**Document Type: Detail Design**

**Document Name:**

**ADSP FRAMEWORK: TDM CAPTURE CLASS**

Renesas Design Vietnam Co. Ltd

R-Car Software Solution 2 Group

Middleware 1 Team

Renesas Electronics Corporation

Table of Contents

[1. Overview 3](#_Toc534270899)

[2. Function list 4](#_Toc534270900)

[3. Detail information 5](#_Toc534270901)

[3.1 Data type and Macro definition 5](#_Toc534270902)

[3.2 Globle variable 6](#_Toc534270903)

[3.3 Function definition 6](#_Toc534270904)

[3.3.1 xa\_tdm\_capture\_prepare\_runtime 6](#_Toc534270905)

[3.3.2 xa\_tdm\_capture\_sufficient\_outport\_check 8](#_Toc534270906)

[3.3.3 xa\_tdm\_capture\_empty\_this\_buffer 9](#_Toc534270907)

[3.3.4 xa\_tdm\_capture\_fill\_this\_buffer 11](#_Toc534270908)

[3.3.5 xa\_tdm\_capture\_memtab 13](#_Toc534270909)

[3.3.6 xa\_tdm\_capture\_preprocess 15](#_Toc534270910)

[3.3.7 xa\_tdm\_capture\_postprocess 17](#_Toc534270911)

[3.3.8 xa\_tdm\_capture\_flush 19](#_Toc534270912)

[3.3.9 xa\_tdm\_capture\_cleanup 21](#_Toc534270913)

[3.3.10 xa\_tdm\_capture\_terminate 23](#_Toc534270914)

[3.3.11 xa\_tdm\_capture\_destroy 24](#_Toc534270915)

[3.3.12 xa\_tdm\_capture\_factory 25](#_Toc534270916)

[4. Revision history 26](#_Toc534270917)

List of Figures

[Figure 1‑1 The software architecture 3](#_Toc534270918)

[Figure 3‑1 xa\_tdm\_capture\_prepare\_runtime flowchart 7](#_Toc534270919)

[Figure 3‑2 xa\_tdm\_capture\_sufficient\_outport\_check flowchart 8](#_Toc534270920)

[Figure 3‑3 xa\_tdm\_capture\_empty\_this\_buffer flowchart 10](#_Toc534270921)

[Figure 3‑4 xa\_tdm\_capture\_fill\_this\_buffer flowchart 12](#_Toc534270922)

[Figure 3‑5 xa\_tdm\_capture\_memtab flowchart 14](#_Toc534270923)

[Figure 3‑6 xa\_tdm\_capture\_preprocess flowchart 16](#_Toc534270924)

[Figure 3‑7 xa\_tdm\_capture\_postprocess flowchart 18](#_Toc534270925)

[Figure 3‑8 xa\_tdm\_capture\_flush flowchart 20](#_Toc534270926)

[Figure 3‑9 xa\_tdm\_capture\_cleanup flowchart 22](#_Toc534270927)

[Figure 3‑10 xa\_tdm\_capture\_terminate flowchart 23](#_Toc534270928)

[Figure 3‑11 xa\_tdm\_capture\_destroy flowchart 24](#_Toc534270929)

[Figure 3‑12 xa\_tmd\_capture\_factory flowchart 25](#_Toc534270930)

List of Table

[Table 2‑1 Function list 4](#_Toc534270931)

[Table 3‑1 XATDMCapture type structure information 5](#_Toc534270932)

[Table 3‑2 Macro definitions 5](#_Toc534270933)

[Table 3‑3 Globle variable xa\_tdm\_capture\_cmd 6](#_Toc534270934)

# Overview

This section provides an overview of the software architecture.

User Application

ADSP Interface

Kernel Space

User Space

ADSP Driver

ARM

Audio HW

**ADSP Framework**

TDM class

Equalizer class

Capture class

Renderer class

ADSP

Plugin

Equalizer Plugin\*

TDM Plugin

Capture Plugin

Renderer Plugin

This document’s target is in side of red square

\* not connect to SCU/SSI ADMA block

SCU/ SSI/

ADMA

DAC/

ADC

Figure 1‑1 ****The software architecture****

# Function list

The following is list of functions:

Table 2‑1 Function list

|  |  |  |
| --- | --- | --- |
| **Type** | **Function Name** | **Outline** |
| API | xa\_tdm\_capture\_factory | This API is to construct generic audio component and set functions pointer for internal functions |
| Internal functions | xa\_tdm\_capture\_prepare\_runtime | This function is to prepare TDM capture for steady operation. |
| xa\_tdm\_capture\_sufficient\_outport\_check | This function is to check the sufficient output port. |
| xa\_tdm\_capture\_empty\_this\_buffer | This function is to handle end of stream and respond message back to user. |
| xa\_tdm\_capture\_fill\_this\_buffer | This function is to put message into all output port. |
| xa\_tdm\_capture\_flush | This function is to flush output port queue. |
| xa\_tdm\_capture\_memtab | This function is to initialize all output port. |
| xa\_tdm\_capture\_preprocess | This function is to prepare data for output buffer. |
| xa\_tdm\_capture\_postprocess | This function is to get number of produced bytes and reschedule processing. |
| xa\_tdm\_capture\_terminate | This is to check if we received output port control message. |
| xa\_tdm\_capture\_destroy | This function is to destroy component. |
| xa\_tdm\_capture\_cleanup | This function is to flush output port queue and cancel internal scheduling. |

# Detail information

This section describes detail information of data types, macro definitions, implemented APIs and internal function units, global variable.

## Data type and Macro definition

The XATDMCapture type structure is the work area used by the TDM capture class. When using this class, secure the area with the application program. It’s not necessary to refer to this area because it only contains the internal variables and working buffers of the class. Make sure not to change the value of this area with the application program.

Table 3‑1 XATDMCapture type structure information

|  |  |
| --- | --- |
| Member name | Outline |
| XACodecBase base | Generic audio codec data |
| xf\_output\_port\_t output[XA\_MEMCNT\_OUTPUT] | Array of output port queue |
| u32 out\_idx[XA\_MEMCNT\_OUTPUT] | Array of output port index |
| u32 output\_total | The number of total output port |
| u32 factor | Time conversion factor (input byte "duration" in timebase units) |

Note: Marro XA\_MEMCNT\_OUTPUT: 4 (number of ouput memory block) in xa-tdm-capture-api.h

Table 3‑2 Macro definitions

|  |  |  |
| --- | --- | --- |
| Macro | Value | Outline |
| XA\_TDM\_CAP\_FLAG\_OUTPUT\_PORT\_INIT\_DONE | \_\_XA\_BASE\_FLAG(1 << 0) | Output port init done flag |
| XA\_TDM\_CAP\_FLAG\_INPUT\_DATA\_OVER | \_\_XA\_BASE\_FLAG(1 << 1) | Signal over flag |

Note: Macro \_\_XA\_BASE\_FLAG(f) ((f) << 6) is in xa-class-base.h file.

## Globle variable

Table 3‑3 Globle variable xa\_tdm\_capture\_cmd

|  |  |
| --- | --- |
| static XA\_ERRORCODE (\* const xa\_tdm\_capture\_cmd[])(XACodecBase \*base, xf\_message\_t \*m) | |
| Description: variable stores function pointers arcoding to opcode index to run in runtime operation. | |
| Array index | Value (function pointer) |
| XF\_OPCODE\_TYPE(XF\_SET\_PARAM) | xa\_base\_set\_param |
| XF\_OPCODE\_TYPE(XF\_GET\_PARAM) | xa\_base\_get\_param |
| XF\_OPCODE\_TYPE(XF\_EMPTY\_THIS\_BUFFER) | xa\_tdm\_capture\_empty\_this\_buffer |
| XF\_OPCODE\_TYPE(XF\_FILL\_THIS\_BUFFER) | xa\_tdm\_capture\_fill\_this\_buffer |
| XF\_OPCODE\_TYPE(XF\_FLUSH) | xa\_tdm\_capture\_flush |

Note: Marco XF\_OPCODE\_TYPE(opcode) : ((opcode) & (0x3F)) in xf-opcode.h file.

## Function definition

### xa\_tdm\_capture\_prepare\_runtime

DD\_FWK\_TDM\_01\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static inline XA\_ERRORCODE xa\_tdm\_capture\_prepare\_runtime(XATDMCapture \*tdm\_cap) | | | |
| **Function** | This function is to prepare TDM capture for steady operation. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XATDMCapture | tdm\_cap | I/O | Pointer to codec instance structure (struct XATDMCapure). |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_TDM\_CAP\_CONFIG\_FATAL\_PCM\_WIDTH | | PCM width is not 16 and 24. | |
| XA\_TDM\_CAP\_CONFIG\_FATAL\_CHANNEL\_MODE | | Channel mode is invalid. | |
| XA\_TDM\_CAP\_CONFIG\_FATAL\_SAMPLE\_RATE | | Sample rate is invalid. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | API structure is NULL (error from plugin). | |
| XA\_API\_FATAL\_MEM\_ALIGN | | API structure is not aligned to 4 bytes (error from plugin). | |
| XA\_TDM\_CAP\_CONFIG\_FATAL\_STATE | | Plugin has not pre-init yet (error from plugin). | |
| **Description** | * xa\_tdm\_capture\_prepare\_runtime command processing:   - Get config parameters: sample rate, pcm width, channels, frame size.  - Calculate sample size.  - Set capture timestamp factor. | | | |

[Covers: FD\_FWK\_CMN\_002]

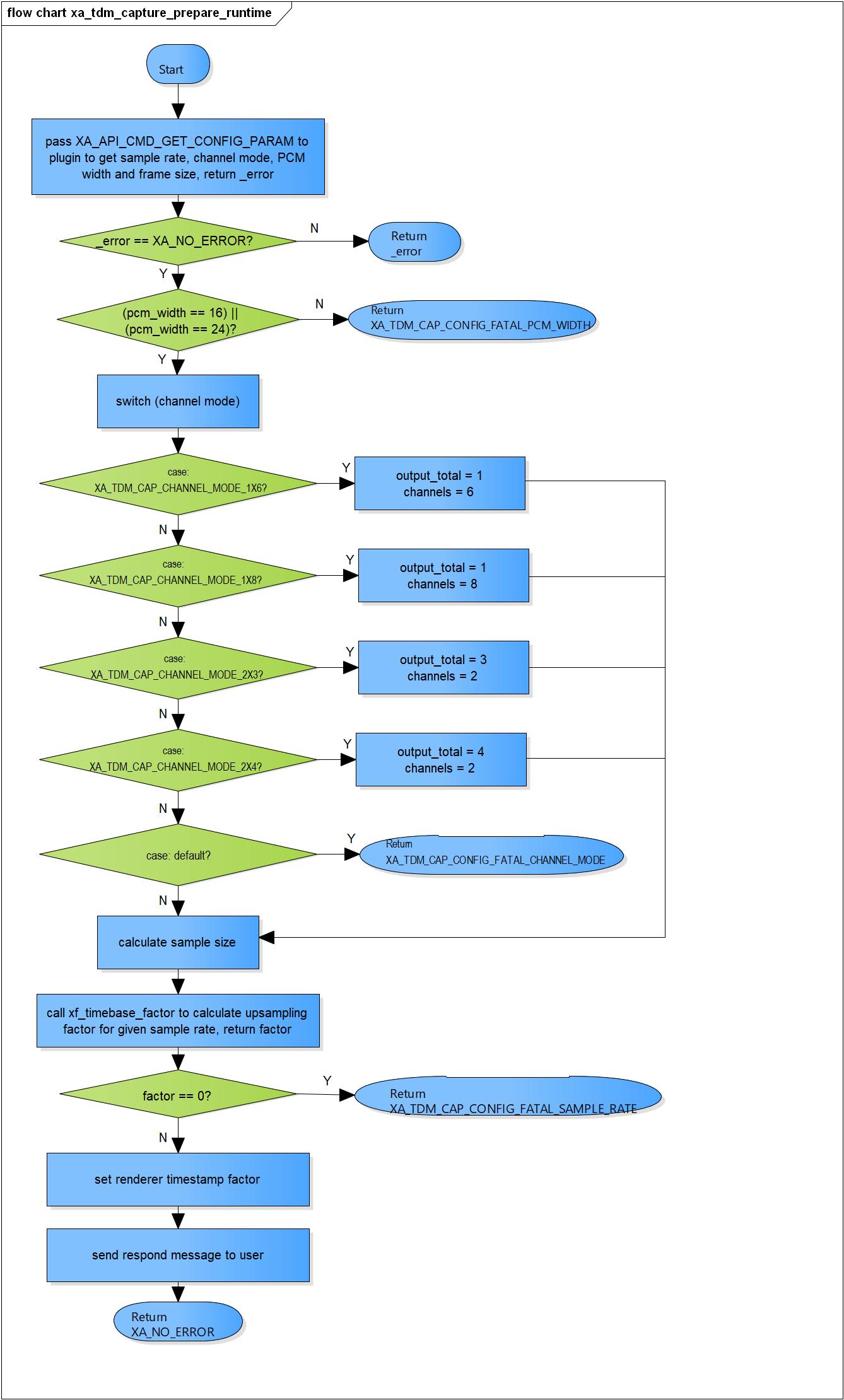


Figure 3‑1 xa\_tdm\_capture\_prepare\_runtime flowchart

### xa\_tdm\_capture\_sufficient\_outport\_check

DD\_FWK\_TDM\_01\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static inline s32 xa\_tdm\_capture\_sufficient\_outport\_check(XACodecBase \*base) | | | |
| **Function** | This function is to check the sufficient output port. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase). |
| **Return value** | 0 | | Indicate ports are not ready yet. | |
| 1 | | Indicate all ports are ready. | |
| **Description** | * xa\_tdm\_capture\_sufficient\_outport\_check command processing:   - Check the sufficient port, if have any port is not ready return 0. | | | |

[Covers: FD\_FWK\_CMN\_003, FD\_FWK\_CMN\_011]

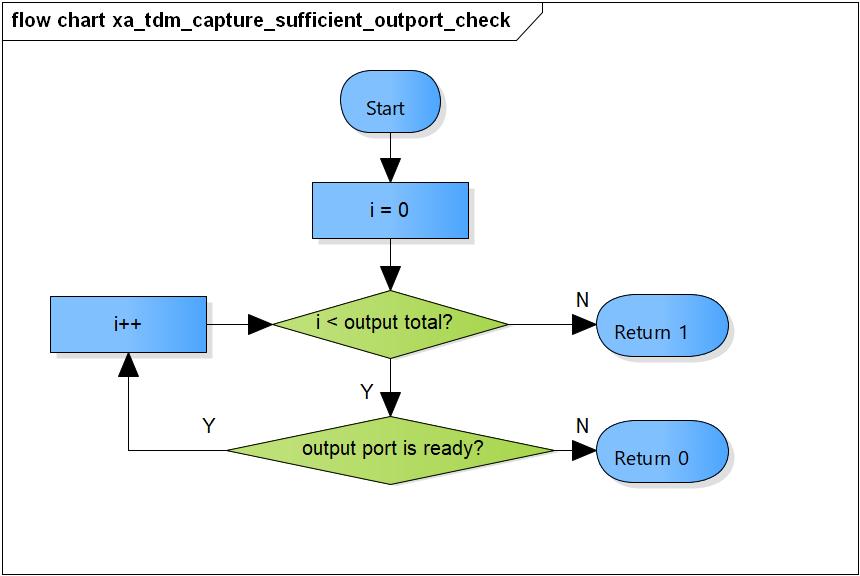


Figure 3‑2 xa\_tdm\_capture\_sufficient\_outport\_check flowchart

### xa\_tdm\_capture\_empty\_this\_buffer

DD\_FWK\_TDM\_01\_003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_capture\_empty\_this\_buffer(XACodecBase \*base,  xf\_message\_t \*m) | | | |
| **Function** | This function is to check end of stream for input port. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase). |
| xf\_message\_t | m | I | Pointer to audio message (struct xf\_message). |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_API\_FATAL\_INVALID\_CMD\_TYPE | | Port is not an input port queue. | |
| XA\_TDM\_CAP\_EXEC\_FATAL\_STATE | | Post initialization state is not completed yet. | |
| **Description** | * xa\_tdm\_capture\_empty\_this\_buffer command processing:   - Check if buffer is end of stream.  - Respond message back to user. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_010]

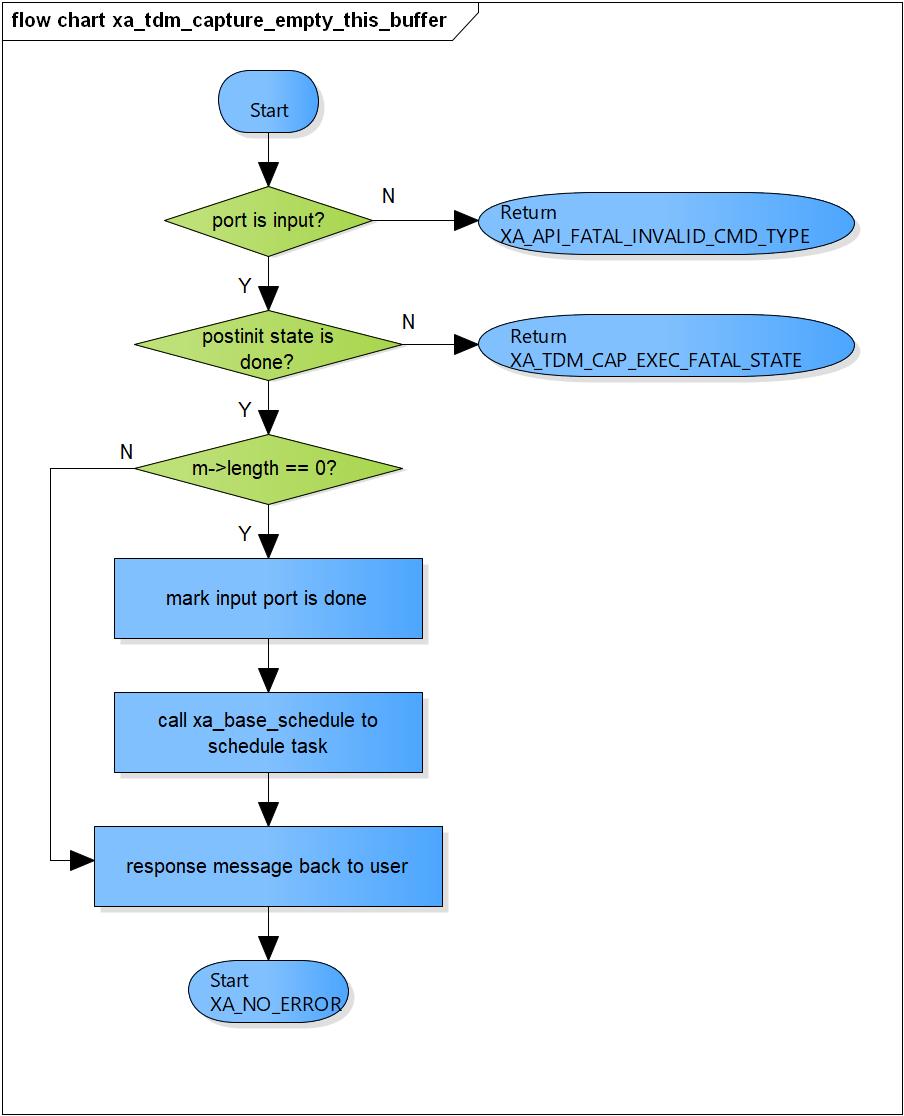


Figure 3‑3 xa\_tdm\_capture\_empty\_this\_buffer flowchart

### xa\_tdm\_capture\_fill\_this\_buffer

DD\_FWK\_TDM\_01\_004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_capture\_fill\_this\_buffer(XACodecBase \*base,  xf\_message\_t \*m) | | | |
| **Function** | This function is to put message into all output port. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase). |
| xf\_message\_t | m | I/O | Pointer to audio message (struct xf\_message). |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_API\_FATAL\_INVALID\_CMD\_TYPE | | Port is invalid. | |
| XA\_CAP\_EXEC\_FATAL\_STATE | | Receive message with non-zero length in runtime initialization state.  Post initialization state is not completed yet.  If receive output port contol message and excution state is not completed yet. | |
| **Description** | * xa\_tdm\_capture\_fill\_this\_buffer command processing:   - Special handling of zero-length buffer.  - Adjust message length.  - Place message into all output port. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_011]

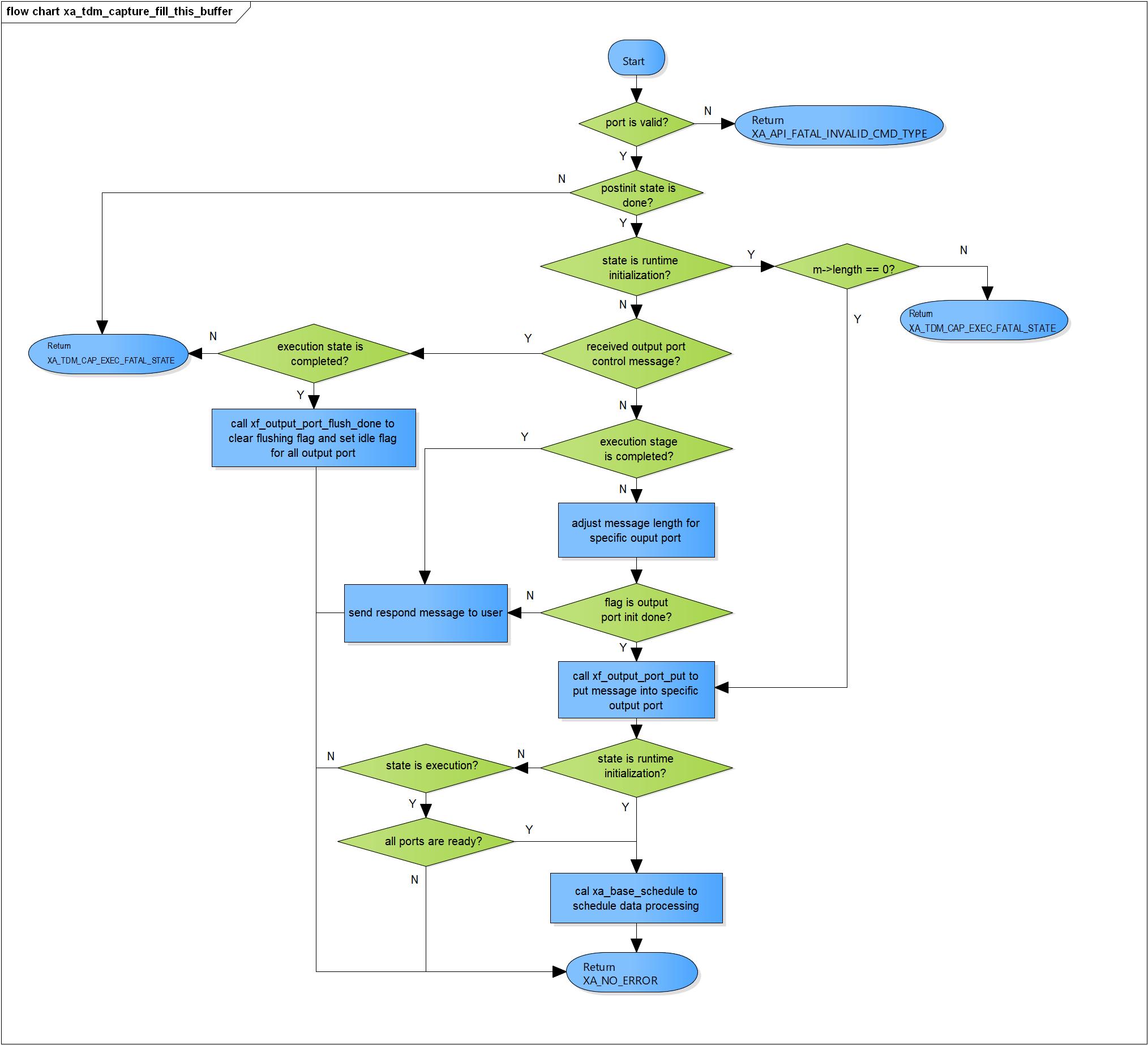


Figure 3‑4 xa\_tdm\_capture\_fill\_this\_buffer flowchart

### xa\_tdm\_capture\_memtab

DD\_FWK\_TDM\_01\_005

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_capture\_memtab(XACodecBase \*base, WORD32 idx, WORD32 type, WORD32 size, WORD32 align, u32 core) | | | |
| **Function** | This function is to initialize all output port. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase) |
| WORD32 | idx | I | Port index  Valid range: [0; 3] |
| WORD32 | type | I | Port type (input or output port)  Valid value: XA\_MEMTYPE\_OUTPUT |
| WORD32 | size | I | Buffer size of port |
| WORD32 | align | x | Port alignment |
| u32 | core | x | Core index of ADSP framework |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_API\_FATAL\_INVALID\_CMD\_TYPE | | Port type is not an output port.  Ports index are invalid. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | Initialize output port is fail. | |
| **Description** | * Memtab command processing:   - Initialize all output port. | | | |

[Covers: FD\_FWK\_CMN\_008, FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_001]

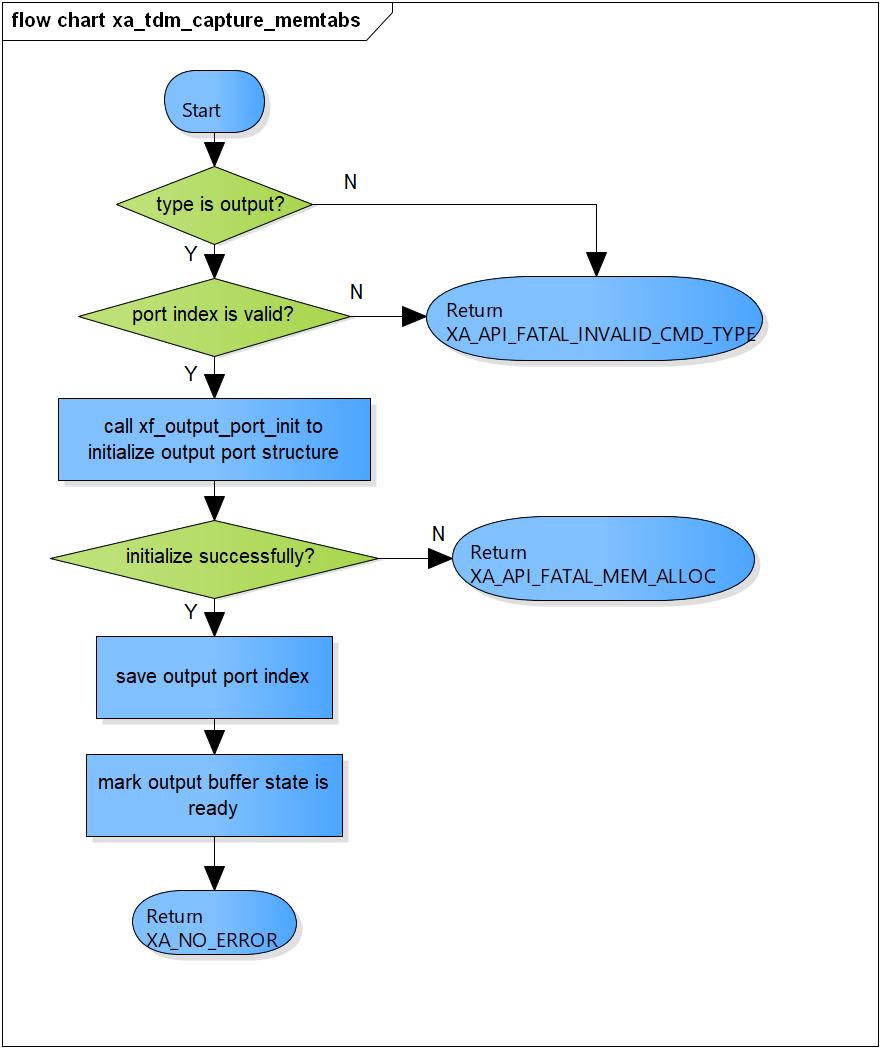


Figure 3‑5 xa\_tdm\_capture\_memtab flowchart

### xa\_tdm\_capture\_preprocess

DD\_FWK\_TDM\_01\_006

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_capture\_preprocess(XACodecBase \*base) | | | |
| **Function** | This function is to prepare data for output buffer. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I | Pointer to codec instance structure (struct XACodecBase) |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_TDM\_CAP\_EXEC\_NONFATAL\_OUTPUT | | No output buffer available. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | API structure is NULL (error from plugin). | |
| XA\_API\_FATAL\_MEM\_ALIGN | | API object is not aligned to 4 bytes (error from plugin). | |
| XA\_TDM\_CAP\_EXEC\_FATAL\_STATE | | TDM-capture is not init yet (error from plugin). | |
| XA\_TDM\_CAP\_CONFIG\_FATAL\_STATE | | TDM-capture is not in post-init yet (error from plugin). | |
| **Description** | * Preprocess command processing:   - Get output pointer from output port queue.  - Set the output buffer pointer. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_002, FD\_FWK\_CMN\_003]

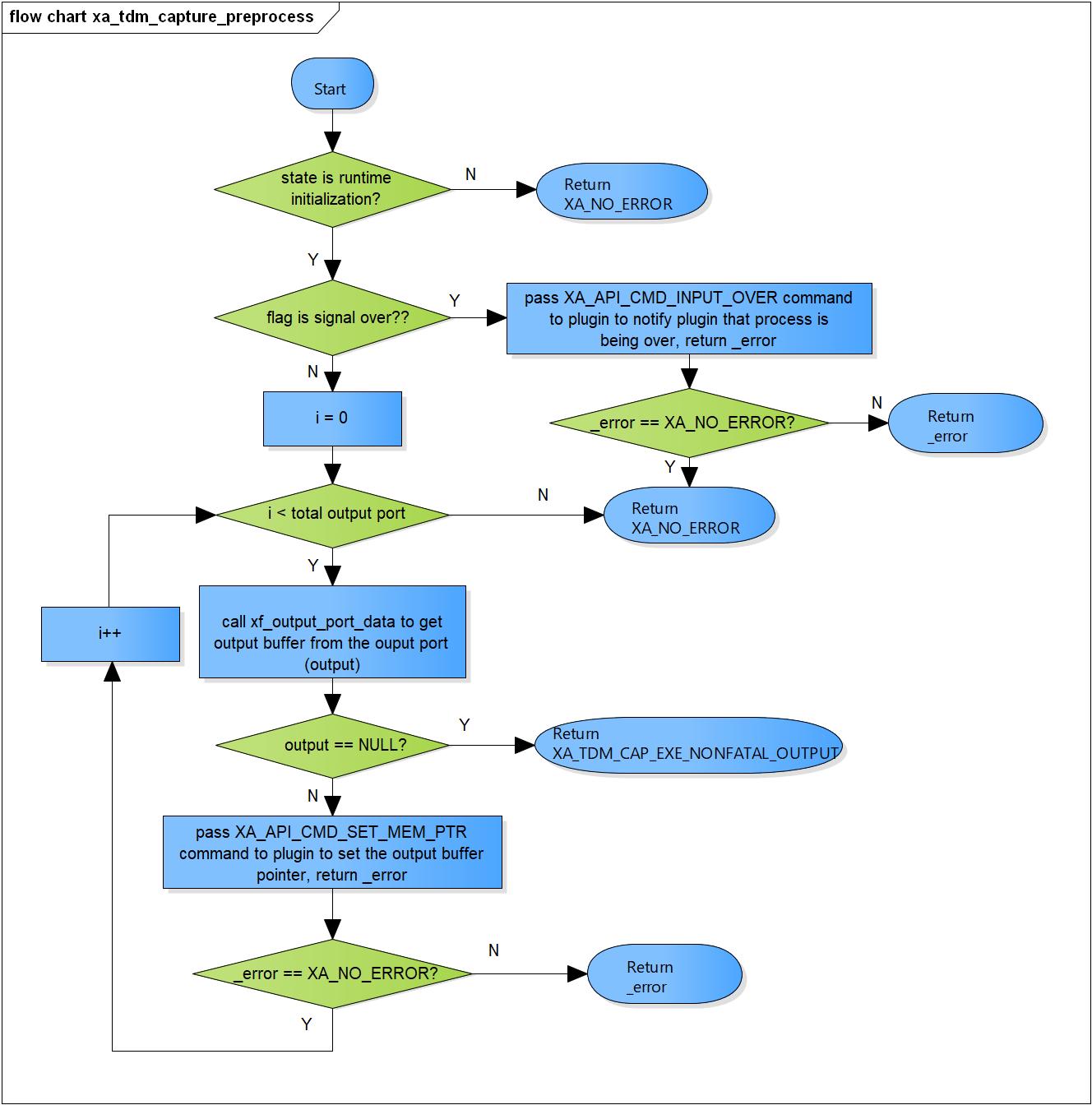


Figure 3‑6 xa\_tdm\_capture\_preprocess flowchart

### xa\_tdm\_capture\_postprocess

DD\_FWK\_TDM\_01\_007

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_capture\_postprocess(XACodecBase \*base,  WORD32 done) | | | |
| **Function** | This function is to get number of produced bytes and reschedule processing. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase). |
| WORD32 | done | I | State of processing of plugin. |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | API structure is NULL (error from plugin). | |
| XA\_API\_FATAL\_MEM\_ALIGN | | API object is not aligned to 4 bytes (error from plugin). | |
| XA\_TDM\_CAP\_EXEC\_FATAL\_STATE | | TDM\_Capture has not init yet (error from plugin).  Output buffer is not ready (error from plugin). | |
| **Description** | * xa\_tdm\_capture\_postprocess command processing:   - Check state if process is done.  - Get number of produced samples from plugin.  - Reschedule execution if we have pending input. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_002, FD\_FWK\_CMN\_003]

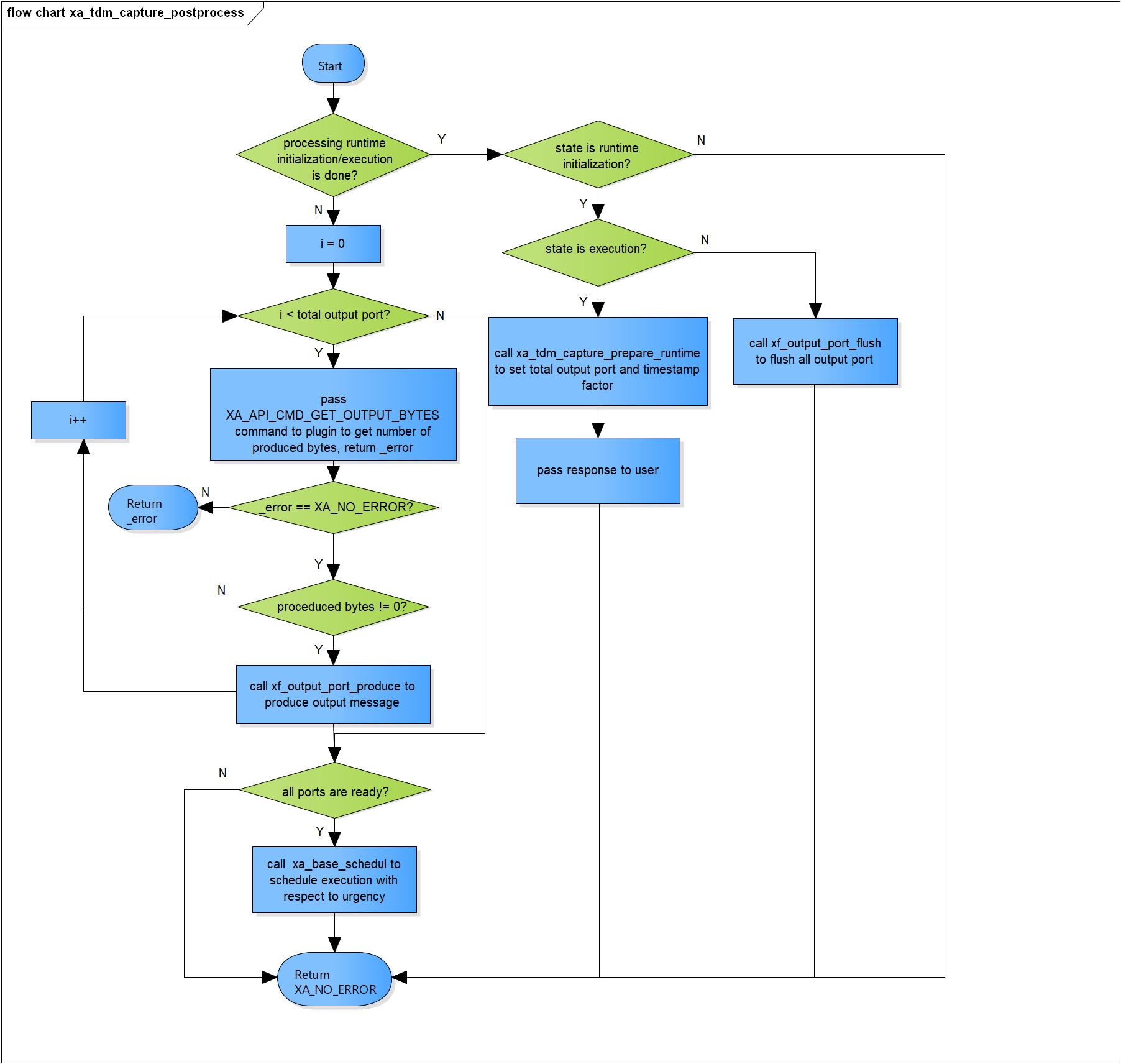


Figure 3‑7 xa\_tdm\_capture\_postprocess flowchart

### xa\_tdm\_capture\_flush

DD\_FWK\_TDM\_01\_008

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_capture\_flush(XACodecBase \*base, xf\_message\_t \*m) | | | |
| **Function** | This function is to flush output port queue. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase). |
| xf\_message\_t | m | I | Pointer to audio message (struct xf\_message). |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_API\_FATAL\_INVALID\_CMD | | Post initialization state is not completed yet. | |
| XA\_API\_FATAL\_INVALID\_CMD\_TYPE | | Message length is not zero.  Port index is invalid. | |
| **Description** | * xa\_tdm\_capture\_flush command processing:   - Check if post initialization state is not completed yet.  - Ensure input message length is zero.  - Flush all output port queue. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_012]

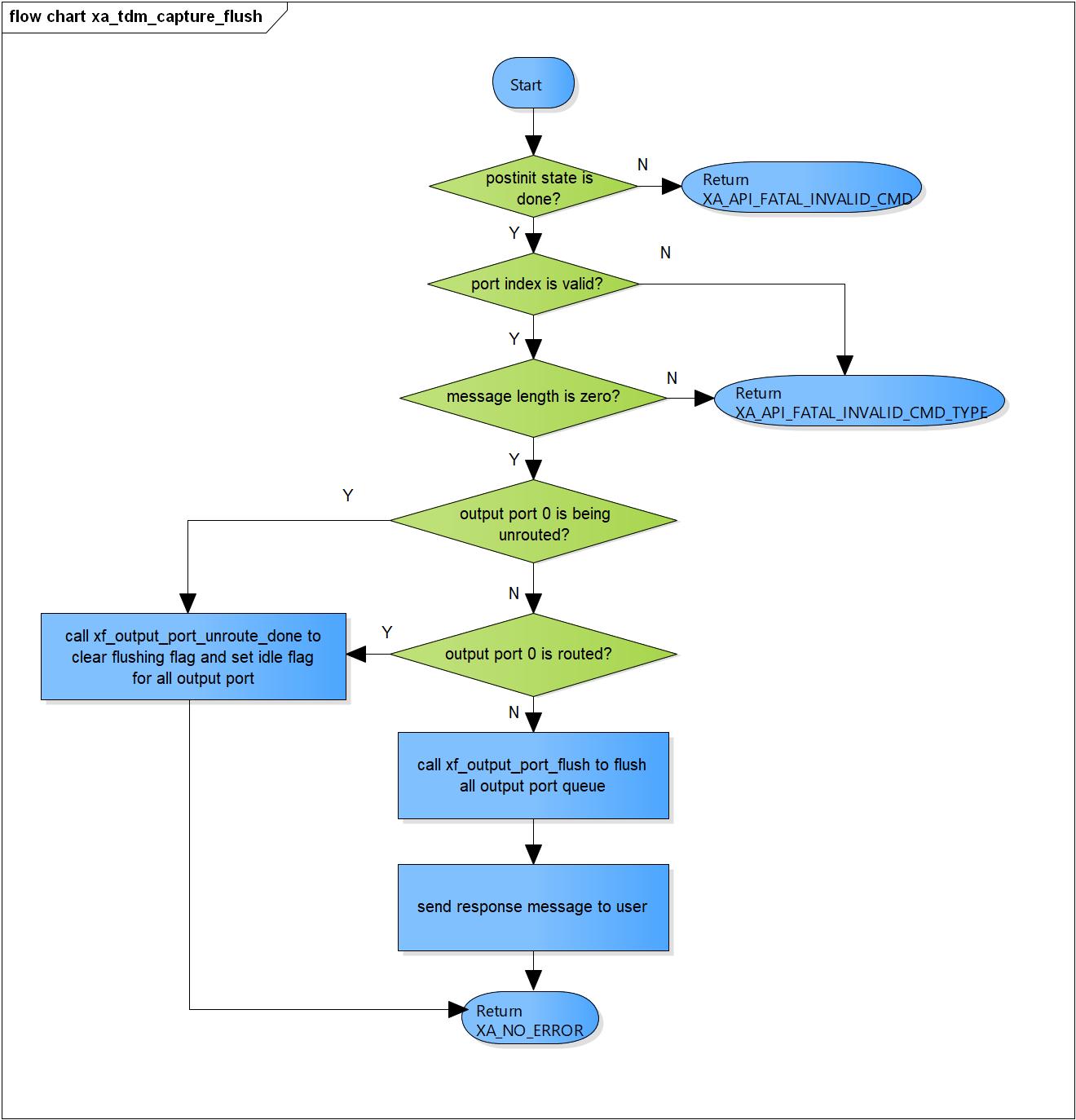


Figure 3‑8 xa\_tdm\_capture\_flush flowchart

### xa\_tdm\_capture\_cleanup

DD\_FWK\_TDM\_01\_009

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_tdm\_capture\_cleanup(xf\_component\_t \*component, xf\_message\_t \*m) | | | |
| **Function** | This function is to flush output port queue and cancel internal scheduling. | | | |
| **Arguments** | Type | Name | I/O | Description |
| xf\_component\_t | component | I/O | Pointer to codec instance structure  (struct xf\_component ). |
| xf\_message\_t | m | I | Pointer to audio message (struct xf\_message). |
| **Return value** | 1 | TDM capture component cannot be destroyed. | | |
| 0 | TDM capture component has been destroyed totally. | | |
| **Description** | * xa\_tdm\_capture\_cleanup command processing:   - Complete message with error response.  - Cancel internal scheduling message if needed.  - Flush and destroy output port. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_004]

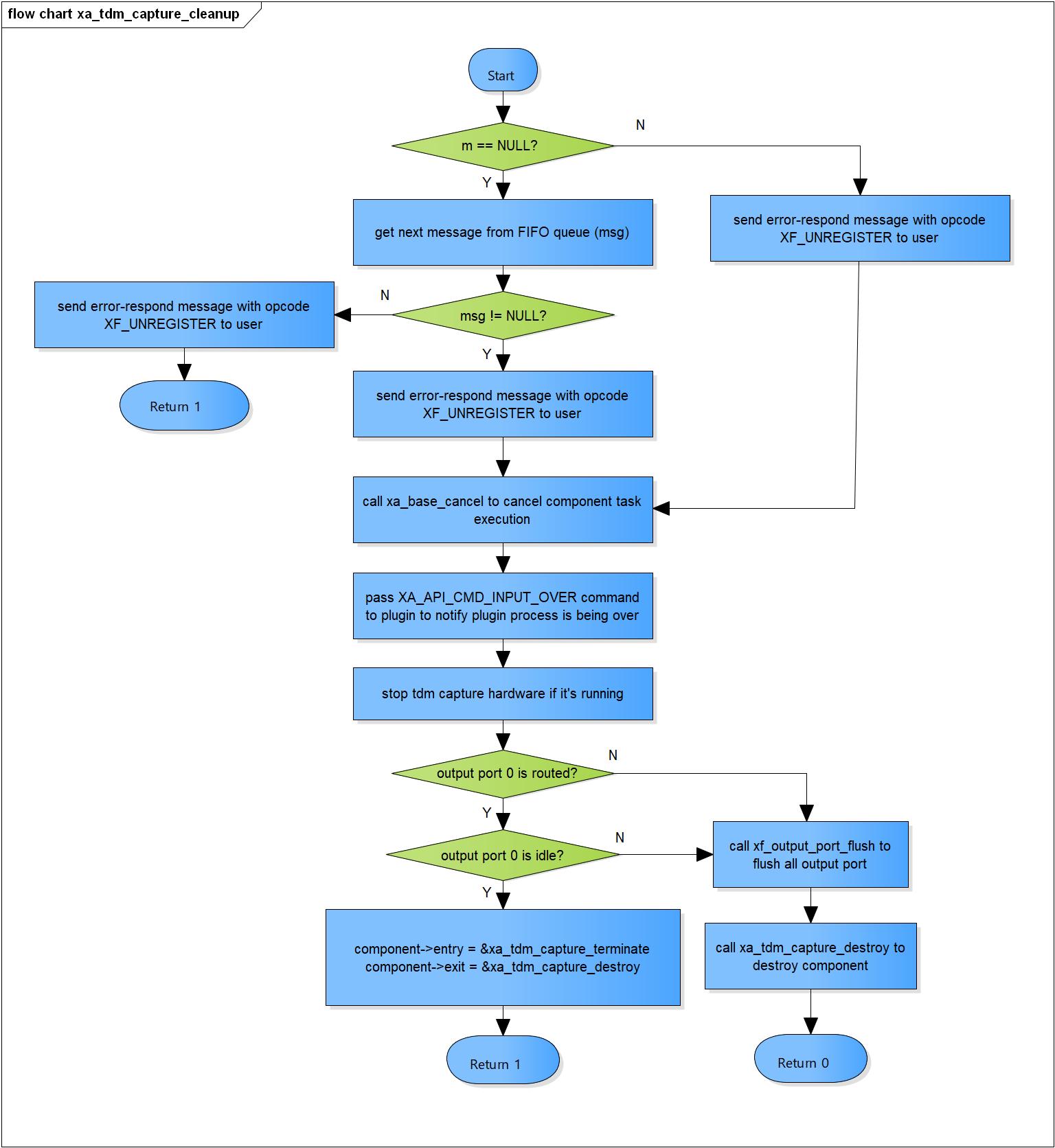


Figure 3‑9 xa\_tdm\_capture\_cleanup flowchart

### xa\_tdm\_capture\_terminate

DD\_FWK\_TDM\_01\_010

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_tdm\_capture\_terminate(xf\_component\_t \*component, xf\_message\_t \*m) | | | |
| **Function** | This is to check if we received output port control message. | | | |
| **Arguments** | Type | Name | I/O | Description |
| xf\_component\_t | component | I | Pointer to codec instance structure  (struct xf\_component ). |
| xf\_message\_t | m | I/O | Pointer to audio message (struct xf\_message). |
| **Return value** | -1 | Output port flushing is completed, component ready to destroy. | | |
| 0 | Normal end. | | |
| **Description** | * xa\_tdm\_capture\_terminate command processing:   - Check if we received output port control message. | | | |

[Covers: FD\_FWK\_CMN\_004]

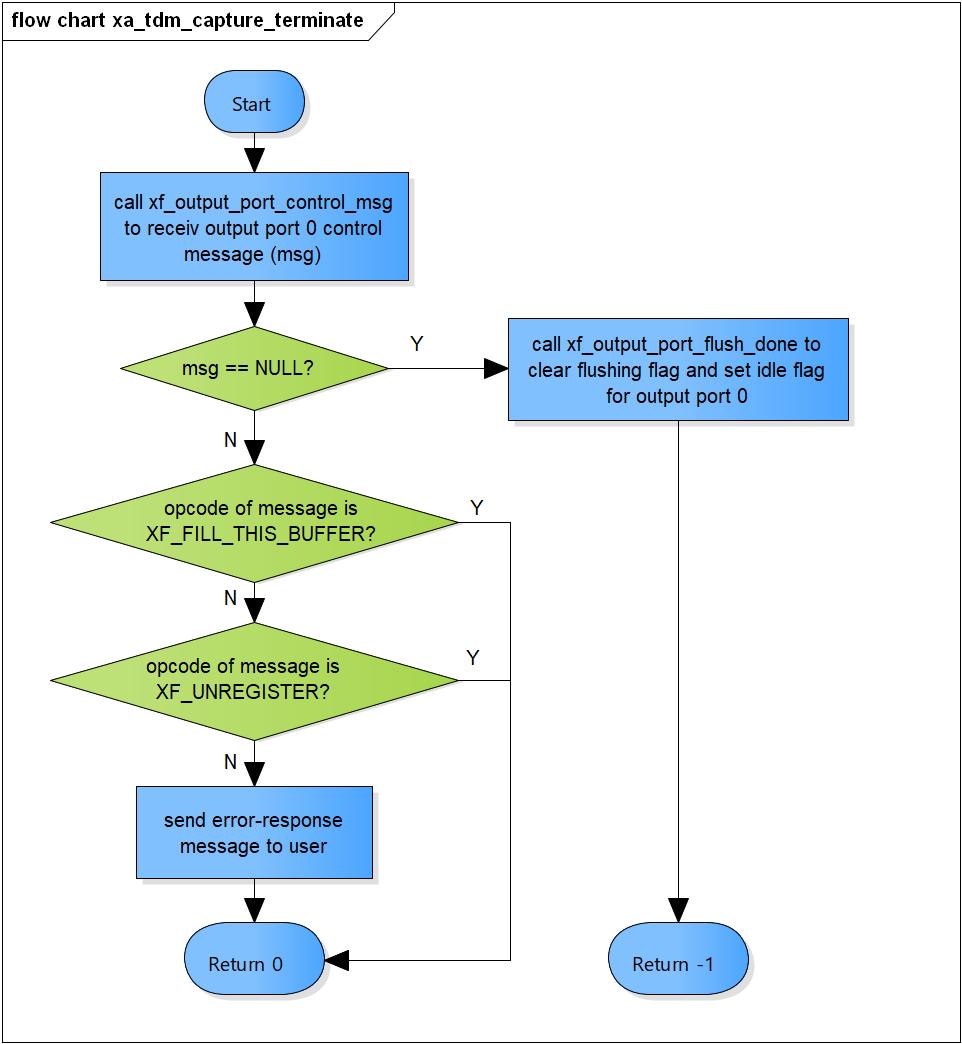


Figure 3‑10 xa\_tdm\_capture\_terminate flowchart

### xa\_tdm\_capture\_destroy

DD\_FWK\_TDM\_01\_011

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_tdm\_capture\_destroy(xf\_component\_t \*component, xf\_message\_t \*m) | | | |
| **Function** | This function is to destroy component. | | | |
| **Arguments** | Type | Name | I/O | Description |
| xf\_component\_t | component | I/O | Pointer to codec instance structure  (struct xf\_component ). |
| xf\_message\_t | m | x | Pointer to audio message (struct xf\_message). |
| 0 | | Indicate the component is destroyed. | |
| **Description** | * xa\_tdm\_capture\_destroy command processing:   - Destroy all output port.  - Destroy base object. | | | |

[Covers: FD\_FWK\_CMN\_004]

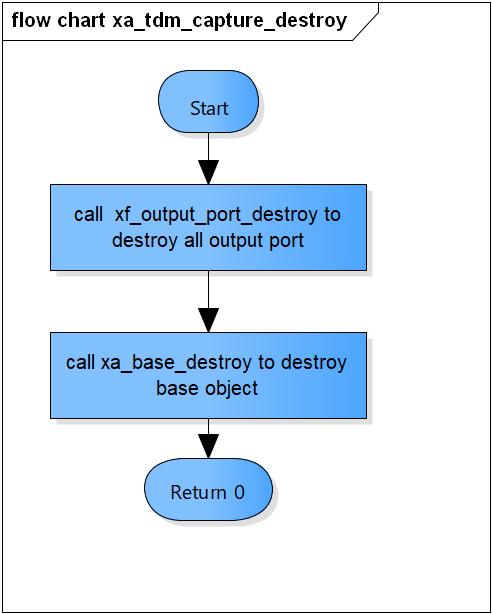


Figure 3‑11 xa\_tdm\_capture\_destroy flowchart

### xa\_tdm\_capture\_factory

DD\_FWK\_TDM\_01\_012

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | xf\_component\_t \* xa\_tdm\_capture\_factory(u32 core, xa\_codec\_func\_t process) | | | |
| **Function** | This function is to initialize TDM Capture component. | | | |
| **Arguments** | Type | Name | I/O | Description |
| u32 | core | I | Core index of ADSP framework. |
| xa\_codec\_func\_t | process | I | Codec API entry point (function). |
| **Return Value** | Return handle to component. | | | |
| **Description** | * xa\_tdm\_capture\_factory command processing:   - Initialize TDM Capture component. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_008, FD\_FWK\_CMN\_009]

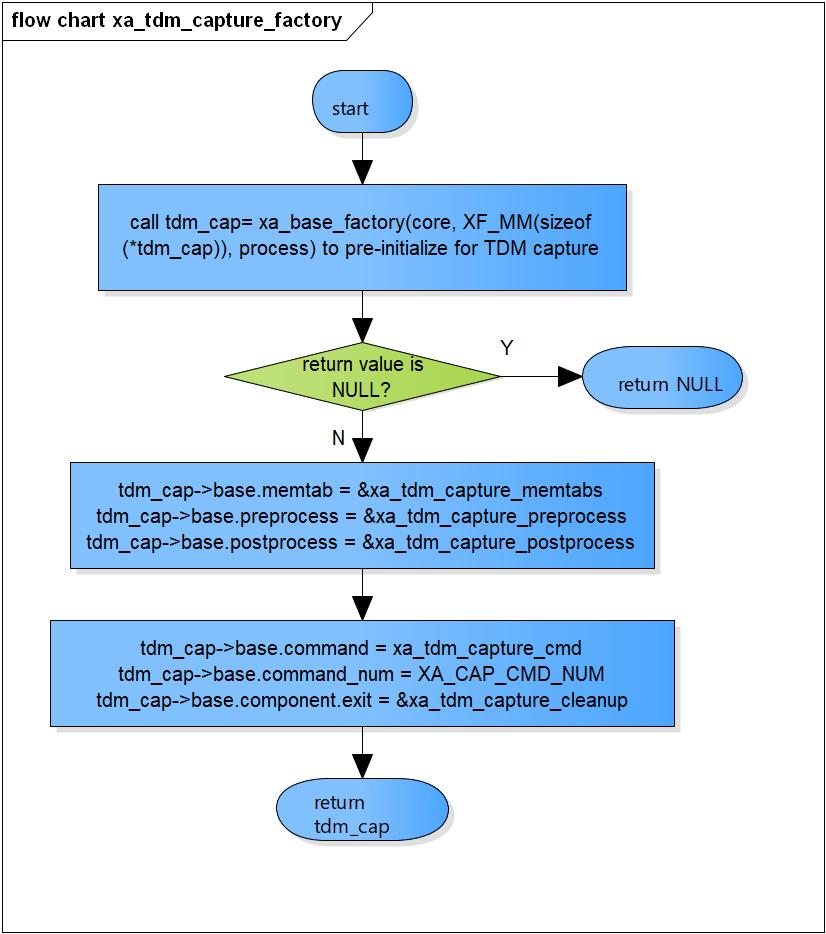


Figure 3‑12 xa\_tmd\_capture\_factory flowchart

# Revision history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Page** | **Content** | **Approved** | **Changed** |
| 1.0.0 | Nov 14 2018 | - | First Edition issued | Vu Phan | Vu Phan |
| 1.1.0 | Dec 10 2018 | - | Add traceability ID | Vu Phan | Ngu Pham |
| 1.2.0 | Jan 03, 2019 | - | Add range for input parameters | Vu Phan | Tien Tran |