32-bit RISC Microcontroller

TXZ+ Family

Reference Manual
FIR calculation Circuit
(FIR-A)

Revision 1.0

2020-10

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Contents

List of Figures

List of Tables

Preface

Related document

Document name
Clock Control and Operation Mode
Product Information
Exception
Multi-Function DMA Controller
I^2S Interface

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3: 0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1]***, ***[XYZ2]***, ***[XYZ3]*** → ***[XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C ...
  Example: ***[ADACR0]***, ***[ADBCR0]***, ***[ADCCR0]*** → ***[ADxCR0]***
  In case of channel, "x" means 0, 1, and 2 ...
  Example: ***[T32A0RUNA]***, ***[T32A1RUNA]***, ***[T32A2RUNA]*** → ***[T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> =0x01 (hexadecimal), ***[XYZn]***<VW> =1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value.
  In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

Terms and Abbreviations

Some of abbreviations used in this document are as follows:

AHB	Advanced High-performance Bus
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
FIFO	First-In First-Out
FIR	Finite Impulse Response
I^2S	Inter-IC Sound
LSB	Least Significant Bit
MSB	Most Significant Bit
TRGSEL	Trigger Selection circuit

1. Outlines

The FIR calculation circuit (FIR) is a function dedicated to I²S. The main functions of FIR calculation circuit are as follows.

Table 1.1 Function outline

Function category	Function	Description
Calculation	Sum-of-products arithmetic processing	When the data is written in a data buffer, the data performs sum-of-products arithmetic processing with filter factor set up beforehand.
	Selection of tap number	Tap number can be set to 1 to 128. (Set value +1 is the actual number of tap.)
	Selection of input data processing	Calculation processing of input data can be selected. - Even numbered data - Odd numbered data - Every data
Data format	Selection of input/output data width	The input/output data width can be selected. (Fixed point number) (Note) - 16-bit - 24-bit - 32-bit
	Selection of number of bits to shift the data	The value of input data which left shifted the setting value is written to a data buffer. (0bit to 16bits can set.)
	Selection of Output data justification	The output format of output data can be selected. - MSB justification - LSB justification
Interlocking function	DMA transfer	The factor of DMA requests are as follows. - Write request of input data - Read request of sum-of-products arithmetic result
	Interruption	The factor of interruptions are as follows - Write input data - Read sum-of-products arithmetic result - Overflow
	Interlocking function of I²S	The trigger of a calculation start can be selected by TRGSEL. - When the I²S receive FIFO exceeds the threshold. - When the I²S transmit FIFO is less than the threshold.
		Only one of L and R of I²S audio data performs calculation.
Special function	Batch initialization	Before the calculation, the data buffer for the number of taps is initialized to the same value at once.

Note: For the fixed point number, the most significant bit expresses a sign and less than it expresses the data of a decimal part.

2. Configuration

The FIR calculation circuit consists of coefficient buffer, data buffer, sum-of-products arithmetic unit, DMA / interrupt control circuit, and each registers.

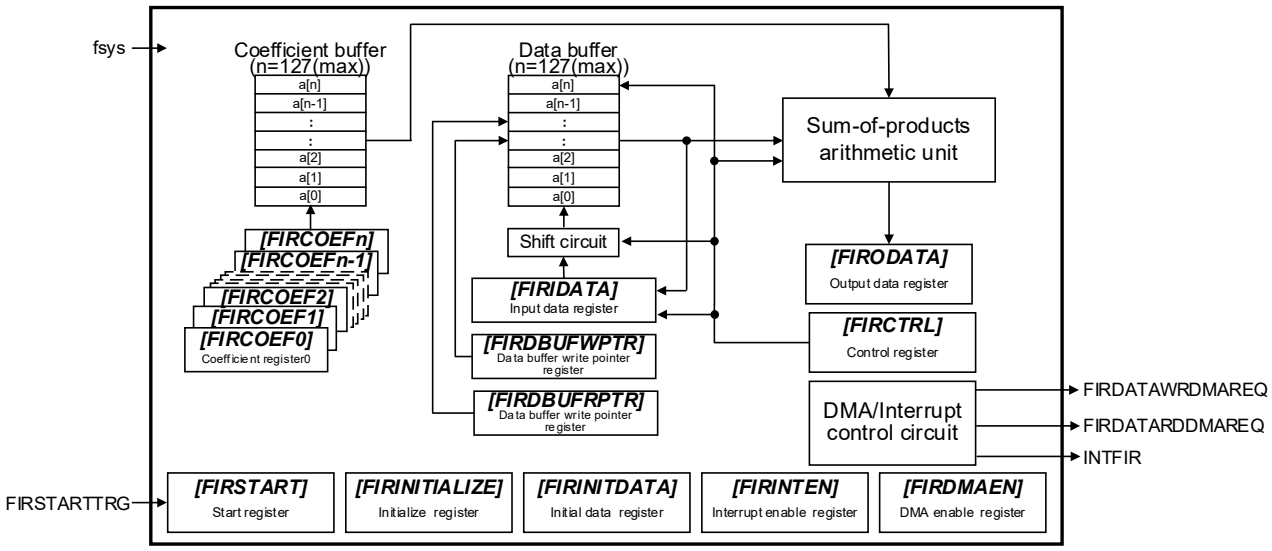

Figure 2.1 FIR calculation circuit block diagram

Table 2.1 List of Signals

No	Symbol	Signal name	I/O	Related Reference manual
1	fsys	System clock	Input	Clock Control and Operation Mode
2	FIRDATAWRDMAREQ	Input data write request	Output	Multi-Function DMA Controller, Product Information
3	FIRDATARDDMAREQ	Arithmetic result data read request	Output	Multi-Function DMA Controller, Product Information
4	INTFIR	Interrupt request signal	Output	Exception
5	FIRSTARTTRG	Arithmetic start trigger signal	Input	Product Information

3. Function

3.1. Clock Supply

When FIR is used, the corresponding clock enable bits should be set to "1" (Clock supply) in fsys supply stop register A (***[CGFSYSENA]*** and ***[CGFSYSMENA]***), fsys supply stop register B (***[CGFSYSENB]*** and ***[CGFSYSMENB])***, f sys supply stop register C (***[CGFSYSMENC]***), and fc supply stop register (***[CGFCEN]***).

The corresponding registers and the bit locations depend on a product. Some products do not have all registers. For the details, refer to "Clock Control and Operation Mode" in Reference manual.

3.2. Operation overview

FIR calculation circuit performs sum-of-products arithmetic processing with up to 128 taps. And the arithmetic result can be read via a register.

3.2.1. Operation when I²S data is received

As an example, Figure 3.1 shows the operation when I²S data is received using the DMAC.

First, set the trigger selection of FIR processing starting (refer to “3.3.1. Trigger selection of FIR processing start”), each control register and filter coefficient, and write "1" to ***[FIRSTART]***<START>. This makes it ready to receive input data. When the number of stages in the I²S receive data FIFO exceeds the threshold,

(1) I²S asserts the receive FIFO status signal of I²S.
(At this time, if the FIR calculation circuit can input data (after the previous sum-of-products arithmetic is completed and the arithmetic result is read))
(2) FIR generates DMA request.
(3) The data is transferred from I²S data FIFO to data buffer (***[FIRIDATA]***<IDATA[31:0]>) of FIR calculation circuit by DMAC.
(4) When data is written to the data buffer, sum-of-products arithmetic (Note) is performed with a pre-set filter coefficient.
(5) After the sum-of-products arithmetic is completed, the FIR calculation circuit generates a DMA request.
(6) The data is transferred from ***[FIRODATA]***<ODATA[31:0]> of arithmetic result to SRAM by DMAC.

The above operation is repeated until order to stop.

To stop the FIR calculation circuit, write "0" to ***[FIRSTART]***<START>. After the internal processing is completed, ***[FIRSTART]***<START> changes from "1" to "0".

Note: Since only one sum-of-products arithmetic unit is contained in the FIR calculation circuit, the sum-of-products arithmetic processing requires cycles for the number of taps.

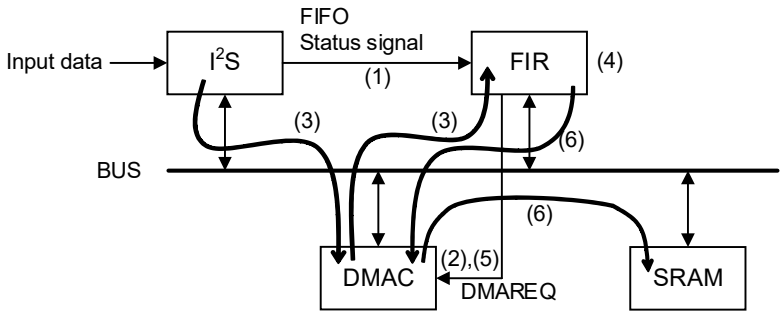

Figure 3.1 Operation when I²S data is received

3.2.2. Operation when I²S data is transmitted

As an example, the operation when I²S data is transmitted using the DMAC is shown.

Figure 3.2 shows the operation of transfer from SRAM to I²S using DMAC. After setting the same as I²S data reception, when the number of stages in I²S transmit FIFO is less than the threshold,

(1) I²S asserts the transmit FIFO status signal of I²S.
(The same as I²S data reception, if the FIR calculation circuit can input data.)
(2) FIR generates DMA request.
(3) Data stored in SRAM is transferred to data buffer ***([FIRIDATA]***<IDATA[31:0]>) of FIR calculation circuit by DMAC.
(4) When data is written to the data buffer, sum-of-products arithmetic (Note) is performed with a pre-set filter coefficient.
(5) After the sum-of-products arithmetic is completed, the FIR calculation circuit generates a DMA request.
(6) The data is transferred from arithmetic result to I²S by DMAC.

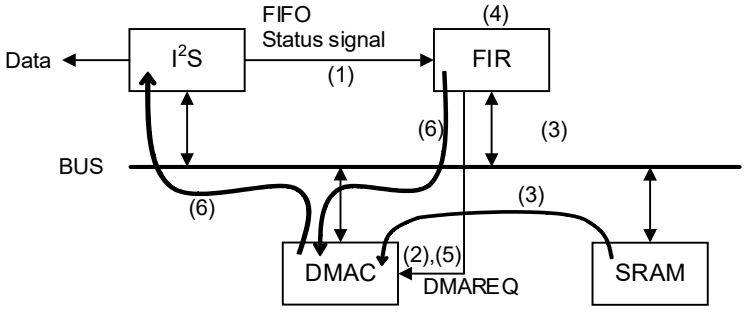

Figure 3.2 Operation when I²S data is transmitted

Note: Since only one sum-of-products arithmetic unit is contained in the FIR calculation circuit, the sum-of-products arithmetic processing requires cycles for the number of taps.

3.2.3. Timing chart of FIR processing

Figure 3.3 shows timing chart of FIR processing. The read of data buffer, sum-of products arithmetic and result writing are performed by pipeline processing.

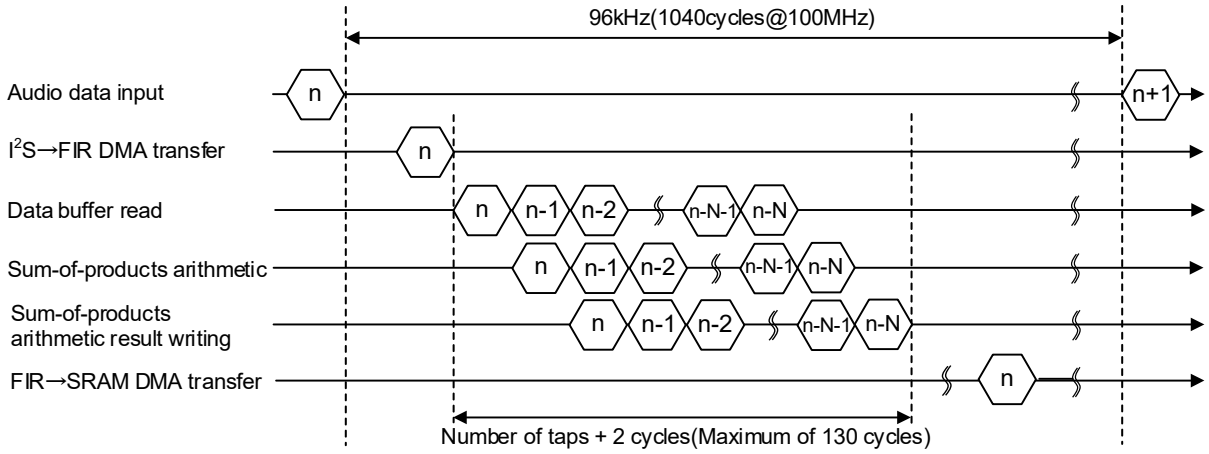

Figure 3.3 Timing chart of FIR processing

3.3. Setting

3.3.1. Trigger selection of FIR processing start

The FIR process start trigger can be selected by setting of TRGSEL (Note).

- When the I^{2}S receive FIFO exceeds the threshold (I^{2}S receive FIFO status signal)
- When the I^{2}S transmit FIFO is less than the threshold (I^{2}S transmit FIFO status signal)

Note: For details, refer to "Product Information" of Reference Manual.

Since the FIR is shared for transmission and receiving, when switching between them, it is necessary to change the settings of TRGSEL, FIR, and DMAC each time.

3.3.2. Batch initialization of data buffer

Before the FIR calculation process, the data buffer entries for the number of taps can be initialized to the same value at once. To start initialization, write the initial value to ***[FIRINITDATA]***<INITDATA[31:0]>, then write "1" to ***[FIRINITIALIZE]***<INITIALISE>. The <INITIALISE> holds "1" during initialization, and becomes "0" after initialization is completed. The number of entries to be initialized is the set value of ***[FIRCTRL]***<TAPNUM[6:0]> +1, so it is necessary to fix <TAPNUM[6:0]> before initialization. However, the initial value of <TAPNUM[6:0]> is maximum in case value of <TAPNUM[6:0]> is not fixed.

When "1" is written to ***[FIRSTART]***<START> during ***[FIRINITIALIZE]***<INITIALISE> is "1", the FIR will be started after initialization is completed. Therefore, it is not necessary to wait for the initialization to be completed by software.

3.3.3. Access to arbitrary data of data buffer

The data buffer has a write pointer that indicates the entry to be written next and a read pointer that indicates the entry to be read next. By manipulating these pointers, arbitrary data can be accessed individually. After writing to ***[FIRIDATA]***<IDATA[31:0]>, the write pointer is incremented and next entries can be written continuously. Also, after reading ***[FIRIDATA]***<IDATA[31:0]>, the read pointer is decremented, and the data can be read one after another in the new order. After writing, the read pointer shows the value of the write pointer before the increment, that is, the read pointer shows the entry that was written.

Table 3.1 Operation of data buffer pointer

	Write	Read
Write pointer	Increment (It becomes "0x0" at the next write that reaches the ***[FIRCTRL]***<TAPNUM[6:0]> set value.)	No change
Read pointer	Value of the write pointer before increment	Decrement (At the next read that reaches 0x0, it becomes ***[FIRCTRL]***<TAPNUM[6:0]> set value.)

The write pointer and read pointer can be rewritten directly by the ***[FIRDBUFWPTR]*** and ***[FIRDBUFRPTR]*** registers respectively (Note1). After rewriting the pointer, arbitrary entry in the data buffer can be accessed by writing / reading ***[FIRIDATA]***<IDATA[31:0]> (Note2).

Note1: Do not write to the pointer of data buffer during FIR processing.

Note2: Access to the data buffer cannot be performed during initialization by ***[FIRINITIALIZE]***<INITIALIZE> in “3.3.2. Batch initialization of data buffer”, even when FIR processing is stopped. Therefore, after initializing the data buffer by <INITIALIZE>, when rewriting the value of arbitrary entry, wait until <INITIALIZE> becomes "0".

The write pointer is automatically initialized to "0" when FIR processing starts. Therefore, the sum-of-products arithmetic after writing (Note) the first data is given by the following formula.

a[0]*x[0]+a[1]*x[N]+ ...+a[N]*x[1]

The next data is written to x[1], and the sum-of-products arithmetic is given by the following formula.

a[0]*x[1]+a[1]*x[0]+a[2]*x[N]+...+a[N]*x[2]

a[n]: The value of the n-th entry in the coefficient buffer
x[n]: The value of the n-th entry in the data buffer
N: The value of ***[FIRCTRL]***<TAPNUM[6:0]>

Note: The first data is written to x[0]. Therefore, x[0] is not necessary to set the initial value.

When FIR processing is completed, the read pointer shows the last written entry. Therefore, the write data can be read in a new order by reading ***[FIRIDATA]***<IDATA [31: 0]> after the completion of FIR processing.

3.3.4. Data format

The format of each data and control register are shown in Table 3.2.

Table 3.2 Data format

Data	Format	Control register
Coefficient(***[FIRCOEF]***<COEF[15:0]>)	16-bit fixed point number (Note1)	-
Input data(***[FIRIDATA]***<IDATA[31:0]>)(Note2)/ Initial value(***[FIRINITDATA]***<INITDATA[31:0]>)	16/24/32-bit fixed point number (Note1)	***[FIRCTRL]*** <IDATASIZE[2:0]>
Output data(***[FIRODATA]***<ODATA[31:0]>)(Note2)	16/24/32-bit fixed point number (Note1)	***[FIRCTRL]*** <ODATASIZE[2:0]>

Note1: For the fixed point number, the most significant bit expresses a sign and less than it expresses the data of a decimal part.

Note2: The coefficient is 16 bits, but the input / output data can also be selected 24 bits or 32 bits. In this case, the arithmetic result with 16 bits or less of the decimal point includes an error.

3.3.5. Bit shift function of input data

As shown in Figure 3.4, when input data is written to ***[FIRIDATA]***<IDATA[31:0]>, the data which left shifted the bits quantity (0 to 16 bits) specified by ***[FIRCTRL]***<IDATASFTAMT[4:0]> is stored in the data buffer.

[FIRIDATA]<IDATA[31:0]> will read the data stored in the data buffer.

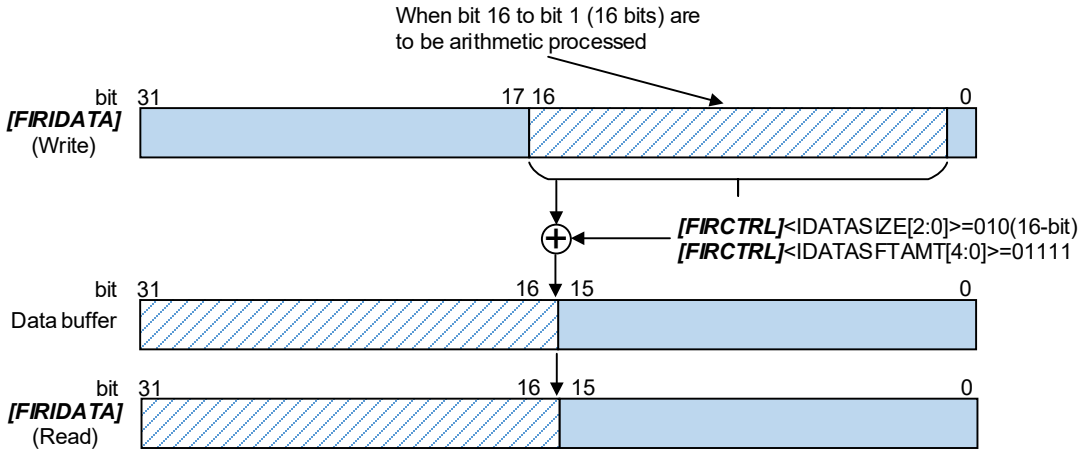

Figure 3.4 Bit shift function of input data

Set ***[FIRCTRL]***<IDATASFTAMT[4:0]> so that the data written in the data buffer is MSB side justification (refer to Figure 3.5).

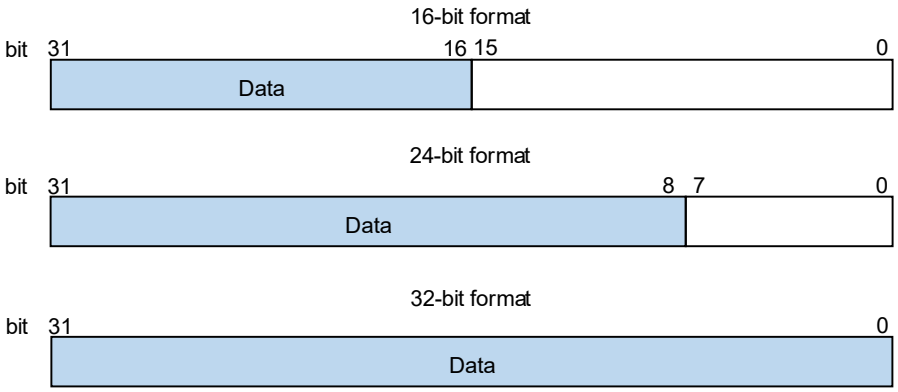

Figure 3.5 Format of input data (MSB justification)

3.3.6. Format of output data

As shown in Figure 3.6, the output data (***[FIRODATA]***<ODATA[31:0]>) can be selected MSB justification or LSB justification by ***[FIRCTRL]***<ODATAFMT>.

Note: In case of LSB justification, sign extension is not performed.

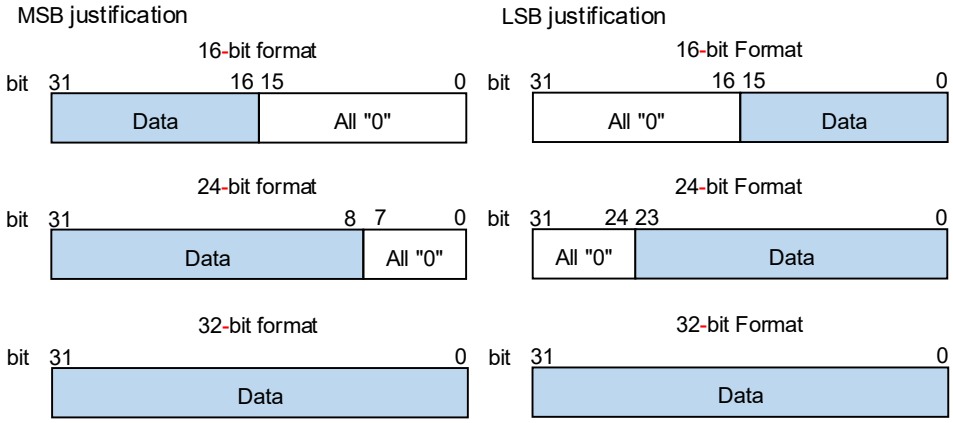

Figure 3.6 Format of output data

3.4. Interrupt

Interrupt is generated by the following factors.

- Input data write request
- Sum-of-products arithmetic result read request
- Sum-of-products arithmetic overflow occurred

When each factor occurs, the corresponding bit of ***[FIRRAWINTSTAT]*** is set to "1". At the time, if the corresponding bit of ***[FIRINTEN]*** is "1", an interrupt is generated. The interrupt is level signal. The interrupt is cleared by writing "1" to the corresponding bit of ***[FIRRAWINTSTAT]***.

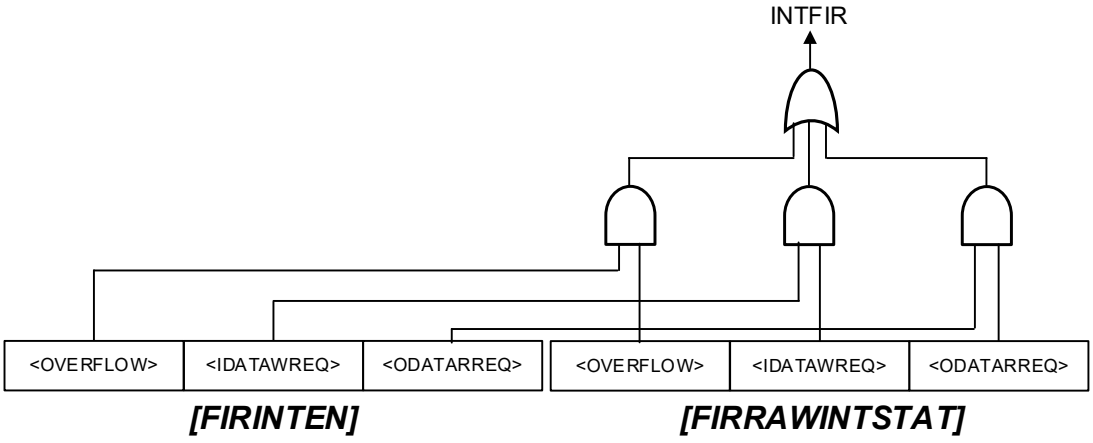

Figure 3.7 Generation logic of interrupt

3.4.1. Input data write request interrupt

When ***[FIRINTEN]***<IDATAWREQ> is set to "1", input data write request interrupt is enabled. When receive and transmit FIFO status signals of I^2S are asserted, if the FIR calculation circuit can receive data (cannot receive when data buffer access occurs during initialization or sum-of-products arithmetic), the interrupt is generated.

3.4.2. Sum-of-products arithmetic result read request interrupt

When ***[FIRINTEN]***<ODATARREQ> is set to "1", sum-of-products arithmetic result read request interrupt is enabled. The Interrupt is generated when sum-of-products arithmetic is completed.

3.4.3. Sum-of-products arithmetic overflow interrupt

When ***[FIRINTEN]***<OVERFLOW> is set to "1", sum-of-products arithmetic overflow interrupt is enabled. When the overflow occurs during sum-of-product arithmetic, the interrupt is generated. Overflow occurs under any of the following conditions. Overflow interrupt occurs when sum-of-products arithmetic is completed.

- Halfway value of sum-of-products arithmetic ≥ 2
- Halfway value of sum-of-products arithmetic < -2
- Sum-of-products arithmetic result ≥ 1
- Sum-of-products arithmetic result < -1

The arithmetic result (value of ***[FIRODATA]***<ODATA[31:0]>) when overflow occurs cannot be used because the upper digits are lost and the result is incorrect.

3.5. DMA request

The FIR calculation circuit generates each DMA request signal, when both each bit of ***[FIRDMAEN]*** and the factor (1 cycle pulse) for asserting the corresponding bit of ***[FIRRAWINTSTAT]*** is "1". The following are factors of DMA request.

- Input data write request
- Sum-of-products arithmetic result read request

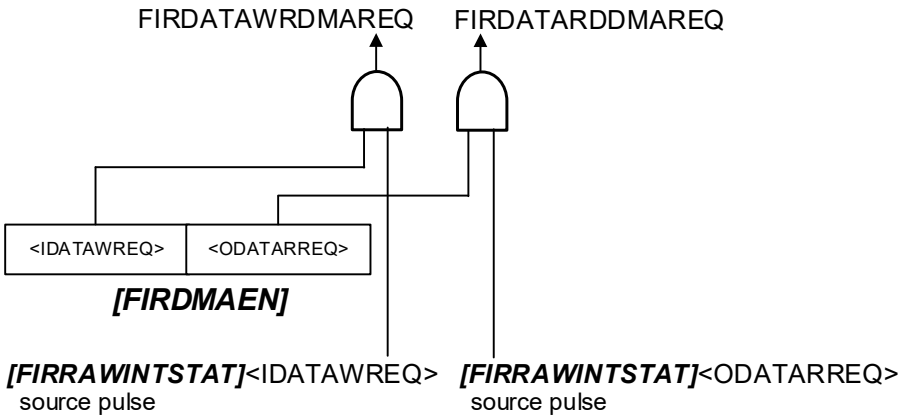

Figure 3.8 Generation logic of DMA request

NOTE: When the bit of ***[FIRDMAEN]*** is "1", the <IDATAWREQ> and <ODATARREQ> of ***[FIRRAWINTSTAT]*** are not set to "1" even if the valid factor occurs.

3.6. Operation flow and processing procedure

3.6.1. Operation flow

Figure 3.9 and Figure 3.10 show the operation flow.

(1) When processing by DMAC

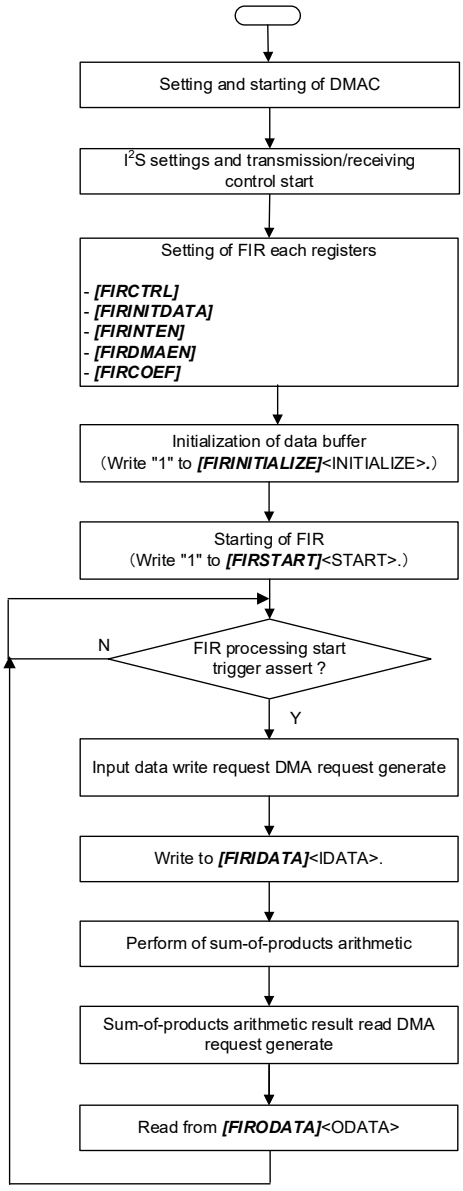

Figure 3.9 Operation flow (1)

(2) When processing by interrupt

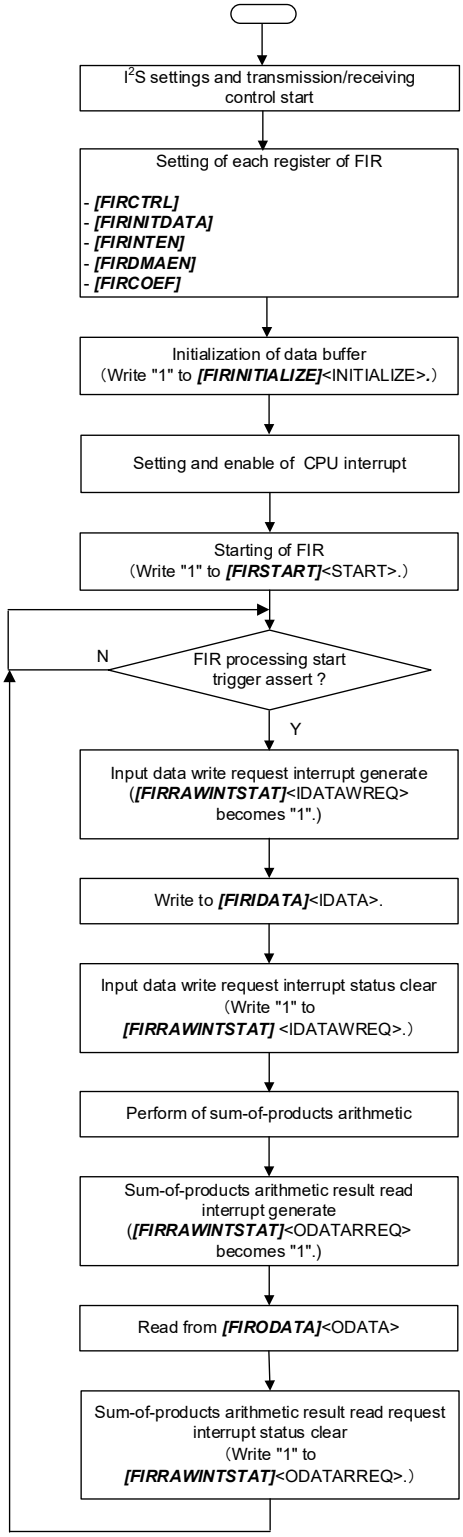

Figure 3.10 Operation flow (2)

NOTE: For details on interrupt, refer to "Exception" of Reference Manual.

3.6.2. Stopping procedure

Table 3.3 shows the stop procedure of FIR calculation circuit. The FIR calculation circuit stops by writing "0" to ***[FIRSTART]***<START> at any time. At that time, input data write request (or sum-of-products arithmetic result read request) DMA request / interrupt is generated, and when the FIR is waiting for writing to ***[FIRIDATA]***<IDATA[31:0]> (or reading ***[FIRODATA]***<ODATA[31:0]>), it will stop after writing (or reading).

Then stop the DMAC.

In case of processing by the DMAC, during FIR calculation circuit is waiting for the writing to <IDATA[31:0]> (or reading from <ODATA[31:0]>), when DMAC has stopped previously, CPU must write to <IDATA[31:0]> (or reading from <ODATA[31:0]>). In this case, it is necessary to distinguish the status using ***[FIRSEQSTAT]***<SEQSTAT[2:0]> and perform the appropriate processing (writing to <IDATA[31:0]> or reading from <ODATA[31:0]>).

Table 3.3 Stopping procedure

	Processing
1	Write "0" to ***[FIRSTART]***<START> at any timing.
2	Polling until <START> becomes "0".
3	Stop I^2S.
4	Stop DMAC (when processing by DMAC).

3.6.3. Processing when I^2S is received

When transferring data received by I^2S to the FIR calculation circuit by DMAC, the processing method differs depending on the data format and data width.

3.6.3.1. Processing when 24-bit and 32-bit stereo data is received

In the 32-bit and 24-bit stereo formats, L and R are alternated one word at a time as shown in Figure 3.11, and two words are transferred from I^2S to ***[FIRIDATA]***<IDATA[31:0]> by the DMAC, and one word of them is used for sum-of-products arithmetic.

Set ***[FIRCTRL]***<IDATASEL[1:0]> to "10" and set the DMAC unit transfer length to two words and fixed address transfer. When processing even-numbered data (with 0 origin) of I^2S reception data, set the transfer destination address of LMEM0(L) to ***[FIRIDATA]***. LMEM1(R) is transferred to ***[FIRIDATA]***+0x4, but address of ***[FIRIDATA]***+0x4 is address of ***[FIRODATA]***, not effective when writing. When processing odd-numbered data, set the transfer destination address of LMEM0(L) to ***[FIRIDATA]***-0x4. LMEM0(L) is transferred to ***[FIRIDATA]***-0x4, but address of ***[FIRIDATA]***-0x4 is the reserved address, not effective when writing. Therefore, either L or R data to be processed is written to ***[FIRIDATA]***<IDATA[31:0]>.

Note: The DMAC contained in this product increments or decrements the address within a unit transfer even if the address fixed transfer is set.

3.6.3.2. Processing when 16-bit stereo data is received

In the case of 16-bit stereo format, the upper or lower 16 bits in a word are L or R data as shown in Figure 3.11. In this case, you can select either one by using the bit shift function of input data in “3.3.5. Bit shift function of input data”. As an example, to select R data in Upper First, set ***[FIRCTRL]***<IDATASFTAMT[4:0]> to “10000” (16 bits left shift).

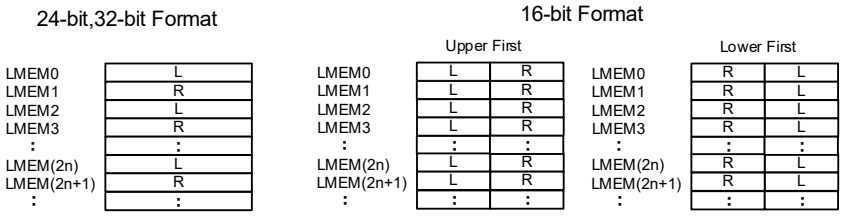

Figure 3.11 Example of I²S data format

3.6.3.3. Processing when 24-bit and 32-bit monaural data is received

Set ***[FIRCTRL]***<IDATASEL[1:0]> to "10" and set the DMAC unit transfer length to one word. Every one word is written to ***[FIRIDATA]***<IDATA[31:0]>, sum-of-products arithmetic is performed.

3.6.3.4. Processing when 16-bit monaural data is received

Set the data width of I²S to 24-bit or 32-bit and receive one data, and process it as 16-bit data with the FIR calculation circuit.

3.6.4. Processing when I²S is transmitted

When transmitting data of I²S processed by the FIR calculation circuit, the processing method differs depending on the data format and data width.

3.6.4.1. Processing when 24-bit and 32-bit stereo data is transmitted

In the case of 32-bit and 24-bit stereo formats, data transfer from SRAM to FIR calculation circuit performs FIR processing for one by one word, as shown in Figure 3.12.

When transferring FIR processed data to I²S, set the unit transfer length of the DMAC to two words and fixed address transfer. When processing even-numbered data (with 0 origin) of I²S transmission data, set the transfer source address to ***[FIRODATA]***. The data of address of ***[FIRODATA]***+0x4 is transferred to R Data 0, but address of ***[FIRODATA]***+0x4 is the reserved address, not effective when reading. When processing odd-numbered transmission data, set the transfer source address to ***[FIRODATA]***-0x4. ***[FIRODATA]***-0x4 is transferred to L Data 0, but address of ***[FIRODATA]***-0x4 is the address of ***[FIRIDATA]***, not effective when reading after FIR starting.

Therefore, either L or R data to be transmitted is read from ***[FIRODATA]***<ODATA[31:0]>. Select the data that you want to process on the device connected by I²S.

Note: The DMAC contained in this product increments or decrements the address within a unit transfer even if the address fixed transfer is set.

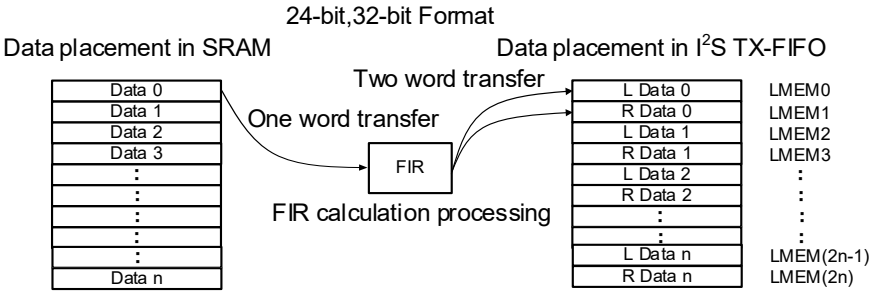

Figure 3.12 Processing when 24-bit and 32-bit stereo data is transmitted

3.6.4.2. Processing when 16-bit stereo data is transmitted

In the case of 16-bit stereo format, data transfer from the SRAM to the FIR calculation circuit is performed half word at a time and FIR processing is performed, as shown in Figure 3.13.

The sum-of-products arithmetic result is justified to the MSB or LSB according to the setting of ***[FIRCTRL]***<ODATAFMT>. When these sum-of-products arithmetic results are transferred to the I2S TX FIFO, the value of the arithmetic result register is transferred by one word, and then the I2S TX FIFO is transmitted with the aligned data which only L or R as shown in Figure 3.13. Select the L or R that has data on the device connected by I2S.

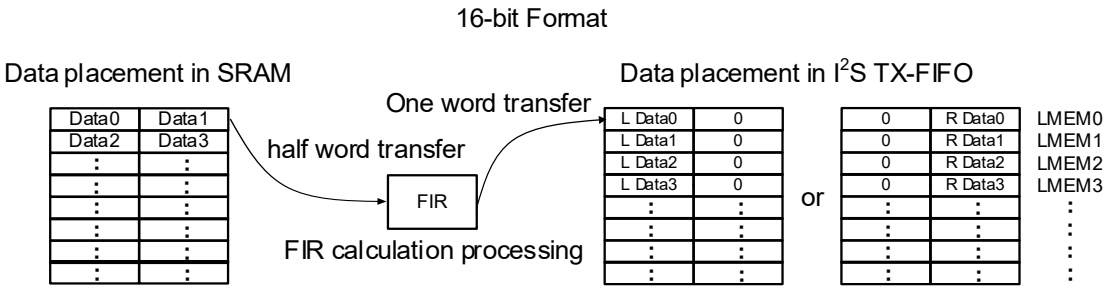

Figure 3.13 Processing when 16-bit stereo data is transmitted

3.6.4.3. Processing when 24-bit and 32-bit monaural data is transmitted

When the data is transmitted by 24-bit or 32-bit monaural format, transfer from FIR to I2S for one by one word in the same way as when 16-bit stereo data is transmitted.

3.6.4.4. Processing when 16-bit monaural data is transmitted

When the data is transmitted by 16-bit monaural format, set the data width of I2S to 24-bit or 32-bit and transmit one data. Select the 16-bit data that you want to process on the device connected by I2S.

4. Registers

4.1. List of Registers

The FIR registers and their addresses are shown as follows.

Function name		Channel/Unit	Base address		
			TYPE1	TYPE2	TYPE3
FIR calculation circuit	**FIR**	-	-	0x400DD000	-

Register name		Address (Base+)
Start register	***[FIRSTART]***	0x0000
Control register	***[FIRCTRL]***	0x0004
Initialize register	***[FIRINITIALIZE]***	0x0008
Initial data register	***[FIRINITDATA]***	0x000C
Interrupt enable register	***[FIRINTEN]***	0x0010
DMA enable register	***[FIRDMAEN]***	0x0014
Interrupt status register	***[FIRRAWINTSTAT]***	0x0018
Sequencer status register	***[FIRSEQSTAT]***	0x001C
Data buffer write pointer register	***[FIRDBUFWPTR]***	0x0020
Data buffer read pointer register	***[FIRDBUFRPTR]***	0x0024
Input data register	***[FIRIDATA]***	0x0030
Output data register	***[FIRODATA]***	0x0034
Coefficient register	***[FIRCOEFn]***	0x0040 to 0x023C

4.2. Details of register

4.2.1. *[FIRSTART]* (Start register)

Bit	Bit Symbol	After Reset	Type	Function
31:1	-	0	R	Read as "0".
0	START	0	W	FIR control 0: Stop FIR 1: Start FIR
			R	Operating status 0: FIR stopping 1: FIR operating

Note: When "1" is written to <START> while ***[FIRINITIALIZE]***<INITIALISE> is "1", FIR will be started after initialization is completed.

4.2.2. *[FIRCTRL]* (Control register)

Bit	Bit Symbol	After Reset	Type	Function
31:29	-	0	R	Read as "0".
28	ODATAFMT	0	R/W	Selection of output data justification 0: MSB justification 1: LSB justification
27	-	0	R	Read as "0".
26:24	ODATASIZE[2:0]	000	R/W	Selection of output data width 010: 16-bit 011: 24-bit 100: 32-bit Other settings are prohibited.
23:22	-	0	R	Read as "0".
21:20	IDATASEL[1:0]	00	R/W	Selection of input data processing 00: Process even numbered data 01: Process odd numbered data 10: Process every data Other settings are prohibited.
19:17	-	0	R	Read as "0".
16:12	IDATASFTAMT [4:0]	00000	R/W	Selection of number of bits to shift the input data 0bit to 16bits can set. 17bits to 31bits are prohibited.
11	-	0	R	Read as "0".
10:8	IDATASIZE[2:0]	000	R/W	Selection of intput data width 010: 16-bit 011: 24-bit 100: 32-bit Other settings are prohibited.
7	-	0	R	Read as "0".
6:0	TAPNUM[6:0]	0x7F	R/W	Selection of tap number Set value + 1 is the tap number (Tap number can be set from 1 to 128)

Note: Set this register when ***[FIRSTART]***<START> = 0.

4.2.3. *[FIRINITIALIZE]* (Initialize register)

Bit	Bit Symbol	After Reset	Type	Function
31:1	-	0	R	Read as "0".
0	INITIALIZE	0	W	Initialization control 0: Not effect 1: Initialization
			R	Operating status 0: Not initializing 1: Initializing

Note: Set this register when ***[FIRSTART]***<START> = 0.

4.2.4. *[FIRINITDATA]* (Initial data register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	INITDATA[31:0]	0x00000000	R/W	Data initial value

Note: Set this register when ***[FIRSTART]***<START> = 0.

4.2.5. *[FIRINTEN]* (Interrupt enable register)

Bit	Bit Symbol	After Reset	Type	Function
31:3	-	0	R	Read as "0".
2	OVERFLOW	0	R/W	Sum-of-products arithmetic overflow interrupt control 0: Interrupt disabled 1: Interrupt enabled
1	ODATARREQ	0	R/W	Sum-of-products arithmetic result read request interrupt control 0: Interrupt disabled 1: Interrupt enabled
0	IDATAWREQ	0	R/W	Input data write request interrupt control 0: Interrupt disabled 1: Interrupt enabled

Note: Set this register when ***[FIRSTART]***<START> = 0.

4.2.6. *[FIRDMAEN]* (DMA enable register)

Bit	Bit Symbol	After Reset	Type	Function
31:2	-	0	R	Read as "0".
1	ODATARREQ	0	R/W	Sum-of-products arithmetic result read DMA request control 0: Disabled 1: Enabled
0	IDATAWREQ	0	R/W	Input data write DMA request control 0: Disabled 1: Enabled

Note: Set this register when ***[FIRSTART]***<START> = 0.

4.2.7. *[FIRRAWINTSTAT]* (Interrupt status register)

Bit	Bit Symbol	After Reset	Type	Function
31:3	-	0	R	Read as "0".
2	OVERFLOW	0	W	Sum-of-Products arithmetic overflow interrupt clear 0: Not effect 1: Interrupt clear
			R	Sum-of-products arithmetic overflow interrupt status 0: Not occurred 1: Occurred
1	ODATARREQ	0	W	Sum-of-products arithmetic result read request interrupt clear 0: Not effect 1: Interrupt clear
			R	Sum-of-products arithmetic result read request interrupt status 0: Not requested 1: Requested
0	IDATAWREQ	0	W	Input data write request interrupt clear 0: Not effect 1: Interrupt clear
			R	Input data write request interrupt status 0: Not requested 1: Requested

4.2.8. *[FIRSEQSTAT]* (Sequencer status register)

Bit	Bit Symbol	After Reset	Type	Function
31:3	-	0	R	Read as "0".
2:0	SEQSTAT[2:0]	000	R	Operating status 000: Idle 001: Initializing data 010: Waiting for trigger 011: Waiting for input data write 100: Calculating 101: Waiting for output data read Others: reserved

4.2.9. *[FIRDBUFWPTR]* (Data buffer write pointer register)

Bit	Bit Symbol	After Reset	Type	Function
31:7	-	0	R	Read as "0".
6:0	DBUFWPTR[6:0]	0000000	R/W	The write pointer value of data buffer

Note: Set this register when ***[FIRSTART]***<START> = 0.

4.2.10. *[FIRDBUFRPTR]* (Data buffer read pointer register)

Bit	Bit Symbol	After Reset	Type	Function
31:7	-	0	R	Read as "0".
6:0	DBUFRPTR[6:0]	1111111	R/W	The read pointer value of data buffer

Note: Set this register when ***[FIRSTART]***<START> = 0.

4.2.11. *[FIRIDATA]* (Input data register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	IDATA[31:0]	Undefined	W	Writes the data to the address of the data buffer indicated by the write pointer. (Note1), (Note2)
			R	Reads the data at the address of the data buffer indicated by the read pointer. Data can be read in new order of the shifted value.

Note1: The number of stages in the data buffer is ***[FIRCTRL]***<TAPNUM[6:0]> set value + 1, and when written, old data is discarded.

Note2: Do not write except during the input data write request by DMA or interrupt. Do not write except during data initialization before starting.

4.2.12. *[FIRODATA]* (Output data register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	ODATA[31:0]	0x00000000	R	Result of sum-of-products arithmetic

4.2.13. *[FIRCOEF0]* (Coefficient register)

The bit configuration of the ***[FIRCOEF1]*** to ***[FIRCOEF127]*** registers is the same as the ***[FIRCOEF0]*** register.

Bit	Bit Symbol	After Reset	Type	Function
31:16	COEF[15:0]	Undefined	R/W	coefficient
15:0	-	0	R	Read as "0".

Note: Set this register when ***[FIRSTART]***<START> = 0.

5. Revision History

Table 5.1 Revision History

Revision	Date	Description
1.0	2020-10-14	First release

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**