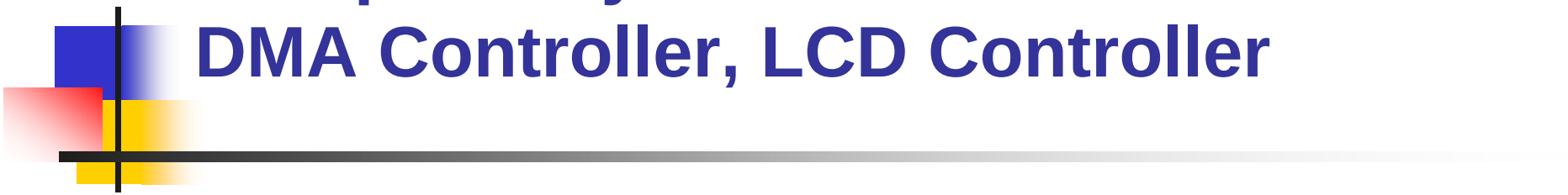
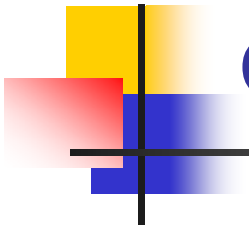


CPE 323 Introduction to Embedded Computer Systems: DMA Controller, LCD Controller

Instructor: Dr Aleksandar Milenkovic

Lecture Notes



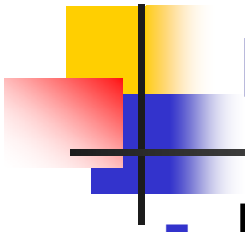
Outline

MSP430: System Architecture

- DMA Controller
- LCD Controller

DMA Controller Introduction

- Direct memory access (DMA) controller
  transfers data from one address to another
  without CPU intervention, across the entire address range.
  - Move data from the ADC12 conversion memory to RAM
  - Move data from RAM to DAC12
- Devices that contain a DMA controller may have one, two, or three DMA channels available
- Using the DMA controller
  - Can increase the throughput of peripheral modules
  - Can reduce system power consumption
    by allowing the CPU to remain in a low-power mode
    without having to awaken to move data to or from a peripheral

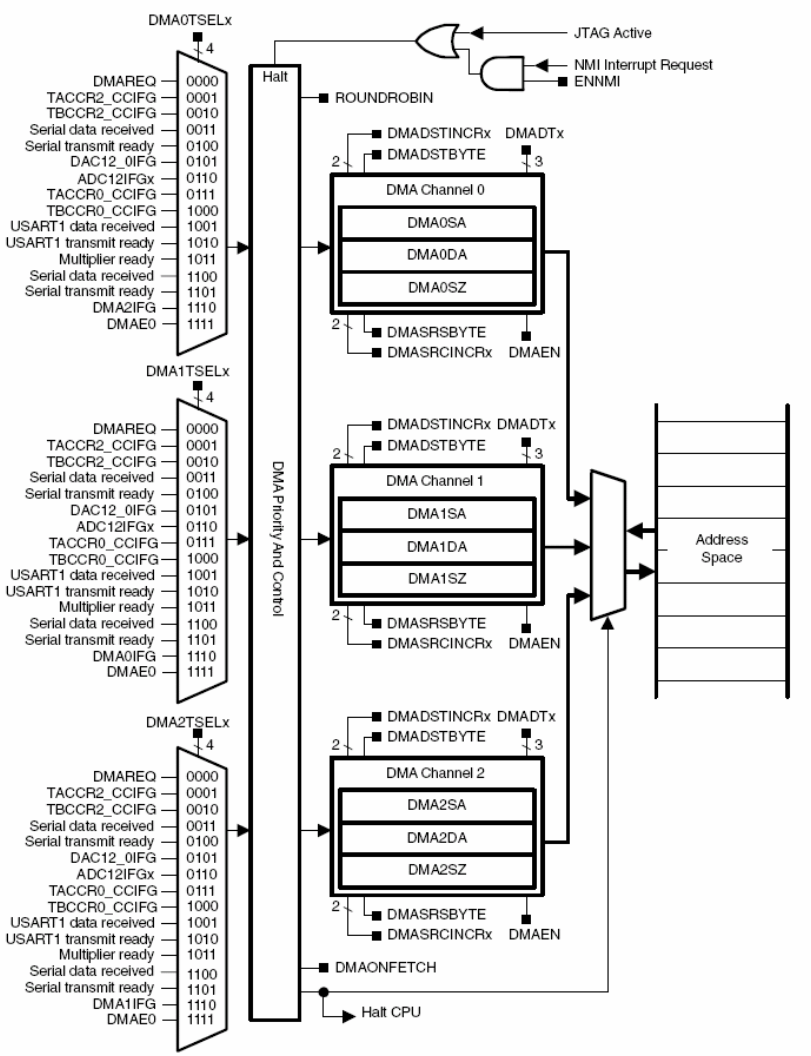


MSP430 DMA Features

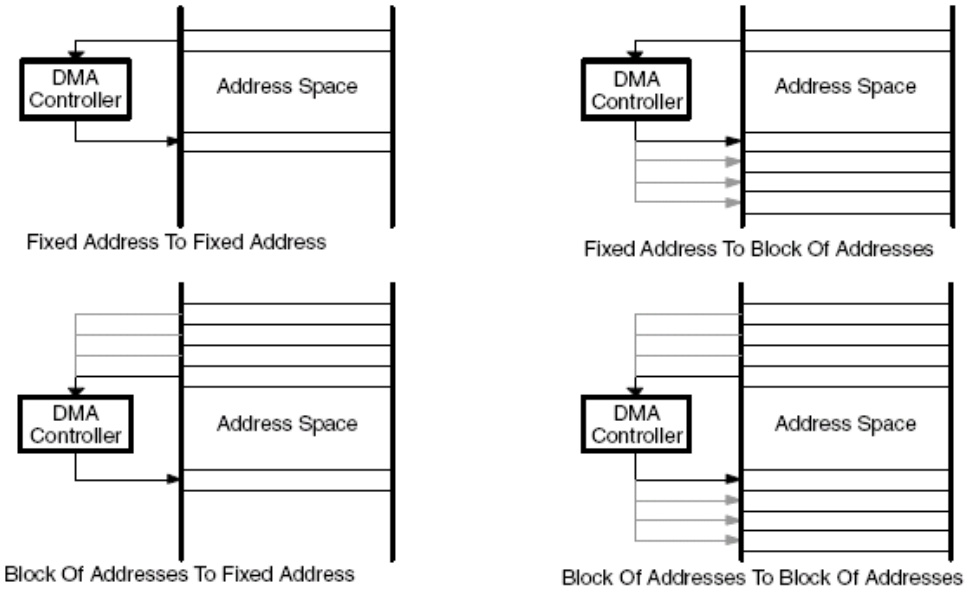
- Up to three independent transfer channels
- Configurable DMA channel priorities
- Requires only two MCLK clock cycles per transfer
- Byte or word and mixed byte/word transfer capability
- Block sizes up to 65535 bytes or words
- Configurable transfer trigger selections
- Selectable edge or level-triggered transfer
- Four addressing modes
- Single, block, or burst-block transfer modes

- Configured from software

DMA Block Diagram



DMA Addressing Modes



- Configured with the DMASRCINCRx and DMADSTINCRx control bits
  - DMASRCINCRx/ DMADSTINCRx bits select if the source/destination address is incremented, decremented, or unchanged after each transfer
- Transfers may be byte-to-byte, word-to-word, byte-to-word, or word-to-byte
  - Word-to-byte: only the lower byte of the source-word is transferred
  - Byte-to-word: the upper byte of the destination-word is cleared when the transfer occurs

DMA Transfer Modes

- Single/Repeated single modes: each byte/word transfer requires a separate trigger
- Block/Repeated block modes: a transfer of a complete block of data occurs after one trigger
  - CPU is halted until the complete block has been transferred
- Burst-block/Repeated burst-block modes: transfers are block transfers with CPU activity interleaved.
  - CPU executes 2 MCLK cycles after every four byte/word transfers of the block resulting in 20% CPU execution capacity

DMADTx	Transfer Mode	Description
000	Single transfer	Each transfer requires a trigger. DMAEN is automatically cleared when DMAxSZ transfers have been made.
001	Block transfer	A complete block is transferred with one trigger. DMAEN is automatically cleared at the end of the block transfer.
010, 011	Burst-block transfer	CPU activity is interleaved with a block transfer. DMAEN is automatically cleared at the end of the burst-block transfer.
100	Repeated single transfer	Each transfer requires a trigger. DMAEN remains enabled.
101	Repeated block transfer	A complete block is transferred with one trigger. DMAEN remains enabled.
110, 111	Repeated burst-block transfer	CPU activity is interleaved with a block transfer. DMAEN remains enabled.

DMA Trigger Operation

- DMAxTSELx bits select trigger
- Edge-sensitive or level-sensitive

Table 10–2. DMA Trigger Operation

DMAxTSELx	Operation
0000	A transfer is triggered when the DMAREQ bit is set. The DMAREQ bit is automatically reset when the transfer starts
0001	A transfer is triggered when the TACCR2 CCIFG flag is set. The TACCR2 CCIFG flag is automatically reset when the transfer starts. If the TACCR2 CCIE bit is set, the TACCR2 CCIFG flag will not trigger a transfer.
0010	A transfer is triggered when the TBCCR2 CCIFG flag is set. The TBCCR2 CCIFG flag is automatically reset when the transfer starts. If the TBCCR2 CCIE bit is set, the TBCCR2 CCIFG flag will not trigger a transfer.
0011	Devices with USART0: A transfer is triggered when the URXIFG0 flag is set. URXIFG0 is automatically reset when the transfer starts. If URXIE0 is set, the URXIFG0 flag will not trigger a transfer. Devices with USCI_A0: A transfer is triggered when the UCA0RXIFG flag is set. UCA0RXIFG is automatically reset when the transfer starts. If UCA0RXIE is set, the UCA0RXIFG flag will not trigger a transfer.
0100	Devices with USART0: A transfer is triggered when the UTXIFG0 flag is set. UTXIFG0 is automatically reset when the transfer starts. If UTXIE0 is set, the UTXIFG0 flag will not trigger a transfer. Devices with USCI_A0: A transfer is triggered when the UCA0TXIFG flag is set. UCA0TXIFG is automatically reset when the transfer starts. If UCA0TXIE is set, the UCA0TXIFG flag will not trigger a transfer.
0101	A transfer is triggered when the DAC12_0CTL DAC12IFG flag is set. The DAC12_0CTL DAC12IFG flag is automatically cleared when the transfer starts. If the DAC12_0CTL DAC12IE bit is set, the DAC12_0CTL DAC12IFG flag will not trigger a transfer.
0110	A transfer is triggered by an ADC12IFGx flag. When single-channel conversions are performed, the corresponding ADC12IFGx is the trigger. When sequences are used, the ADC12IFGx for the last conversion in the sequence is the trigger. A transfer is triggered when the conversion is completed and the ADC12IFGx is set. Setting the ADC12IFGx with software will not trigger a transfer. All ADC12IFGx flags are automatically reset when the associated ADC12MEMx register is accessed by the DMA controller.

DMA Trigger Operation (cont'd)

0111	A transfer is triggered when the TACCR0 CCIFG flag is set. The TACCR0 CCIFG flag is automatically reset when the transfer starts. If the TACCR0 CCIE bit is set, the TACCR0 CCIFG flag will not trigger a transfer.
1000	A transfer is triggered when the TBCCR0 CCIFG flag is set. The TBCCR0 CCIFG flag is automatically reset when the transfer starts. If the TBCCR0 CCIE bit is set, the TBCCR0 CCIFG flag will not trigger a transfer.
1001	A transfer is triggered when the URXIFG1 flag is set. URXIFG1 is automatically reset when the transfer starts. If URXIE1 is set, the URXIFG1 flag will not trigger a transfer.
1010	A transfer is triggered when the UTXIFG1 flag is set. UTXIFG1 is automatically reset when the transfer starts. If UTXIE1 is set, the UTXIFG1 flag will not trigger a transfer.
1011	A transfer is triggered when the hardware multiplier is ready for a new operand.
1100	A transfer is triggered when the UCB0RXIFG flag is set. UCB0RXIFG is automatically reset when the transfer starts. If UCB0RXIE is set, the UCB0RXIFG flag will not trigger a transfer.
1101	A transfer is triggered when the UCB0TXIFG flag is set. UCB0TXIFG is automatically reset when the transfer starts. If UCB0TXIE is set, the UCB0TXIFG flag will not trigger a transfer.
1110	A transfer is triggered when the DMAxIFG flag is set. DMA0IFG triggers channel 1, DMA1IFG triggers channel 2, and DMA2IFG triggers channel 0. None of the DMAxIFG flags are automatically reset when the transfer starts.
1111	A transfer is triggered by the external trigger DMAE0.

DMA Channel Priorities

- Default DMA channel priorities are DMA0−DMA1−DMA2
  - If two or three triggers happen simultaneously or are pending, the channel with the highest priority completes its transfer (single, block or burst-block transfer) first, then the second priority channel, then the third priority channel.
- Transfers in progress are not halted if a higher priority channel is triggered
  - The higher priority channel waits until the transfer in progress completes before starting
- DMA channel priorities are configurable with the ROUNDROBIN bit (see below)

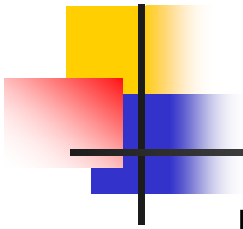
DMA Priority	Transfer Occurs	New DMA Priority
DMA0 – DMA1 – DMA2	DMA1	DMA2 – DMA0 – DMA1
DMA2 – DMA0 – DMA1	DMA2	DMA0 – DMA1 – DMA2
DMA0 – DMA1 – DMA2	DMA0	DMA1 – DMA2 – DMA0

DMA Transfer Cycle Times

- DMA requires 1 or 2 MCLK cc to synchronize before each single transfer or complete block or burst-block transfer
- Each byte/word transfer requires 2 MCLK after synchronization, and one cycle of wait time after the transfer
- DMA cycle time is dependent on the MSP430 operating mode and clock system setup (use MCLK)
  - If the MCLK source is active, but the CPU is off, the DMA controller will use the MCLK source for each transfer, without re-enabling the CPU.
  - If the MCLK source is off, the DMA controller will temporarily restart MCLK, sourced with DCOCLK, for the single transfer or complete block or burst-block transfer
  - The CPU remains off, and after the transfer completes, MCLK is turned off.

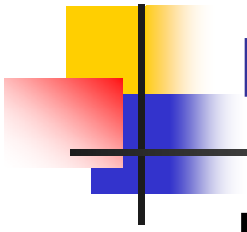
CPU Operating Mode	Clock Source	Maximum DMA Cycle Time
Active mode	MCLK=DCOCLK	4 MCLK cycles
Active mode	MCLK=LFXT1CLK	4 MCLK cycles
Low-power mode LPM0/1	MCLK=DCOCLK	5 MCLK cycles
Low-power mode LPM3/4	MCLK=DCOCLK	5 MCLK cycles + 6 µs†
Low-power mode LPM0/1	MCLK=LFXT1CLK	5 MCLK cycles
Low-power mode LPM3	MCLK=LFXT1CLK	5 MCLK cycles
Low-power mode LPM4	MCLK=LFXT1CLK	5 MCLK cycles + 6 µs†

† The additional 6 µs are needed to start the DCOCLK. It is the $t_{(LPMx)}$ parameter in the data sheet.



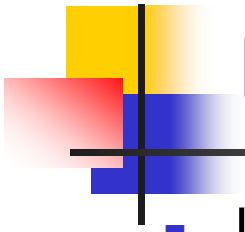
DMA and Interrupts

- DMA transfers are not interruptible by system interrupts.
  - System interrupts remain pending until the completion of the transfer
  - NMI interrupts can interrupt the DMA controller if the ENNMI bit is set
- System interrupt service routines are interrupted by DMA transfers
  - If an interrupt service routine or other routine must execute with no interruptions, the DMA controller should be disabled prior to executing the routine
- Each DMA channel has its own DMAIFG flag
  - Each DMAIFG flag is set in any mode, when the corresponding DMAxSZ register counts to zero. If the corresponding DMAIE and GIE bits are set, an interrupt request is generated



DMA and Other Devices

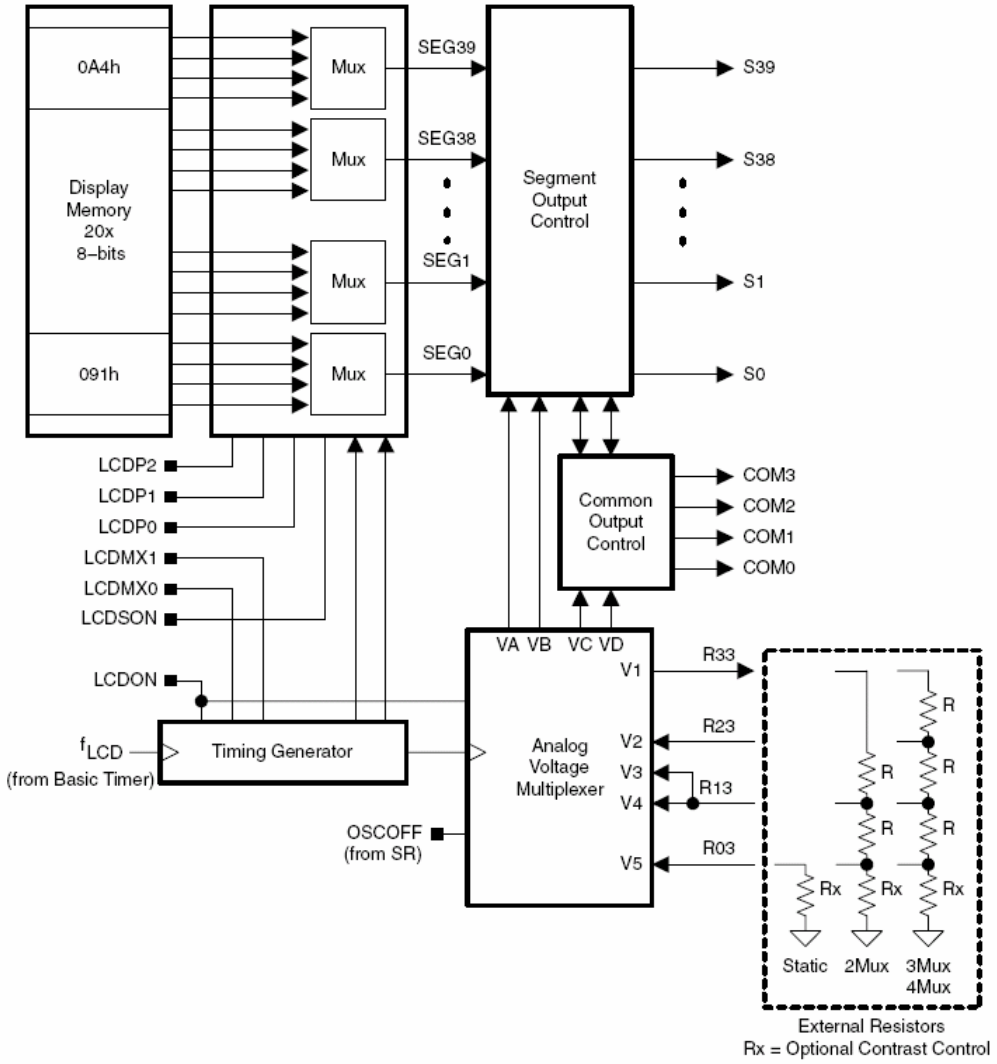
- USCI_B I2C Module
- ADC12
- DAC12
- Writing to Flash



LCD Controller

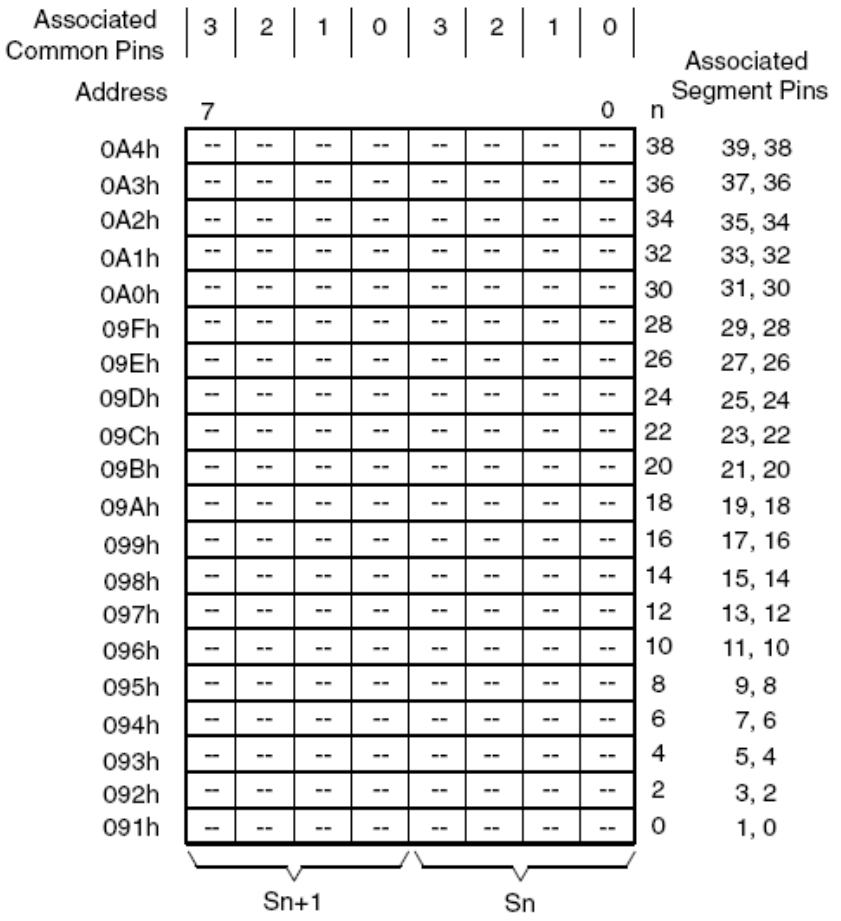
- Liquid Crystal Display (LCD) controller
  - Included in several devices of the MSP430 families (’3xx and ’4xx)
  - Allows a rapid and simple way to interface with the program
- LCD controller commands the LCD panels generating voltage signals to the segments. It supports static, and multiplex rates up to 4 (2 mux, 3 mux and 4 mux) LCD panels
- Features
  - Display memory
  - Automatic signal generation
  - Configurable frame frequency
  - Blinking capability
  - Support for 4 types of LCDs:
    - Static
    - 2-mux, 1/2 bias
    - 3-mux, 1/3 bias
    - 4-mux, 1/3 bias

LCD Controller Block Diagram



LCD Memory Map

- Each memory bit corresponds to one LCD segment, or is not used, depending on the mode.
- To turn on an LCD segment, its corresponding memory bit is set



Associated Common Pins	3	2	1	0	3	2	1	0		Associated Segment Pins
Address	7							0	n	
0A4h	--	--	--	--	--	--	--	--	38	39, 38
0A3h	--	--	--	--	--	--	--	--	36	37, 36
0A2h	--	--	--	--	--	--	--	--	34	35, 34
0A1h	--	--	--	--	--	--	--	--	32	33, 32
0A0h	--	--	--	--	--	--	--	--	30	31, 30
09Fh	--	--	--	--	--	--	--	--	28	29, 28
09Eh	--	--	--	--	--	--	--	--	26	27, 26
09Dh	--	--	--	--	--	--	--	--	24	25, 24
09Ch	--	--	--	--	--	--	--	--	22	23, 22
09Bh	--	--	--	--	--	--	--	--	20	21, 20
09Ah	--	--	--	--	--	--	--	--	18	19, 18
099h	--	--	--	--	--	--	--	--	16	17, 16
098h	--	--	--	--	--	--	--	--	14	15, 14
097h	--	--	--	--	--	--	--	--	12	13, 12
096h	--	--	--	--	--	--	--	--	10	11, 10
095h	--	--	--	--	--	--	--	--	8	9, 8
094h	--	--	--	--	--	--	--	--	6	7, 6
093h	--	--	--	--	--	--	--	--	4	5, 4
092h	--	--	--	--	--	--	--	--	2	3, 2
091h	--	--	--	--	--	--	--	--	0	1, 0

Sn+1 Sn

LCD Controller Operation

- LCD controller supports blinking
  - The LCDSON bit is ANDed with each segment's memory bit.
    - When LCDSON = 1, each segment is on or off according to its bit value
    - When LCDSON = 0, each LCD segment is off
- Timing generation
  - Uses the f_{LCD} signal from the Basic Timer1 to generate the timing for common and segment lines
    - Proper frequency f_{LCD} depends on the LCD's requirement for framing frequency and LCD multiplex rate.
    - See the *Basic Timer1* chapter for more information on configuring the f_{LCD} frequency

LCD Controller Operation

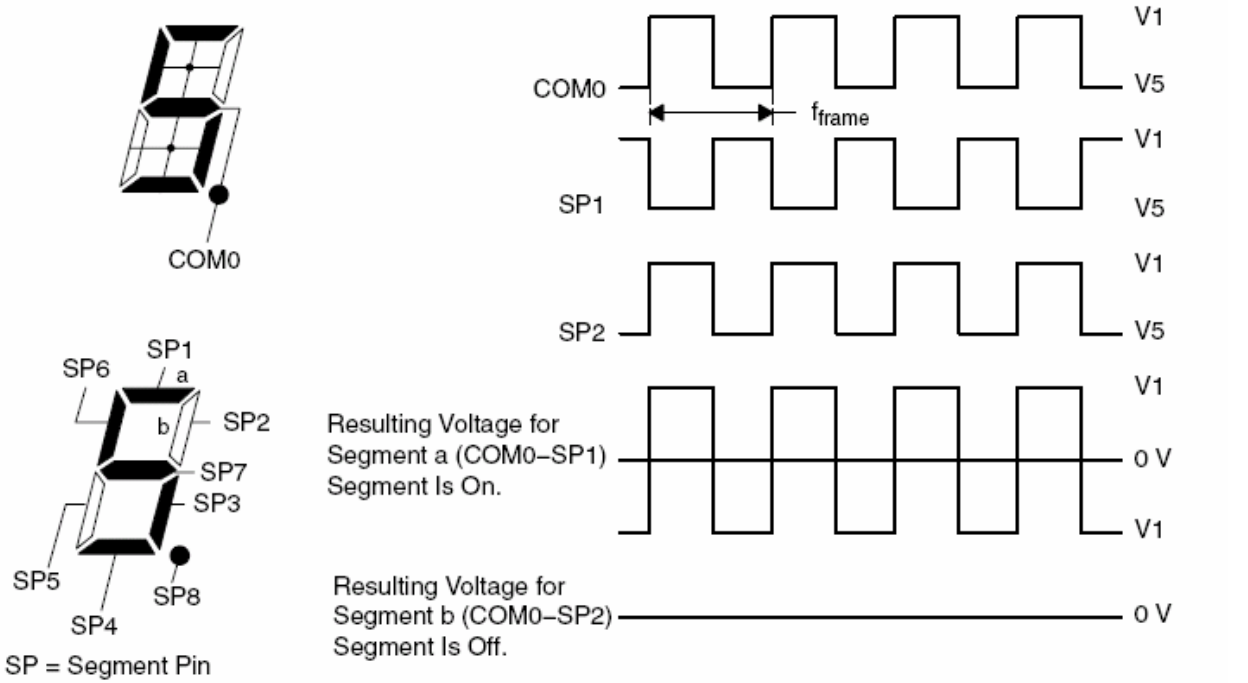
- LCD voltage generation
  - Voltages required for the LCD signals are supplied externally to pins R33, R23, R13, and R03
  - Using an equally weighted resistor divider ladder between these pins establishes the analog voltages as shown in Table 24−1
  - The resistor value R is typically 680 k
    - Values of R from 100k to 1M can be used depending on LCD requirements.
  - R33 is a switched-VCC output. This allows the power to the resistor ladder to be turned off eliminating current consumption when the LCD is not used.

Table 24−1. External LCD Module Analog Voltage

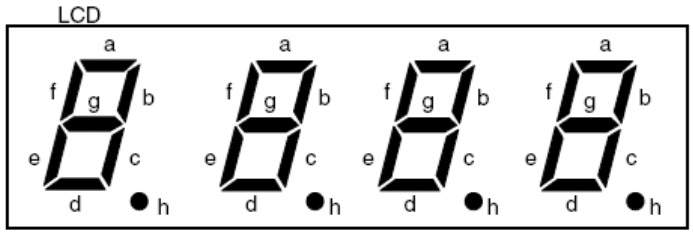
OSCOFF	LCDMXx	LCDON	VA	VB	VC	VD	R33
x	xx	0	0	0	0	0	Off
1	xx	x	0	0	0	0	Off
0	00	1	V5/V1	V1/V5	V5/V1	V1/V5	On
0	01	1	V5/V1	V1/V5	V3/V3	V1/V5	On
0	1x	1	V5/V1	V2/V4	V4/V2	V1/V5	On

Static Mode

- Each MSP430 segment pin drives one LCD segment
- One common line, COM0, is used.



Static LCD Example



Pinout and Connections

Connections

430 Pins		LCD Pinout PIN	COM0
S0	↔	1	1a
S1	↔	2	1b
S2	↔	3	1c
S3	↔	4	1d
S4	↔	5	1e
S5	↔	6	1f
S6	↔	7	1g
S7	↔	8	1h
S8	↔	9	2a
S9	↔	10	2b
S10	↔	11	2c
S11	↔	12	2d
S12	↔	13	2e
S13	↔	14	2f
S14	↔	15	2g
S15	↔	16	2h
S16	↔	17	3a
S17	↔	18	3b
S18	↔	19	3c
S19	↔	20	3d
S20	↔	21	3e
S21	↔	22	3f
S22	↔	23	3g
S23	↔	24	3h
S24	↔	25	4a
S25	↔	26	4b
S26	↔	27	4c
S27	↔	28	4d
S28	↔	29	4e
S29	↔	30	4f
S30	↔	31	4g
S31	↔	32	4h
COM0	↔	33	COM0
COM1	NC		
COM2	NC		
COM3	NC		

Display Memory

COM	3	2	1	0	3	2	1	0		
MAB 0A0h	--	--	--	h	--	--	--	g	n = 30	
09Fh	--	--	--	f	--	--	--	e	28	
09Eh	--	--	--	d	--	--	--	c	26	Digit 4
09Dh	--	--	--	b	--	--	--	a	24	
09Ch	--	--	--	h	--	--	--	g	22	
09Bh	--	--	--	f	--	--	--	e	20	Digit 3
09Ah	--	--	--	d	--	--	--	c	18	
099h	--	--	--	b	--	--	--	a	16	
098h	--	--	--	h	--	--	--	g	14	
097h	--	--	--	f	--	--	--	e	12	Digit 2
096h	--	--	--	d	--	--	--	c	10	
095h	--	--	--	b	--	--	--	a	8	
094h	--	--	--	h	--	--	--	g	6	
093h	--	--	--	f	--	--	--	e	4	Digit 1
092h	--	--	--	d	--	--	--	c	2	
091h	--	--	--	b	--	--	--	a	0	

Static Mode Software Example

```
;  All eight segments of a digit are often located in four
;  display memory bytes with the static display method.
;
a      EQU    001h
b      EQU    010h
c      EQU    002h
d      EQU    020h
e      EQU    004h
f      EQU    040h
g      EQU    008h
h      EQU    080h
;  The register content of Rx should be displayed.
:  The Table represents the 'on'-segments according to the
;  content of Rx.
       MOV.B  Table (Rx),RY    ; Load segment information
                               ; into temporary memory.
                               ; (Ry) = 0000 0000 hfdb geca
       MOV.B  Ry,&LCDn         ; Note:
                               ; All bits of an LCD memory
                               ' byte are written
       RRA    Ry               ; (Ry) = 0000 0000 0hfd bgec
       MOV.B  Ry,&LCDn+1       ; Note:
                               ; All bits of an LCD memory
                               ; byte are written
       RRA    Ry               ; (Ry) = 0000 0000 00hf dbge
       MOV.B  Ry,&LCDn+2       ; Note:
                               ; All bits of an LCD memory
                               ' byte are written
       RRA    Ry               ; (Ry) = 0000 0000 000h fdbg
       MOV.B  Ry,&LCDn+3       ; Note:
                               ; All bits of an LCD memory
                               ' byte are written

       ...........
       ...........
;
Table  DB     a+b+c+d+e+f      ; displays "0"
       DB     b+c;             ; displays "1"
       ...........
       ...........
       DB
       ...........
```

2-MUX Mode

- Each MSP430 segment pin drives two LCD segments
- Two common lines, COM0 and COM1, are used
- 2-mux example waveforms

a=COM1-SP1

b=COM1-SP2

c=COM1-SP3

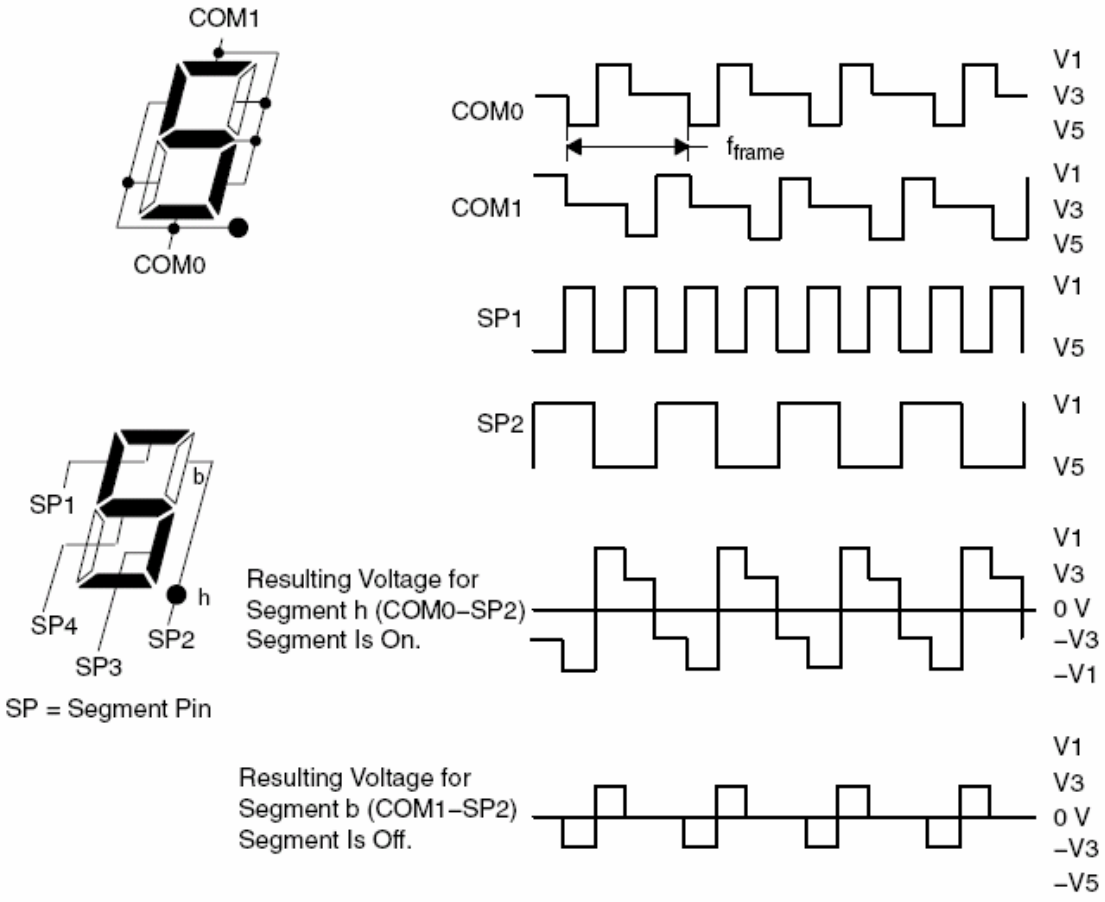
d=COM0-SP3

e=COM0-SP4

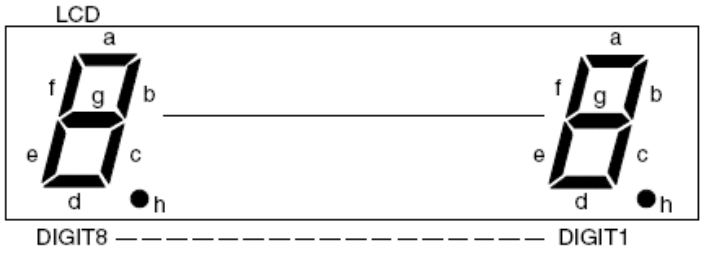
f=COM0-SP1

g=COM1-SP4

h=COM0-SP2



2-MUX LCD Example



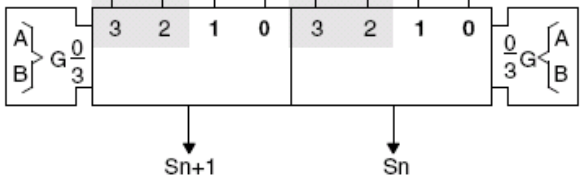
Pinout and Connections

Connections

'430 Pins		LCD Pinout PIN	COM0	COM1
S0	↔	1	1f	1a
S1	↔	2	1h	1b
S2	↔	3	1d	1c
S3	↔	4	1e	1g
S4	↔	5	2f	2a
S5	↔	6	2h	2b
S6	↔	7	2d	2c
S7	↔	8	2e	2g
S8	↔	9	3f	3a
S9	↔	10	3h	3b
S10	↔	11	3d	3c
S11	↔	12	3e	3g
S12	↔	13	4f	4a
S13	↔	14	4h	4b
S14	↔	15	4d	4c
S15	↔	16	4e	4g
S16	↔	17	5f	5a
S17	↔	18	5h	5b
S18	↔	19	5d	5c
S19	↔	20	5e	5g
S20	↔	21	6f	6a
S21	↔	22	6h	6b
S22	↔	23	6d	6c
S23	↔	24	6e	6g
S24	↔	25	7f	7a
S25	↔	26	7h	7b
S26	↔	27	7d	7c
S27	↔	28	7e	7g
S28	↔	29	8f	8a
S29	↔	30	8h	8b
S30	↔	31	8d	8c
S31	↔	32	8e	8g
COM0	↔	33	COM0	
COM1	↔	34		COM1
COM2	NC			
COM3	NC			

Display Memory

	COM	3	2	1	0	3	2	1	0		
MAB	0A0h	--	--	g	e	--	--	c	d	n = 30	1/2 Digit 8
	09Fh	--	--	b	h	--	--	a	f	28	
	09Eh	--	--	g	e	--	--	c	d	26	Digit 7
	09Dh	--	--	b	h	--	--	a	f	24	
	09Ch	--	--	g	e	--	--	c	d	22	Digit 6
	09Bh	--	--	b	h	--	--	a	f	20	
	09Ah	--	--	g	e	--	--	c	d	18	Digit 5
	099h	--	--	b	h	--	--	a	f	16	
	098h	--	--	g	e	--	--	c	d	14	Digit 4
	097h	--	--	b	h	--	--	a	f	12	
	096h	--	--	g	e	--	--	c	d	10	Digit 3
	095h	--	--	b	h	--	--	a	f	8	
	094h	--	--	g	e	--	--	c	d	6	Digit 2
	093h	--	--	b	h	--	--	a	f	4	
	092h	--	--	g	e	--	--	c	d	2	Digit 1
	091h	--	--	b	h	--	--	a	f	0	

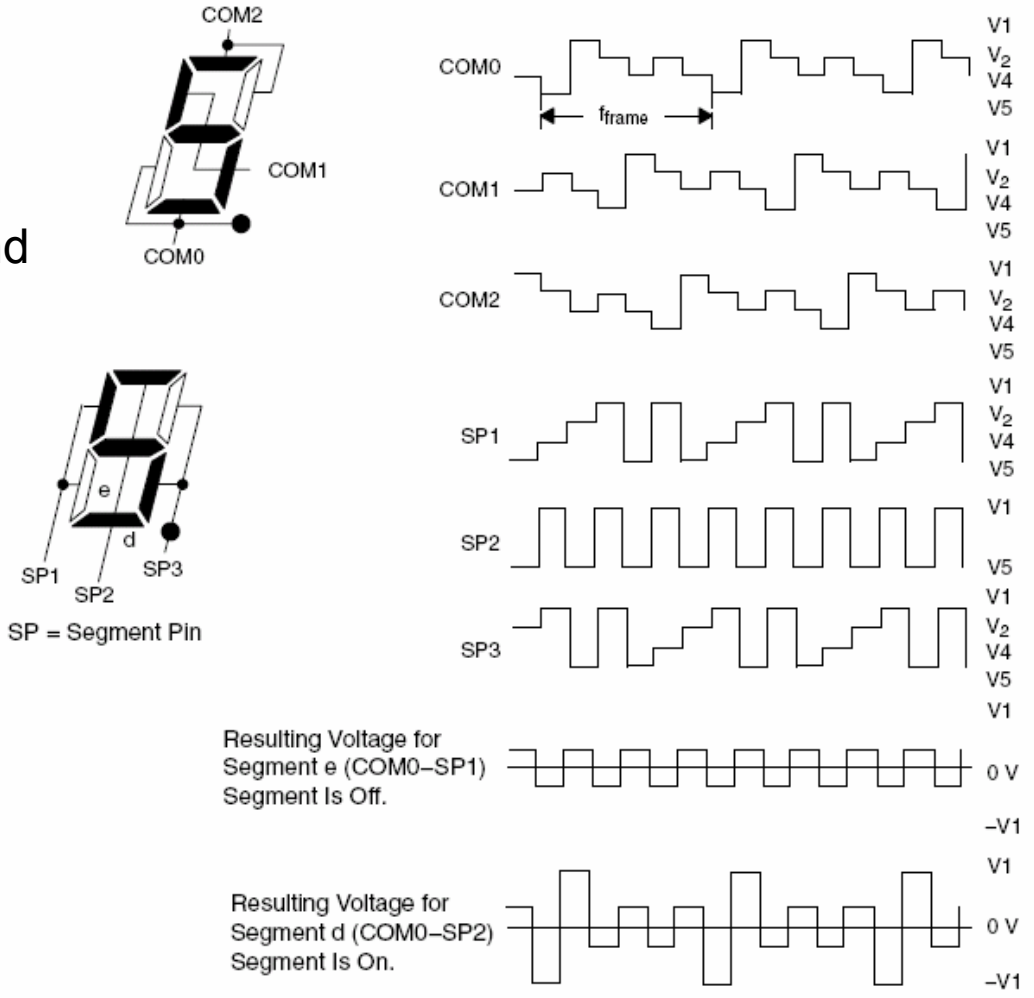


2-MUX Software Example

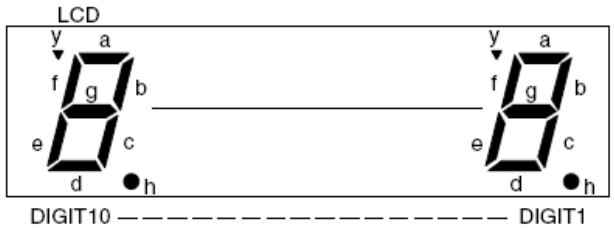
```
;  All eight segments of a digit are often located in two
;  display memory bytes with the 2mux display rate
;
a      EQU    002h
b      EQU    020h
c      EQU    008h
d      EQU    004h
e      EQU    040h
f      EQU    001h
g      EQU    080h
h      EQU    010h
;  The register content of Rx should be displayed.
;  The Table represents the 'on'-segments according to the
;  content of Rx.
;
       ...........
       ...........
       MOV.B  Table(Rx),Ry ; Load segment information into
                           ; temporary memory.
       MOV.B  Ry,&LCDn     ; (Ry) = 0000  0000  gebh  cdaf
                           ; Note:
                           ; All bits of an LCD memory byte
                           ; are written
       RRA    Ry           ; (Ry) = 0000  0000  0geb  hcda
       RRA    Ry           ; (Ry) = 0000  0000  00ge  bhcd
       MOV.B  Ry,&LCDn+1     ; Note:
                           ; All bits of an LCD memory byte
                           ; are written
       ...........
       ...........
       ...........
Table  DB     a+b+c+d+e+f     ; displays "0"
       ...........
       DB     a+b+c+d+e+f+g+  ; displays "8"
       ...........
       ...........
       DB
       ...........
;
```

3-MUX Mode Waveforms

- Each MSP430 segment pin drives three LCD segments
- Three common lines, COM0 and COM1, and COM2 are used
- 3-mux example waveforms



3-MUX LCD Example

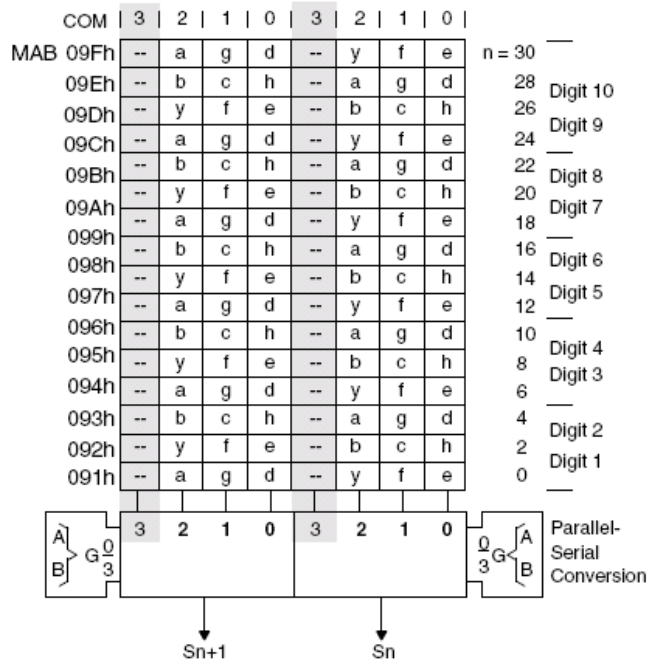


Pinout and Connections

Connections

'430 Pins		LCD Pinout			
		PIN	COM0	COM1	COM2
S0	↔	1	1e	1f	1y
S1	↔	2	1d	1g	1a
S2	↔	3	1h	1c	1b
S3	↔	4	2e	2f	2y
S4	↔	5	2d	2g	2a
S5	↔	6	2h	2c	2b
S6	↔	7	3e	3f	3y
S7	↔	8	3d	3g	3a
S8	↔	9	3h	3c	3b
S9	↔	10	4e	4f	4y
S10	↔	11	4d	4g	4a
S11	↔	12	4h	4c	4b
S12	↔	13	5e	5f	5y
S13	↔	14	5d	5g	5a
S14	↔	15	5h	5c	5b
S15	↔	16	6e	6f	6y
S16	↔	17	6d	6g	6a
S17	↔	18	6h	6c	6b
S18	↔	19	7e	7f	7y
S19	↔	20	7d	7g	7a
S20	↔	21	7h	7c	7b
S21	↔	22	8e	8f	8y
S22	↔	23	8d	8g	8a
S23	↔	24	8h	8c	8b
S24	↔	25	9e	9f	9y
S25	↔	26	9d	9g	9a
S26	↔	27	9h	9c	9b
S27	↔	28	10e	10f	10y
S28	↔	29	10d	10g	10a
S29	↔	30	10h	10c	10b
COM0	↔	31	COM0		
COM1	↔	32		COM1	
COM2	↔	33			COM2
COM3	NC				

Display Memory



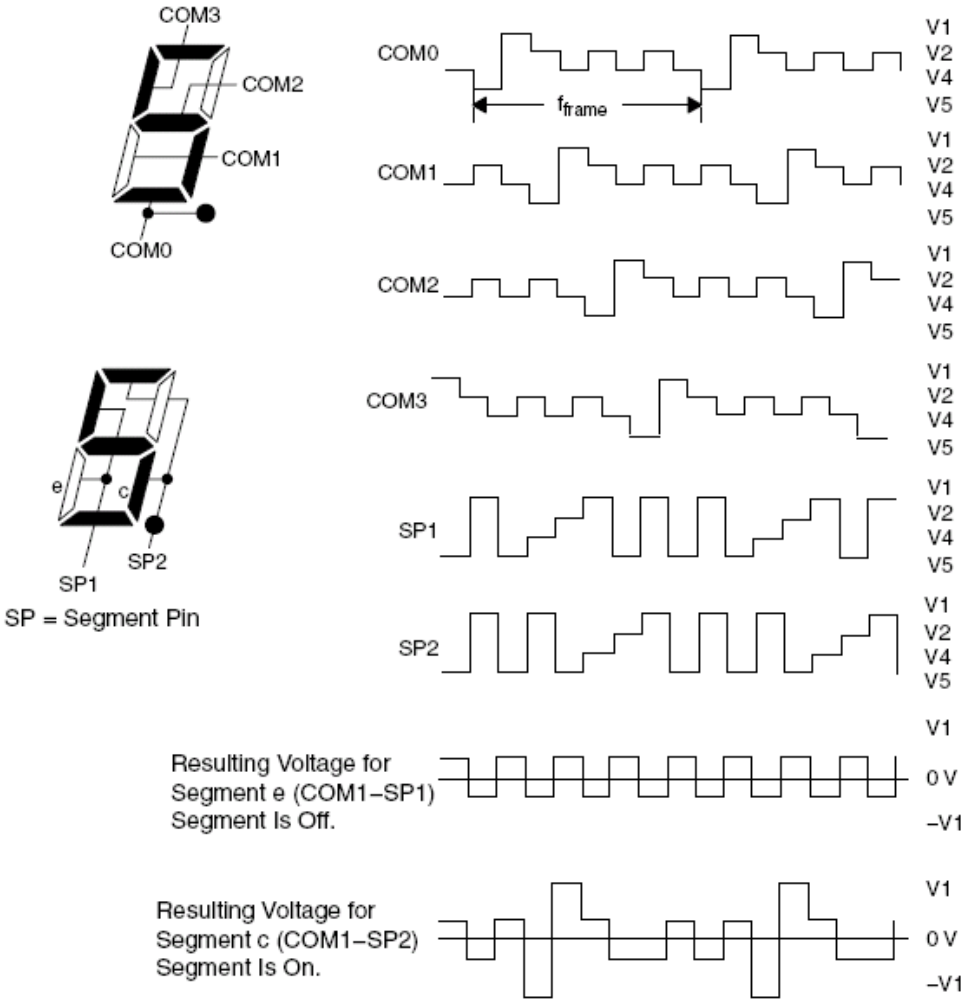
3-MUX Software Example

```
;  The 3mux rate can support nine segments for each
;  digit. The nine segments of a digit are located in
;  1 1/2 display memory bytes.
;
a       EQU    0040h
b       EQU    0400h
c       EQU    0200h
d       EQU    0010h
e       EQU    0001h
f       EQU    0002h
g       EQU    0020h
h       EQU    0100h
Y       EQU    0004h
;  The  LSDigit of register Rx should be displayed.
;  The Table represents the 'on'-segments according to the
;  LSDigit of register of Rx.
;  The register Ry is used for temporary memory
;
ODDDIGRLA    Rx             ; LCD in 3mux has 9 segments per
                            ; digit; word table required for
                            ; displayed characters.
       MOV   Table(Rx),Ry   ; Load segment information to
                            ; temporary mem.
                            ; (Ry) = 0000  0bch  0agd  0yfe
       MOV.B Ry,&LCDn       ; write 'a, g, d, y, f, e' of
                            ; Digit n (LowByte)
       SWPB  Ry             ; (Ry) =  0agd  0yfe  0000  0bch
       BIC.B #07h,&LCDn+1   ; write 'b, c, h' of Digit n
                            ; (HighByte)
       BIS.B Ry,&LCD_n+1
       .....
EVNDIGRLA    Rx             ; LCD in 3mux has 9 segments per
                            ; digit; word table required for
                            ; displayed characters.
       MOV   Table(Rx),Ry   ; Load segment information to
                            ; temporary mem.
                            ; (Ry) = 0000  0bch  0agd  0yfe
       RLA   Ry             ; (Ry) = 0000  bch0  agd0  yfe0
       RLA   Ry             ; (Ry) = 000b  ch0a  gd0y  fe00
       RLA   Ry             ; (Ry) = 00bc  h0ag  d0yf  e000
       RLA   Ry             ; (Ry) = 0bch  0agd  0yfe  0000
       BIC.B #070h,&LCD_n+1
       BIS.B Ry,&LCD_n+1    ; write 'y, f, e' of Digit n+1
                            ; (LowByte)
       SWPB  Ry             ; (Ry) = 0yfe  0000  0bch  0agd
       MOV.B Ry,&LCD_n+2    ; write 'b, c, h, a, g, d' of
                            ; Digit n+1 (HighByte)
       ...........
Table  DW    a+b+c+d+e+f    ; displays "0"
       DW    b+c            ; displays "1"
       ...........
       ...........
       DW    a+e+f+g        ; displays "F"
```

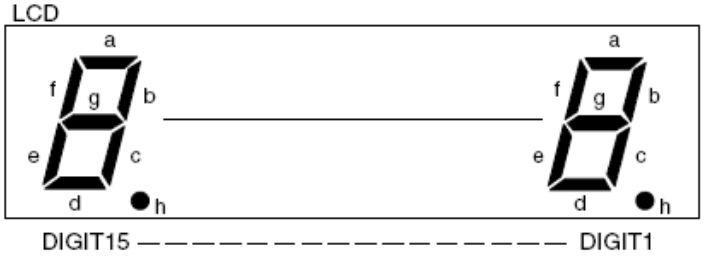
C

4-MUX Mode Waveforms

- Each MSP430 segment pin drives four LCD segments
- Four common lines, COM0, COM1, COM2, and COM3 are used
- 4-mux example waveforms



4-MUX LCD Example



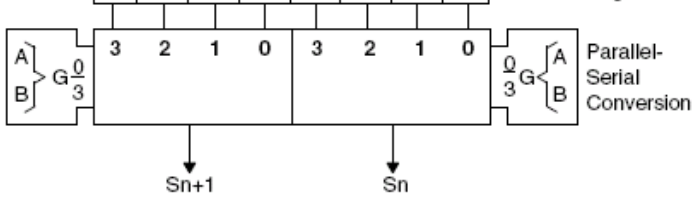
Pinout and Connections

Connections

'430 Pins		LCD Pinout PIN	COM0	COM1	COM2	COM3
S0	↔	1	1d	1e	1g	1f
S1	↔	2	1h	1c	1b	1a
S2	↔	3	2d	2e	2g	2f
S3	↔	4	2h	2c	2b	2a
S4	↔	5	3d	3e	3g	3f
S5	↔	6	3h	3c	3b	3a
S6	↔	7	4d	4e	4g	4f
S7	↔	8	4h	4c	4b	4a
S8	↔	9	5d	5e	5g	5f
S9	↔	10	5h	5c	5b	5a
S10	↔	11	6d	6e	6g	6f
S11	↔	12	6h	6c	6b	6a
S12	↔	13	7d	7e	7g	7f
S13	↔	14	7h	7c	7b	7a
S14	↔	15	8d	8e	8g	8f
S15	↔	16	8h	8c	8b	8a
S16	↔	17	9d	9e	9g	9f
S17	↔	18	9h	9c	9b	9a
S18	↔	19	10d	10e	10g	10f
S19	↔	20	10h	10c	10b	10a
S20	↔	21	11d	11e	11g	11f
S21	↔	22	11h	11c	11b	11a
S22	↔	23	12d	12e	12g	12f
S23	↔	24	12h	12c	12b	12a
S24	↔	25	13d	13e	13g	13f
S25	↔	26	13h	13c	13b	13a
S26	↔	27	14d	14e	14g	14f
S27	↔	28	14h	14c	14b	14a
S28	↔	29	15d	15e	15g	15f
S29	↔	30	15h	15c	15b	15a
COM0	↔	31	COM0			
COM1	↔	32		COM1		
COM2	↔	33			COM2	
COM3	↔	34				COM3

Display Memory

COM	3	2	1	0	3	2	1	0	
MAB 09Fh	a	b	c	h	f	g	e	d	n = 30 Digit 16
09Eh	a	b	c	h	f	g	e	d	28 Digit 15
09Dh	a	b	c	h	f	g	e	d	26 Digit 14
09Ch	a	b	c	h	f	g	e	d	24 Digit 13
09Bh	a	b	c	h	f	g	e	d	22 Digit 12
09Ah	a	b	c	h	f	g	e	d	20 Digit 11
099h	a	b	c	h	f	g	e	d	18 Digit 10
098h	a	b	c	h	f	g	e	d	16 Digit 9
097h	a	b	c	h	f	g	e	d	14 Digit 8
096h	a	b	c	h	f	g	e	d	12 Digit 7
095h	a	b	c	h	f	g	e	d	10 Digit 6
094h	a	b	c	h	f	g	e	d	8 Digit 5
093h	a	b	c	h	f	g	e	d	6 Digit 4
092h	a	b	c	h	f	g	e	d	4 Digit 3
091h	a	b	c	h	f	g	e	d	2 Digit 2
									0 Digit 1



4-MUX Software Example

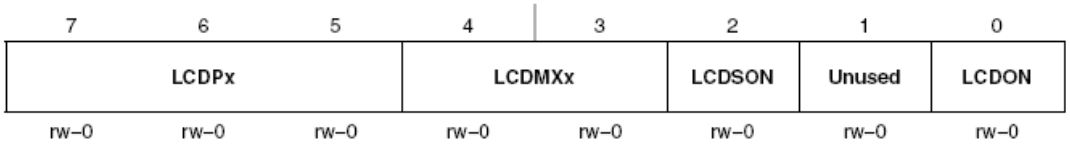
```
;  The 4mux rate supports eight segments for each digit.
;  All eight segments of a digit can often be located in
;  one display memory byte
a      EQU    080h
b      EQU    040h
c      EQU    020h
d      EQU    001h
e      EQU    002h
f      EQU    008h
g      EQU    004h
h      EQU    010h
;
;  The  LSDigit of register Rx should be displayed.
;  The Table represents the 'on'-segments according to the
;  content of Rx.
;
       MOV.B  Table(Rx),&LCDn ; n = 1 ..... 15
                              ; all eight segments are
                              ; written to the display
                              ; memory
       ...........
       ...........

Table  DB     a+b+c+d+e+f     ; displays "0"
       DB     b+c             ; displays "1"
       ...........
       ...........
       DB     b+c+d+e+g       ; displays "d"
       DB     a+d+e+f+g       ; displays "E"
       DB     a+e+f+g         ; displays "F"
```

LCD Control Registers

Register	Short Form	Register Type	Address	Initial State
LCD control register	LCDCTL	Read/write	090h	Reset with PUC
LCD memory 1	LCDM1	Read/write	091h	Unchanged
LCD memory 2	LCDM2	Read/write	092h	Unchanged
LCD memory 3	LCDM3	Read/write	093h	Unchanged
LCD memory 4	LCDM4	Read/write	094h	Unchanged
LCD memory 5	LCDM5	Read/write	095h	Unchanged
LCD memory 6	LCDM6	Read/write	096h	Unchanged
LCD memory 7	LCDM7	Read/write	097h	Unchanged
LCD memory 8	LCDM8	Read/write	098h	Unchanged
LCD memory 9	LCDM9	Read/write	099h	Unchanged
LCD memory 10	LCDM10	Read/write	09Ah	Unchanged
LCD memory 11	LCDM11	Read/write	09Bh	Unchanged
LCD memory 12	LCDM12	Read/write	09Ch	Unchanged
LCD memory 13	LCDM13	Read/write	09Dh	Unchanged
LCD memory 14	LCDM14	Read/write	09Eh	Unchanged
LCD memory 15	LCDM15	Read/write	09Fh	Unchanged
LCD memory 16	LCDM16	Read/write	0A0h	Unchanged
LCD memory 17	LCDM17	Read/write	0A1h	Unchanged
LCD memory 18	LCDM18	Read/write	0A2h	Unchanged
LCD memory 19	LCDM19	Read/write	0A3h	Unchanged
LCD memory 20	LCDM20	Read/write	0A4h	Unchanged

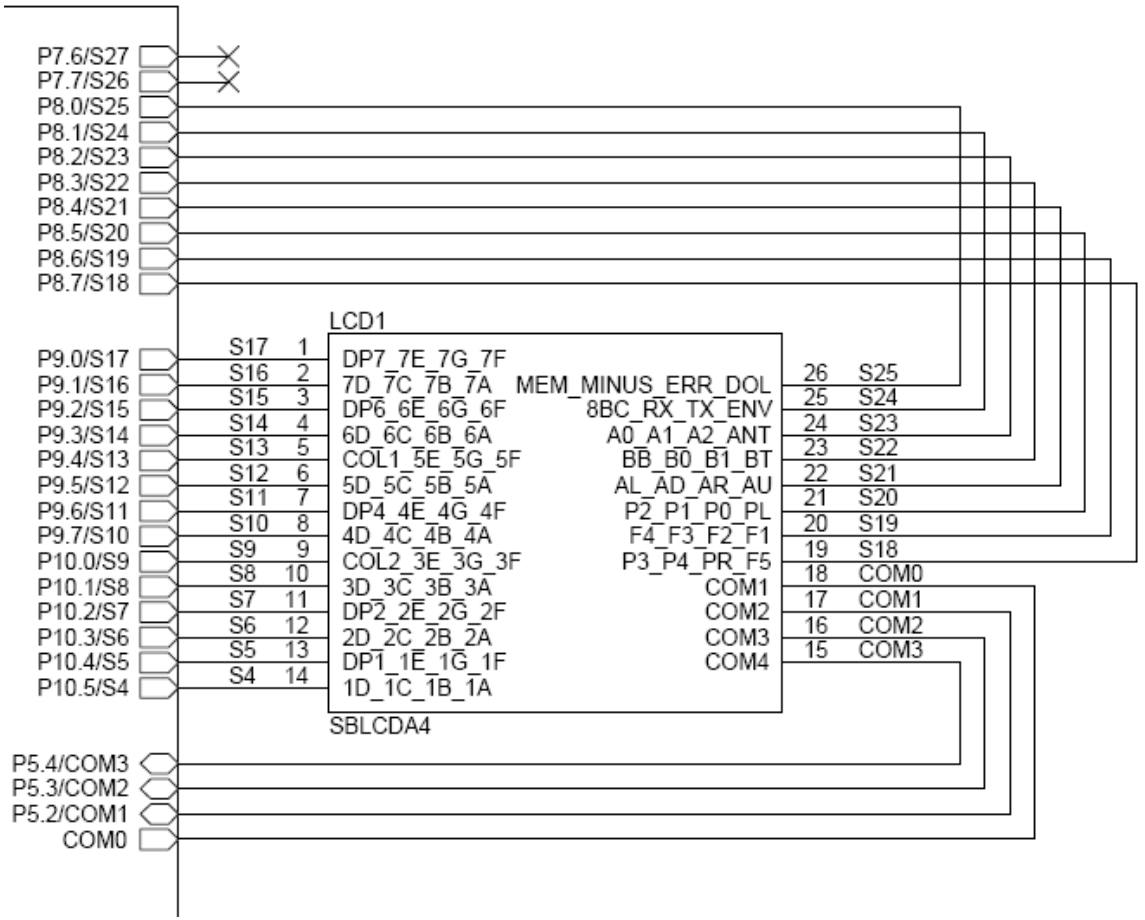
LCD Control Register



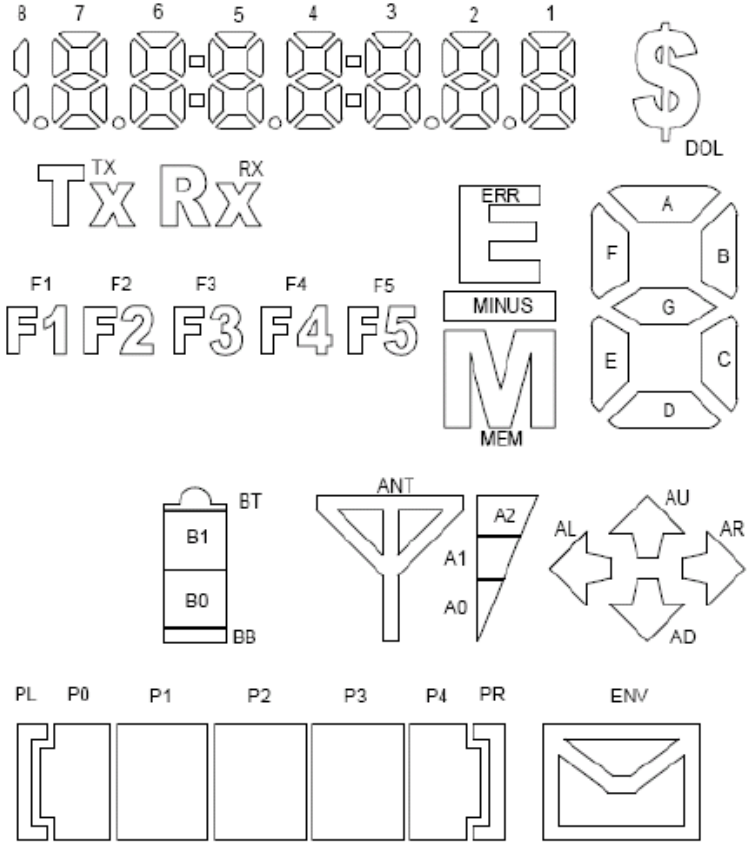
7	6	5	4	3	2	1	0
LCDPx			LCDMXx		LCDSON	Unused	LCDON
rw–0	rw–0	rw–0	rw–0	rw–0	rw–0	rw–0	rw–0

LCDPx	Bits 7-5	LCD Port Select. These bits select the pin function to be port I/O or LCD function for groups of segments pins. These bits ONLY affect pins with multiplexed functions. Dedicated LCD pins are always LCD function. 000 No multiplexed pins are LCD function 001 S0-S15 are LCD function 010 S0-S19 are LCD function 011 S0-S23 are LCD function 100 S0-S27 are LCD function 101 S0-S31 are LCD function 110 S0-S35 are LCD function 111 S0-S39 are LCD function
LCDMXx	Bits 4-3	LCD mux rate. These bits select the LCD mode. 00 Static 01 2-mux 10 3-mux 11 4-mux
LCDSON	Bit 2	LCD segments on. This bit supports flashing LCD applications by turning off all segment lines, while leaving the LCD timing generator and R33 enabled. 0 All LCD segments are off 1 All LCD segments are enabled and on or off according to their corresponding memory location.
Unused	Bit 1	Unused
LCDON	Bit 0	LCD On. This bit turns on the LCD timing generator and R33. 0 LCD timing generator and Ron are off 1 LCD timing generator and Ron are on

DRFG4618 LCD Interface



Softbaugh LCD SBLCDA4: Segment Description



SBLCDA4 Display