DAC12_A

NOTE: This chapter is an excerpt from the *MSP430x5xx and MSP430x6xx Family User's Guide*.

The DAC12_A module is a 12-bit voltage-output digital-to-analog converter (DAC). This chapter describes the operation and use of the DAC12_A module.

Topic **Page**

1.1 DAC12_A Introduction

The DAC12_A module is a 12-bit voltage-output DAC. The DAC12_A can be configured in 8-bit or 12-bit mode and can be used in conjunction with the DMA controller. When multiple DAC12_A modules are present, they can be grouped together for synchronous operation.

Features of the DAC12_A include:

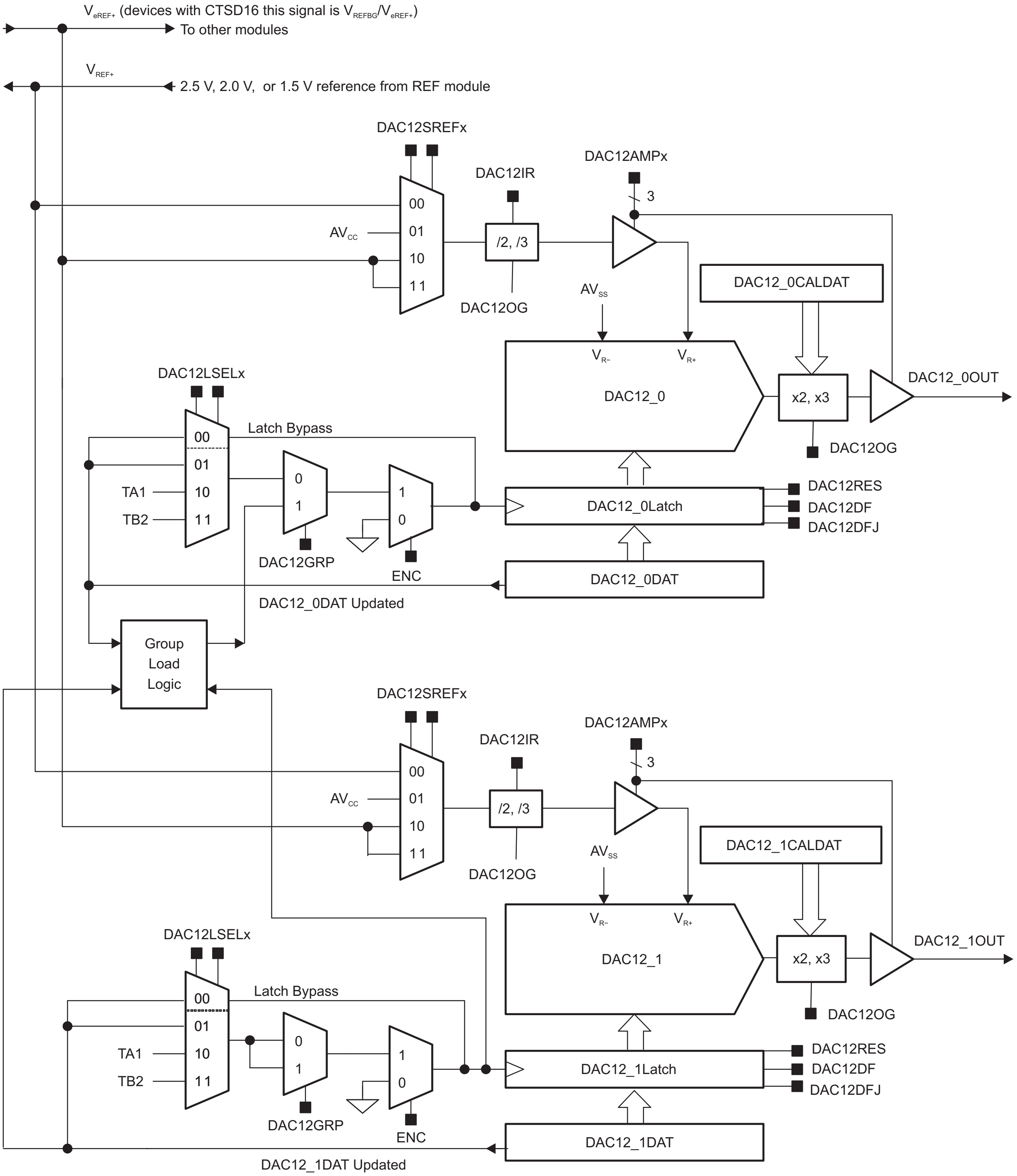
- 12-bit monotonic output
- 8-bit or 12-bit voltage-output resolution
- Programmable settling time and power consumption
- Internal or external reference selection
- Straight binary or twos-complement data format, right or left justified
- Self-calibration option for offset correction
- Synchronized update capability for multiple DAC12_A modules

NOTE: Multiple DAC12_A Modules

Some devices may integrate more than one DAC12_A module. In the case where more than one DAC12_A is present on a device, the multiple DAC12_A modules operate identically.

Throughout this chapter, nomenclature appears such as DAC12_xDAT or DAC12_xCTL to describe register names. When this occurs, the x is used to indicate which DAC12 module is being discussed. In cases where operation is identical, the register is simply referred to as DAC12_xCTL.

Figure 1-1 shows the block diagram for a device with two DAC12_A modules. Figure 1-2 shows the block diagram for a device with one DAC12_A module.



Copyright © 2017, Texas Instruments Incorporated

Figure 1-1. DAC12_A Block Diagram for a Device With Two Modules

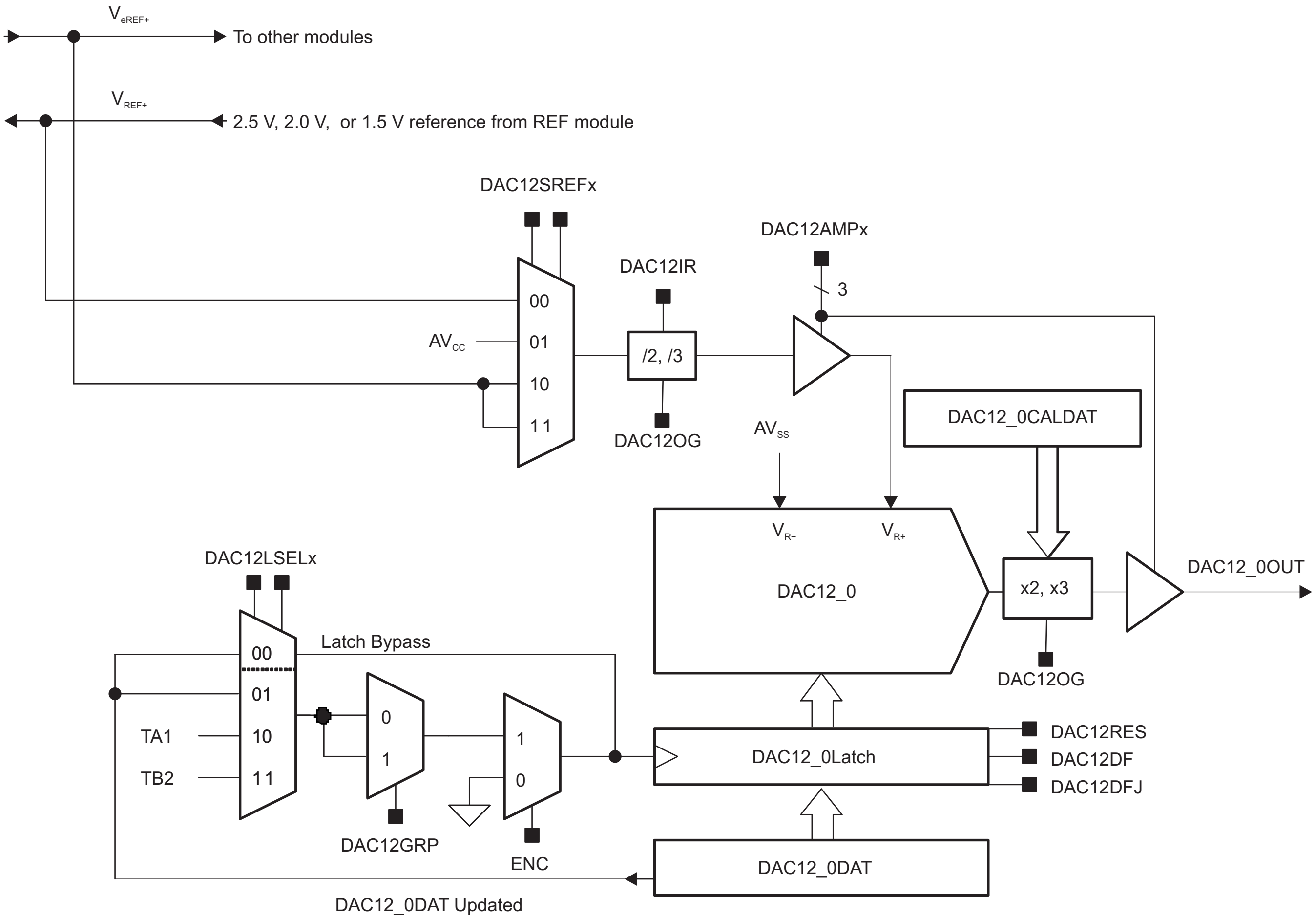


Figure 1-2. DAC12_A Block Diagram For a Device With One Module

1.2 DAC12_A Operation

The DAC12_A module is configured with user software. The setup and operation of the DAC12_A is discussed in the following sections.

1.2.1 DAC12_A Core

The DAC12_A can be configured to operate in 8-bit or 12-bit mode using the DAC12RES bit. The full-scale output is programmable to be 1x, 2x, or 3x the selected reference voltage using the DAC12IR and DAC12OG bits. This feature lets the user control the dynamic range of the DAC. The DAC12DF bit lets the user select between straight-binary data and twos-complement data for the DAC. When using straight-binary data format, the formula for the output voltage is given in Table 1-1.

Table 1-1. DAC Full-Scale Range (V_{ref} = V_{eREF+} or V_{REF+} or V_{REFBG} [(1)])

Resolution	DAC12RES	DAC12OG	DAC12IR	Output Voltage Formula
12 bit	0	0	0	$V_{out} = V_{ref} \times 3 \times (DAC12\_xDAT / 4096)$
		1		$V_{out} = V_{ref} \times 2$ x $(DAC12\_xDAT / 4096)$
		x	1	$V_{out} = V_{ref} \times (DAC12\_xDAT / 4096)$
8 bit	1	0	0	$V_{out} = V_{ref} \times 3 \times (DAC12\_xDAT / 256)$
		1		$V_{out} = V_{ref} \times 2 \times (DAC12\_xDAT / 256)$
		x	1	$V_{out} = V_{ref} \times (DAC12\_xDAT / 256)$

[(1)] V_{REFBG} is not available on all devices; see the device-specific data sheet for availability.

In 8-bit mode, the maximum useable value for DAC12_xDAT is 0FFh. In 12-bit mode, the maximum useable value for DAC12_xDAT is 0FFFh. Values greater than these may be written to the register, but all leading bits are ignored.

1.2.2 DAC12_A Port Selection

On most devices, the DAC outputs are multiplexed with the port Px pins and potentially other secondary functions. When DAC12AMPx > 0, the DAC function is automatically selected for the pin, regardless of the state of the associated PxSEL.y and PxDIR.y bits. See the port pin diagrams in the device-specific data sheet for more details.

1.2.3 DAC12_A Reference

The DAC12SREFx bits select the reference for the DAC12_A from:

- AV_{CC}
- An external reference voltage
- The internal 1.16-V (V_{REFBG}) voltage reference (V_{REFBG} is available only on devices with a CTSD16 module; see the device-specific data sheet)
- The internal 1.5-V, 2.0-V, or 2.5-V reference from the REF module.

When DAC12SREFx = {0} and DAC12AMPx > {1}, V_{REF+} is used as the reference, which is supplied from the REF module. See the REF chapter for further information.

When DAC12SREFx = {1}, AV_{CC} is used as the reference.

When DAC12SREFx = {2,3} the external reference, V_{eREF+} signal, is used as the reference unless the device has a CTSD16 module.

On devices that have a CTSD16 module (see the device-specific data sheet), when DAC12SREFx = {2,3} either the external reference signal (V_{eREF+}) or the internally generated reference signal (V_{REFBG}) is selected. See Table 1-2 for details on which bits to set to select between the V_{eREF+} and V_{REFBG} signals.

Table 1-2. DAC12SREFx = {2,3} Signal Selection Requirements for Devices With a CTSD16 Module

Signal Selected	REFOUT	REFON	PxSEL.y	CTSD16REFS[1]
V_{eREF+}	0	x	1	If the CTSD16 module is used, CTSD16REFS must be set to 0.
V_{REFBG}	1	1	1[2]	If the CTSD16 module is used, CTSD16REFS must be set to 1.

[1] If CTSD16 is used, CTSD16REFS bit must be set according to this table as external voltage reference V_{eREF+} and internal voltage reference V_{REFBG} cannot both be used at the same time as they use the same pin.

[2] When V_{REFBG} is selected, PxSEL.y must always be set even if V_{REFBG} is only used inside the device.

1.2.3.1 DAC12_A Reference Input and Voltage Output Buffers

The reference input and voltage output buffers of the DAC can be configured to optimize the balance of settling time and power consumption. Eight combinations are selected using the DAC12AMPx bits. In the low setting, the settling time is the slowest, and the current consumption of both buffers is the lowest. The medium and high settings have faster settling times, but the current consumption increases. See the device-specific data sheet for parameters.

1.2.4 Updating the DAC12_A Voltage Output

The DAC12_xDAT register can be connected directly to the DAC core or double buffered. The DAC12LSELx bits select the trigger for updating the DAC voltage output.

When DAC12LSELx = 0, the data latch is transparent and the DAC12_xDAT register is applied directly to the DAC core. The DAC output updates immediately when new DAC data is written to the DAC12_xDAT register, regardless of the state of the DAC12ENC bit.

When DAC12LSELx = 1, DAC data is latched and applied to the DAC core after new data is written to DAC12_xDAT.

When DAC12LSELx = 2 or 3, data is latched on the rising edge from the Timer_A CCR1 output or Timer_B CCR2 output, respectively. DAC12ENC must be set to latch the new data when DAC12LSELx > 0.

1.2.5 DAC12_xDAT Data Formats

The DAC12_A supports both straight-binary and twos-complement data formats. When using straight-binary data format, the full-scale output value is 0FFFh in 12-bit mode (0FFh in 8-bit mode) (see Figure 1-3).

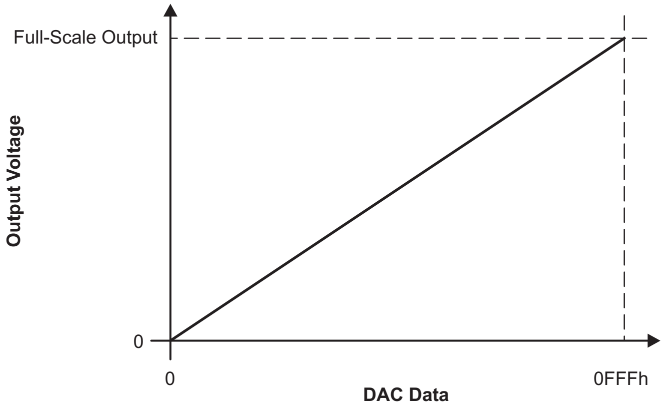


Figure 1-3. Output Voltage vs DAC Data, 12-Bit Straight-Binary Mode

When using twos-complement data format, the range is shifted so that a DAC12_xDAT value of 0800h (0080h in 8-bit mode) results in a zero output voltage, 0000h is the mid-scale output voltage, and 07FFh (007Fh for 8-bit mode) is the full-scale voltage output (see Figure 1-4).

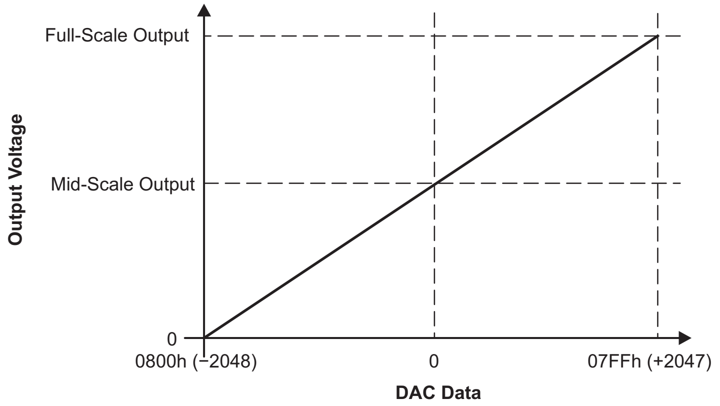


Figure 1-4. Output Voltage vs DAC Data, 12-Bit Twos-Complement Mode

1.2.6 DAC12_A Output Amplifier Offset Calibration

The offset voltage of the DAC output amplifier can be positive or negative. When the offset is negative, the output amplifier attempts to drive the voltage negative but cannot do so. The output voltage remains at zero until the DAC digital input produces a sufficient positive output voltage to overcome the negative offset voltage, resulting in the transfer function in Figure 1-5.

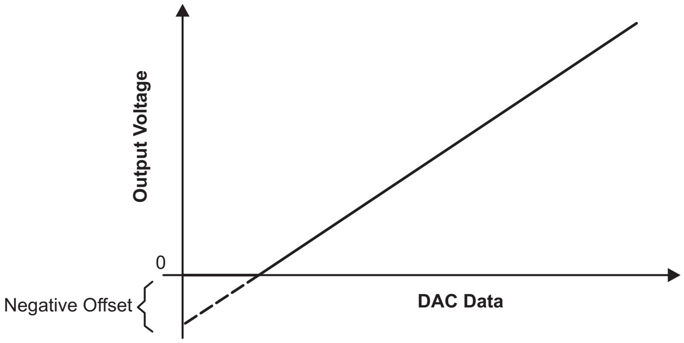


Figure 1-5. Negative Offset

When the output amplifier has a positive offset, a digital input of zero does not result in a zero output voltage. The DAC12_A output voltage reaches the maximum output level before the DAC12_A data reaches the maximum code (see Figure 1-6).

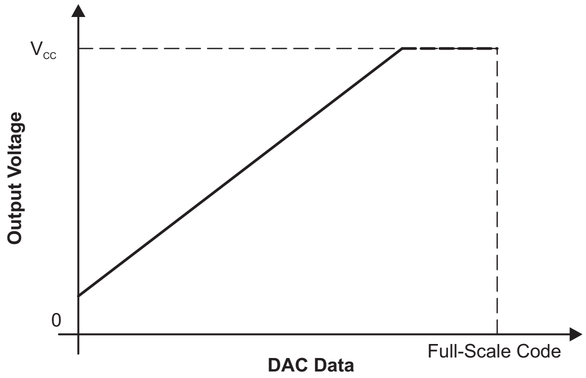


Figure 1-6. Positive Offset

The DAC12_A can calibrate the offset voltage of the output amplifier. Setting the DAC12CALON bit initiates the offset calibration. Allow the calibration to complete before using the DAC. This can be checked by monitoring the DAC12CALON bit. When the calibration is complete, the DAC12CALON bit is automatically reset. The DAC12AMPx bits should be configured before calibration. For best calibration results, port and CPU activity should be minimized during calibration.

The contents of DAC12x_CALDAT can be protected from inadvertent write access by a lock mechanism controlled by the DAC12x_CALCTL register. At power up, the LOCK bit is set, and calibration is not possible and writing to the DAC12x_CALDAT is ignored. To perform calibration, the LOCK bit must be cleared by writing the correct password to DAC12x_CALCTL and clearing the LOCK bit. After LOCK is cleared, calibration or writing to the DAC12x_CALDAT can be performed. After calibration is performed, lock the calibration registers by writing the correct password to DAC12x_CALCTL and setting the LOCK bit.

Reading DAC12_xCALDAT should only be performed while the DAC12CALON bit is cleared, otherwise incorrect values may be read. The DAC12xCAL data format is twos complement. Only the lower byte is used and the upper byte has no effect on the calibration.

1.2.7 Grouping Multiple DAC12_A Modules

Multiple DAC12_A modules can be grouped together with the DAC12GRP bit to synchronize the update of each DAC output. Hardware ensures that all DAC12_A modules in a group update simultaneously independent of any interrupt or NMI event.

On devices that contain more than one DAC, DAC12_0 and DAC12_1 are grouped by setting the DAC12GRP bit of DAC12_0. The DAC12GRP bit of DAC12_1 is don't care. When DAC12_0 and DAC12_1 are grouped:

- The DAC12_0 and DAC12_1 DAC12LSELx bits select the update trigger for both DACs.
- The DAC12LSELx bits for both DACs must be the same.
- The DAC12LSELx bits for both DACs must be > 0
- The DAC12ENC bits of both DACs must be set to 1

When DAC12_0 and DAC12_1 are grouped, both DAC12_xDAT registers must be written before the outputs update, even if data for one or both of the DACs is not changed. Figure 1-7 shows a latch update timing example for grouped DAC12_0 and DAC12_1.

When DAC12_0 DAC12GRP = 1 and both DAC12_x DAC12LSELx > 0 and either DAC12ENC = 0, neither DAC updates.

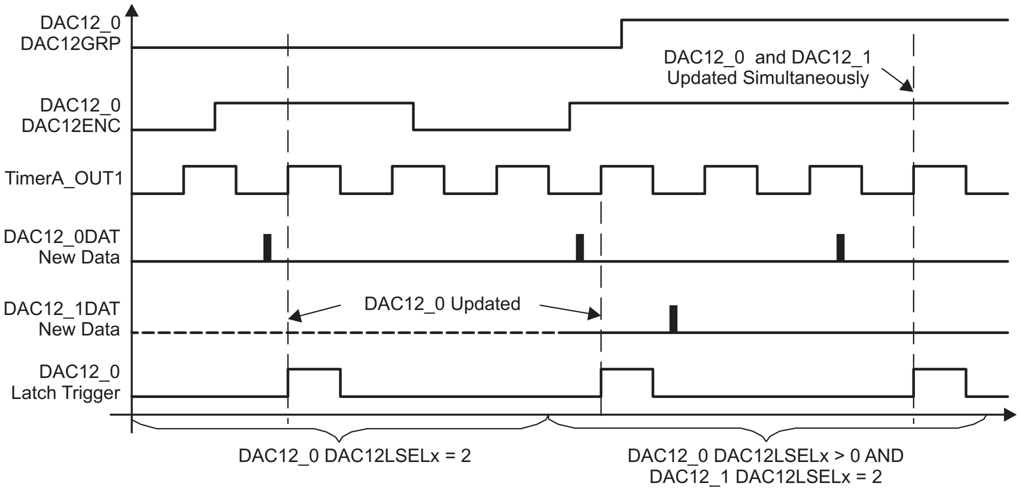


Figure 1-7. DAC12_A Group Update Example, Timer_A3 Trigger

NOTE: DAC12_A Settling Time

The DMA controller can transfer data to the DAC12_A faster than the DAC12_A output can settle. The user must make sure that the DAC12_A settling time is not violated when using the DMA controller. See the device-specific data sheet for parameters.

1.2.8 DAC12_A Interrupts

The DAC12IFG bit is set when DAC12LSELx > 0 and DAC data is latched from the DAC12_xDAT register into the data latch. When DAC12LSELx = 0, the DAC12IFG flag is not set.

A set DAC12IFG bit indicates that the DAC is ready for new data. If both the DAC12IE and GIE bits are set, the DAC12IFG generates an interrupt request. The DAC12IFG flag must be reset by software or can be reset automatically by accessing the DAC12IV register. See the DAC12IV description for further information.

For devices that contain a DMA, each DAC channel has a DMA trigger associated with it. When DAC12IFG is set, it can trigger a DMA transfer to the DAC12_xDAT register. DAC12IFG is automatically reset when the transfer begins. If the DAC12IE is also set, no DMA transfer occurs when the DAC12IFG is set.

1.2.8.1 DAC12IV, Interrupt Vector Generator

The DAC12_A flags are prioritized and combined to source a single interrupt vector. The interrupt vector register DAC12IV is used to determine which flag requested an interrupt.

The highest priority enabled interrupt generates a number in the DAC12IV register (see register description). This number can be evaluated or added to the program counter to automatically enter the appropriate software routine. Disabled DAC interrupts do not affect the DAC12IV value.

Any access, read or write, of the DAC12IV register automatically resets the highest pending interrupt flag. If another interrupt flag is set, another interrupt is immediately generated after servicing the initial interrupt. For example, if the DAC12IFG_0 and DAC12IFG_1 flags are set when the interrupt service routine accesses the DAC12IV register, the DAC12IFG_0 flag is reset automatically. After the RETI instruction of the interrupt service routine is executed, the DAC12IFG_1 flag generates another interrupt.

1.2.8.1.1 DAC12IV Software Example

The following software example shows the recommended use of DAC12IV and the handling overhead. The DAC12IV value is added to the PC to automatically jump to the appropriate routine.

The numbers at the right margin show the necessary CPU cycles for each instruction. The software overhead for different interrupt sources includes interrupt latency and return-from-interrupt cycles, but not the task handling itself. The latencies are:

```
; Interrupt handler for DAC12_A.                Cycles

DAC12_HND             ; Interrupt latency         6

   ADD &DAC12IV,PC    ; Add offset to Jump table   3
   RETI               ; Vector 0: No interrupt     5
   JMP DAC12IFG_0_HND ; Vector 2: DAC12_0          2
   JMP DAC12IFG_1_HND ; Vector 4: DAC12_1          2
   RETI               ; Vector 6: Reserved         5
   RETI               ; Vector 8: Reserved         5
   RETI               ; Vector 8: Reserved         5
   RETI               ; Vector 10: Reserved        5
DAC12IFG_1_HND ; Vector 4: DAC12_1
   ...  ; Task starts here
   RETI ; Back to main program

DAC12IFG_0_HND ; Vector 2: DAC12_0
   ...  ; Task starts here
   RETI ; Back to main program                     5
```

1.3 DAC Outputs

On most devices, each DAC channel can be output to two different port pins selected by the DAC12OPS bit. When DAC12OPS = 0, one of the two ports is selected as the DAC output. Similarly, when DAC12OPS = 1, the other port is selected as the DAC output. Table 1-3 summarizes this for a single DAC channel that can be output to either ports Pm.y or Pn.z.

Table 1-3. DAC Output Selection

DAC12OPS	DAC12AMP	Pm.y Function	Pn.z Function
0	{0}	I/O	I/O
0	{1}	I/O	DAC output, 0 V
0	{>1}	I/O	DAC output
1	{0}	I/O	I/O
1	{1}	DAC output, 0 V	I/O
1	{>1}	DAC output	I/O

1.4 DAC12_A Registers

The DAC12_A registers are listed in Table 1-4. The base address can be found in the device-specific data sheet. The address offset is listed in Table 1-4.

Table 1-4. DAC12_A Registers

Offset	Acronym	Register Name	Type	Access	Reset	Section
00h	DAC12_0CTL0	DAC12_0 Control 0	Read/write	Word	0000h	Section 1.4.1
02h	DAC12_0CTL1	DAC12_0 Control 1	Read/write	Word	0000h	Section 1.4.2
04h	DAC12_0DAT	DAC12_0 Data	Read/write	Word	0000h	Section 1.4.3 through Section 1.4.10
06h	DAC12_0CALCTL	DAC12_0 Calibration Control	Read/write	Word	9601h	Section 1.4.11
08h	DAC12_0CALDAT	DAC12_0 Calibration Data	Read/write	Word	0000h	Section 1.4.12
10h	DAC12_1CTL0	DAC12_1 Control 0	Read/write	Word	0000h	Section 1.4.1
12h	DAC12_1CTL1	DAC12_1 Control 1	Read/write	Word	0000h	Section 1.4.2
14h	DAC12_1DAT	DAC12_1 Data	Read/write	Word	0000h	Section 1.4.3 through Section 1.4.10
16h	DAC12_1CALCTL	DAC12_1 Calibration Control	Read/write	Word	0000h	Section 1.4.11
18h	DAC12_1CALDAT	DAC12_1 Calibration Data	Read/write	Word	0000h	Section 1.4.12
1Eh	DAC12IV	DAC12IV	Read	Word	0000h	Section 1.4.13

1.4.1 DAC12_xCTL0 Register

DAC12 Control Register 0

Figure 1-8. DAC12_xCTL0 Register

15	14	13	12	11	10	9	8
DAC12OPS	DAC12SREFx		DAC12RES	DAC12LSELx		DAC12CALON	DAC12IR
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAC12AMPx			DAC12DF	DAC12IE	DAC12IFG	DAC12ENC	DAC12GRP
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

▨ Can be modified only when DAC12ENC = 0

Table 1-5. DAC12_xCTL0 Register Description

Bit	Field	Type	Reset	Description
15	DAC12OPS	RW	0h	DAC output select 0b = DAC12_x channel output on Pm.y 1b = DAC12_x channel output on Pn.z
14-13	DAC12SREFx	RW	0h	DAC select reference voltage 00b = VREF+ 01b = AVCC 10b = VeREF+; for devices with CTSD16 = VeREF+/VREFBG, see Table 1-2 for details on the selection between external reference V_{eREF+} and the internally generated V_{REFBG} signal 11b = VeREF+; for devices with CTSD16 = VeREF+/VREFBG, see Table 1-2 for details on the selection between external reference V_{eREF+} and the internally generated V_{REFBG} signal
12	DAC12RES	RW	0h	DAC resolution select 0b = 12-bit resolution 1b = 8-bit resolution
11-10	DAC12LSELx	RW	0h	DAC load select. Selects the load trigger for the DAC latch. DAC12ENC must be set for the DAC to update, except when DAC12LSELx = 0. 00b = DAC latch loads when DAC12_xDAT written (DAC12ENC is ignored) 01b = DAC latch loads when DAC12_xDAT written, or, when grouped, when all DAC12_xDAT registers in the group have been written. 10b = Rising edge of Timer_A.OUT1 (TA1) 11b = Rising edge of Timer_B.OUT2 (TB2)
9	DAC12CALON	RW	0h	DAC calibration on. This bit initiates the DAC offset calibration sequence and is automatically reset when the calibration completes. 0b = Calibration is not active 1b = Initiate calibration or calibration in progress
8	DAC12IR	RW	0h	DAC input range. The DAC12IR bit along with the DAC12OG bit set the reference input and voltage output range. 0b = If DAC12OG = 0, then DAC12 full-scale output = 3x reference voltage; if DAC12OG = 1, then DAC12 full-scale output = 2x reference voltage 1b = DAC12 full-scale output = 1x reference voltage

Table 1-5. DAC12_xCTL0 Register Description (continued)

Bit	Field	Type	Reset	Description
7-5	DAC12AMPx	RW	0h	DAC amplifier setting. These bits select settling time vs current consumption for the DAC input and output amplifiers. 000b = Input Buffer: Off; Output Buffer: DAC Off, output high impedance 001b = Input Buffer: Off; Output Buffer: DAC Off, output 0 V 010b = Input Buffer: Low speed and current; Output Buffer: Low speed and current 011b = Input Buffer: Low speed and current; Output Buffer: Medium speed and current 100b = Input Buffer: Low speed and current; Output Buffer: High speed and current 101b = Input Buffer: Medium speed and current; Output Buffer: Medium speed and current 110b = Input Buffer: Medium speed and current; Output Buffer: High speed and current 111b = Input Buffer: High speed and current; Output Buffer: High speed and current
4	DAC12DF	RW	0h	DAC data format 0b = Straight binary 1b = Twos complement
3	DAC12IE	RW	0h	DAC interrupt enable 0b = Disabled 1b = Enabled
2	DAC12IFG	RW	0h	DAC interrupt flag 0b = No interrupt pending 1b = Interrupt pending
1	DAC12ENC	RW	0h	DAC enable conversion. This bit enables the DAC12_A module when DAC12LSELx > 0. when DAC12LSELx = 0, DAC12ENC is ignored. 0b = DAC disabled 1b = DAC enabled
0	DAC12GRP	RW	0h	DAC group. Groups DAC12_x with the next higher DAC12_x. Not used for DAC12_1 on dual DAC devices. 0b = Not grouped 1b = Grouped

1.4.2 DAC12_xCTL1 Register

DAC12 Control Register 1

Figure 1-9. DAC12_xCTL1 Register

15	14	13	12	11	10	9	8
Reserved							
r0	r0	r0	r0	r0	r0	r0	r0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DAC12OG	DAC12DFJ
r0	r0	r0	r0	r0	r0	rw-(0)	rw-(0)

▨ Can be modified only when DAC12ENC = 0

Table 1-6. DAC12_xCTL1 Register Description

Bit	Field	Type	Reset	Description
15-2	Reserved	R	0h	Reserved. Always reads as 0.
1	DAC12OG	RW	0h	DAC output buffer gain. Can be modified only when DAC12ENC = 0. 0b = 3x gain 1b = 2x gain
0	DAC12DFJ	RW	0h	DAC data format justification. Can be modified only when DAC12ENC = 0. 0b = Data format right justified 1b = Data format left justified

1.4.3 DAC12_xDAT Register, Unsigned 12-Bit Binary Format, Right Justified

DAC12 Data Register unsigned 12-bit binary format, right justified (DAC12RES = 0, DAC12DF = 0, DAC12DFJ = 0)

Figure 1-10. DAC12_xDAT Register

15	14	13	12	11	10	9	8
Reserved				DAC12 Data			
r(0)	r(0)	r(0)	r(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
DAC12 Data							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-7. DAC12_xDAT Register Description

Bit	Field	Type	Reset	Description
15-12	Reserved	R	0h	Reserved. Always reads as 0.
11-0	DAC12 Data	RW	0h	DAC data in unsigned format. Bit 11 represents the MSB.

1.4.4 DAC12_xDAT Register, Unsigned 12-Bit Binary Format, Left Justified

DAC12 Data Register unsigned 12-bit binary format, left justified (DAC12RES = 0, DAC12DF = 0, DAC12DFJ = 1)

Figure 1-11. DAC12_xDAT Register

15	14	13	12	11	10	9	8
DAC12 Data							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
DAC12 Data				Reserved			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	r(0)	r(0)	r(0)	r(0)

Table 1-8. DAC12_xDAT Register Description

Bit	Field	Type	Reset	Description
15-4	DAC12 Data	RW	0h	DAC data in unsigned format. Bit 15 represents the MSB.
3-0	Reserved	R	0h	Reserved. Always reads as 0.

1.4.5 DAC12_xDAT Register, Twos-Complement 12-Bit Binary Format, Right Justified

DAC12 Data Register twos complement 12-bit binary format, right justified (DAC12RES = 0, DAC12DF = 1, DAC12DFJ = 0)

Figure 1-12. DAC12_xDAT Register

15	14	13	12	11	10	9	8
Bit 11	Bit 11	Bit 11	Bit 11	DAC12 Data			
r(0)	r(0)	r(0)	r(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
DAC12 Data							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-9. DAC12_xDAT Register Description

Bit	Field	Type	Reset	Description
15-12	Bit 11	R	0h	These bits are sign extension bits and are equal to the contents of bit 11. These bits are automatically updated with the contents of bit 11.
11-0	DAC12 Data	RW	0h	DAC data in twos complement format. Bit 11 represents the sign bit of the twos complement value.

1.4.6 DAC12_xDAT Register, Twos-Complement 12-Bit Binary Format, Left Justified

DAC12 Data Register twos complement 12-bit binary format, left justified (DAC12RES = 0, DAC12DF = 1, DAC12DFJ = 1)

Figure 1-13. DAC12_xDAT Register

15	14	13	12	11	10	9	8
DAC12 Data							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
DAC12 Data				Reserved			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	r(0)	r(0)	r(0)	r(0)

Table 1-10. DAC12_xDAT Register Description

Bit	Field	Type	Reset	Description
15-4	DAC12 Data	RW	0h	DAC data in twos complement format. Bit 15 represents the sign bit of the twos complement value.
3-0	Reserved	R	0h	Reserved. Always reads as 0.

1.4.7 DAC12_xDAT Register, Unsigned 8-Bit Binary Format, Right Justified

DAC12 Data Register unsigned 8-bit binary format, right justified (DAC12RES = 1, DAC12DF = 0, DAC12DFJ = 0)

Figure 1-14. DAC12_xDAT Register

15	14	13	12	11	10	9	8
Reserved							
r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)
7	6	5	4	3	2	1	0
DAC12 Data							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-11. DAC12_xDAT Register Description

Bit	Field	Type	Reset	Description
15-8	Reserved	R	0h	Reserved. Always reads as 0.
7-0	DAC12 Data	RW	0h	DAC data in unsigned format. Bit 7 represents the MSB.

1.4.8 DAC12_xDAT Register, Unsigned 8-Bit Binary Format, Left Justified

DAC12 Data Register unsigned 8-bit binary format, left justified (DAC12RES = 1, DAC12DF = 0, DAC12DFJ = 1)

Figure 1-15. DAC12_xDAT Register

15	14	13	12	11	10	9	8
DAC12 Data							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
Reserved							
r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)

Table 1-12. DAC12_xDAT Register Description

Bit	Field	Type	Reset	Description
15-8	DAC12 Data	RW	0h	DAC data in unsigned format. Bit 15 represents the MSB.
7-0	Reserved	R	0h	Reserved. Always reads as 0.

1.4.9 DAC12_xDAT Register, Twos-Complement 8-Bit Binary Format, Right Justified

DAC12 Data Register twos complement 8-bit binary format, right justified (DAC12RES = 1, DAC12DF = 1, DAC12DFJ = 0)

Figure 1-16. DAC12_xDAT Register

15	14	13	12	11	10	9	8
Bit 7	Bit 7	Bit 7	Bit 7	Bit 7	Bit 7	Bit 7	Bit 7
r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)
7	6	5	4	3	2	1	0
DAC12 Data							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-13. DAC12_xDAT Register Description

Bit	Field	Type	Reset	Description
15-8	Bit 7	R	0h	These bits are sign extension bits and are equal to the contents of bit 7. These bits are automatically updated with the contents of bit 7.
7-0	DAC12 Data	RW	0h	DAC data in twos complement format. Bit 7 represents the sign bit of the twos complement value.

1.4.10 DAC12_xDAT Register, Twos-Complement 8-Bit Binary Format, Left Justified

DAC12 Data Register twos complement 8-bit binary format, left justified (DAC12RES = 1, DAC12DF = 1, DAC12DFJ = 1)

Figure 1-17. DAC12_xDAT Register

15	14	13	12	11	10	9	8
DAC12 Data							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
Reserved							
r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)

Table 1-14. DAC12_xDAT Register Description

Bit	Field	Type	Reset	Description
15-8	DAC12 Data	RW	0h	DAC data in twos complement format. Bit 15 represents the sign bit of the twos complement value.
7-0	Reserved	R	0h	Reserved. Always reads as 0.

1.4.11 DAC12_xCALCTL Register

DAC12 Calibration Control Register

Figure 1-18. DAC12_xCALCTL Register

15	14	13	12	11	10	9	8
DAC12KEY							
rw	rw	rw	rw	rw	rw	rw	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							LOCK
r0	r0	r0	r0	r0	r0	r0	rw-(1)

Table 1-15. DAC12_xCALCTL Register Description

Bit	Field	Type	Reset	Description
15-8	DAC12KEY	RW	96h	DAC calibration lock password. Always reads as 0x96. Must be written with 0xA5 for LOCK bit to be set or cleared. An incorrect key results in the LOCK bit being set, thereby disabling write access to DAC12_xCALDAT.
7-1	Reserved	R	0h	Reserved. Always reads as 0.
0	LOCK	RW	1h	DAC calibration lock. To enable write access to the DAC12 calibration data register, this bit must be cleared by writing 0xA5 to DAC12KEY along with LOCK = 0. Writing an incorrect key or writing to DAC12x_CALCTL using byte mode causes the LOCK bit to be automatically set. If the LOCK bit is set, write access to the calibration data registers and hardware calibration is not possible. All previous values in the DAC12_xCALDAT are retained. 0b = Calibration data register write access enabled, calibration can be performed. 1b = Calibration data register write access disabled, calibration disabled.

1.4.12 DAC12_xCALDAT Register

DAC12 Calibration DATA Register

Figure 1-19. DAC12_xCALDAT Register

15	14	13	12	11	10	9	8
Reserved							
r0	r0	r0	r0	r0	r0	r0	r0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAC12 Calibration Data							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-16. DAC12_xCALDAT Register Description

Bit	Field	Type	Reset	Description
15-8	Reserved	R	0h	Reserved. Always reads as 0.
7-0	DAC12 Calibration Data	RW	0h	DAC calibration data. The DAC calibration data is represented in twos complement format providing a range of –128 to +127.

1.4.13 DAC12IV Register

DAC12 Interrupt Vector Register

Figure 1-20. DAC12IV Register

15	14	13	12	11	10	9	8
DAC12IVx							
r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)	r(0)
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAC12IVx							
r(0)	r(0)	r(0)	r(0)	r-(0)	r-(0)	r-(0)	r(0)

Table 1-17. DAC12IV Register Description

Bit	Field	Type	Reset	Description
15-0	DAC12IVx	R	0h	DAC interrupt vector value 00h = No interrupt pending 02h = Interrupt Source: DAC12 channel 0; Interrupt Flag: DAC12IFG_0; Interrupt Priority: Highest 04h = Interrupt Source: DAC12 channel 1; Interrupt Flag: DAC12IFG_1; Interrupt Priority: Lowest 06h = Reserved 08h = Reserved 0Ah = Reserved 0Ch = Reserved 0Eh = Reserved

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265