Lecture 10
Digital-to-Analogue Conversion & Direct Memory Access (DMA)

Texas Instruments Incorporated
University of Beira Interior (PT)

Pedro Dinis Gaspar, António Espírito Santo, Bruno Ribeiro, Humberto Santos
University of Beira Interior, Electromechanical Engineering Department
www.msp430.ubi.pt

>> Contents

Contents (1/2)

Contents (2/2)

Introduction (1/3)

- The final stage in digital processing is to convert the digital output value to a signal that can be used by the real world e.g. a voltage or current;
- A Digital-to-Analogue converter (DAC) is an electronic device or circuit that converts a digital representation of a quantity to a discrete analogue value;
- The inputs to a DAC are the digital value and a reference voltage V_{REF} to set the analogue output level;

Introduction (2/3)

- **Provides a continuous time output signal, mathematically often treated as discrete Dirac pulses into a zero-order hold and consisting of a series of fixed steps;**

- **Filtering the discrete output signal can be used to approximate a continuous time signal, as well as:**
  - Increasing the resolution;
  - Increasing the number of discrete levels and;
  - Reducing the level size (reduces the quantization error).

Introduction (3/3)

- **Ideal DAC output:**
  - A sequence of impulses filtered to construct a continuous time analogue signal;
  - Precise reproduction of the sampled signal up to the Nyquist frequency.

- **Real DAC output: Reconstruction is not precise**
  - Filter has infinite phase delay;
  - There will be quantization errors.

- **The digital data sequence is usually converted into an analogue voltage at a uniform update rate;**

- **The clock signal latches the actual data of the digital input data sequence and the DAC holds the output analogue voltage until the next clock signal latches new data.**

DAC types (1/2)

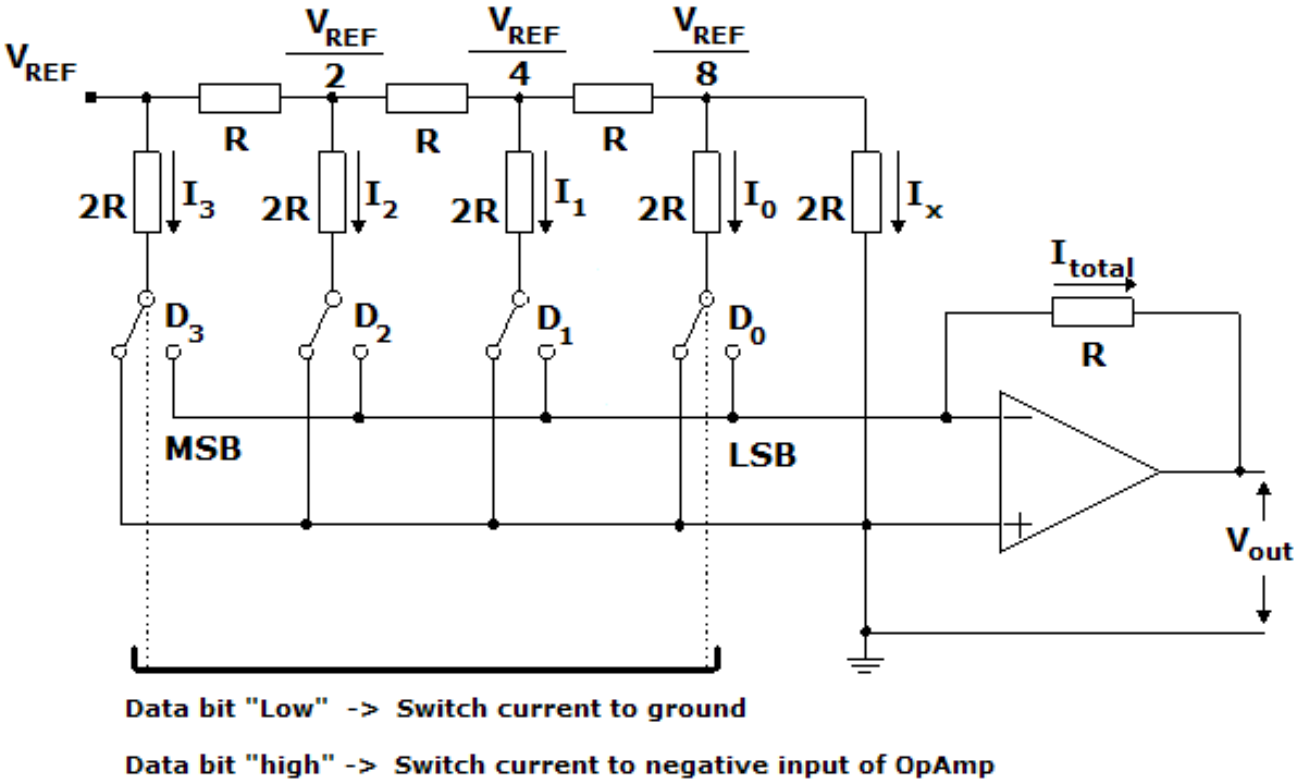
- **Binary Weighted DAC:**
  - Contains one resistor (or current source) for each bit of the DAC connected to a common voltage source V_{REF};
  - There are accuracy problems (high precision resistors are required);

- **R/2R Ladder DAC:**
  - Binary weighted DAC that uses a repeating cascaded structure of resistors of value R and 2R;
  - The MSP430's DAC12 module uses this architecture.

DAC types (2/2)

R/2R Ladder DAC:

- Example: R/2R 4 bit DAC architecture:



- Switch current to negative input of Op-Amp which is a virtual ground

DAC characteristic parameters (1/2)

- **Resolution (*n*):**
  - Number of possible DAC output levels, 2^n (*n*: no. of bits);
  - The Effective Number Of Bits (ENOB) is the actual resolution achieved by the DAC, taking into account errors like nonlinearity, signal-to noise ratio.

- **Integral Non-Linearity (INL):**
  - Deviation of a DAC's transfer function from a straight line.

- **Differential Non-Linearity (DNL):**
  - Difference between an actual step height and the ideal value of 1 LSB;
  - DNL < 1 LSB, the DAC is monotonic, that is, no loss of data.

DAC characteristic parameters (2/2)

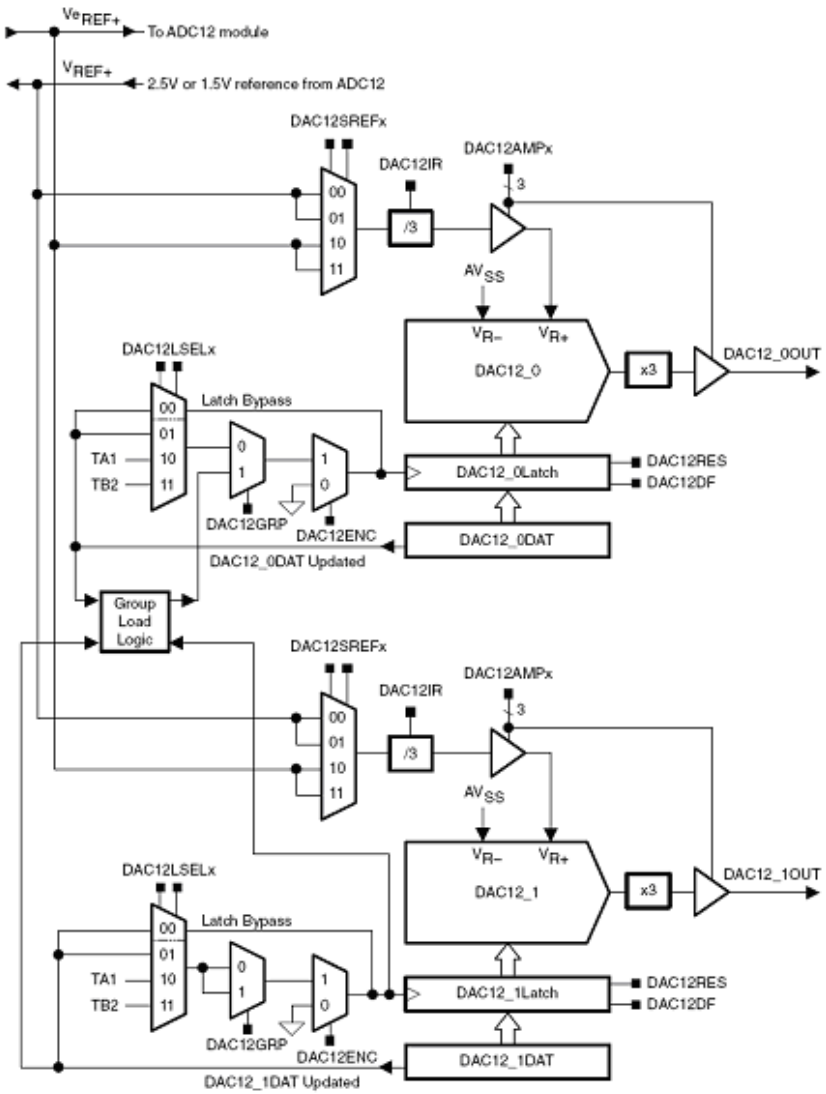
- **Offset error:**
  - Analogue output voltage when the digital input is zero.
- **Gain error:**
  - Difference between the ideal maximum output voltage and the actual maximum value of the transfer function, after subtracting the offset error.
- **Monotonicity:**
  - Ability of the analogue output of the DAC to increase with an increase in digital code or the converse.
- **Total Harmonic Distortion (THD):**
  - Distortion and noise introduced to the signal by the DAC.
- **Dynamic range:**
  - Difference between the largest and the smallest signals.

DAC12 module

- The 12 bit DAC12 module is a voltage output DAC;
- All the MSP430 hardware development tools contain this module;
- The MSP430FG4618 device on the Experimenter's board has two DAC12 modules, allowing them to be grouped together for synchronous update operation.

DAC12 module

- **DAC12 block diagram:**



DAC12 features

- 12 bit monotonic output;
- 8-bit or 12-bit voltage output resolution;
- Programmable settling time *vs.* power consumption;
- Internal or external reference selection;
- Straight binary or Two's complement data format;
- Self-calibration option for offset correction;
- Synchronized update capability for multiple DAC12s;
- Direct Memory Access (DMA) enable.

DAC12 operation (1/4)

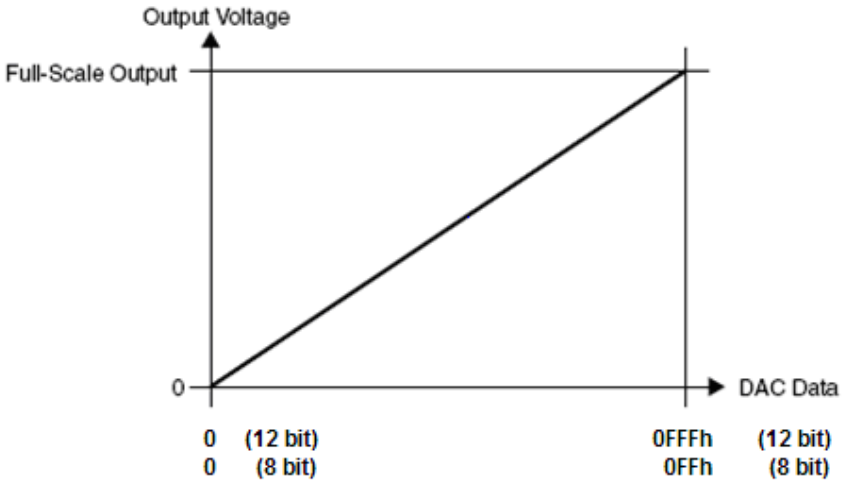
- **DAC12 core:**
  - Dynamic range controlled by:
    - DAC's resolution: 8 bits or 12 bits (DAC12RES bit);
    - Full-scale output: 1xV_{REF} or 3xV_{REF} (DAC12IR bit);
    - Input data format: straight binary or two's complement (DAC12DF bit).
  - The output voltage (straight binary data format):

Resolution	DAC12RES	DAC12IR	Output voltage
12 bit	0	0	$V_{OUT} = V_{REF} \times 3 \times \frac{DAC12\_xDAT}{4096}$
12 bit	0	1	$V_{OUT} = V_{REF} \times \frac{DAC12\_xDAT}{4096}$
8 bit	1	0	$V_{OUT} = V_{REF} \times 3 \times \frac{DAC12\_xDAT}{256}$
8 bit	1	1	$V_{OUT} = V_{REF} \times \frac{DAC12\_xDAT}{256}$

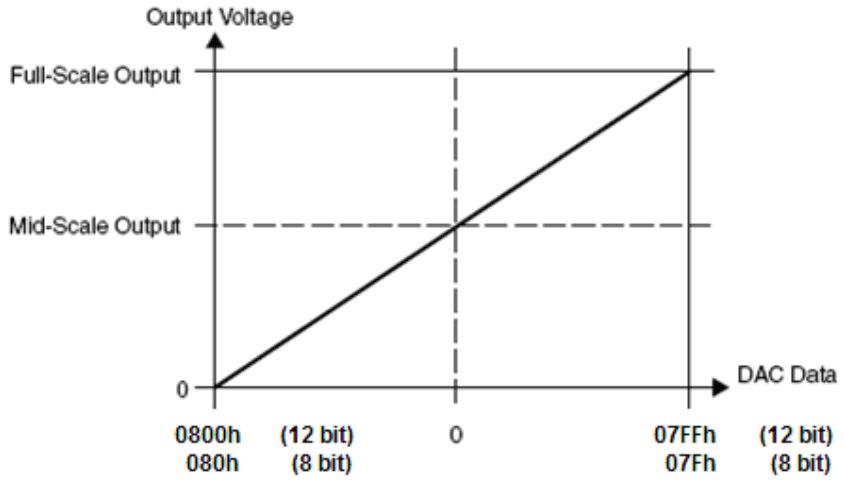
DAC12 operation (2/4)

DAC12_xDAT Data Format:

- The data format modifies the full-scale output voltage:



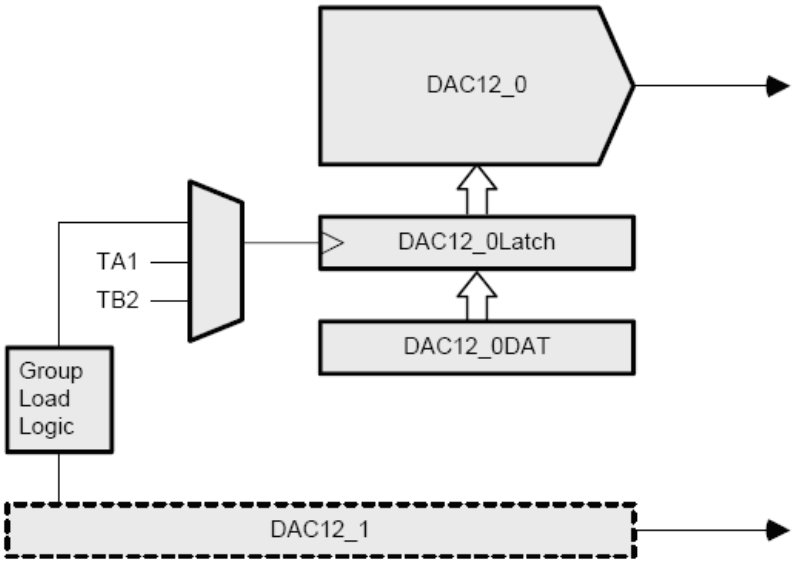
Straight binary



Two's complement

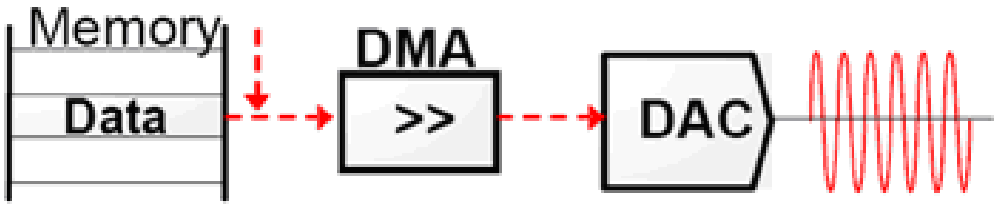
DAC12 operation (3/4)

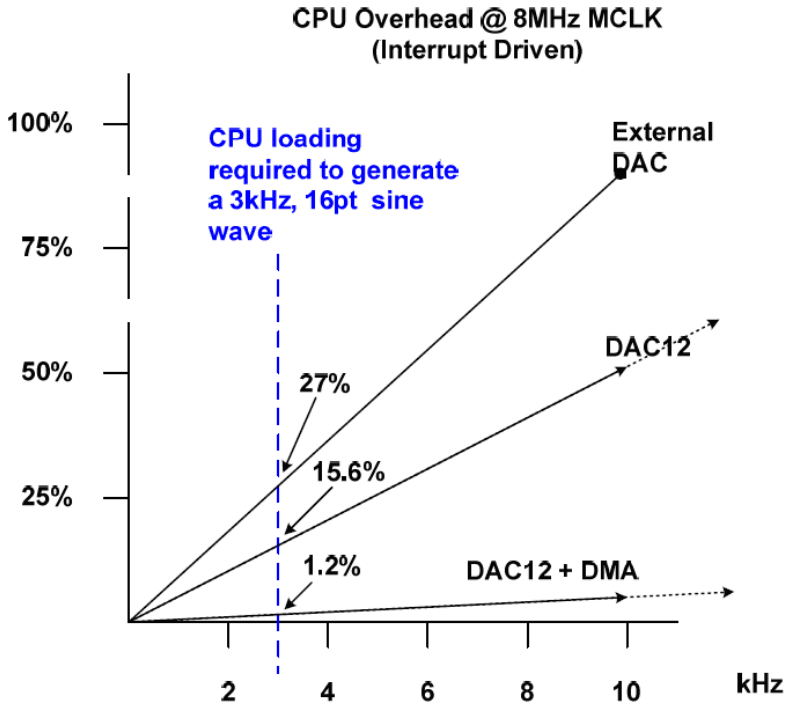
- **Updating the DAC12 voltage output (DAC12_xDAT reg.):**
  - Configurable with the DAC12LSELx bits:
    - DAC12LSELx = 0: Immediate when new data is written;
    - DAC12LSELx = 1: Grouped (data is latched);
    - DAC12LSELx = 2: Rising edge from the Timer_A CCR1;
    - DAC12LSELx = 3: Rising edge from the Timer_B CCR2.



DAC12 operation (4/4)

- **DAC12 Interrupts:**
  - The DAC12IV is shared with the DMA controller;
  - This structure provides:
    - Increased system flexibility;
    - No code execution required;
    - Lower power;
    - Higher efficiency.





DAC12 Registers (1/3)

❑ DAC12_xCTL, DAC12 Control Register

15	14	13	12	11	10	9	8
DAC12OPS	DAC12SREFx		DAC12RES	DAC12LSELx		DAC12CALON	DAC12IR

7	6	5	4	3	2	1	0
DAC12AMPx			DAC12DF	DAC12IE	DAC12IFG	DAC12ENC	DAC12GRP

Bit		Description
15	DAC12OPS	DAC12 output: DAC12OPS = 0 ⇒ DAC12_0 on P6.6, DAC12_1 on P6.7 DAC12OPS = 1 ⇒ DAC12_0 on V_{eREF+}, DAC12_1 on P5.1
14-13	DAC12REFx	DAC12 reference voltage: DAC12REF1 DAC12REF0 = 00 ⇒ V_{REF+} DAC12REF1 DAC12REF0 = 01 ⇒ V_{REF+} DAC12REF1 DAC12REF0 = 10 ⇒ Ve_{REF+} DAC12REF1 DAC12REF0 = 11 ⇒ Ve_{REF+}
12	DAC12RES	DAC12 resolution: DAC12RES = 0 ⇒ 12 bit resolution DAC12RES = 1 ⇒ 8 bit resolution
11-10	DAC12LSELx	DAC12 load: DAC12LSEL1 DAC12LSEL0 = 00 ⇒DAC12_xDAT written DAC12LSEL1 DAC12LSEL0 = 01 ⇒all grouped DAC12_xDAT written DAC12LSEL1 DAC12LSEL0 = 10 ⇒Rising edge of Timer_A.OUT1 (TA1) DAC12LSEL1 DAC12LSEL0 = 11 ⇒Rising edge of Timer_B.OUT2 (TB2)
9	DAC12CALON	DAC12 calibration initialized or in progress when DAC12CALON = 1
8	DAC12IR	DAC12 input range: DAC12IR = 0 ⇒ DAC12 full-scale output: 3x reference voltage DAC12IR = 1 ⇒ DAC12 full-scale output: 1x reference voltage

DAC12 Registers (2/3)

❑ DAC12_xCTL, DAC12 Control Register

15	14	13	12	11	10	9	8
DAC12OPS	DAC12SREFx		DAC12RES	DAC12LSELx		DAC12CALON	DAC12IR

7	6	5	4	3	2	1	0
DAC12AMPx			DAC12DF	DAC12IE	DAC12IFG	DAC12ENC	DAC12GRP

Bit		Description		
7-5	DAC12AMPx	DAC12 amplifier setting: AMP2 AMP1 AMP0 = 000 ⇒ AMP2 AMP1 AMP0 = 001 ⇒ AMP2 AMP1 AMP0 = 010 ⇒ AMP2 AMP1 AMP0 = 011 ⇒ AMP2 AMP1 AMP0 = 100 ⇒ AMP2 AMP1 AMP0 = 101 ⇒ AMP2 AMP1 AMP0 = 110 ⇒ AMP2 AMP1 AMP0 = 111 ⇒ f: frequency (speed) I: current	Input buffer: Off Off Low f / I Low f / I Low f / I Medium f / I Medium f / I High f / I	Output buffer: DAC12 off (high Z) DAC12 off (0 V) Low f / I Medium f / I High f / I Medium f / I High f / I High f / I
4	DAC12DF	DAC12 data format: DAC12DF = 0 ⇒ Straight binary DAC12DF = 1 ⇒ Two's complement		
3	DAC12IE	DAC12 interrupt enable when DAC12IE = 1		
2	DAC12IFG	DAC12 Interrupt flag DAC12IFG = 1 when interrupt pending		
1	DAC12ENC	DAC12 enable when DAC12ENC = 1.		
0	DAC12GRP	Groups DAC12_x with the next higher DAC12_x when DAC12GRP = 1 (exception for DAC12_1)		

DAC12 Registers (3/3)

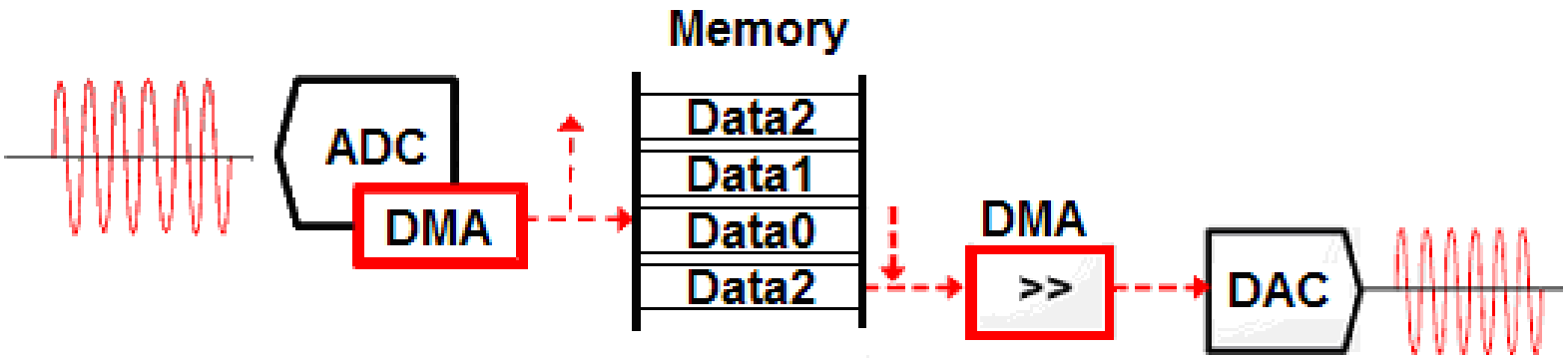
- **DAC12_xDAT, DAC12 Data Register**
  - The four most significant bits (bits 15 – 12) are always zero;
  - The twelve least significant bits store the DAC12 data (bits 11 – 0);
  - The DAC12 data is right justified, but the MSB depends on:
    - Resolution:
      - 8 bit: Bit 7;
      - 12 bit: Bit 11.
    - Data format:
      - Straight binary: MSB is data;
      - Two's complement: MSB is sign.

DMA capability (1/2)

- **The MSP430 has been designed for applications requiring low power;**
- **When the application requires data-handling, the direct memory access (DMA) capability included in some devices is useful:**
  - 5xxx; FG4xx(x); F261x; F16x(x) and F15x;
  - Among these: MSP430FG4618 (Experimenter's board).
- **DMA automatically handles data;**
- **DMA does not require CPU intervention;**
- **DMA helps reduce the power consumption (CPU remains sleeping).**

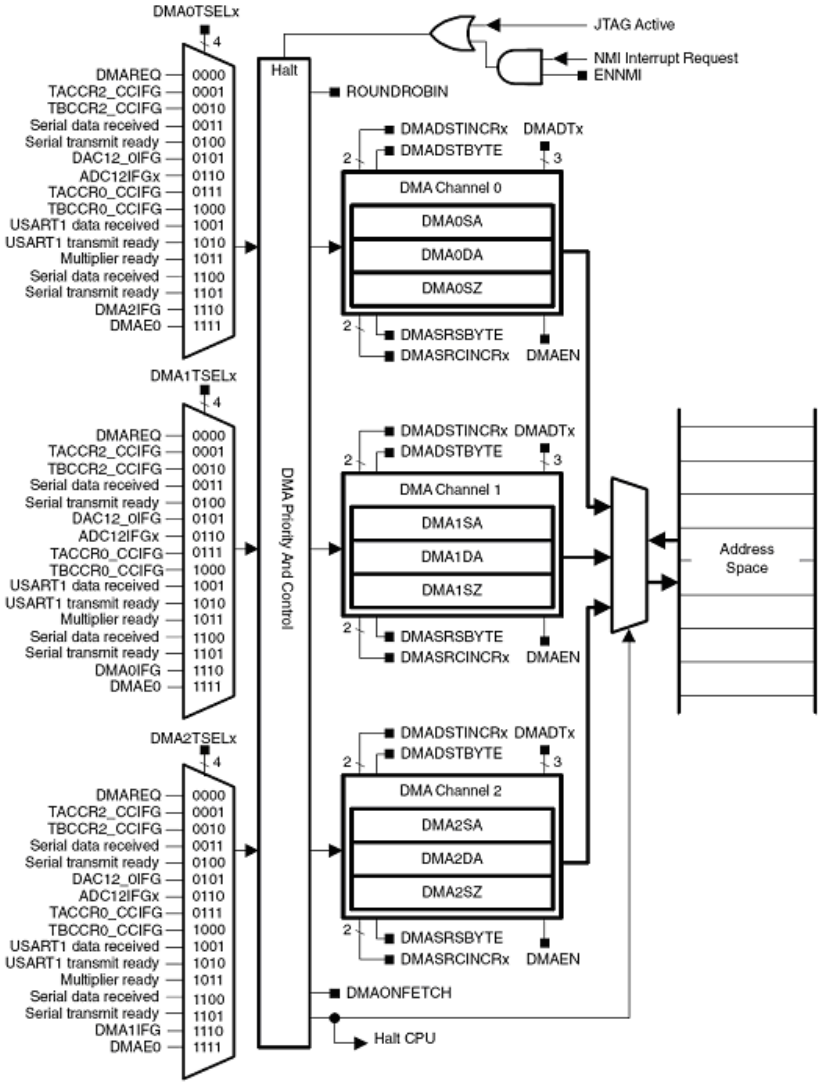
DMA capability (2/2)

- **Concept of DMA: move functionality to peripherals:**
  - Peripherals use less current than the CPU;
  - Delegating control to peripherals allows the CPU to shut down (saves power);
  - "Intelligent" peripherals are more capable, providing a better opportunity for CPU shutoff;
  - DMA can be enabled for repetitive data handling, increasing the throughput of peripheral modules;
  - Minimal software requirements and CPU cycles.



DMA configuration and operation (1/9)

- **Block diagram:**



DMA configuration and operation (2/9)

- **DMA controller features:**
  - Three independent transfer channels;
  - Configurable (ROUNDROBIN bit) DMA channel priorities:
    - Default: DMA0–DMA1–DMA2;
  - DMA Transfer cycle time:
    - Requires only two MCLK clock cycles per transfer;
    - Each byte/word transfer requires:
      - 2 MCLK cycles after synchronization;
      - 1 MCKL cycle of wait time after transfer.

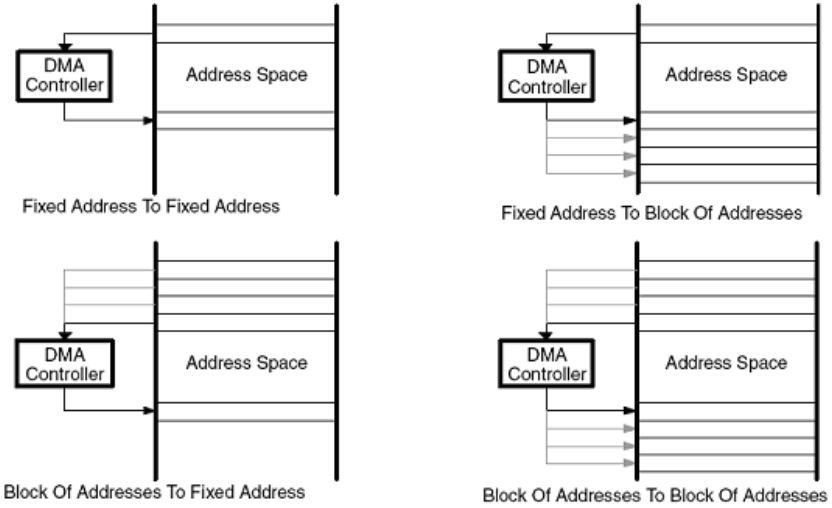
DMA configuration and operation (3/9)

- **DMA controller features:**
  - Block sizes up to 65535 bytes or words;
  - Configurable edge/level-triggered transfer (DMALEVEL bit).
  - Byte or word and mixed byte/word transfer capability:
    - Byte-to-byte;
    - Word-to-word;
    - Byte-to-word (upper byte of the destination word is cleared);
    - Word-to-byte (lower byte of the source word is transferred).

DMA configuration and operation (4/9)

DMA controller features:

- Four addressing modes for each DMA channel are independently configurable (DMASRCINCRx and DMADSTINCRx control bits):
  - Fixed address to fixed address;
  - Fixed address to block of addresses;
  - Block of addresses to fixed address;
  - Block of addresses to block of addresses.



DMA configuration and operation (5/9)

- **DMA controller features:**
  - Six transfer modes (each channel is individually configurable by the DMADTx bits):

DMADTx	Transfer mode	Description	DMAEN after transfer
000	Single transfer	Each transfer requires a trigger	0
001	Block transfer	A complete block is transferred with one trigger	0
010, 011	Burst-block transfer	CPU activity is interleaved with a block transfer	0
100	Repeated single transfer	Each transfer requires a trigger	1
101	Repeated block transfer	A complete block is transferred with one trigger	1
110, 111	Repeated burst-block transfer	CPU activity is interleaved with a block transfer	1

DMA configuration and operation (6/9)

- **System interrupts:**
  - DMA transfers are not interruptible by system interrupts, but system ISRs can be interrupted by DMA transfers;
  - Only NMI interrupts can be given priority over the DMA controller (ENNMI bit is set). If the ENNMI bit is not set, system interrupts remain pending until the completion of the transfer.
- **DMA controller interrupts:**
  - Each DMA channel has its own DMAIFG flag that is set when the corresponding DMAxSZ register counts to zero (all modes);
  - If the corresponding DMAIE and GIE bits are set, an interrupt request is generated.

DMA configuration and operation (7/9)

- **DMA controller interrupts:**
  - The MSP430FG4618 implements the interrupt vector register DMAIV;
  - All DMAIFG flags are prioritized and combined to source a single interrupt vector;
  - DMAIV is used to determine which flag requested an interrupt.

DMA configuration and operation (8/9)

- **DMA transfers:**
  - **USCI_B I2C module with DMA:**
    - Two trigger sources for the DMA controller;
    - Triggers a transfer when new I2C data is received and when data is required for transmit.
  - **ADC12 with DMA:**
    - Automatically moves data from any ADC12MEMx register to another location.
  - **DAC12 with DMA:**
    - Automatically moves data to the DAC12_xDAT register.

DMA configuration and operation (9/9)

- **DMA transfers:**
  - **DMA with flash memory:**
    - Automatically moves data to the Flash memory;
    - Supports word/byte data transfers to the flash memory;
    - The write timing control is done by the Flash controller;
    - Write transfers to the Flash memory succeed if the Flash controller set-up is done before the DMA transfer and if the Flash is not busy.
  - **All DMA transfers:**
    - Occur without CPU intervention;
    - Operate independently of any low-power modes;
    - Increase throughput of modules.

DMA Registers (1/11)

❑ DMACTL0, DMA Control Register 0 (FG4618)

15	14	13	12	11	10	9	8
Reserved				DMA2TSELx			

7	6	5	4	3	2	1	0
DMA1TSELx				DMA0TSELx			

- All DMAxTSELx registers are the same.

DMAxTSELx	Transfer triggered
0000	when DMAREQ = 1 (DMAREQ = 0 automatically when the transfer starts)
0001	<Timer_A> when TACCR2 CCIFG = 1 (CCIFG = 0 automatically when the transfer starts) If CCIE = 1, CCIFG won't trigger a transfer
0010	<Timer_B> when TBCCR2 CCIFG = 1 (CCIFG = 0 automatically when the transfer starts) If CCIE = 1, CCIFG won't trigger a transfer

DMA Registers (2/11)

❑ DMACTL0, DMA Control Register 0 (FG4618) (continued)

DMAxTSELx	Transfer triggered
0011	<USART0>: when URXIFG0 = 1 (URXIFG0 = 0 automatically when the transfer starts) If URXIE0 = 1, URXIFG0 flag won't trigger a transfer <USCI_A0>: when UCA0RXIFG = 1 (UCA0RXIFG = 0 automatically when the transfer starts) If UCA0RXIE = 1, UCA0RXIFG flag won't trigger a transfer
0100	<USART0>: when UTXIFG0 =1 (UTXIFG0 = 0 automatically when the transfer starts) If UTXIE0 = 1, UTXIFG0 flag won't trigger a transfer <USCI_A0>: when UCA0TXIFG = 1 (UCA0TXIFG = 0 automatically when the transfer starts) UCA0TXIE = 1, UCA0TXIFG flag won't trigger a transfer
0101	<DAC12> when DAC12_0CTL DAC12IFG = 1 (DAC12IFG = 0 automatically when the transfer starts) If DAC12IE = 1, DAC12IFG won't trigger a transfer

DMA Registers (3/11)

❑ DMACTL0, DMA Control Register 0 (FG4618) (continued)

DMAxTSELx	Transfer triggered
0110	<ADC12> when ADC12IFGx = 1 (corresponding ADC12IFGx flag for single-channel conversions, and the ADC12IFGx for the last conversion for sequence conversions) (All ADC12IFGx = 0 automatically when the associated ADC12MEMx register is accessed by the DMA controller)
0111	<Timer_A> when TACCR0 CCIFG = 1: CCIFG = 0 automatically when the transfer starts If CCIE = 1, CCIFG flag won't trigger a transfer
1000	<Timer_B> when TBCCR0 CCIFG = 1 (CCIFG = 0 automatically when the transfer starts) If CCIE = 1, CCIFG won't trigger a transfer
1001	<USART1>: when URXIFG1 = 1 (URXIFG1 = 0 automatically when the transfer starts) If URXIE1 = 1, URXIFG0 flag won't trigger a transfer

DMA Registers (4/11)

❑ DMACTL0, DMA Control Register 0 (FG4618) (continued)

DMAxTSELx	Transfer triggered
1010	<USART1>: when UTXIFG1 =1 (UTXIFG1 = 0 automatically when the transfer starts) If UTXIE1 = 1, UTXIFG0 flag won't trigger a transfer
1011	<Hardware Multiplier> when the hardware multiplier is ready for a new operand
1100	<USCI_B0>: when UCB0RXIFG = 1 (UCB0RXIFG = 0 automatically when the transfer starts) If UCB0RXIE = 1, UCB0RXIFG flag won't trigger a transfer
1101	<USCI_B0>: when UCB0TXIFG = 1 (UCB0TXIFG = 0 automatically when the transfer starts) UCB0TXIE = 1, UCB0TXIFG flag won't trigger a transfer
1110	when the DMAxIFG = 1: DMA0IFG triggers channel 1 DMA1IFG triggers channel 2 DMA2IFG triggers channel 0 (None of the DMAxIFG = 0 automatically when the transfer starts)
1111	When an external trigger DMAE0 = 1

DMA Registers (5/11)

❑ DMACTL1, DMA Control Register 1 (FG4618)

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

7	6	5	4	3	2	1	0
0	0	0	0	0	DMAONFETCH	ROUNDROBIN	ENNMI

Bit		Description
2	DMAONFETCH	DMA on fetch: DMAONFETCH = 0 ⇒ DMA transfer occurs immediately DMAONFETCH = 1 ⇒ DMA transfer occurs on next instruction fetch after the trigger
1	ROUNDROBIN	Round robin: ROUNDROBIN = 0 ⇒ DMA channel priority is DMA0 − DMA1 − DMA2 ROUNDROBIN = 1 ⇒ DMA channel priority changes with each transfer
0	ENNMI	Enable NMI when ENNMI = 1, allowing NMI interrupt to interrupt a DMA transfer

DMA Registers (6/11)

❑ DMAxCTL, DMA Channel x Control Register (FG4618)

15	14	13	12	11	10	9	8
Reserved	DMADTx			DMADSTINCRx		DMASRCINCRx	

7	6	5	4	3	2	1	0
DMADSTBYTE	DMASRCBYTE	DMALEVEL	DMAEN	DMAIFG	DMAIE	DMAABORT	DMAREQ

Bit		Description
14-12	DMADTx	DMA transfer mode: DMADT2 DMADT1 DMADT0 = 000 ⇒ Single transfer DMADT2 DMADT1 DMADT0 = 001 ⇒ Block transfer DMADT2 DMADT1 DMADT0 = 010 ⇒ Burst-block transfer DMADT2 DMADT1 DMADT0 = 011 ⇒ Burst-block transfer DMADT2 DMADT1 DMADT0 = 100 ⇒ Repeated single transfer DMADT2 DMADT1 DMADT0 = 101 ⇒ Repeated block transfer DMADT2 DMADT1 DMADT0 = 110 ⇒ Repeated burst-block transfer DMADT2 DMADT1 DMADT0 = 111 ⇒ Repeated burst-block transfer

DMA Registers (7/11)

❑ DMAxCTL, DMA Channel x Control Register (FG4618) (continued)

15	14	13	12	11	10	9	8
Reserved	DMADTx			DMADSTINCRx		DMASRCINCRx	

7	6	5	4	3	2	1	0
DMADSTBYTE	DMASRCBYTE	DMALEVEL	DMAEN	DMAIFG	DMAIE	DMAABORT	DMAREQ

Bit		Description
11-10	DMADSTINCRx	DMA destination address increment/decrement 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. DMADSTINCR1 DMADSTINCR0 = 00 ⇒ Address unchanged DMADSTINCR1 DMADSTINCR0 = 01 ⇒ Address unchanged DMADSTINCR1 DMADSTINCR0 = 10 ⇒ Address decremented DMADSTINCR1 DMADSTINCR0 = 11 ⇒ Address increment

DMA Registers (8/11)

❑ DMAxCTL, DMA Channel x Control Register (FG4618) (continued)

15	14	13	12	11	10	9	8
Reserved	DMADTx			DMADSTINCRx		DMASRCINCRx	

7	6	5	4	3	2	1	0
DMADSTBYTE	DMASRCBYTE	DMALEVEL	DMAEN	DMAIFG	DMAIE	DMAABORT	DMAREQ

Bit		Description
9-8	DMASRCINCRx	DMA source address increment/decrement after 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. DMASRCINCR1 DMASRCINCR0 = 00 ⇒ Address unchanged DMASRCINCR1 DMASRCINCR0 = 01 ⇒ Address unchanged DMASRCINCR1 DMASRCINCR0 = 10 ⇒ Address decremented DMASRCINCR1 DMASRCINCR0 = 11 ⇒ Address increment

DMA Registers (9/11)

❑ DMAxCTL, DMA Channel x Control Register (FG4618) (continued)

15	14	13	12	11	10	9	8
Reserved	DMADTx			DMADSTINCRx		DMASRCINCRx	

7	6	5	4	3	2	1	0
DMADSTBYTE	DMASRCBYTE	DMALEVEL	DMAEN	DMAIFG	DMAIE	DMAABORT	DMAREQ

Bit		Description
7	DMADSTBYTE	DMA destination length (byte or word): DMADSTBYTE = 0 ⇒ Word DMADSTBYTE = 1 ⇒ Byte
6	DMASRCBYTE	DMA source length (byte or word): DMASRCBYTE = 0 ⇒ Word DMASRCBYTE = 1 ⇒ Byte
5	DMALEVEL	DMA level: DMALEVEL = 0 ⇒ Edge sensitive trigger (rising edge) DMALEVEL = 1 ⇒ Level sensitive trigger (high level)
4	DMAEN	DMA enable when DMAEN = 1
3	DMAIFG	DMA interrupt flag DMAIFG = 1 when interrupt pending
2	DMAIE	DMA interrupt enable when DMAIE = 1
1	DMAABORT	DMA Abort DMAABORT = 1 when a DMA transfer is interrupted by NMI
0	DMAREQ	DMA request DMAREQ = 1 starts DMA

DMA Registers (10/11)

- **DMAxSA, DMA Source Address Register (FG4618)**
  - 32-bit register points to the DMA source address for single transfers or the first source address for block transfers.

- **DMAxDA, DMA Destination Address Register (FG4618)**
  - 32-bit register points to the DMA destination address for single transfers or the first source address for block transfers.

  - For both registers (DMAxSA and DMAxDA):
    - Bits 31−20 are reserved and always read as zero;
    - Reading or writing to bits 19-16 requires the use of extended instructions;
    - When writing to DMAxSA or DMAxDA with word instructions, bits 19-16 are cleared.

DMA Registers (11/11)

- **DMAxSZ, DMA Size Address Register (FG4618)**
  - The 16-bit DMA size address register defines the number of bytes/words of data per block transfer:
    - DMAxSZ decrements with each word or byte transfer;
    - When DMAxSZ = 0, it is immediately and automatically reloaded with its previously initialized value.

- **DMAIV, DMA Interrupt Vector Register (FG4618)**
  - 16 bit DMAIV value only uses bits 3 to 1 (other bits = 0);
  - DMAIV content provides the interrupt source priority:

    DMAIV = 02h: DMA channel 0 (highest priority);

    DMAIV = 04h: DMA channel 1;

    DMAIV = 06h: DMA channel 2;

    …

    DMAIV = 0Eh: Reserved (lowest priority).