AES Accelerator

NOTE: This chapter is an excerpt from the *MSP430x5xx and MSP430x6xx Family User's Guide*. See the latest version of the full user's guide for additional information and revision history.

The AES accelerator module performs AES128 encryption or decryption in hardware. This chapter describes the AES accelerator.

Topic **Page**

1.1 AES Accelerator Introduction

The AES accelerator module performs encryption and decryption of 128-bit data with 128-bit keys according to the advanced encryption standard (AES) (FIPS PUB 197) in hardware.

The AES accelerator features are:

- Encryption and decryption according to AES FIPS PUB 197 with 128-bit key
- On-the-fly key expansion for encryption and decryption
- Off-line key generation for decryption
- Byte and word access to key, input, and output data
- AES ready interrupt flag

Figure 1-1 shows the AES accelerator block diagram.

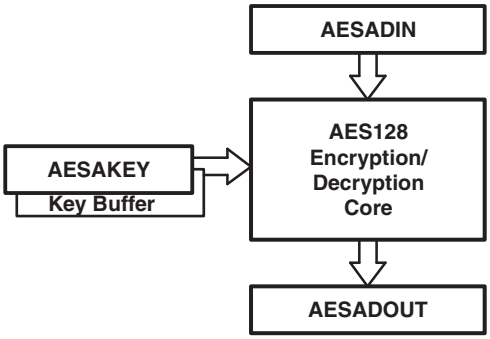

Figure 1-1. AES Accelerator Block Diagram

1.2 AES Accelerator Operation

The AES accelerator is configured with user software. The following sections describe the setup and operation.

Internally, the AES algorithm's operations are performed on a two-dimensional array of bytes called the State. For AES-128, the State consists of four rows of bytes, each containing four bytes. Figure 1-2 shows how the input is assigned to the State array, with in[0] being the first data byte written into the AES accelerator data input register, AESADIN. The encrypt or decrypt operations are then conducted on the State array, after which its final values can be read from the output with out[0] being the first data byte read from the AES accelerator data output register, AESADOUT.

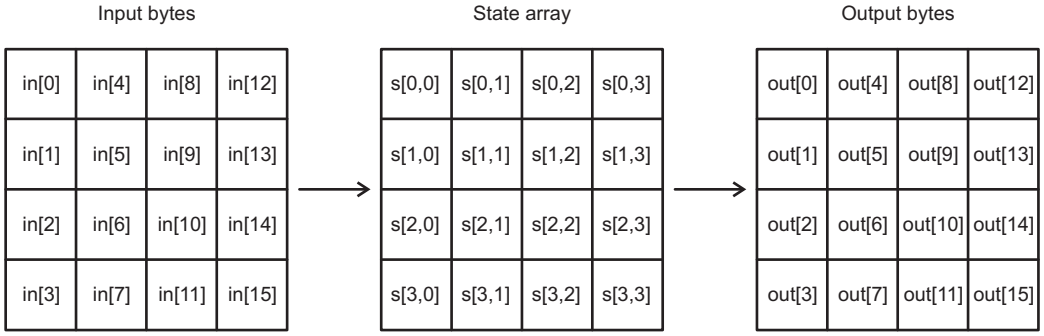

Figure 1-2. AES State Array Input and Output

The module allows word and byte access to all data registers, AESAKEY, AESADIN, and AESADOUT. Word and byte access should not be mixed while reading from or writing into one of the registers. However, it is possible to write one of the registers using byte access and another using word access.

NOTE: Access Restrictions

While the AES accelerator is busy (AESBUSY = 1), AESADOUT always reads as 0, the AESDOUTCNTx counter, the AESDOUTRD flag, and the AESDINWR flag are reset, any attempt to change AESOPx, AESDINWR, or AESKEYWR is ignored, and writing to AESAKEY or AESADIN aborts the current operation, resets the complete module (except for AESRDYIE and AESOPx), and sets the AES error flag AESERRFG.

AESADIN and AESAKEY are write-only registers and always read as 0.

Writing data into AESADIN influences the content of the corresponding output data; for example, writing in[0] alters out[0], writing in[1] alters out[1], and so on, but interleaved operation is possible; for example, first reading out[0], then writing in[0], and continuing with reading out[1], writing in[1], and so on.

NOTE: When using a code debugger, the AES module does not stop its operation when program code is halted or single stepped.

1.2.1 Encryption

Figure 1-3 shows the encryption process with the cipher being a series of transformations that converts the plaintext written into the AESADIN register to a ciphertext that can be read from the AESADOUT register using the cipher key provided in the AESAKEY register.

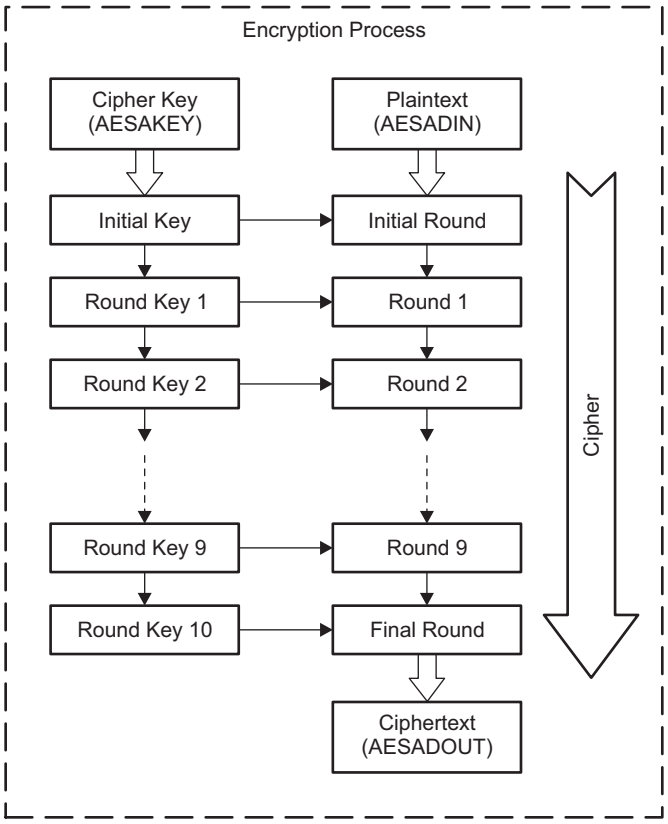

Figure 1-3. AES-128 Encryption Process

The steps to perform encryption are:

1. Set AESOPx = 00 to select encryption. Changing the AESOPx bits clears the AESKEYWR flag, and a new key must be loaded in the next step.
2. Load the 128-bit key into AESAKEY or set the AESKEYWR flag by software, if the key from a previous operation should be used. When all 16 bytes are written, the AESKEYWR flag indicates completion.

   If a key was loaded previously without changing AESOPx, the AESKEYWR flag is cleared with the first write access to AESAKEY. Loading the key must be completed before the next step is performed.
3. Load 128-bit data into AESADIN, or set the AESDINWR flag by software if the output data from a previous operation should be encrypted. When all 16 bytes are written, the AESDINWR flag indicates completion. The module starts encrypting the presented data when AESDINWR = 1.
4. While the AES module is performing encryption, the AESBUSY bit is 1. The encryption takes 167 MCLK clock cycles. After its completion, the AESRDYIFG is set, and the result can be read from AESADOUT. When all 16 bytes are read, the AESDOUTRD flag indicates completion.

   The AESRDYIFG flag is cleared when reading AESADOUT or writing to AESAKEY or AESADIN.
5. If additional data should be encrypted with the same key loaded in step 2, new data can be written into AESADIN after the results of the operation on the previous data were read from AESADOUT. When an additional 16 data bytes are written, the module automatically starts the encryption using the key loaded in step 2.

   When using the output feedback (OFB) cipher block chaining mode, setting the AESDINWR flag is sufficient to trigger the next encryption, and the module starts the encryption automatically using the output data from the previous encryption as input data.

1.2.2 Decryption

Figure 1-4 shows the decryption process with the inverse cipher being a series of transformations that convert the ciphertext written into the AESADIN register to a plaintext that can be read from the AESADOUT register using the cipher key provided via the AESAKEY register.

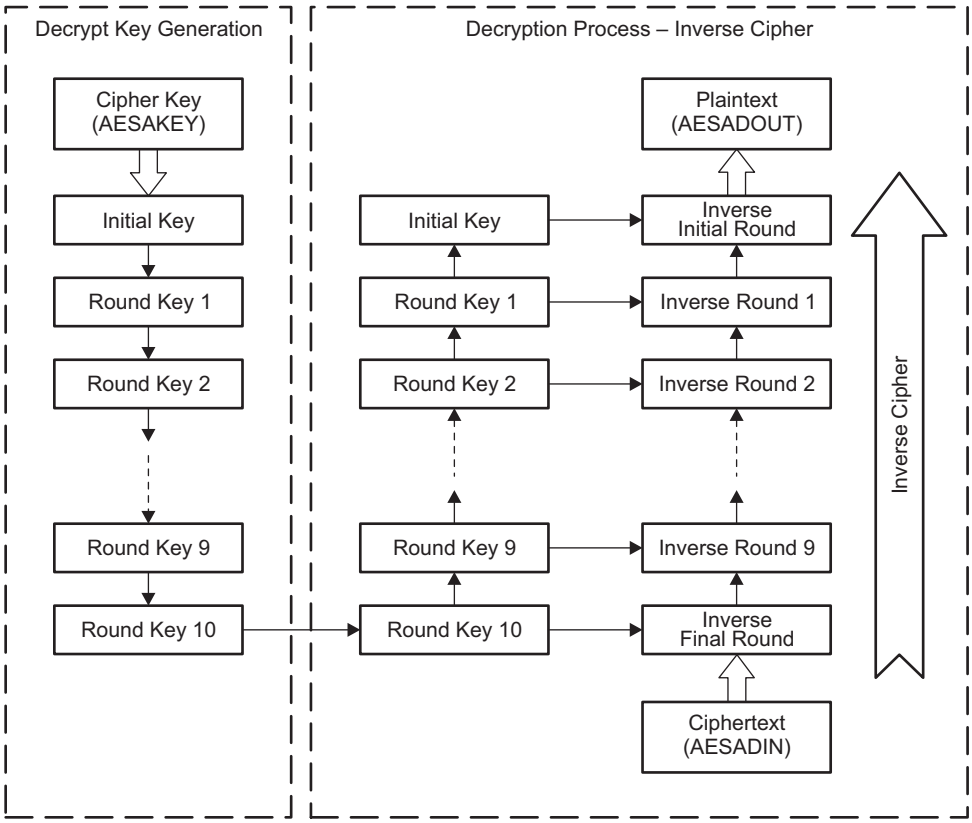

Figure 1-4. AES-128 Decryption Process using AESOPx = 01

The steps to perform decryption are:

1. Set AESOPx = 01 to select decryption using the same key used for encryption. Set AESOPx = 11 if the first-round key required for decryption (round key 10) is already generated and is loaded in step 2. Changing the AESOPx bits clears the AESKEYWR flag, and a new key must be loaded in step 2.
2. Load the 128-bit key into AESAKEY, or set the AESKEYWR flag by software, if the key from a previous operation should be used. When all 16 bytes are written, the AESKEYWR flag indicates completion.

   If a key was loaded previously without changing AESOPx, the AESKEYWR flag is cleared with the first write access to AESAKEY. Loading the key must be completed before the next step is performed.
3. Load 128-bit data into AESADIN or set the AESDINWR flag by software if the output data from a previous operation should be decrypted. When all 16 bytes are written, the AESDINWR flag indicates completion. The module starts decrypting the presented data as soon as AESDINWR = 1.
4. While the AES module is performing decryption, the AESBUSY bit is 1. The decryption takes 214 MCLK clock cycles with AESOPx = 01 and 167 MCLK clock cycles with AESOPx = 11. After its completion, the AESRDYIFG is set, and the result can be read from AESADOUT. When all 16 bytes are read the AESDOUTRD flag indicates completion.

   The AESRDYIFG flag is cleared when reading AESADOUT or writing to AESAKEY or AESADIN.
5. If additional data should be decrypted with the same key loaded in step 2, new data can be written into AESADIN after the results of the operation on the previous data were read from AESADOUT. When additional 16 data bytes are written, the module automatically starts the decryption using the key loaded in step 2.

1.2.3 Decryption Key Generation

Figure 1-5 shows the decryption process with a pregenerated decryption key. In this case, the decryption key is calculated first with AESOPx = 10, then the precalculated key can be used together with the decryption operation AESOPx = 11.

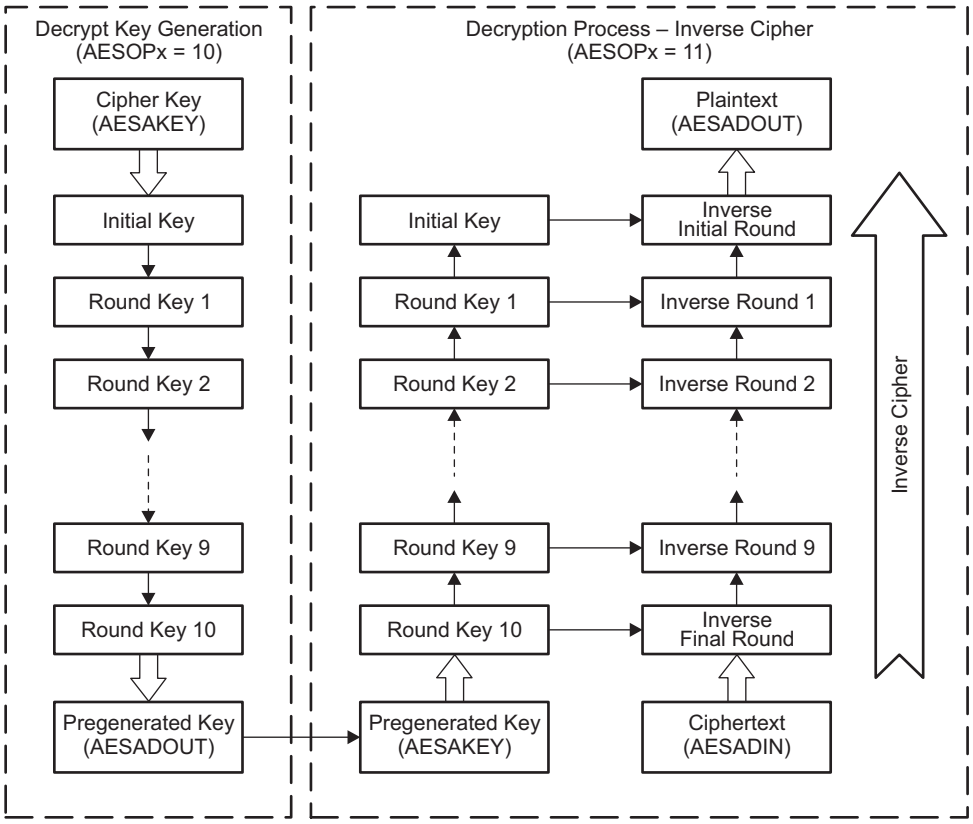

Figure 1-5. AES-128 Decryption Process using AESOPx = 10 and 11

To generate the decryption key independent from the actual decryption, the following steps are required:

1. Set AESOPx = 10 to select decryption key generation. Changing the AESOPx bits clears the AESKEYWR flag, and a new key must be loaded in step 2.
2. Load the 128-bit key into AESAKEY, or set the AESKEYWR flag by software if the key from a previous operation should be used. When all 16 bytes are written, the AESKEYWR flag indicates completion.

   The generation of the first round key required for decryption starts immediately.
3. While the AES module is performing the key generation, the AESBUSY bit is 1. It takes 52 CPU clock cycles to complete the key generation. After its completion, the AESRDYIFG is set, and the result can be read from AESADOUT. When all 16 bytes are read, the AESDOUTRD flag indicates completion.

   The AESRDYIFG flag is cleared when reading AESADOUT or writing to AESAKEY or AESADIN.
4. If data should be decrypted with the generated key, AESOPx must be set to 11. Then the generated key must be loaded or, if it was just generated with AESOPx = 10, it is sufficient to set the AESKEYWR flag by software to indicate that the key is already valid. Afterward, the steps described in Section 1.2.2 to load the data and the rest of the process must be followed.

1.2.4 Using the AES Accelerator With Low-Power Modes

The AES accelerator module provides automatic clock activation for MCLK for use with low-power modes. When the AES accelerator is busy, it automatically activates MCLK, regardless of the control-bit settings for the clock source. The clock remains active until the AES accelerator completes its operation.

1.2.5 AES Accelerator Interrupts

The AESRDYIFG interrupt flag is set when the AES module completes the selected operation on the provided data. An interrupt request is generated if AESRDYIE and GIE are also set. AESRDYIFG is automatically reset if the AES interrupt is serviced, if AESADOUT is read, or if AESADIN or AESAKEY are written. AESRDYIFG is reset after a PUC or with AESSWRST = 1. AESRDYIE is reset after a PUC but is not reset by AESSWRST = 1.

1.2.6 Implementing Block Cipher Modes

All block cipher modes must be implemented in software. The AES accelerator supports only encrypt and decrypt functionality.

1.3 AES_ACCEL Registers

The AES Accelerator registers are listed in Table 1-1.

Table 1-1. AES_ACCEL Registers

Offset	Acronym	Register Name	Type	Access	Reset	Section
000h	AESACTL0	AES accelerator control register 0	Read/write	Word	00h	Section 1.3.1
002h	Reserved					
004h	AESASTAT	AES accelerator status register	Read only	Word	00h	Section 1.3.2
006h	AESAKEY	AES accelerator key register	Read/write	Word	00h	Section 1.3.3
008h	AESADIN	AES accelerator data in register	Read/write	Word	00h	Section 1.3.4
00Ah	AESADOUT	AES accelerator data out register	Read/write	Word	00h	Section 1.3.5

1.3.1 AESACTL0 Register

AES Accelerator Control Register 0

AESACTL0 is shown in Figure 1-6 and described in Table 1-2.

Figure 1-6. AESACTL0 Register

15	14	13	12	11	10	9	8
Reserved			AESRDYIE	AESERRFG	Reserved		AESRDYIFG
r0	r0	r0	rw-0	rw-0	r0	r0	rw-0
7	6	5	4	3	2	1	0
AESSWRST	Reserved					AESOPx	
rw-0	r0	r0	r0	rw-0	rw-0	rw-0	rw-0

Table 1-2. AESACTL0 Register Description

Bit	Field	Type	Reset	Description
15-13	Reserved	R	0h	Reserved
12	AESRDYIE	RW	0h	AES ready interrupt enable. AESRDYIE is not reset by AESSWRST = 1. 0 = Interrupt disabled 1 = Interrupt enabled
11	AESERRFG	RW	0h	AES error flag. AESAKEY or AESADIN were written while an AES operation was in progress. The bit must be cleared by software. 0 = No error 1 = Error occurred
10-9	Reserved	R	0h	Reserved
8	AESRDYIFG	RW	0h	AES ready interrupt flag. Set when the selected AES operation was completed and the result can be read from AESADOUT. Automatically cleared when AESADOUT is read or AESAKEY or AESADIN is written. 0 = No interrupt pending 1 = Interrupt pending
7	AESSWRST	RW	0h	AES software reset. Immediately resets the complete AES accelerator module even when busy except for the AESRDYIE and AESOPx bits. The AESSWRST bit is automatically reset and always reads as zero. 0 = No reset 1 = Reset AES accelerator module
6-2	Reserved	R	0h	Reserved
1-0	AESOPx	RW	0h	AES operation. The AESOPx bits are not reset by AESSWRST = 1. 00 = Encryption 01 = Decryption. The provided key is the same key used for encryption. 10 = Generate first round key required for decryption. 11 = Decryption. The provided key is the first round key required for decryption.

1.3.2 AESASTAT Register

AES Accelerator Status Register

AESASTAT is shown in Figure 1-7 and described in Table 1-3.

Figure 1-7. AESASTAT Register

15	14	13	12	11	10	9	8
AESDOUTCNTx				AESDINCNTx			
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
7	6	5	4	3	2	1	0
AESKEYCNTx				AESDOUTRD	AESDINWR	AEKEYWR	AESBUSY
r-0	r-0	r-0	r-0	r-0	rw-0	rw-0	r-0

Table 1-3. AESASTAT Register Description

Bit	Field	Type	Reset	Description
15-12	AESDOUTCNTx	R	0h	Bytes read from AESADOUT. Reset when AESDOUTRD is reset. If AESDOUTCNTx = 0 and AESDOUTRD = 0, no bytes were read. If AESDOUTCNTx = 0 and AESDOUTRD = 1, all bytes were read.
11-8	AESDINCNTx	R	0h	Bytes written by AESADIN. Reset when AESDINWR is reset. If AESDINCNTx = 0 and AESDINWR = 0, no bytes were written. If AESDINCNTx = 0 and AESDINWR = 1, all bytes were written.
7-4	AESKEYCNTx	R	0h	Bytes written by AESAKEY. Reset when AESKEYWR is reset. If AESKEYCNTx = 0 and AESKEYWR = 0, no bytes were written. If AESKEYCNTx = 0 and AESKEYWR = 1, all bytes were written.
3	AESDOUTRD	R	0h	All 16 bytes read from AESADOUT. AESDOUTRD is reset by PUC, AESSWRST, an error condition, changing AESOPx, when the AES accelerator is busy, and when the output data is read again. 0 = Not all bytes read 1 = All bytes read
2	AESDINWR	RW	0h	All 16 bytes written to AESADIN. This bit can be modified by software. Changing its state by software also resets the AEDINCNTx bits. AESDINWR is reset by PUC, AESSWRST, an error condition, changing AESOPx, starting to write or overwrite the data, and when the AES accelerator is busy. Because this bit is reset when AESOPx is changed, AESDINWR can be set by software again to indicate that the current data is still valid. 0 = Not all bytes written 1 = All bytes written
1	AESKEYWR	RW	0h	All 16 bytes written to AESAKEY. This bit can be modified by software. Changing its state by software also resets the AESKEYCNTx bits. AESKEYWR is reset by PUC, AESSWRST, an error condition, changing AESOPx, and starting to write or overwrite a new key. Because it is reset when AESOPx is changed it can be set by software again to indicate that the loaded key is still valid. 0 = Not all bytes written 1 = All bytes written
0	AESBUSY	R	0h	AES accelerator module busy; encryption, decryption, or key generation in progress. 0 = Not busy 1 = Busy

1.3.3 AESAKEY Register

AES Accelerator Key Register

AESAKEY is shown in Figure 1-8 and described in Table 1-4.

Figure 1-8. AESAKEY Register

15	14	13	12	11	10	9	8
AESKEY1x (Key Byte n+1)							
w-0	w-0	w-0	w-0	w-0	w-0	w-0	w-0
7	6	5	4	3	2	1	0
AESKEY0x (Key Byte n)							
w-0	w-0	w-0	w-0	w-0	w-0	w-0	w-0

Table 1-4. AESAKEY Register Description

Bit	Field	Type	Reset	Description
15-8	AESKEY1x	W	0	AES key byte n+1 when AESAKEY is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero. The key is reset by PUC or by AESSWRST = 1.
7-0	AESKEY0x	W	0	AES key byte n when AESAKEY is written as word. AES next key byte when AESAKEY_L is written as byte. Do not mix word and byte access. Always reads as zero. The key is reset by PUC or by AESSWRST = 1.

1.3.4 AESADIN Register

AES Accelerator Data In Register

AESADIN is shown in Figure 1-9 and described in Table 1-5.

Figure 1-9. AESADIN Register

15	14	13	12	11	10	9	8
AESDIN1x (DIN Byte n+1)							
w-0	w-0	w-0	w-0	w-0	w-0	w-0	w-0
7	6	5	4	3	2	1	0
AESDIN0x (DIN Byte n)							
w-0	w-0	w-0	w-0	w-0	w-0	w-0	w-0

Table 1-5. AESADIN Register Description

Bit	Field	Type	Reset	Description
15-8	AESDIN1x	W	0	AES data in byte n+1 when AESADIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.
7-0	AESDIN0x	W	0	AES data in byte n when AESADIN is written as word. AES next data in byte when AESADIN_L is written as byte. Do not mix word and byte access. Always reads as zero.

1.3.5 AESADOUT Register

AES Accelerator Data Out Register

AESADOUT is shown in Figure 1-10 and described in Table 1-6.

Figure 1-10. AESADOUT Register

15	14	13	12	11	10	9	8
AESDOUT1x (DOUT Byte n+1)							
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
7	6	5	4	3	2	1	0
AESDOUT0x (DOUT Byte n)							
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0

Table 1-6. AESADOUT Register Description

Bit	Field	Type	Reset	Description
15-8	AESDOUT1x	R	0	AES data out byte n+1 when AESADOUT is read as word. Do not use these bits for byte access. Do not mix word and byte access.
7-0	AESDOUT0x	R	0	AES data out byte n when AESADOUT is read as word. AES next data out byte when AESADOUT_L is read as byte. Do not mix word and byte access.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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