

Chapter 9

Design Example: text LCD controller logic

Text LCD 출력장치

Text LCD 출력장치

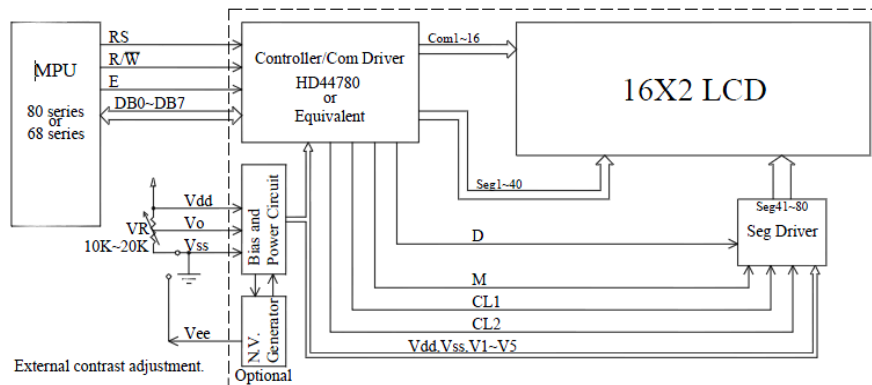
- 1-2 줄의 문자를 표시할 수 있는 LCD 출력장치
- Altera DE2 보드는 16 문자 2줄의 text LCD 장치를 제공



- 8비트 또는 4비트 마이크로프로세서의 제어신호와 쉽게 연결하도록 되어 있음 (E, R/W 제어신호)
- 내부의 HD44780 dot matrix LCD controller에서 출력 표시를 제어함
- backlight 기능

Text LCD 출력 장치 – Block Diagram

Block Diagram



Text LCD 출력 장치 – Interface Pin 기능

Interface Pin기능

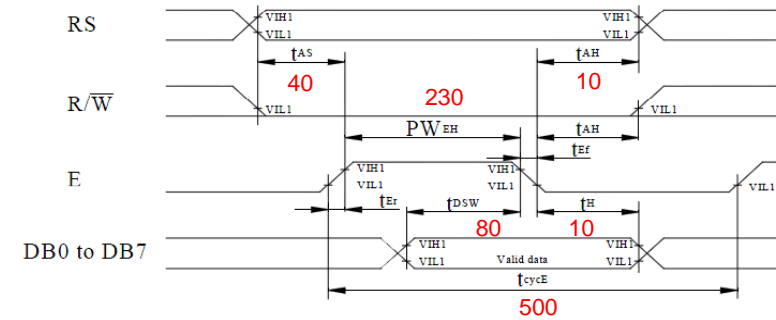
핀	신호명	기능
1	V _{SS}	전원 GND
2	V _{DD}	전원 +5V
3	V _{EE}	Contrast 제어 전압레벨 (V _{DD} -V _{EE} =13.5~0V)
4	RS	Register Select (0=instruction, 1=data)
5	R/W	Read/Write (0=FPGA->LCD, 1:FPGA<-LCD)
6	E	Enable Signal for read/write LCD
7	DB0(LSB)	Data
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	A	+LED (backlight LED용 전원 +4.4~4.7V)
16	K	-LED (backlight LED용 전원 GND)

Text LCD: Altera-DE2에서의 pin 배치

DE2		DE2-70	
Signal Name	FPGA Pin No.	Description	FPGA Pin No.
LCD_DATA[0]	PIN_J1	LCD Data[0]	PIN_E1
LCD_DATA[1]	PIN_J2	LCD Data[1]	PIN_E3
LCD_DATA[2]	PIN_H1	LCD Data[2]	PIN_D2
LCD_DATA[3]	PIN_H2	LCD Data[3]	PIN_D3
LCD_DATA[4]	PIN_J4	LCD Data[4]	PIN_C1
LCD_DATA[5]	PIN_J3	LCD Data[5]	PIN_C2
LCD_DATA[6]	PIN_H4	LCD Data[6]	PIN_C3
LCD_DATA[7]	PIN_H3	LCD Data[7]	PIN_B2
LCD_RW	PIN_K4	LCD Read/Write Select, 0 = Write, 1 = Read	PIN_F3
LCD_EN	PIN_K3	LCD Enable	PIN_E2
LCD_RS	PIN_K1	LCD Command/Data Select, 0 = Command, 1 = Data	PIN_F2
LCD_ON	PIN_L4	LCD Power ON/OFF	PIN_F1
LCD_BLON	PIN_K2	LCD Back Light ON/OFF	PIN_G3

Text LCD 출력장치 – Timing

Write timing도



$t_{AS} = 40\text{ns min}$ (address setup time)

$PW_{EH} = 230\text{ns min}$ (enable pulse width)

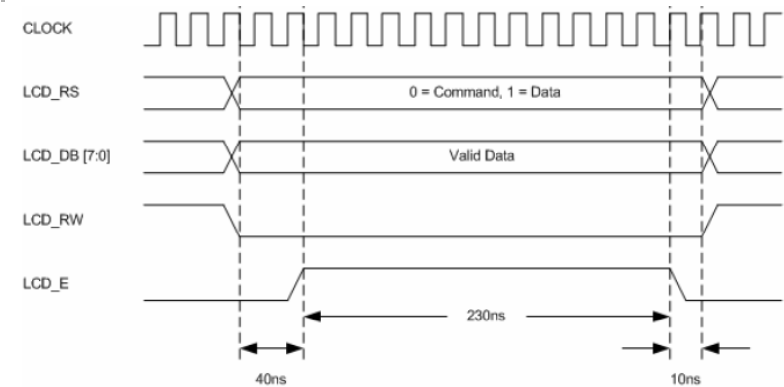
$t_{AH} = 10\text{ns min}$ (address hold time)

Text LCD 출력장치 – Timing(2)

Write 동작 timing parameter

Item	Symbol	Min	Typ	Max	Unit
Enable cycle time	t_{cycE}	500	—	—	ns
Enable pulse width (high level)	PW_{EH}	230	—	—	ns
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	20	ns
Address set-up time (RS, R/W to E)	t_{AS}	40	—	—	ns
Address hold time	t_{AH}	10	—	—	ns
Data set-up time	t_{DSW}	80	—	—	ns
Data hold time	t_H	10	—	—	ns

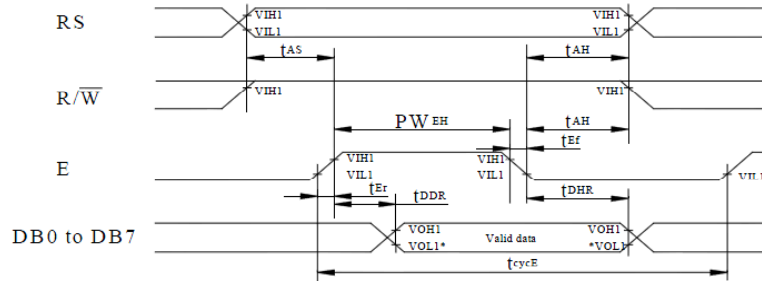
Text LCD 출력장치 – Timing(3)



RW=0: write

Text LCD 출력장치 – Timing(4)

Read timing도



RW=1: read

- 예제 설계에서는 read operation은 사용하지 않음

제어 명령

LCD장치 제어 명령표

기능	제어 신호		제어 명령							
	Rs	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear Display	0	0	0	0	0	0	0	0	0	1
Return home	0	0	0	0	0	0	0	0	0	1
Entry mode set	0	0	0	0	0	0	0	1	I/D	S
Display ON/OFF control	0	0	0	0	0	0	1	D	C	B
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	0	0
Function set	0	0	0	0	1	DL	N	F	0	0
Set CG RAM address	0	0	0	1	CG RAM address					
Set DD RAM address	0	0	1	DD RAM address						
Read busy flag and address	0	1	BF	Address Counter						
Data write to CG RAM or DD RAM	1	0	Data							
Data read from CG RAM or DD RAM	1	1	Data							

write command

read status

write RAM

read RAM

제어 명령 설명

기능	bit	1	0	
clear display	화면을 지우고, 커서 Home으로			
return home	화면은 그대로, 커서만 Home으로			
entry mode set	I/D(inc/dec)	커서이동방향	증가	감소
	S (shift)	화면이동(data 입력 시)	on	off
display on/off control	D (display)	화면 표시	on	off
	C (cursor)	커서 표시	on	off
	B (blinking)	깜박임	on	off
display or cursor shift	S/C	이동 대상	screen	cursor
	R/L	이동 방향	right	left
function set	DL	data length	8-bit	4-bit
	N	number of display	2	1
	F	font size	5 x 10	5 x 7

제어 명령 설명

기능	내용
set CG RAM address	CG RAM 주소 설정 이후의 read/write data는 CG RAM을 사용
set DD RAM address	DD RAM 주소 설정 이후의 read/write data는 DD RAM을 사용
read busy flag and address	busy flag(BF)와 address counter(AC) 값을 read BF=0일 때에만 제어명령의 write가 가능함

- DDRAM(Display data RAM) – 출력 문자를 저장함
- CG RAM(character generator RAM) – 사용자 정의 문자 폰트 저장

Data의 read, write

- data는 address counter(AC)가 지정하는 주소의 CG RAM 또는 DD RAM 에 read/write 됨

DDRAM 주소와 출력 위치

Display Data RAM (DD RAM)

- 출력 문자들의 ASCII 코드를 저장함 (1줄당 40문자 저장)

Display position	1	2	3	4	5	39	40
DDRAM address (hexadecimal)	00	01	02	03	04	26	27
	40	41	42	43	44	66	67

DD RAM 주소와 출력 위치

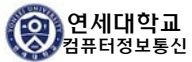
Character located	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DDRAM address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

(기본)

Display Shift

For shift left	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50

For shift right	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E



Text LCD 출력 장치 - 표시문자와 코드값

	00H	10H	20H	30H	40H	50H	60H	70H	80H	90H	A0H	B0H	C0H	D0H	E0H	F0H	
0			0	@	P	'	p					-	タ	ミ	α	ρ	
1			!	1	A	Q	a	q				。	ア	チ	ム	à	q
2				2	B	R	b	r				「	イ	ツ	メ	β	θ
3			#	3	C	S	c	s				」	ウ	テ	モ	ξ	∞
4			\$	4	D	T	d	t				「	エ	ト	ヤ	μ	Ω
5			%	5	E	U	e	u				」	オ	ナ	ユ	σ	ι
6			&	6	F	V	f	v				ヲ	カ	ニ	ヨ	ρ	Σ
7				7	G	W	g	w				ァ	キ	ヌ	ラ	g	Π
8			(8	H	X	h	x				ィ	ク	ネ	リ	√	×
9)	9	I	Y	i	y				ウ	ケ	ノ	ル	-1	y
A			*	:	J	Z	j	z				エ	コ	ハ	レ	j	千
B			+	;	K	[k	{				ォ	サ	ヒ	ロ	*	カ
C			,	<	L	¥	l					ャ	シ	フ	ワ	©	円
D			-	=	M]	m	}				ユ	ス	ヘ	ン	〒	÷
E			.	>	N	^	n	→				ヨ	ゼ	ホ		ñ	
F			/	?	O	_	o	←				ッ	ソ	マ	°	ö	■

ASCII코드



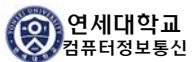
Text LCD 출력장치 - 초기화

초기화 과정

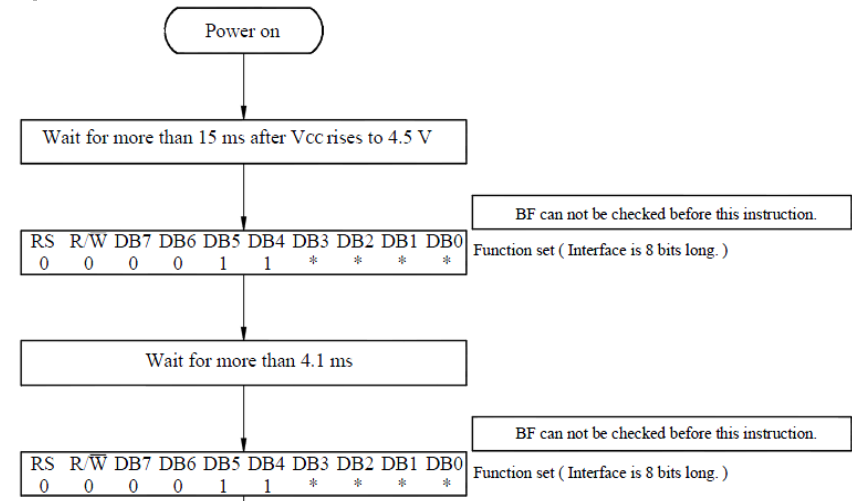
- 전원공급 50ms 후에 LCD 모듈이 reset이 완료됨
- Function set 명령 설정 (command = 001xxx00)
- Display on/off 명령 설정 (command = 00001xxx)
- Entry mode set 명령 설정 (command = 000001xx)
- DD RAM address 설정 (command = 1aaaaaaa)

- 문자 데이터를 write하면 화면에 출력됨 (write할 때마다 address counter가 이동됨)

- 명령어의 수행 시간이 명령어에 따라서 40μs에서 1.5ms 정도 소요되므로 text LCD에 대한 두 명령어 쓰기 사이에 이 시간 이상의 시간 지연이 있어야 한다.



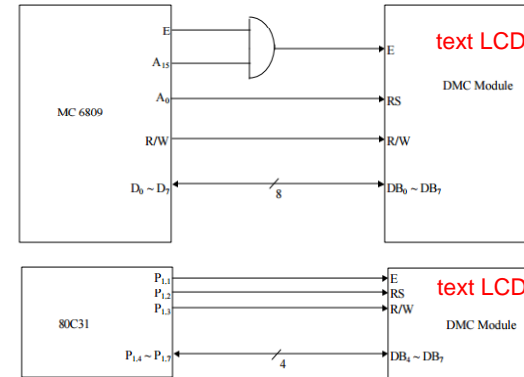
Text LCD 출력장치 - 초기화(2)



Text LCD 회로 설계

■ text LCD

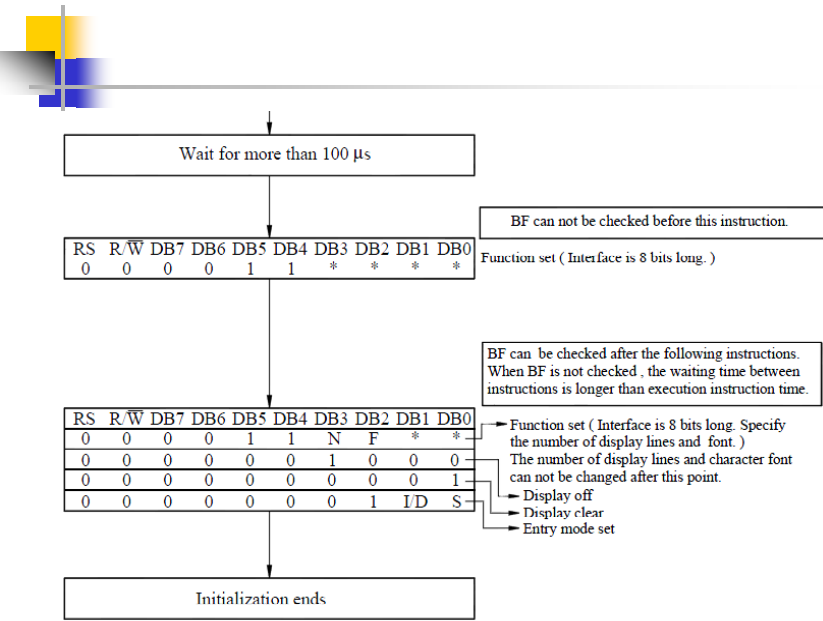
- 원래 4/8-bit microprocessor의 버스 또는 microcontroller의 I/O 포트와 연결하고, 프로그램의 제어로 사용하도록 되어 있다.



CPU에서 제어신호의 타이밍을 하드웨어로 제공함

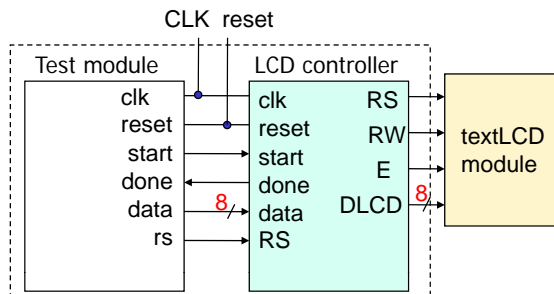
프로그램에서 명령과 데이터를 제공함

microcontroller에서 프로그램으로 제어신호의 타이밍과 명령/데이터를 제공함



■ FPGA에서의 text LCD 제어

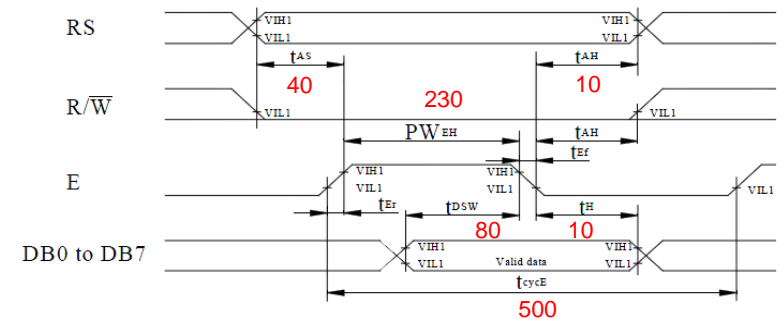
- CPU가 없으므로 제어신호의 타이밍과 명령/데이터를 순서대로 제공하는 기능을 모두 하드웨어로 설계해야 함
 - lcd controller 모듈 : 제어신호 타이밍 제공
 - lcd test 모듈 : text lcd 명령과 데이터(출력문자)를 적절한 순서로 lcd controller로 제공



Text LCD controller 설계

■ LCD write 동작 제어신호 생성

- CLK: 50MHz 클럭을 사용하는 경우 (T = 20ns)

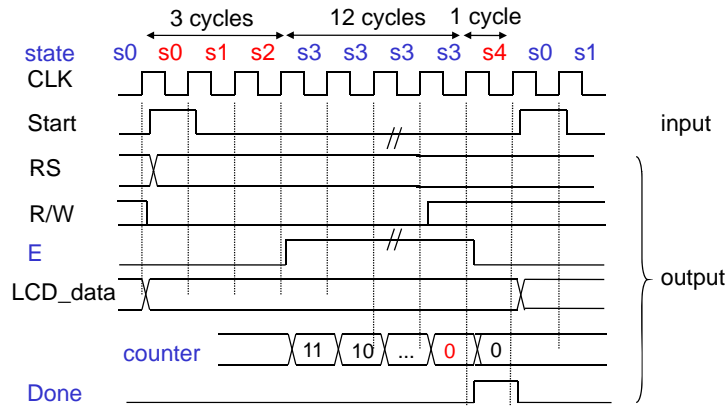


$t_{AS} = 40\text{ns min}$
 $PW_{EH} = 230\text{ns min}$
 $t_{AH} = 10\text{ns min}$

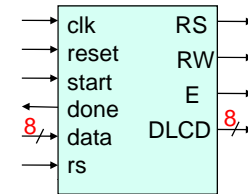
→ 3 CLK 이상
 → 12 CLK 이상
 → 1 CLK 이상

Text LCD controller timing 도

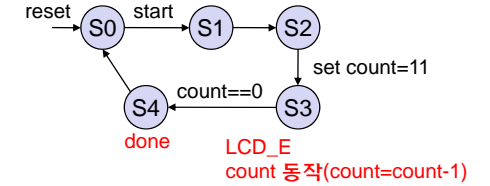
LCD write timing



Text LCD controller 블록도 및 상태도



block diagram



state diagram

Text LCD Controller 코드

```

module lcd_controller(clk, reset, start, RS, data, done,
                    LCD_RS, LCD_RW, LCD_E, LCD_data);
// Host side
input clk; // 50MHz clock
input reset, start;
input RS; // register select
input [7:0] data;
output done; // operation done
// text LCD side
output LCD_RS, LCD_RW, LCD_E;
output [7:0] LCD_data;

reg [2:0] state;
reg [3:0] count;
localparam S0=0, S1=1, S2=2, S3=4, S4=3; // state
localparam WIDTH_E = 12; // pulse width = 12 clk

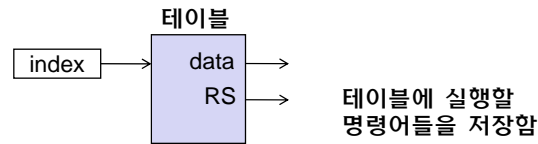
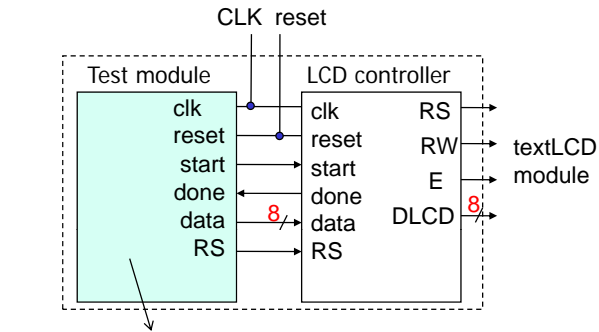
assign LCD_RS = RS; // register select
assign LCD_RW = 1'b0; // write only
assign LCD_data = data; // 8-bit data output
    
```

```

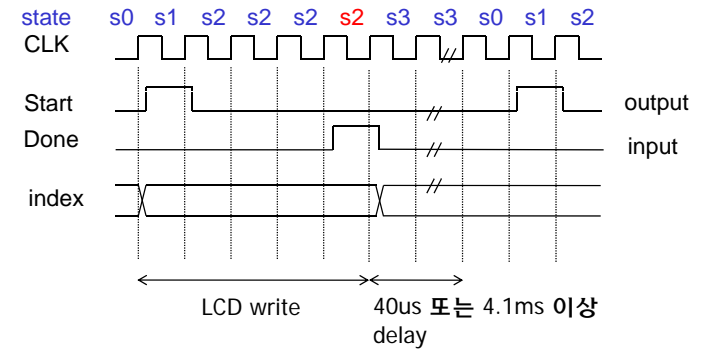
// Timing width
// 50MHz clk -> T = 20ns
// LCD_RS, LCD_RW to LCD_E >= 40ns 3 clk
// LCD_E pulse width >= 230ns 12 clk
// LCD_E to LCD_RS, LCD_RW >= 10ns 1 clk
always @(posedge clk or posedge reset) begin
    if (reset) begin state <= S0; count <= 0; end
    else
        case (state)
            S0: if (start) state <= S1;
            S1: state <= S2;
            S2: if (count==0) begin state <= S3; count = WIDTH_E-1; end
                else count = count - 1;
            S3: if (count==0) state <= S4;
                else count <= count - 1;
            S4: state <= S0;
            default: state <= S0;
        endcase
end

assign LCD_E = (state==S3);
assign done = (state==S2);
endmodule
    
```

Text LCD Controller Test 모듈 설계



Test module 타이밍



Test Module

```

module lcd_test(clk, reset, start, RS, data, done);
    input clk; // 50 MHz (T=20ns)
    input reset;
    output start, RS;
    output [7:0] data;
    input done;

    reg [5:0] index;
    reg [7:0] data;
    reg [1:0] state;
    reg [1:0] delay; // 0: 40us, 1: 4.1ms
    reg [17:0] count;
    reg RS, halt;
    wire start;

    localparam INIT = 0;
    localparam LINE1 = INIT + 4;
    localparam LINE2 = LINE1 + 7;
    localparam LAST = LINE2 + 11;
    localparam S0=0, S1=1, S2=2, S3=3;

    localparam DELAY0 = 40000/20, // 40us
                DELAY1 = 4100000/20; // 4.1ms
    // localparam DELAY0 = 40/20, // for simulation
    // localparam DELAY1 = 120/20;
    
```

```

// S0 => S1 => S2 =>(done)=> S3 => S0
// start delay

// state, count, index
always @(posedge clk or posedge reset) begin
    if (reset) begin
        state <= S0;
        count <= 0;
        index <= INIT;
    end
    else begin
        case (state)
            S0: if (!halt) state <= S1; // idle
            S1: state <= S2; // start
            S2: if (done) begin // LCD write
                state <= S3;
                index <= index + 1; // next index
                if (delay) count <= DELAY1;
                else count <= DELAY0;
            end
            S3: if (count==0) state <= S0; // delay
                else count <= count - 1;
            default: state <= 0;
        endcase
    end
end

assign start = (state==S1);
    
```

```

// instruction table: data, RS, halt, delay
always @* begin
    data = "";
    halt = 0;
    delay = 0;
    RS = 1; // default: data
    case (index)
    // initialize
    INIT: begin data=8'b0011_1100; RS=0; delay = 1; end // function set
    INIT+1: begin data=8'b0000_1100; RS=0; delay = 1; end // display on/off
    INIT+2: begin data=8'b0000_0110; RS=0; delay = 1; end // entry mode set
    INIT+3: begin data=8'b0000_0001; RS=0; delay = 1; end // clear display
    // line 1
    LINE1: begin data=8'b1000_0000; RS=0; end // set DDRAM addr
    LINE1+1: data = "Y";
    LINE1+2: data = "o";
    LINE1+3: data = "n";
    LINE1+4: data = "s";
    LINE1+5: data = "e";
    LINE1+6: data = "i";
    // line 2
    LINE2: begin data=8'b1100_0000; RS=0; end // set DDRAM addr
    LINE2+1: data = "U";
    LINE2+2: data = "n";
    LINE2+3: data = "i";
    LINE2+4: data = "v";
    LINE2+5: data = "e";
    LINE2+6: data = "r";
    LINE2+7: data = "s";
    LINE2+8: data = "i";
    LINE2+9: data = "t";
    LINE2+10: data = "y";
    LAST: halt=1;
    default: halt=1;
    endcase
end
endmodule

```

통합 모듈 코드

```

module textLCD(iCLK_50, iKEY, oLCD_RS, oLCD_RW, oLCD_EN, LCD_D,
              oLCD_ON, oLCD_BLON);
    input iCLK_50;
    input [3:0] iKEY;
    // text LCD side
    output oLCD_RS, oLCD_RW, oLCD_EN;
    output [7:0] LCD_D;
    output oLCD_ON, oLCD_BLON;

    wire start, RS, done;
    wire [7:0] data;

    wire clk_50 = iCLK_50;
    wire reset = ~iKEY[0];

    lcd_test u1 (clk_50, reset, start, RS, data, done);
    lcd_controller u2 (clk_50, reset, start, RS, data, done,
                    oLCD_RS, oLCD_RW, oLCD_EN, LCD_D);

    assign oLCD_ON = 1'b1;
    assign oLCD_BLON = 1'b1;
endmodule

```