THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

CPE 323: DMA Controller

Aleksandar Milenkovic

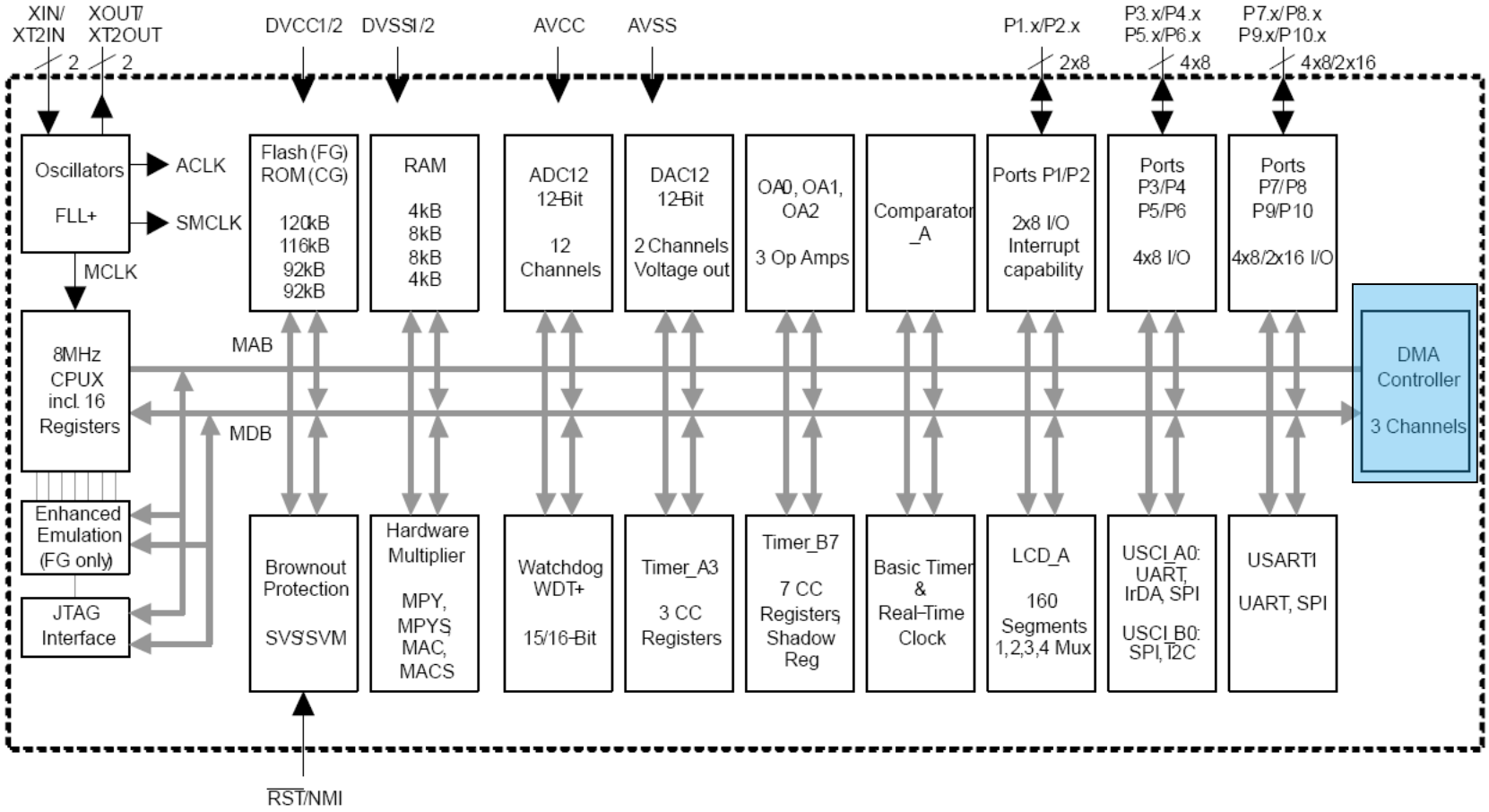
Electrical and Computer Engineering
The University of Alabama in Huntsville

milenka@ece.uah.edu

http://www.ece.uah.edu/~milenka

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

System View: MSP430FG461x



LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

I/O Interfacing: A Review

- Three principal software approaches to interfacing peripheral I/O devices
  - Polling
  - Interrupts
  - DMA Transfers
- Polling
  - Check the status bit (e.g., is USCI.RX.IFG bit set – new character is received); If it is set, read the data from RXBUF; If it is not set, read the status bit again;
    - CPU does polling continually (long waits, no useful work done)
- Interrupts
  - When the status bit is set, ISR is requested
    - CPU does the transfer when needed
- DMA
  - Transfer takes place when everything is ready with no CPU intervention

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMA Controller Introduction

- Direct memory access (DMA) controller transfers data from one address to another without CPU intervention, across the entire address range
- Examples
  - Move data from the ADC12 conversion memory to RAM, move data from RAM to DAC12, move a message to USCI, receive a message from USCI
- Devices that contain a DMA controller may have one, two, or three DMA channels available
- Benefits of 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

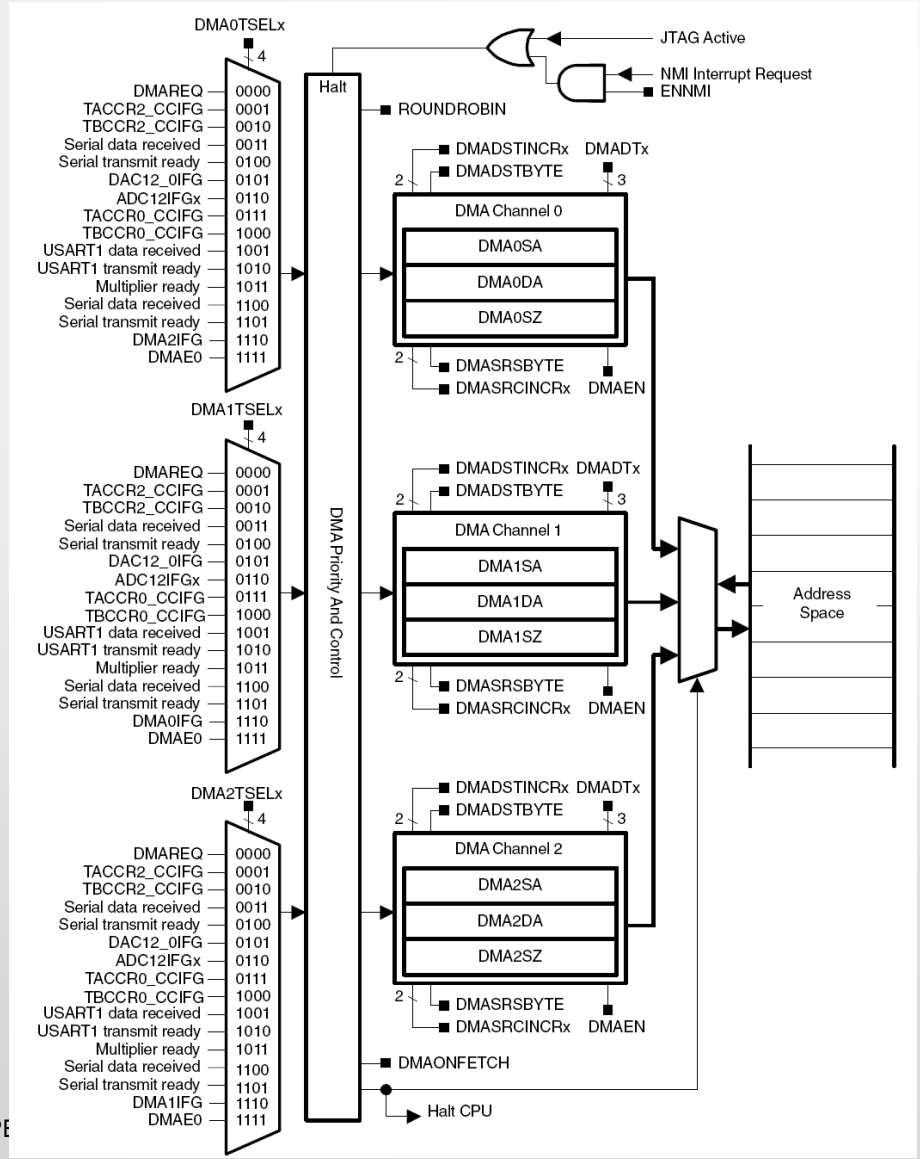
LaCASA

MSP430 DMA Controller 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

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMA Block Diagram



LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMA Operation

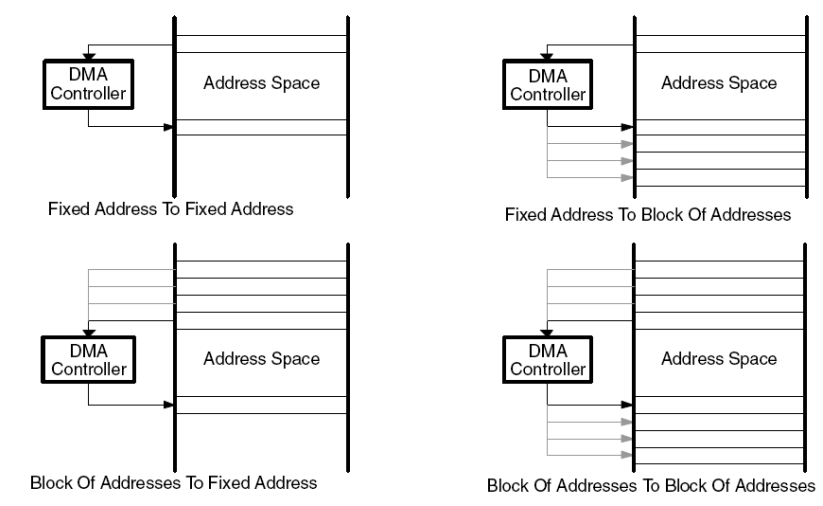
- 3 Channels (DMA0, DMA1, DMA2) for independent transfers
- Initialize block of data transfer from software, carry it out in hardware
- DMA Registers
  - Starting Address (SA)
  - Destination Address (DA)
  - Block Size (SZ)

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMA Addressing Modes

- Configured with the DMASRCINCRx and DMADSTINCRx control bits
  - Select if the source/destination address is incremented, decremented, or unchanged after each transfer
- Four transfer modes
  - Fixed address to fixed address (e.g., comm2comm)
  - Fixed address to block of addresses (e.g. comm2mem)
  - Block of addresses to fixed address (e.g., mem2comm)
  - Block of addresses to block of addresses (mem2mem)



- 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

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

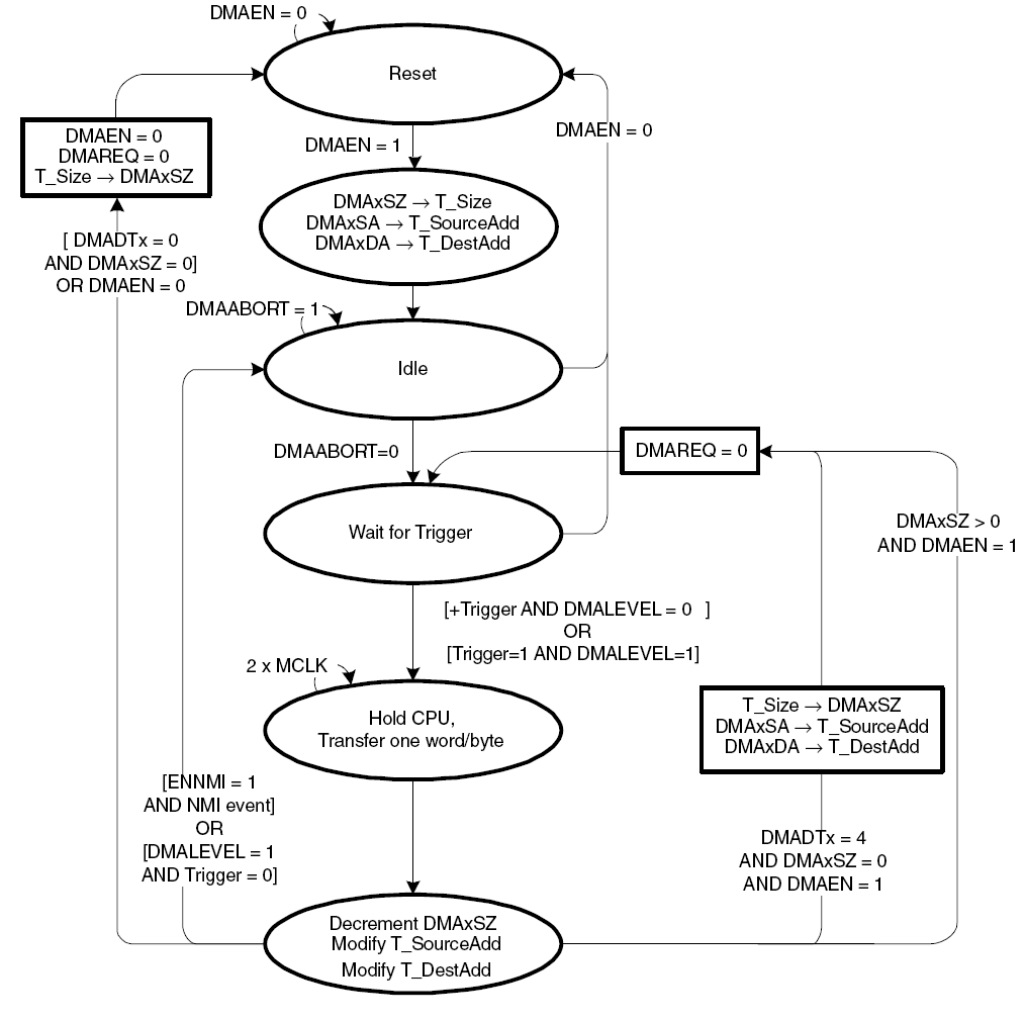
DMA Transfer Modes

- 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.

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

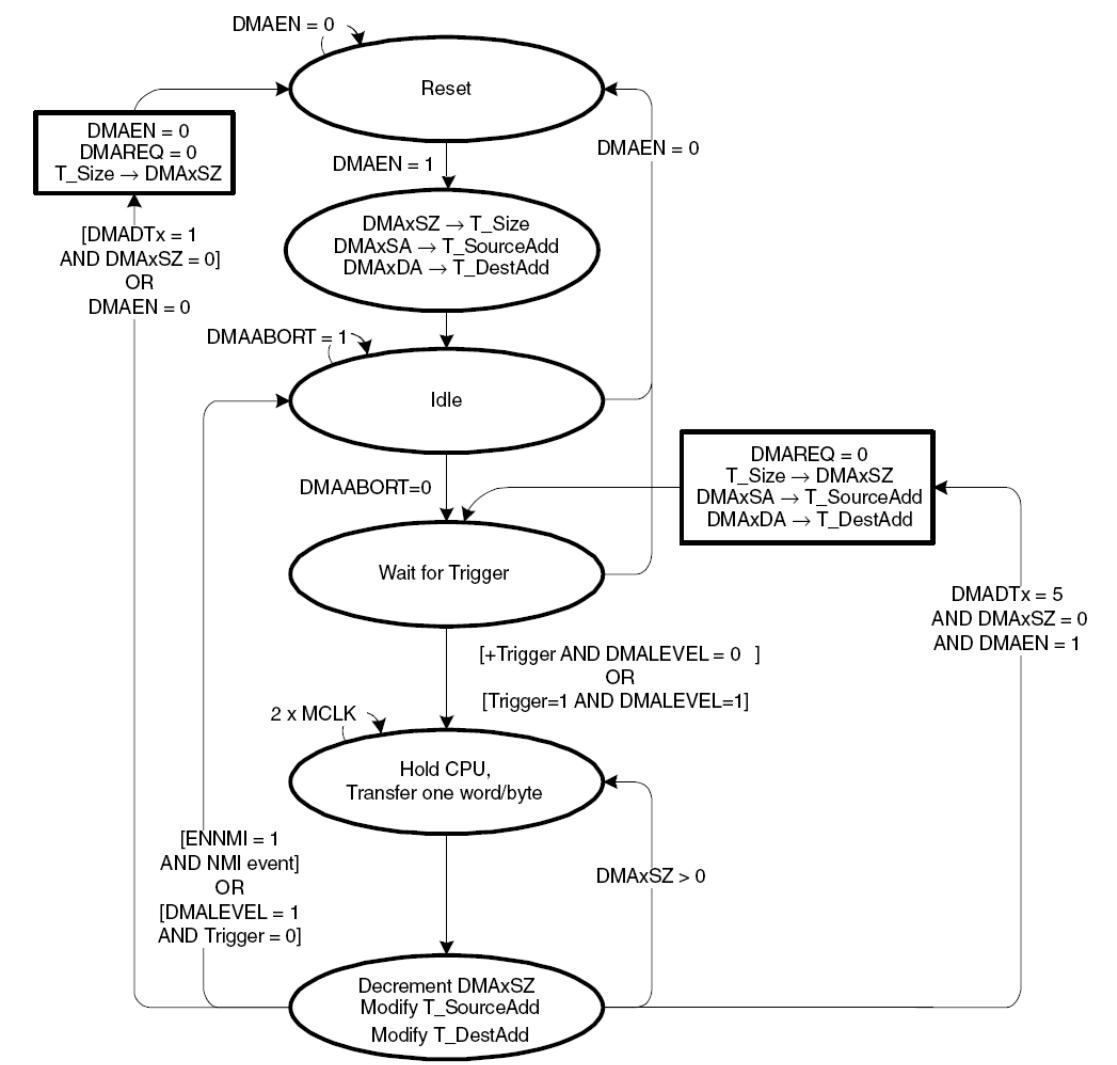
DMA Single Transfer



LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMA Block Transfer



LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMA Trigger Operation

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

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.

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMA Trigger Operation (cont'd)

0101	Devices with DAC12: 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	Devices with ADC12: 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. Devices with SD16 or SD16_A: A transfer is triggered by the SD16IFG flag of the master channel in grouped mode or of channel 0. Setting the SD16IFG with software will not trigger a transfer. All SD16IFG flags are automatically reset when the associated SD16MEMx register is accessed by the DMA controller. If the SD16IE of the master channel is set, the SD16IFG will not trigger a transfer.
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	Devices with USART1: 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. Devices with USCI_A1: A transfer is triggered when the UCA1RXIFG flag is set. UCA1RXIFG is automatically reset when the transfer starts. If UCA1RXIE is set, the UCA1RXIFG flag will not trigger a transfer.

LaCASA

DMA Trigger Operation (cont'd)

DMAxTSELx	Operation
1010	Devices with USART1: 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. Devices with USCI_A1: A transfer is triggered when the UCA1TXIFG flag is set. UCA1TXIFG is automatically reset when the transfer starts. If UCA1TXIE is set, the UCA1TXIFG 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.

Stopping DMA Transfers

- Two ways to stop DMA transfers in progress:
  - A single, block, or burst-block transfer may be stopped with an NMI interrupt, if the ENNMI bit is set in register DMACTL1
  - A burst-block transfer may be stopped by clearing the DMAEN bit

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

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

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

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.

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

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

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMA Interrupts

- 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
- All DMAIFG flags source only one DMA controller interrupt vector and the interrupt vector may be shared with the other modules
  - software must check the DMAIFG and other flags to determine the source of the interrupt
  - The DMAIFG flags are not reset automatically and must be reset by software

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMAIV Register

```
;Interrupt handler for DMA0IFG, DMA1IFG, DMA2IFG    Cycles
DMA_HND     ...             ; Interrupt latency          6
          ADD    &DMAIV,PC ; Add offset to Jump table    3
          RETI             ; Vector  0: No interrupt
5
          JMP    DMA0_HND  ; Vector  2: DMA channel 0    2
          JMP    DMA1_HND  ; Vector  4: DMA channel 1    2
          JMP    DMA2_HND  ; Vector  6: DMA channel 2    2
          RETI             ; Vector  8: Reserved         5
          RETI             ; Vector 10: Reserved         5
          RETI             ; Vector 12: Reserved         5
          RETI             ; Vector 14: Reserved         5

DMA2_HND                   ; Vector 6: DMA channel 2
          ...              ; Task starts here
          RETI             ; Back to main program       5

DMA1_HND                   ; Vector 4: DMA channel 1
          ...              ; Task starts here
          RETI             ; Back to main program       5

DMA0_HND                   ; Vector 2: DMA channel 0
          ...              ; Task starts here
          RETI             ; Back to main program       5
```


LaCASA

DMA and ADC12

- DMA can automatically move data from any ADC12MEMx register to another location
  - No CPU intervention, independently from LPMs
  - => increases throughput of the ADC12 module, and saves energy
- DMA transfers can be triggered from any ADC12IFGx flag
  - When CONSEQx = {0,2} the ADC12IFGx flag for the ADC12MEMx used for the conversion can trigger a DMA transfer
  - When CONSEQx = {1,3}, the ADC12IFGx flag for the last ADC12MEMx in the sequence can trigger a DMA transfer
  - Any ADC12IFGx flag is automatically cleared when the DMA controller accesses the corresponding ADC12MEMx

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMA Registers

Table 10–5. DMA Registers, MSP430FG461x, MSP430F471xx devices

Register	Short Form	Register Type	Address	Initial State
DMA control 0	DMACTL0	Read/write	0122h	Reset with POR
DMA control 1	DMACTL1	Read/write	0124h	Reset with POR
DMA interrupt vector	DMAIV	Read only	0126h	Reset with POR
DMA channel 0 control	DMA0CTL	Read/write	01D0h	Reset with POR
DMA channel 0 source address	DMA0SA	Read/write	01D2h	Unchanged
DMA channel 0 destination address	DMA0DA	Read/write	01D6h	Unchanged
DMA channel 0 transfer size	DMA0SZ	Read/write	01DAh	Unchanged
DMA channel 1 control	DMA1CTL	Read/write	01DCh	Reset with POR
DMA channel 1 source address	DMA1SA	Read/write	01DEh	Unchanged
DMA channel 1 destination address	DMA1DA	Read/write	01E2h	Unchanged
DMA channel 1 transfer size	DMA1SZ	Read/write	01E6h	Unchanged
DMA channel 2 control	DMA2CTL	Read/write	01E8h	Reset with POR
DMA channel 2 source address	DMA2SA	Read/write	01EAh	Unchanged
DMA channel 2 destination address	DMA2DA	Read/write	01EEh	Unchanged
DMA–channel 2 transfer size	DMA2SZ	Read/write	01F2h	Unchanged

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMACTL0

DMACTL0, DMA Control Register 0

15	14	13	12	11	10	9	8
Reserved				DMA2TSELx			
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)

7	6	5	4	3	2	1	0
DMA1TSELx				DMA0TSELx			
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)

Reserved Bits 15–12 Reserved

DMA2TSELx Bits 11–8 DMA trigger select. These bits select the DMA transfer trigger. The trigger selection is device-specific. For MSP430FG43x and MSP430FG461x devices it is given below; for other devices, see the device-specific data sheet.

0000 DMAREQ bit (software trigger)
0001 TACCR2 CCIFG bit
0010 TBCCR2 CCIFG bit
0011 URXIFG0 (MSP430FG43x), UCA0RXIFG (MPS430FG461x)
0100 UTXIFG0 (MSP430FG43x), UCA0TXIFG (MSP430FG461x)
0101 DAC12_0CTL DAC12IFG bit
0110 ADC12 ADC12IFGx bit
0111 TACCR0 CCIFG bit
1000 TBCCR0 CCIFG bit
1001 URXIFG1 bit
1010 UTXIFG1 bit
1011 Multiplier ready
1100 No action (MSP430FG43x), UCB0RXIFG (MSP430FG461x)
1101 No action (MSP430FG43x), UCB0TXIFG (MSP430FG461x)
1110 DMA0IFG bit triggers DMA channel 1
DMA1IFG bit triggers DMA channel 2
DMA2IFG bit triggers DMA channel 0
1111 External trigger DMAE0

DMA1TSELx Bits 7–4 Same as DMA2TSELx

DMA0TSELx Bits 3–0 Same as DMA2TSELx

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMACTL1

DMACTL1, DMA Control Register 1

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
r0	r0	r0	r0	r0	r0	r0	r0

7	6	5	4	3	2	1	0
0	0	0	0	0	DMA ONFETCH	ROUND ROBIN	ENNMI
r0	r0	r0	r0	r0	rw–(0)	rw–(0)	rw–(0)

Reserved	Bits 15–3	Reserved. Read only. Always read as 0.
DMA ONFETCH	Bit 2	DMA on fetch 0 The DMA transfer occurs immediately 1 The DMA transfer occurs on next instruction fetch after the trigger
ROUND ROBIN	Bit 1	Round robin. This bit enables the round-robin DMA channel priorities. 0 DMA channel priority is DMA0 – DMA1 – DMA2 1 DMA channel priority changes with each transfer
ENNMI	Bit 0	Enable NMI. This bit enables the interruption of a DMA transfer by an NMI interrupt. When an NMI interrupts a DMA transfer, the current transfer is completed normally, further transfers are stopped, and DMAABORT is set. 0 NMI interrupt does not interrupt DMA transfer 1 NMI interrupt interrupts a DMA transfer

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMAxCTL

DMAxCTL, DMA Channel x Control Register

15	14	13	12	11	10	9	8
Reserved	DMADTx			DMADSTINCRx		DMASRCINCRx	
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)

7	6	5	4	3	2	1	0
DMA DSTBYTE	DMA SRCBYTE	DMALEVEL	DMAEN	DMAIFG	DMAIE	DMA ABORT	DMAREQ
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)

Reserved Bit 15 Reserved

DMADTx Bits 14–12 DMA Transfer mode.
000 Single transfer
001 Block transfer
010 Burst-block transfer
011 Burst-block transfer
100 Repeated single transfer
101 Repeated block transfer
110 Repeated burst-block transfer
111 Repeated burst-block transfer

DMA DSTINCRx Bits 11–10 DMA destination increment. This bit selects automatic incrementing or decrementing of the destination address after each byte or word transfer. When DMADSTBYTE=1, the destination address increments/decrements by one. When DMADSTBYTE=0, the destination address increments/decrements by two. The DMAxDA is copied into a temporary register and the temporary register is incremented or decremented. DMAxDA is not incremented or decremented.
00 Destination address is unchanged
01 Destination address is unchanged
10 Destination address is decremented
11 Destination address is incremented

DMA SRCINCRx Bits 9–8 DMA source increment. This bit selects automatic incrementing or decrementing of the source address for each byte or word transfer. When DMASRCBYTE=1, the source address increments/decrements by one. When DMASRCBYTE=0, the source address increments/decrements by two. The DMAxSA is copied into a temporary register and the temporary register is incremented or decremented. DMAxSA is not incremented or decremented.
00 Source address is unchanged
01 Source address is unchanged
10 Source address is decremented
11 Source address is incremented

DMA DSTBYTE Bit 7 DMA destination byte. This bit selects the destination as a byte or word.
0 Word
1 Byte

DMA SRCBYTE Bit 6 DMA source byte. This bit selects the source as a byte or word.
0 Word
1 Byte

DMA LEVEL Bit 5 DMA level. This bit selects between edge-sensitive and level-sensitive triggers.
0 Edge sensitive (rising edge)
1 Level sensitive (high level)

DMAEN Bit 4 DMA enable
0 Disabled
1 Enabled

DMAIFG Bit 3 DMA interrupt flag
0 No interrupt pending
1 Interrupt pending

DMAIE Bit 2 DMA interrupt enable
0 Disabled
1 Enabled

DMA ABORT Bit 1 DMA Abort. This bit indicates if a DMA transfer was interrupt by an NMI.
0 DMA transfer not interrupted
1 DMA transfer was interrupted by NMI

DMAREQ Bit 0 DMA request. Software-controlled DMA start. DMAREQ is reset automatically.
0 No DMA start
1 Start DMA

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMAxSA

DMAxSA, DMA Source Address Register

31	30	29	28	27	26	25	824
Reserved							
r0	r0	r0	r0	r0	r0	r0	r0

23	22	21	20	19	18	17	16
Reserved				DMAxSAx			
r0	r0	r0	r0	rw	rw	rw	rw

15	14	13	12	11	10	9	8
DMAxSAx							
rw	rw	rw	rw	rw	rw	rw	rw

7	6	5	4	3	2	1	0
DMAxSAx							
rw	rw	rw	rw	rw	rw	rw	rw

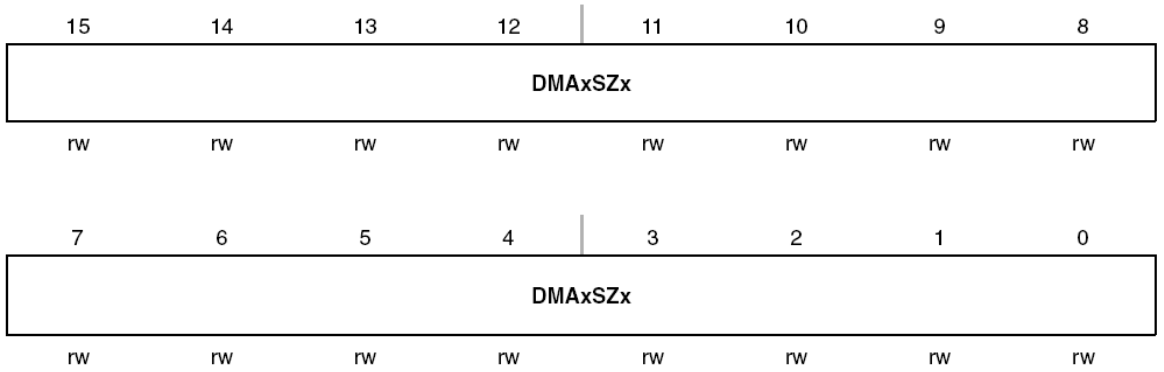
Reserved	Bits 31–20	Reserved
DMAxSAx	Bits 19–0	DMA source address. The source address register points to the DMA source address for single transfers or the first source address for block transfers. The source address register remains unchanged during block and burst-block transfers. Devices that have addressable memory range 64–KB or below contain a single word for the DMAxSA. MSP430FG461x and MSP430F471xx devices implement two words for the DMAxSA register as shown. Bits 31–20 are reserved and always read as zero. Reading or writing bits 19-16 requires the use of extended instructions. When writing to DMAxSA with word instructions, bits 19-16 are cleared.

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMAxSZ

DMAxSZ, DMA Size Address Register



DMAxSZx	Bits 15–0	DMA size. The DMA size register defines the number of byte/word data per block transfer. DMAxSZ register decrements with each word or byte transfer. When DMAxSZ decrements to 0, it is immediately and automatically reloaded with its previously initialized value. 00000h Transfer is disabled 00001h One byte or word to be transferred 00002h Two bytes or words have to be transferred : 0FFFFh 65535 bytes or words have to be transferred

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DMAIV

DMAIV, DMA Interrupt Vector Register

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
r0	r0	r0	r0	r0	r0	r0	r0

7	6	5	4	3	2	1	0
0	0	0	0	DMAIVx			0
r0	r0	r0	r0	r–(0)	r–(0)	r–(0)	r0

DMAIVx Bits 15-0 DMA Interrupt Vector value

DMAIV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
00h	No interrupt pending	–	
02h	DMA channel 0	DMA0IFG	Highest
04h	DMA channel 1	DMA1IFG	
06h	DMA channel 2	DMA2IFG	
08h	Reserved	–	
0Ah	Reserved	–	
0Ch	Reserved	–	
0Eh	Reserved	–	Lowest

LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Mem2Mem DMA Transfer

```
//******************************************************************************
//   MSP430xG461x Demo - DMA0, Repeated Burst to-from RAM, Software Trigger
//
//  Description: A 16 word block from 1400-141fh is transferred to 1420h-143fh
//  using DMA0 in a burst block using software DMAREQ trigger.
//  After each transfer, source, destination and DMA size are
//  reset to initial software setting because DMA transfer mode 5 is used.
//  P5.1 is toggled during DMA transfer only for demonstration purposes.
//  ** RAM location 0x1400 - 0x143f used - make sure no compiler conflict **
//   ACLK = 32kHz, MCLK = SMCLK = default DCO 1048576Hz
//
//               MSP430xG461x
//            -----------------
//        /|\|              XIN|-
//         | |                 | 32kHz
//         --|RST          XOUT|-
//           |                 |
//           |             P5.1|-->LED
//
//   A. Dannenberg/ M. Mitchell
//   Texas Instruments Inc.
//   October 2006
//   Built with IAR Embedded Workbench Version: 3.41A
//******************************************************************************
```

```
#include "msp430xG46x.h"


void main(void)

{

  WDTCTL = WDTPW + WDTHOLD;      // Stop WDT

  P5DIR |= 0x002;  // P1.0  output

  DMA0SA = 0x1400; // Start block address

  DMA0DA = 0x1420; // Destination block address

  DMA0SZ = 0x0010; // Block size

  DMA0CTL = DMADT_5 + DMASRCINCR_3 + DMADSTINCR_3 +
    DMAEN; // Rpt, inc

  DMA0CTL |= DMAEN; // Enable DMA0


  while(1)

  {

    P5OUT |= 0x02;   // P5.1 = 1, LED on

    DMA0CTL |= DMAREQ; // Trigger block transfer

    P5OUT &= ~0x02;    // P5.1 = 0, LED off

  }

}
```


LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Mem2USCI DMA Transfer

```
//*******************************************************************************
//   MSP430xG461x Demo - DMA0, Block Mode UART1 9600, ACLK
//
//   Description: DMA0 is used to transfer a string as a block to U1TXBUF.
//   UTXIFG1 WILL trigger DMA0. "Hello World" is TX'd via 9600 baud on UART1.
//   Watchdog in interval mode triggers block transfer every 1000ms.
//   Level sensitive trigger used for UTXIFG1 to prevent loss of initial edge
//   sensitive triggers - UTXIFG1 which is set at POR.
//   ACLK = UCLK 32768Hz, MCLK = SMCLK = default DCO 1048576Hz
//   Baud rate divider with 32768hz XTAL @9600 = 32768Hz/9600 = 3.41 (000Dh 4Ah)
//
//                MSP430xG461x
//             -----------------
//         /|\|              XIN|-
//          | |                 | 32768Hz
//          --|RST          XOUT|-
//            |                 |
//            |             P4.0|------------> "Hello World"
//            |                 | 9600 - 8N1
//
//   A. Dannenberg/ M. Mitchell
//   Texas Instruments Inc.
//   October 2006
//   Built with IAR Embedded Workbench Version: 3.41A
//*******************************************************************************
```

```
#include "msp430xG46x.h"

const char String1[13] = "\nHello World";

void main(void)
{
  WDTCTL = WDT_ADLY_1000; // WDT 1000ms, ACLK, int
  IE1 |= WDTIE; // Enable WDT interrupt
  P4SEL |= 0x03; // P4.0,1 = USART1 TXD/RXD
  ME2 |= UTXE1 + URXE1; // Enable USART1 TXD/RXD
  UCTL1 |= CHAR;     // 8-bit characters
  UTCTL1 = SSEL0;    // BRCLK = ACLK
  UBR01 = 0x03;      // 32k/9600=3.41
  UBR11 = 0x00;
  UMCTL1 = 0x04A;    // Modulation
  UCTL1 &= ~SWRST;  // Release USART state machine
  DMACTL0 = DMA0TSEL_10;  // UTXIFG1 trigger
  DMA0SA = (int)String1;  // Source block address
  DMA0DA = TXBUF1_; // Destination single address
  DMA0SZ = 0014;     // Block size
  DMA0CTL = DMASRCINCR_3 + DMASBDB + DMALEVEL;
// Repeat, inc src
  __bis_SR_register(LPM3_bits + GIE);
// Enter LPM3 w/ interrupts
}
#pragma vector = WDT_VECTOR // Trigger transfer
__interrupt void WDT_ISR(void)
{
  DMA0CTL |= DMAEN;       // Enable
}
```


LaCASA

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Repeated Single Transfer Demo

```
//*****************************************************************************
//   MSP430xG461x Demo - DMA0, Repeated Block to P5OUT, TACCR2 Trigger
//
//   Description: DMA0 is used to transfer a string byte-by-byte as a repeating
//   block to P5OUT. Timer_A operates continuously with CCR2IFG
//   triggering DMA0. The effect is P5.0/5.1 toggling at different frequencies.
//   ACLK = 32kHz, MCLK = SMCLK = TACLK = default DCO 1048576Hz
//
//              MSP430xG461x
//            -----------------
//        /|\|              XIN|-
//         | |                 | 32kHz
//         --|RST          XOUT|-
//           |                 |
//           |             P5.0|-->
//           |             P5.1|--> LED
//
//   A. Dannenberg/ M. Mitchell
//   Texas Instruments Inc.
//   October 2006
//   Built with IAR Embedded Workbench Version: 3.41A
//*****************************************************************************
```

```
#include "msp430xG46x.h"


const char testconst[6] = { 0x0, 0x3, 0x2, 0x3, 0x0, 0x1 };


void main(void)

{

  WDTCTL = WDTPW + WDTHOLD;  // Stop WDT

  P5DIR |= 0x003;            // P5.0/5.1 output

  DMACTL0 = DMA0TSEL_1;      // CCR2 trigger

  DMA0SA = (int)testconst;   // Source block address

  DMA0DA = (int)&P5OUT; // Destination single address

  DMA0SZ = 0x06;         // Block size

  DMA0CTL = DMADT_4 + DMASRCINCR_3 + DMASBDB + DMAEN; // Rpt,
    inc src

  TACTL = TASSEL_2 + MC_2;  // SMCLK/4, contmode

  __bis_SR_register(LPM0_bits + GIE); // Enter LPM0

}
```


LaCASA