF cannot be checked before this instruction.

Function set (Interface is 8 bits long.)

Wait for more than 100 μs

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

BF cannot be checked before this instruction.

Function set (Interface is 8 bits long.)

BF can be checked after the following instructions. When BF is not checked, the waiting time between instructions is longer than the execution instuction time. (See Table 6.)

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	N	F	*	*
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	I/D	S

Function set (Interface is 8 bits long. Specify the number of display lines and character font.) The number of display lines and character font cannot be changed after this point.

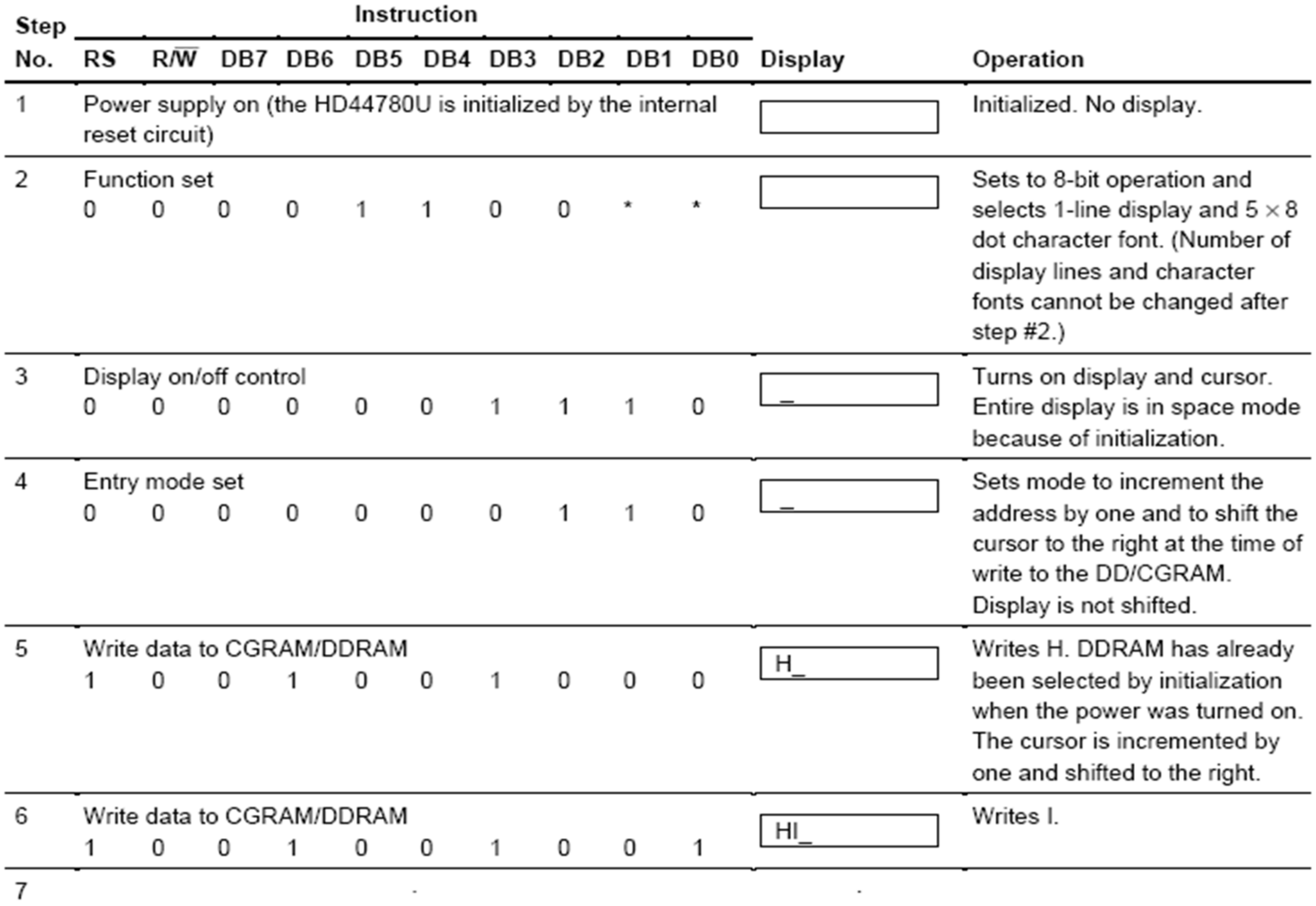
Display off

Display clear

Entry mode set

Initialization ends

Displaying “HI”



Step	Instruction											
No.	RS	R/$\overline{\text{W}}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
1	Power supply on (the HD44780U is initialized by the internal reset circuit)											Initialized. No display.
2	Function set											Sets to 8-bit operation and selects 1-line display and 5 × 8 dot character font. (Number of display lines and character fonts cannot be changed after step #2.)
	0	0	0	0	1	1	0	0	*	*		
3	Display on/off control										_	Turns on display and cursor. Entire display is in space mode because of initialization.
	0	0	0	0	0	0	1	1	1	0		
4	Entry mode set										_	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
	0	0	0	0	0	0	0	1	1	0		
5	Write data to CGRAM/DDRAM										H_	Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
	1	0	0	1	0	0	1	0	0	0		
6	Write data to CGRAM/DDRAM										HI_	Writes I.
	1	0	0	1	0	0	1	0	0	1		
7					.						.	

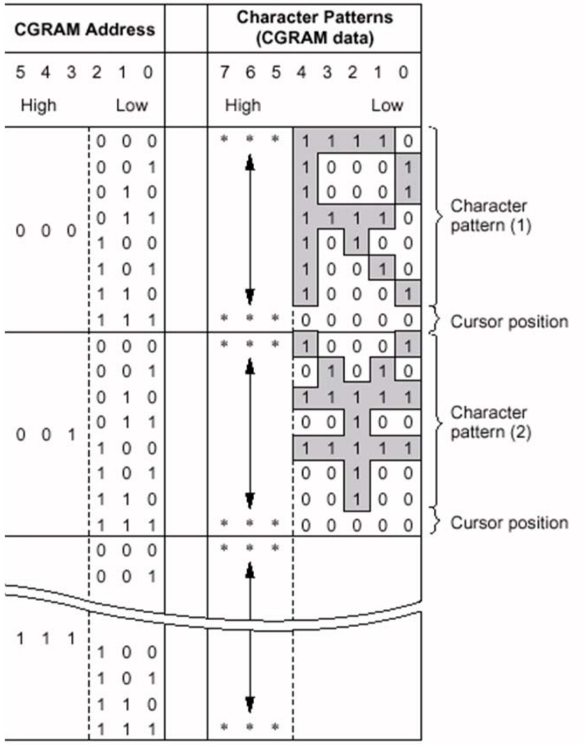
Displaying Custom Characters with CGRAM

❒ CGRAM address is represented by 6 bits

- **Recall from the instruction format that 6 bits are available**
- **8 characters can be stored for 5 x 8 dots**

❒ Storing custom characters

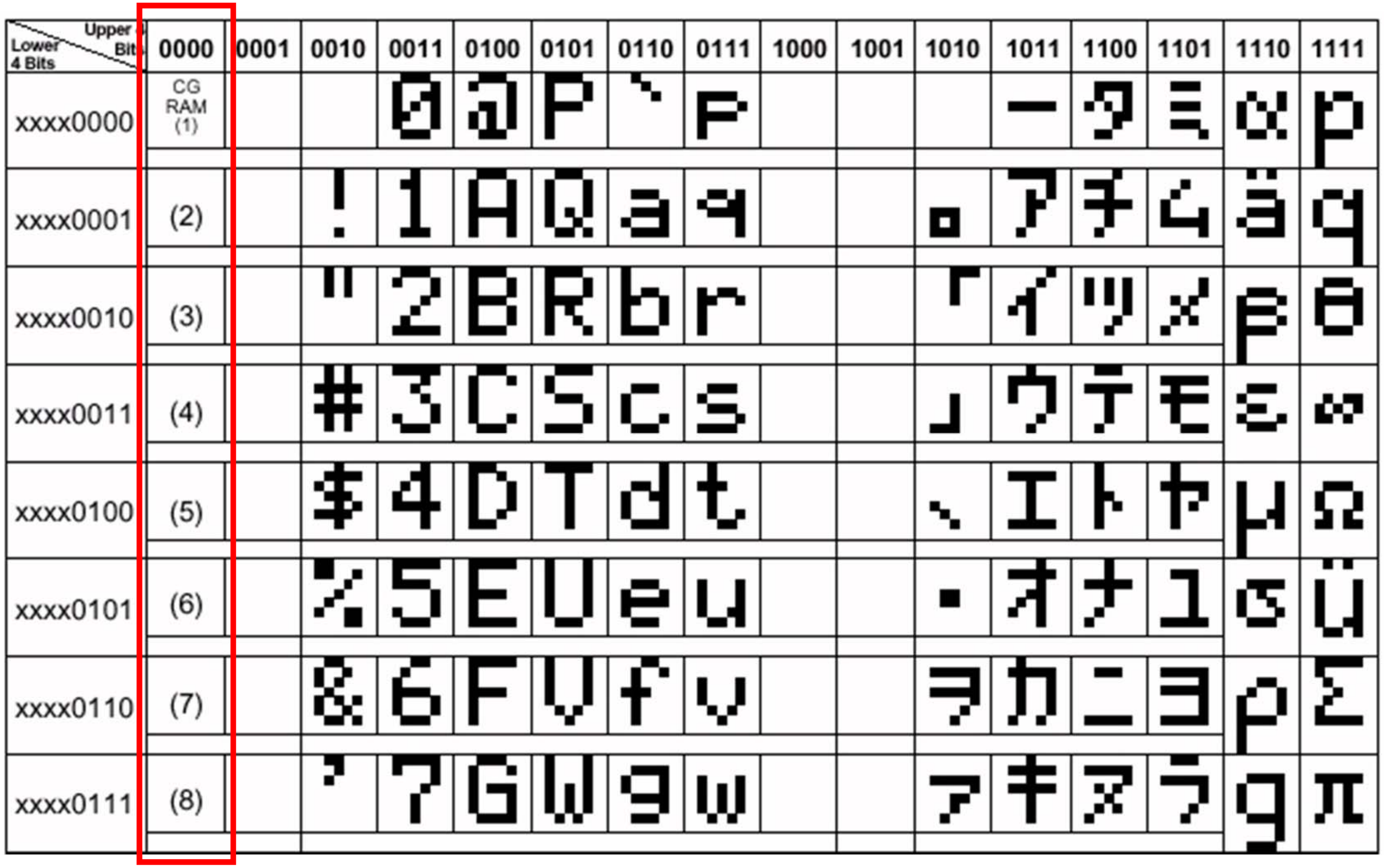
- **Write data to each CGRAM address**
  - First set CGRAM address by using an instruction “00 01xx xxxx”
  - Then write data using an instruction “10 ***x xxxx”
- **Note that upper 3 bits are not used for 5 x 8 dots**



CGRAM Address						Character Patterns (CGRAM data)								
5	4	3	2	1	0	7	6	5	4	3	2	1	0	
High					Low	High							Low	
0	0	0	0	0	0	*	*	*	1	1	1	1	0	Character pattern (1)
			0	0	1				1	0	0	0	1	
			0	1	0				1	0	0	0	1	
			0	1	1				1	1	1	1	0	
			1	0	0				1	0	1	0	0	
			1	0	1				1	0	0	1	0	
			1	1	0				1	0	0	0	1	
			1	1	1	*	*	*	0	0	0	0	0	Cursor position
0	0	1	0	0	0	*	*	*	1	0	0	0	1	Character pattern (2)
			0	0	1				0	1	0	1	0	
			0	1	0				1	1	1	1	1	
			0	1	1				0	0	1	0	0	
			1	0	0				1	1	1	1	1	
			1	0	1				0	0	1	0	0	
			1	1	0				0	0	1	0	0	
			1	1	1	*	*	*	0	0	0	0	0	Cursor position
			0	0	0	*	*	*						
			0	0	1									
1	1	1												
			1	0	0									
			1	0	1									
			1	1	0									
			1	1	1	*	*	*						

Displaying Custom Characters with CGRAM

- Eight entries are reserved for CGRAM characters

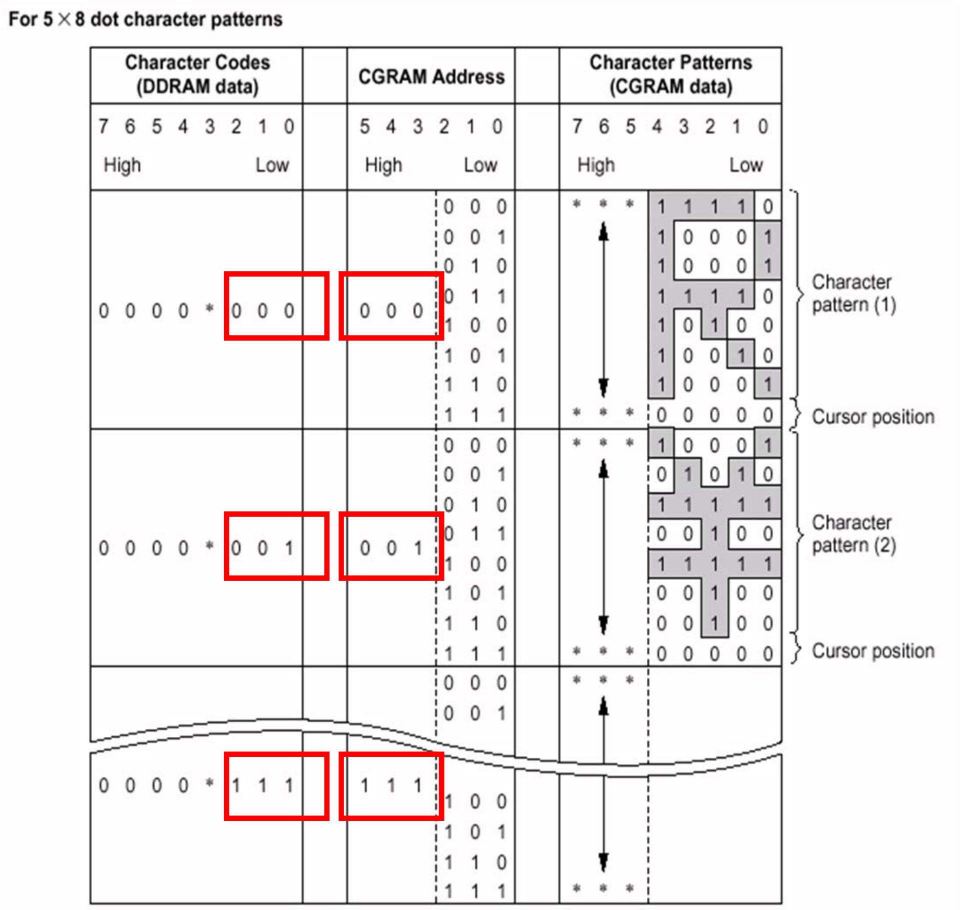


Upper Bits / Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	メ	β	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π

Displaying Custom Characters with CGRAM

- ❒ **Mapping between character code and CGRAM address**

- ❒ **CGRAM character can be displayed in the same manner as CGROM**
  - ▪ **First set DDRAM address (display position) by using an instruction “00 1xxx xxxx”**
  - ▪ **Then write data using an instruction “10 0000 *xyz”**



For 5×8 dot character patterns

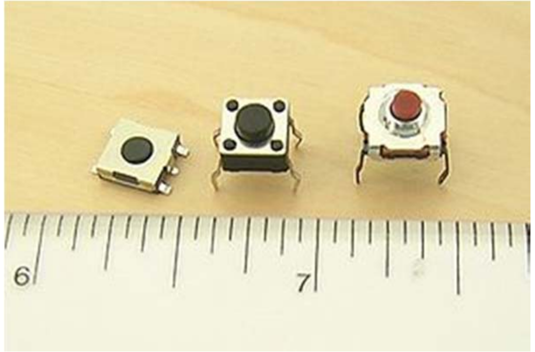
Character Codes (DDRAM data) 7 6 5 4 3 2 1 0 (High … Low)	CGRAM Address 5 4 3 (High)	CGRAM Address 2 1 0 (Low)	Character Patterns (CGRAM data) 7 6 5 (High)	Character Patterns 4 3 2 1 0 (Low)	
0 0 0 0 * 0 0 0	0 0 0	0 0 0	* * *	1 1 1 1 0	Character pattern (1)
		0 0 1		1 0 0 0 1	
		0 1 0		1 0 0 0 1	
		0 1 1		1 1 1 1 0	
		1 0 0		1 0 1 0 0	
		1 0 1		1 0 0 1 0	
		1 1 0		1 0 0 0 1	
		1 1 1	* * *	0 0 0 0 0	Cursor position
0 0 0 0 * 0 0 1	0 0 1	0 0 0	* * *	1 0 0 0 1	Character pattern (2)
		0 0 1		0 1 0 1 0	
		0 1 0		1 1 1 1 1	
		0 1 1		0 0 1 0 0	
		1 0 0		1 1 1 1 1	
		1 0 1		0 0 1 0 0	
		1 1 0		0 0 1 0 0	
		1 1 1	* * *	0 0 0 0 0	Cursor position
		0 0 0	* * *		
		0 0 1			
0 0 0 0 * 1 1 1	1 1 1	1 0 0			
		1 0 1			
		1 1 0			
		1 1 1	* * *		

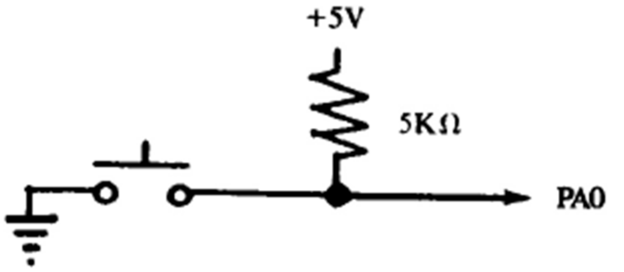
Topics Covered

1. Discrete LEDs
2. Seven-Segment LEDs
3. Character LCD
4. Keypads
5. Touch Screens

Switch

- A switch is an electrical component that can break an electrical circuit, interrupting the current or diverting it from one conductor to another
  - The most familiar form of switch is a manually operated electromechanical device with one or more sets of electrical contacts
  - Each set of contacts can be in one of two states: either 'closed‘ or 'open’





Switch open: Out = Vcc
Switch closed: Out = GND

Interfacing with pull-up resistor

Contact Bounce (Chatter)

- Contact bounce is a common problem with mechanical switches
  - Switch contacts are usually made of springy metals that are forced into contact by an actuator
  - When the contacts strike together, their momentum and elasticity act together to cause bounce
  - The result is a rapidly pulsed electrical current instead of a clean transition from zero to full current

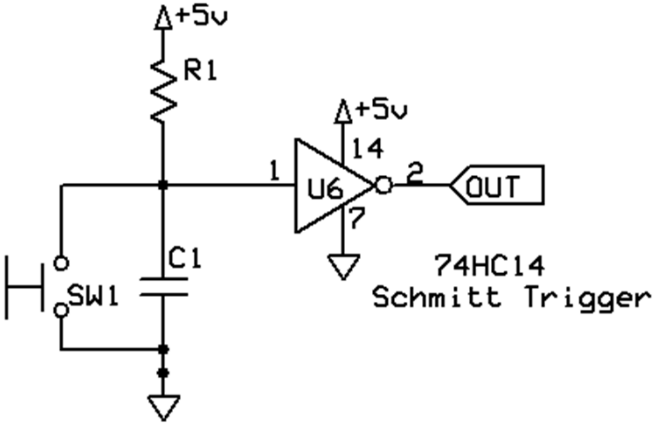
Debouncing Solutions

❒ Wet contacts

- Mercury wetted switch maintains a conductive path despite the mechanical bounce, but are prohibited by RoHS (Restriction of Hazardous Substances Directive) because of mercury's toxicity

❒ Pull up (or pull down) resistor and capacitor

- When the switch is closed the capacitor will almost instantly discharge through the switch



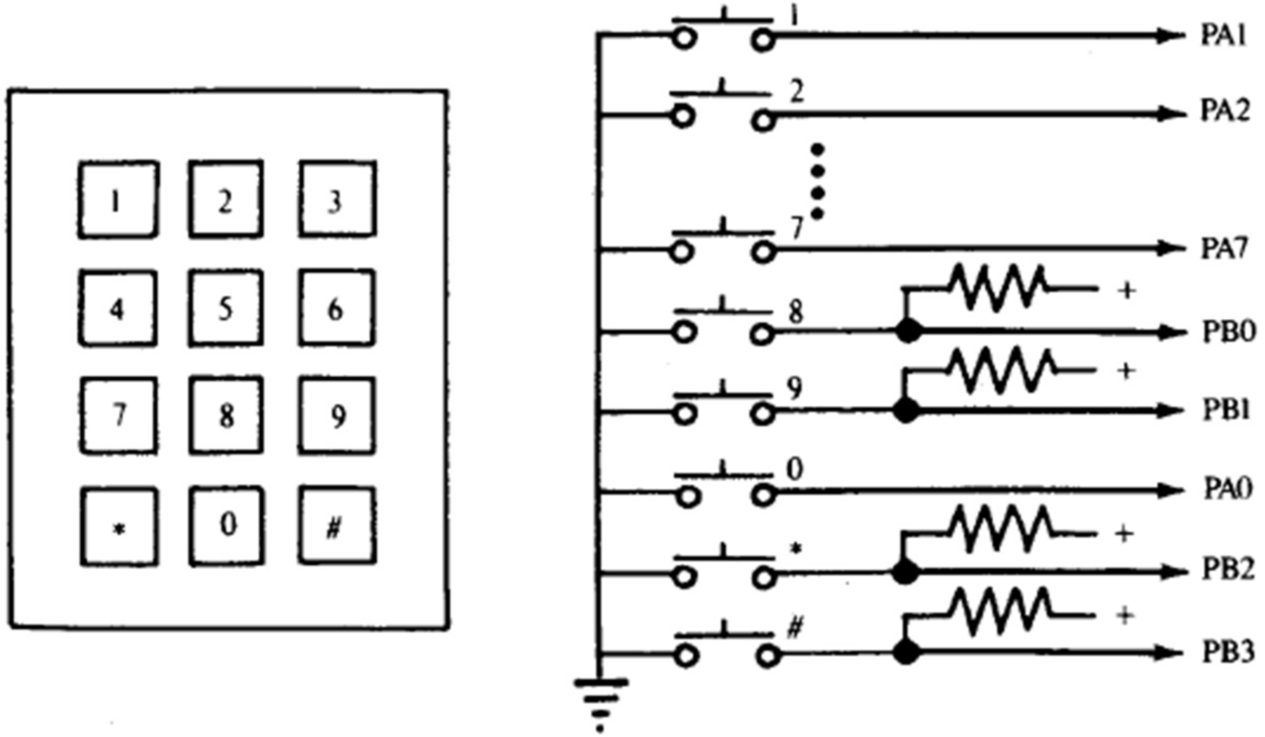
Choose RC > duration of bounce, in seconds

❒ Sampling

- The simplest way to debounce a switch transition, either in hardware or software, is merely to sample the switch state at intervals longer than any possible train of bounces

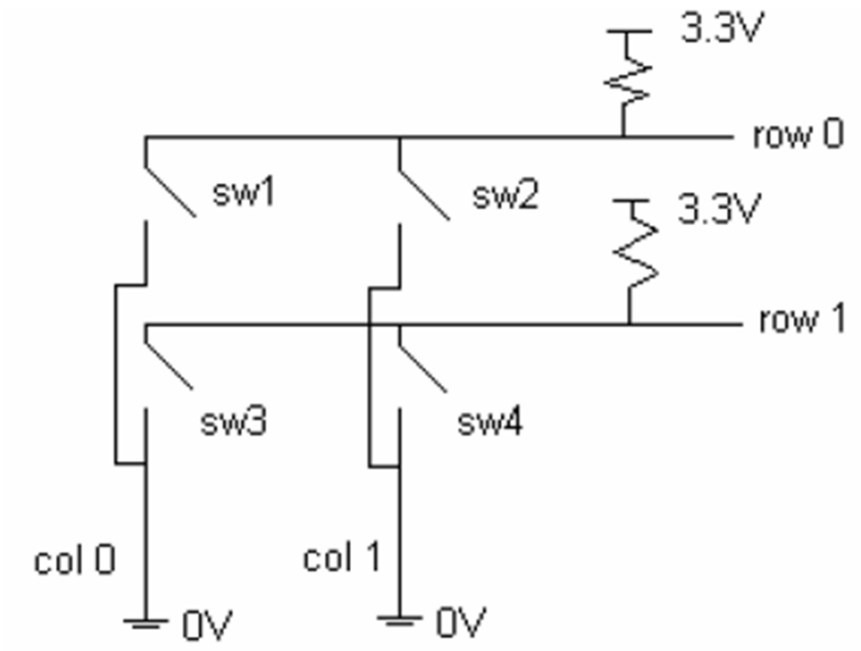
12-Button Keyboard (Phone)

- 12 buttons
- 12 parallel input bits are required



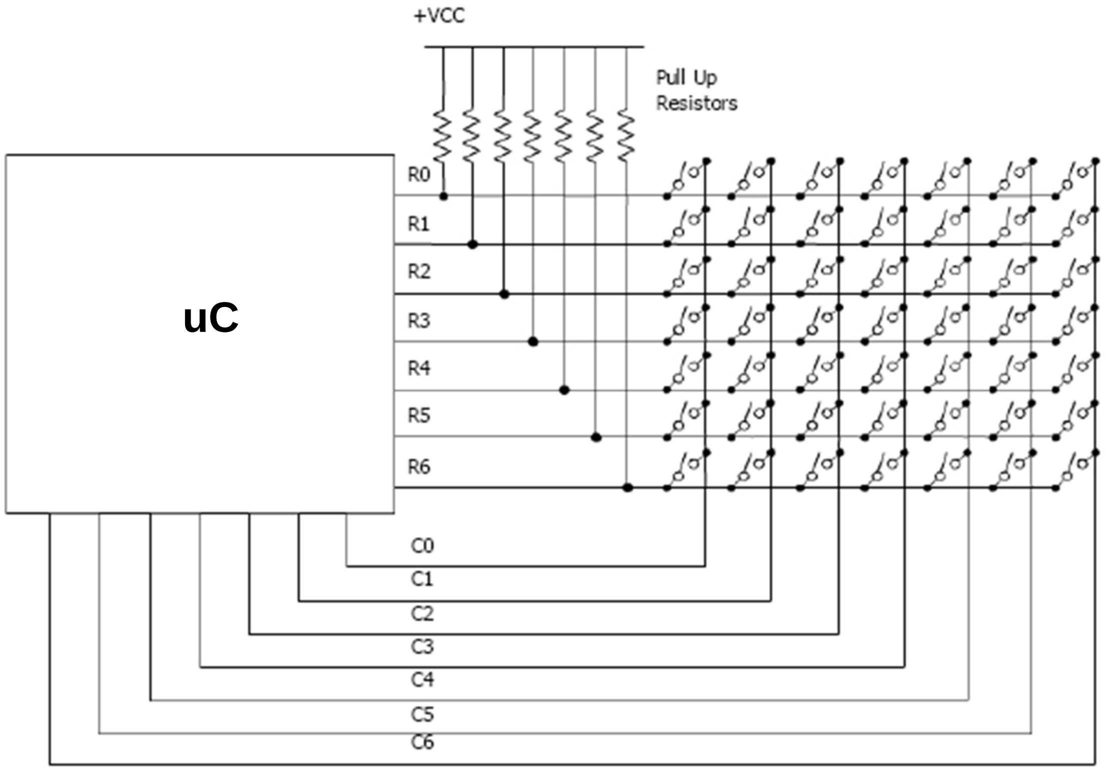
Matrix Keypad

- ❒ A matrix keypad is a set of switches arranged so that they fall into rows and columns (a matrix)
  - The top terminals of every switch in the same row are internally connected together and also connected to an external pin
  - The bottom terminals of every switch in the same column are internally connected together and also connected to a different external pin



Keypad Scanning

- The columns are configured as outputs
  - On power-up or on reset, all the columns are driven low
- The rows are configured as inputs and pulled up



Keypad Scanning

- Whenever a key is pressed, the row and column corresponding to that key get connected, causing its row input to go low

- SW or HW should then start actively scanning each column one by one to determine the location of the key that was pressed
  - The column which is being scanned is made low while others are tri-stated
  - If the key which is pressed lies in this column, one of the row inputs will go low, revealing its location
  - When the required column is detected, it should maintain the column at ground and checks whether the same row is still low after a small delay
    - This is done in order to debounce the key press

Topics Covered

1. **Discrete LEDs**
2. **Seven-Segment LEDs**
3. **Character LCD**
4. **Keypads**
5. **Touch Screens**

Touchscreen

❒ A touchscreen is a display which can detect the presence and location of a touch within the display area

- The term generally refers to touch or contact to the display of the device by a finger or hand
- Touchscreens can also sense other passive objects, such as a stylus

Touchscreen Technologies

- ❒ Resistive
  - A resistive touchscreen panel is composed of several layers
  - The most important are two thin metallic electrically conductive and resistive layers separated by thin space
  - When some object touches this kind of touch panel, the layers are connected at a certain point; the panel then electrically acts similar to two voltage dividers with connected outputs
  - This causes a change in the electrical current which is registered as a touch event and sent to the controller for processing

- ❒ Surface acoustic wave
  - Surface acoustic wave (SAW) technology uses ultrasonic waves that pass over the touchscreen panel
  - When the panel is touched, a portion of the wave is absorbed
  - This change in the ultrasonic waves registers the position of the touch event and sends this information to the controller for processing

Touchscreen Technologies

❒ Capacitive

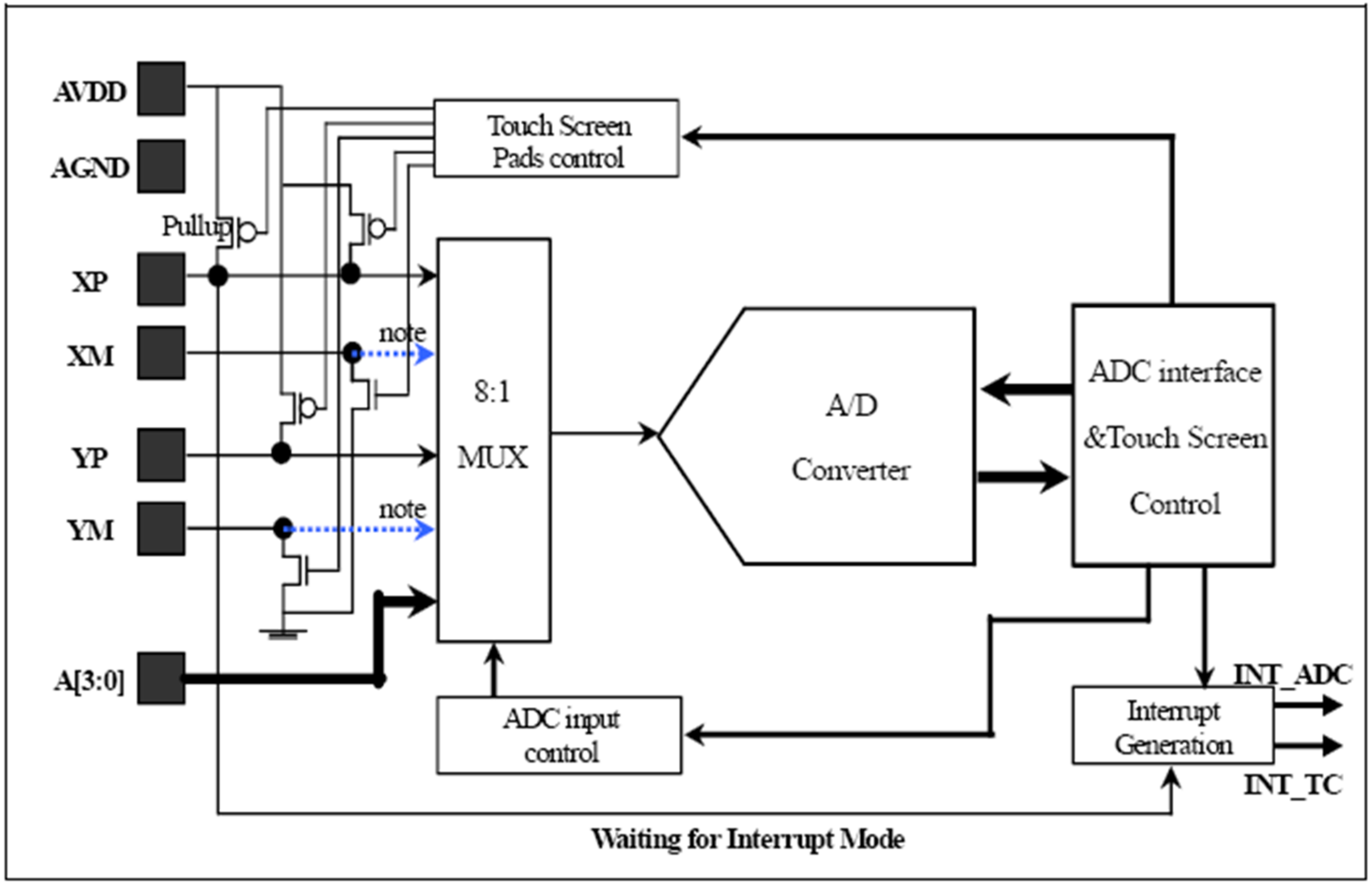
- A capacitive touchscreen panel is coated with a material, typically indium tin oxide, that conducts a continuous electrical current across the sensor
- The sensor therefore exhibits a precisely controlled field of stored electrons in both the horizontal and vertical axes - it achieves capacitance
- The human body is also an electrical device which has stored electrons and therefore also exhibits capacitance
- Capacitive sensors work based on proximity, and do not have to be directly touched to be triggered
- It is a durable technology that is used in a wide range of applications
- It has a higher clarity than Resistive technology, but it only responds to finger contact and will not work with a gloved hand or pen stylus unless the stylus is conductive and transmits the user's capacitance
- Capacitive touch screens can also support Multitouch (several fingers)
- Examples include Apple Inc.'s iPhone and iPod touch, and HTC's G1 & HTC Magic

S3C2440A ADC & Touchscreen

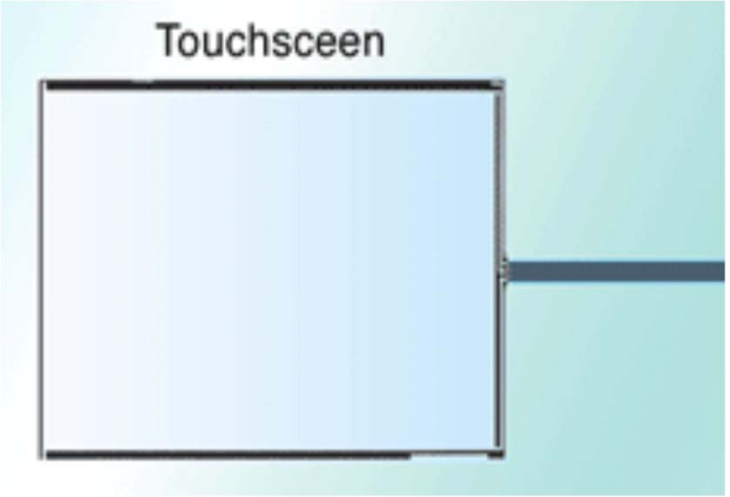
❒ S3C2440A ADC & touchscreen

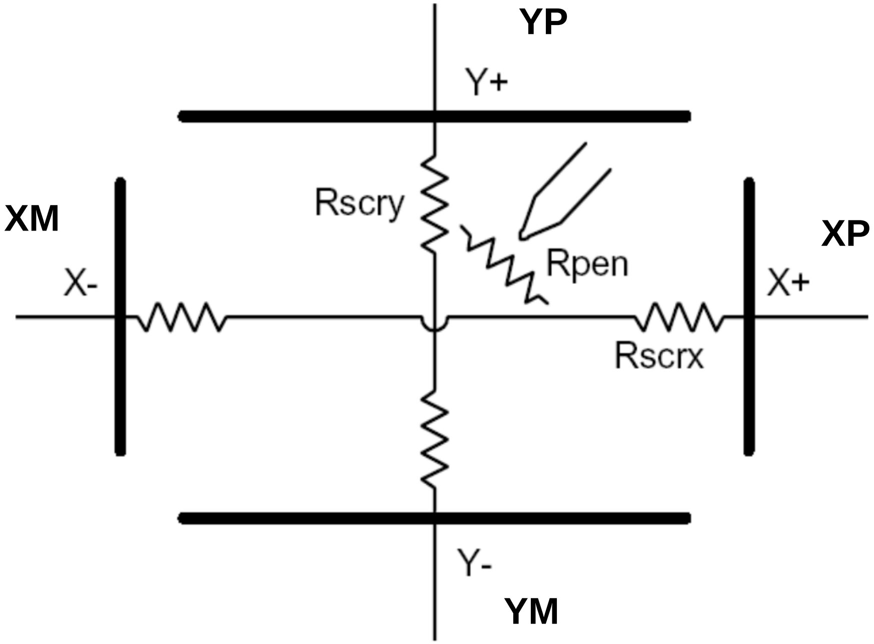
- 8-channel analog inputs
- Converting the analog input to 10-bit digital codes
- Maximum 500K SPS (samples per second)
- 2.5 MHz A/D converter clock
- Touch screen interface control pads (XP, XM, YP, YM) for X, Y position conversion

Block Diagram



Resistive Touchscreen and Equivalent Circuit





- The position is obtained from X and Y resistance values