**Document Type: Detail Design**

**Document Name:**

**ADSP FRAMEWORK: 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](#_Toc6381561)

[2. Function list 4](#_Toc6381562)

[3. Detail information 5](#_Toc6381563)

[3.1 Data type and Macro definition 5](#_Toc6381564)

[3.2 Globle variable 6](#_Toc6381565)

[3.3 Function definition 7](#_Toc6381566)

[3.3.1 xa\_capture\_prepare\_runtime 7](#_Toc6381567)

[3.3.2 xa\_capture\_empty\_this\_buffer 9](#_Toc6381568)

[3.3.3 xa\_capture\_fill\_this\_buffer 10](#_Toc6381569)

[3.3.4 xa\_capture\_memtab 12](#_Toc6381570)

[3.3.5 xa\_capture\_port\_route 14](#_Toc6381571)

[3.3.6 xa\_capture\_port\_unroute 16](#_Toc6381572)

[3.3.7 xa\_capture\_preprocess 17](#_Toc6381573)

[3.3.8 xa\_capture\_postprocess 19](#_Toc6381574)

[3.3.9 xa\_capture\_flush 21](#_Toc6381575)

[3.3.10 xa\_capture\_cleanup 23](#_Toc6381576)

[3.3.11 xa\_capture\_terminate 25](#_Toc6381577)

[3.3.12 xa\_capture\_destroy 26](#_Toc6381578)

[3.3.13 xa\_capture\_factory 27](#_Toc6381579)

[3.3.14 xa\_capture\_mmap\_this\_buffer 29](#_Toc6381580)

[4. Revision history 30](#_Toc6381581)

List of Figures

[Figure 1‑1 The software architecture 3](#_Toc6381582)

[Figure 3‑1 xa\_capture\_prepare\_runtime flowchart 8](#_Toc6381583)

[Figure 3‑2 xa\_capture\_empty\_this\_buffer flowchart 9](#_Toc6381584)

[Figure 3‑3 xa\_capture\_fill\_this\_buffer flowchart 11](#_Toc6381585)

[Figure 3‑4 xa\_capture\_memtab flowchart 13](#_Toc6381586)

[Figure 3‑5 xa\_capture\_port\_route flowchart 15](#_Toc6381587)

[Figure 3‑6 xa\_capture\_port\_unroute flowchart 16](#_Toc6381588)

[Figure 3‑7 xa\_capture\_preprocess flowchart 18](#_Toc6381589)

[Figure 3‑8 xa\_capture\_postprocess flowchart 20](#_Toc6381590)

[Figure 3‑9 xa\_capture\_flush flowchart 22](#_Toc6381591)

[Figure 3‑10 xa\_capture\_cleanup flowchart 24](#_Toc6381592)

[Figure 3‑11 xa\_capture\_terminate flowchart 25](#_Toc6381593)

[Figure 3‑12 xa\_capture\_destroy flowchart 26](#_Toc6381594)

[Figure 3‑13 xa\_capture\_factory flowchart 27](#_Toc6381595)

[Figure 3‑13 xa\_capture\_mmap\_this\_buffer flowchart 29](#_Toc6381596)

List of Table

[Table 2‑1 Function list 4](#_Toc6381597)

[Table 3‑1 XACapture type structure information 5](#_Toc6381598)

[Table 3‑2 Macro definitions 5](#_Toc6381599)

[Table 3‑3 Globle variable xa\_capture\_cmd 6](#_Toc6381600)

# 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\_capture\_factory | This API is to construct generic audio component and set functions pointer for internal functions. |
| Internal functions | xa\_capture\_prepare\_runtime | This function is to prepare capture for steady operation. |
| xa\_capture\_empty\_this\_buffer | This function is to check end of stream for input port. |
| xa\_capture\_fill\_this\_buffer | This function is to place message into output port. |
| xa\_capture\_flush | This function is to purge port queue. |
| xa\_capture\_memtab | This function is to initialize output port. |
| xa\_capture\_port\_route | This function is to route output port - allocate buffers. |
| xa\_capture\_port\_unroute | This function is to unroute output port and destroy all buffers allocated. |
| xa\_capture\_preprocess | This function is to set the output buffer pointer. |
| xa\_capture\_postprocess | This function is to get number of produced bytes and reschedule processing. |
| xa\_capture\_terminate | This is capture termination-state command processor function. |
| xa\_capture\_destroy | This function is to destroy component. |
| xa\_capture\_cleanup | This function is to purge port queue and cancel component task execution. |
| xa\_capture\_mmap\_this\_buffer | This function is to map buffer address from user to output port index. |

# 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 XACapture type structure is the work area used by the 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 XACapture type structure information

|  |  |
| --- | --- |
| Member name | Outline |
| XACodecBase base | Generic audio codec data |
| xf\_input\_port\_t input | Input port queue |
| xf\_output\_port\_t output | Output port queue |
| WORD32 out\_idx | Output port index |
| u32 factor | Time conversion factor (input byte "duration" in timebase units) |
| u32 sample\_size | Sample size in bytes |
| u32 buf\_size | Mapped buffer size |

Table 3‑2 Macro definitions

|  |  |  |
| --- | --- | --- |
| Macro | Value | Outline |
| XA\_CAP\_FLAG\_ RUNNING | \_\_XA\_BASE\_FLAG(1 << 0) | Capture is performed |
| XA\_CAP\_FLAG\_SILENCE | \_\_XA\_BASE\_FLAG(1 << 1) | Capture is idle and produces silence |
| XA\_CAP\_FLAG\_INPUT\_READY | \_\_XA\_BASE\_FLAG(1 << 2) | Input data is ready |

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

## Globle variable

Table 3‑3 Globle variable xa\_capture\_cmd

|  |  |
| --- | --- |
| static XA\_ERRORCODE (\* const xa\_capture\_cmd[])(XACodecBase \*base, xf\_message\_t \*m) | |
| Description: variable stores function pointers according 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\_ROUTE) | xa\_capture\_port\_route |
| XF\_OPCODE\_TYPE(XF\_UNROUTE) | xa\_capture\_port\_unroute |
| XF\_OPCODE\_TYPE(XF\_EMPTY\_THIS\_BUFFER) | xa\_capture\_empty\_this\_buffer |
| XF\_OPCODE\_TYPE(XF\_FILL\_THIS\_BUFFER) | xa\_capture\_fill\_this\_buffer |
| XF\_OPCODE\_TYPE(XF\_FLUSH) | xa\_capture\_flush |
| [XF\_OPCODE\_TYPE(XF\_MMAP\_THIS\_BUFFER)] | xa\_capture\_mmap\_this\_buffer |

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

## Function definition

### xa\_capture\_prepare\_runtime

DD\_FWK\_RDR\_01\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static inline XA\_ERRORCODE xa\_capture\_prepare\_runtime(XACapture \*capture) | | | |
| **Function** | This function is to prepare capture for steady operation. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACapture | capture | I/O | Pointer to codec instance structure (struct XACapture). |
| **Return value** | 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\_CAP\_CONFIG\_FATAL\_STATE | | Pre-initialization is not completed yet (error from plugin). | |
| XA\_CAP\_CONFIG\_NONFATAL\_ERR\_PCM\_WIDTH | | PCM width is not 16 or 24 bit. | |
| XA\_CAP\_CONFIG\_NONFATAL\_ERR\_SAMPLE\_RATE | | Sample rate is invalid. | |
| XA\_NO\_ERROR | | Nomally end. | |
| **Description** | * xa\_equlizer\_prepare\_runtime command processing:   - Get config parameters: sample rate, pcm width, channels.  - Set capture timestamp factor. | | | |

[Covers: FD\_FWK\_CMN\_002]

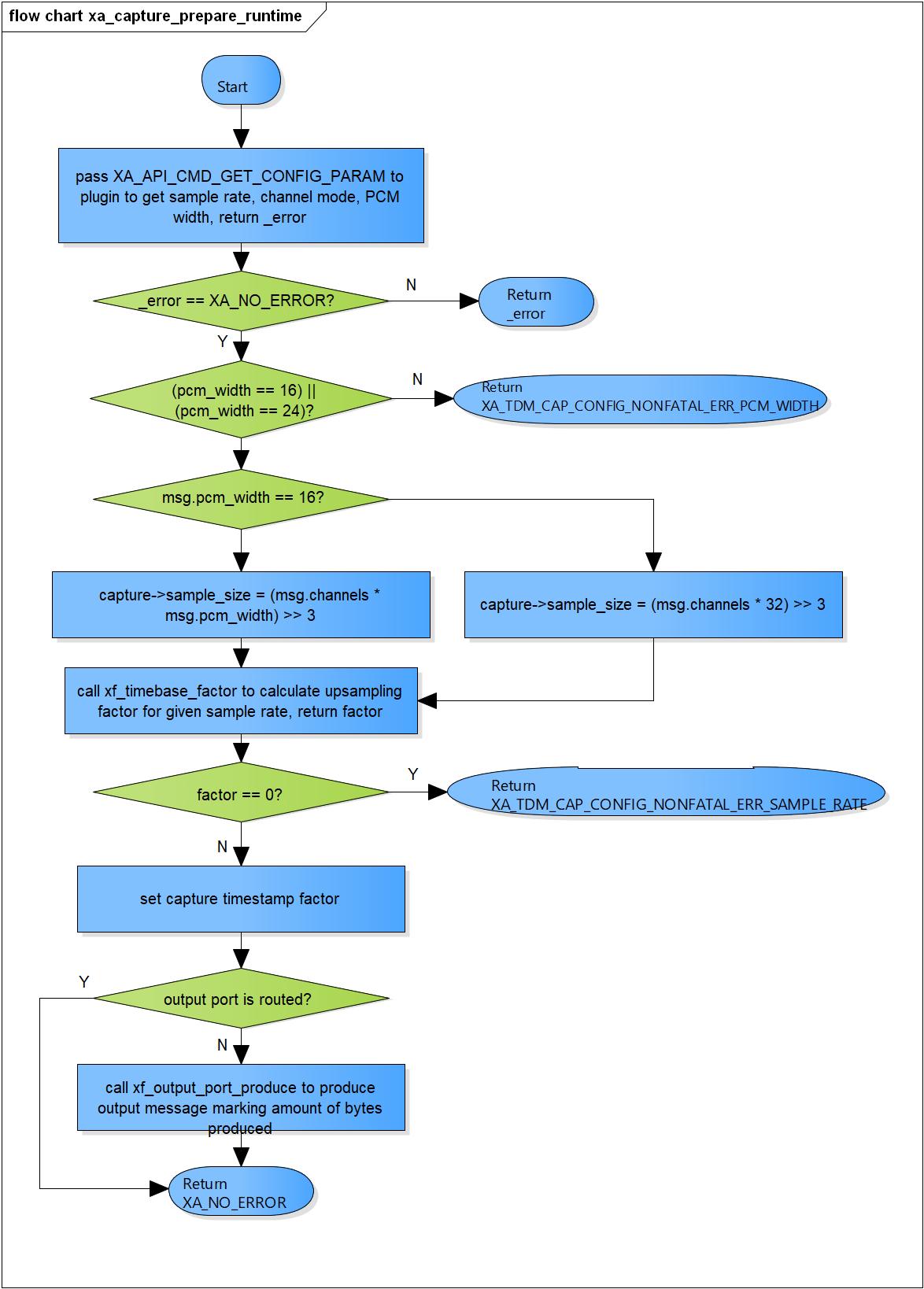


Figure 3‑1 xa\_capture\_prepare\_runtime flowchart

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

DD\_FWK\_RDR\_01\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_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 | 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. | |
| **Description** | * xa\_capture\_empty\_this\_buffer command processing:   - Check if buffer is end of stream.  - Response message back to user. | | | |

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

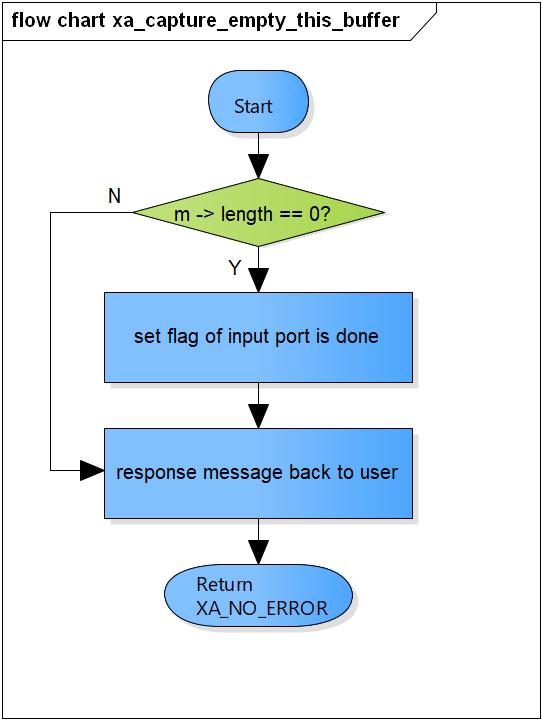


Figure 3‑2 xa\_capture\_empty\_this\_buffer flowchart

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

DD\_FWK\_RDR\_01\_003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_capture\_fill\_this\_buffer(XACodecBase \*base,  xf\_message\_t \*m); | | | |
| **Function** | This function is to place message into 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 | | The port is not an output port. | |
| XA\_API\_FATAL\_INVALID\_CMD | | Post initialization state is not completed yet. | |
| XA\_CAP\_EXEC\_FATAL\_STATE | | Receive message with non-zero length in runtime initialization state.  Execution state is not completed yet. | |
| **Description** | * xa\_capture\_fill\_this\_buffer command processing:   - Special handling of zero-length buffer.  - Adjust message length.  - Put message into output port. | | | |

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

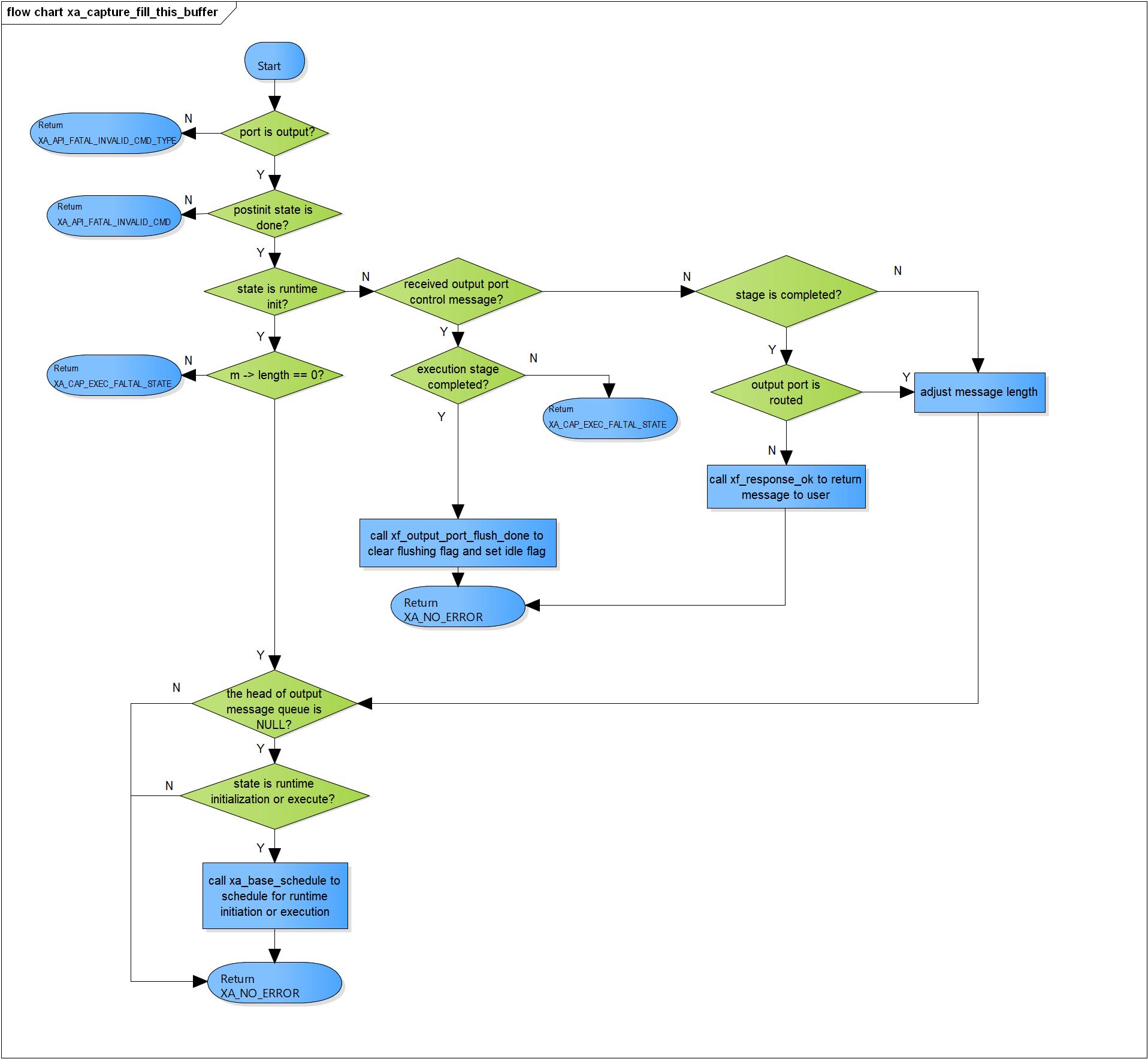


Figure 3‑3 xa\_capture\_fill\_this\_buffer flowchart

### xa\_capture\_memtab

DD\_FWK\_RDR\_01\_004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_capture\_memtab(XACodecBase \*base, WORD32 idx,  WORD32 type, WORD32 size, WORD32 align, u32 core) | | | |
| **Function** | This function is to initialize 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 value: 1 |
| 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.  Port index is invalid. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | Initialize output port is fail. | |
| **Description** | * Memtab command processing:   - Initialize output port structure.  - Mark capture input buffer is ready. | | | |

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

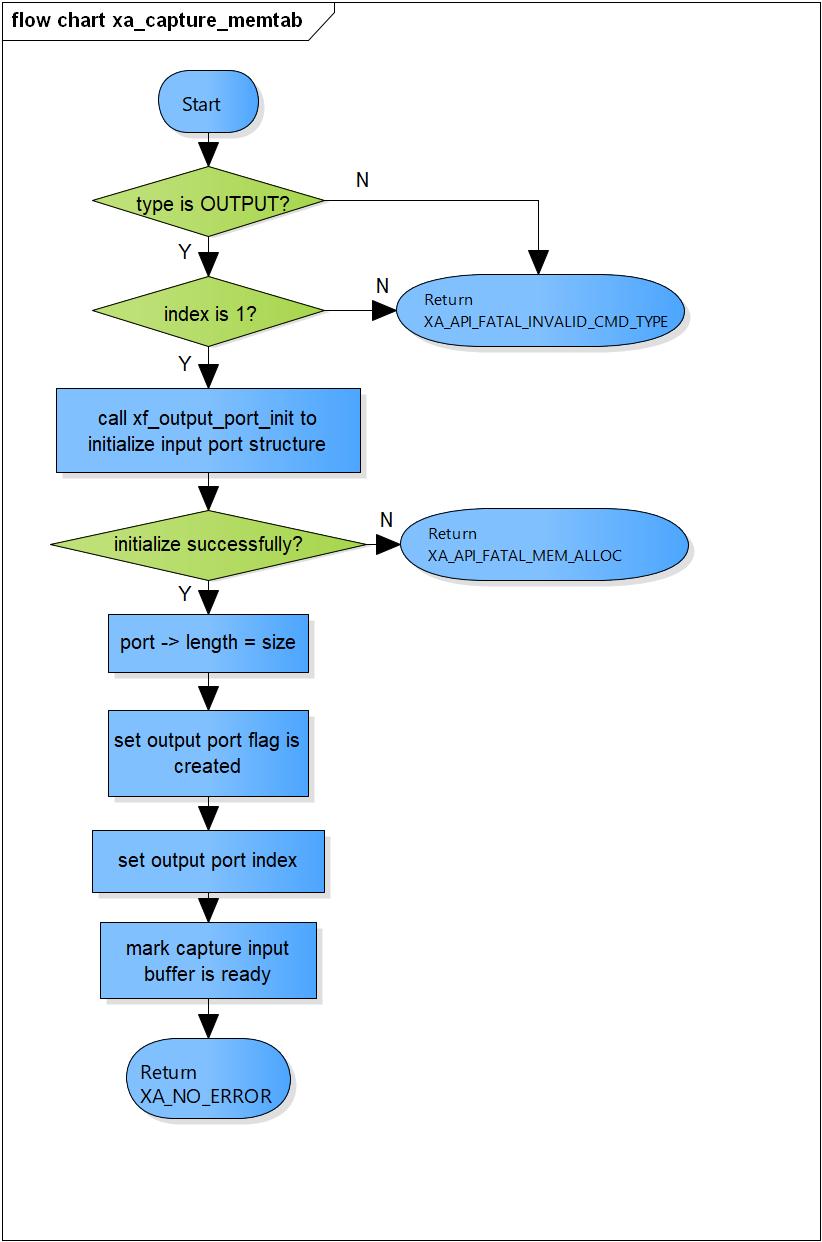


Figure 3‑4 xa\_capture\_memtab flowchart

### xa\_capture\_port\_route

DD\_FWK\_RDR\_01\_005

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_capture\_port\_route(XACodecBase \*base, xf\_message\_t \*m) | | | |
| **Function** | This function is to route output port - allocate memory buffers. | | | |
| **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 | | Ports have been routed before yet.  Port is not an output. | |
| XA\_API\_FATAL\_INVALID\_CMD | | Post initialization state is not completed yet. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | Route port is not successful. | |
| **Description** | * Output port routing command processing:   - Make sure port is not routed yet.  - Request destination component for memory map function.  - Schedule processing. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_006]

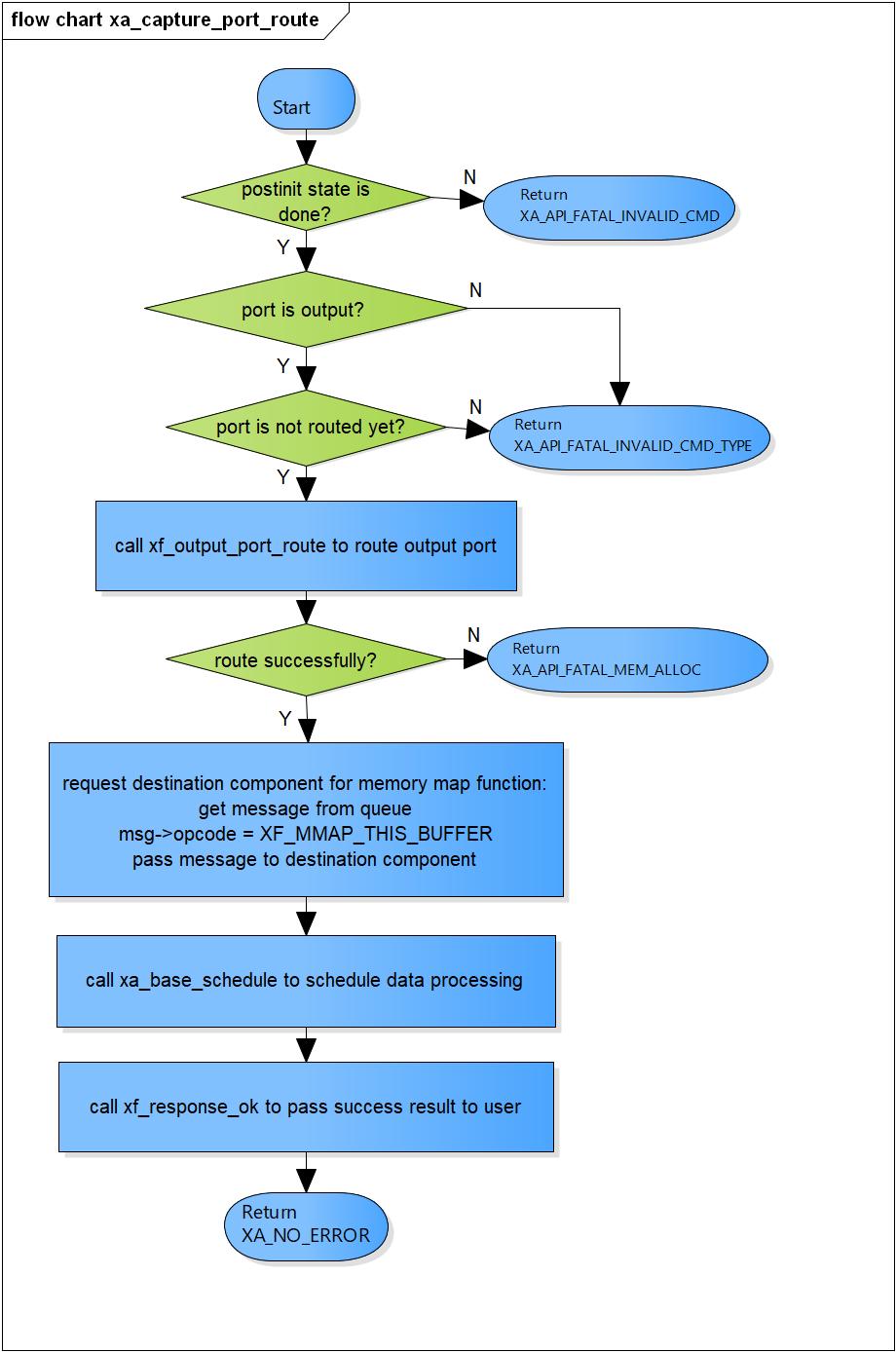


Figure 3‑5 xa\_capture\_port\_route flowchart

### xa\_capture\_port\_unroute

DD\_FWK\_RDR\_01\_006

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_capture\_port\_unroute(XACodecBase \*base, xf\_message\_t \*m) | | | |
| **Function** | This function is to unroute output port and destroy all buffers allocated. | | | |
| **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 output port. | |
| XA\_API\_FATAL\_INVALID\_CMD | | Post initialization state is not completed yet. | |
| **Description** | * Port unroute command command processing:   - Cancel any pending processing.  - Unroute output port. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_007]

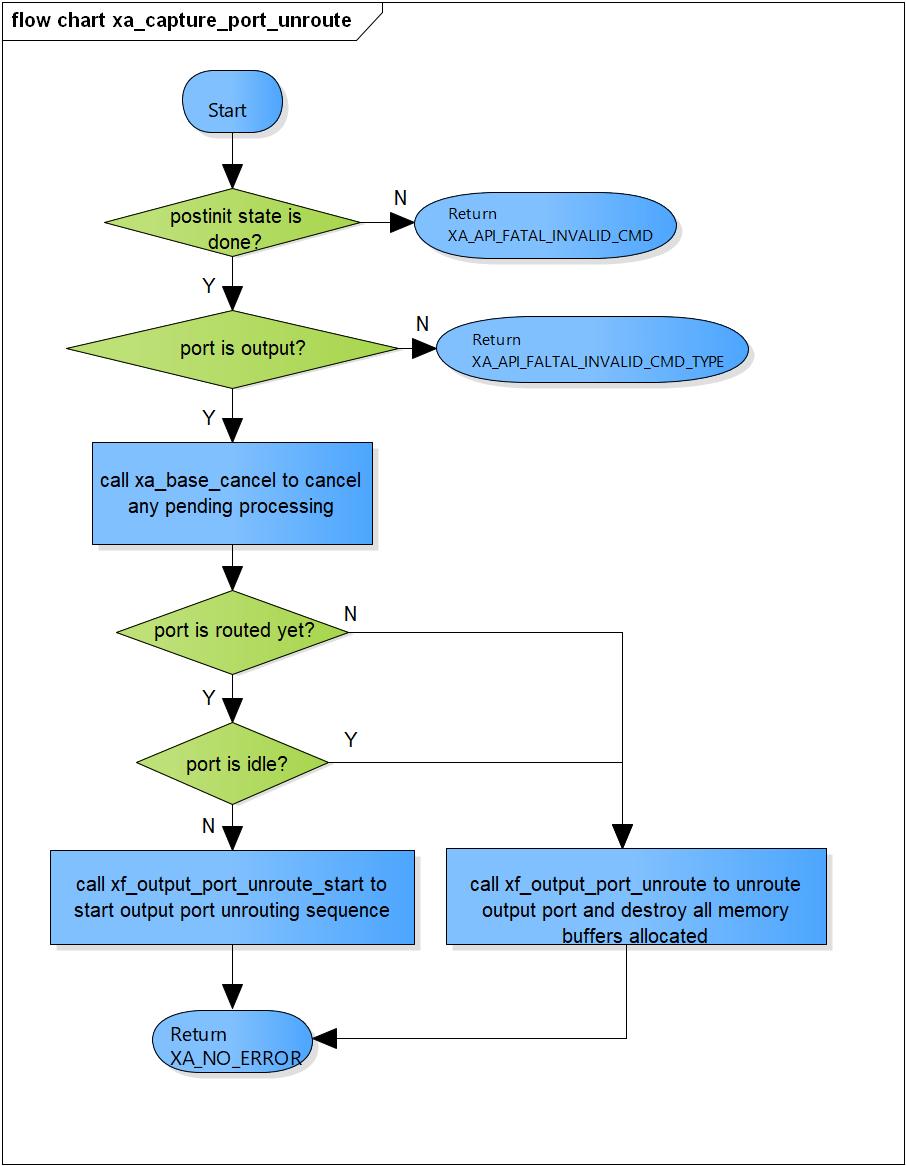


Figure 3‑6 xa\_capture\_port\_unroute flowchart

### xa\_capture\_preprocess

DD\_FWK\_RDR\_01\_007

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_capture\_preprocess(XACodecBase \*base) | | | |
| **Function** | This function is to set the output buffer pointer. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I | Pointer to codec instance structure (struct XACodecBase). |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_CAP\_EXEC\_NONFATAL\_OUTPUT | | No output buffer available. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | API structure or output buffer is NULL (error from plugin). | |
| XA\_API\_FATAL\_MEM\_ALIGN | | API structure is not aligned to 4 bytes (error from plugin). | |
| XA\_API\_FATAL\_INVALID\_CMD\_TYPE | | Output port index is an invalid memory index (error from plugin). | |
| XA\_CAP\_CONFIG\_FATAL\_STATE | | Capture is not in post-init state (error from plugin). | |
| **Description** | * Preprocess command processing:   - Check current execution stage is runtime intialization.  - Set the output buffer pointer.  - Check if input buffer is over.  - Set total number of bytes we have in buffer. | | | |

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

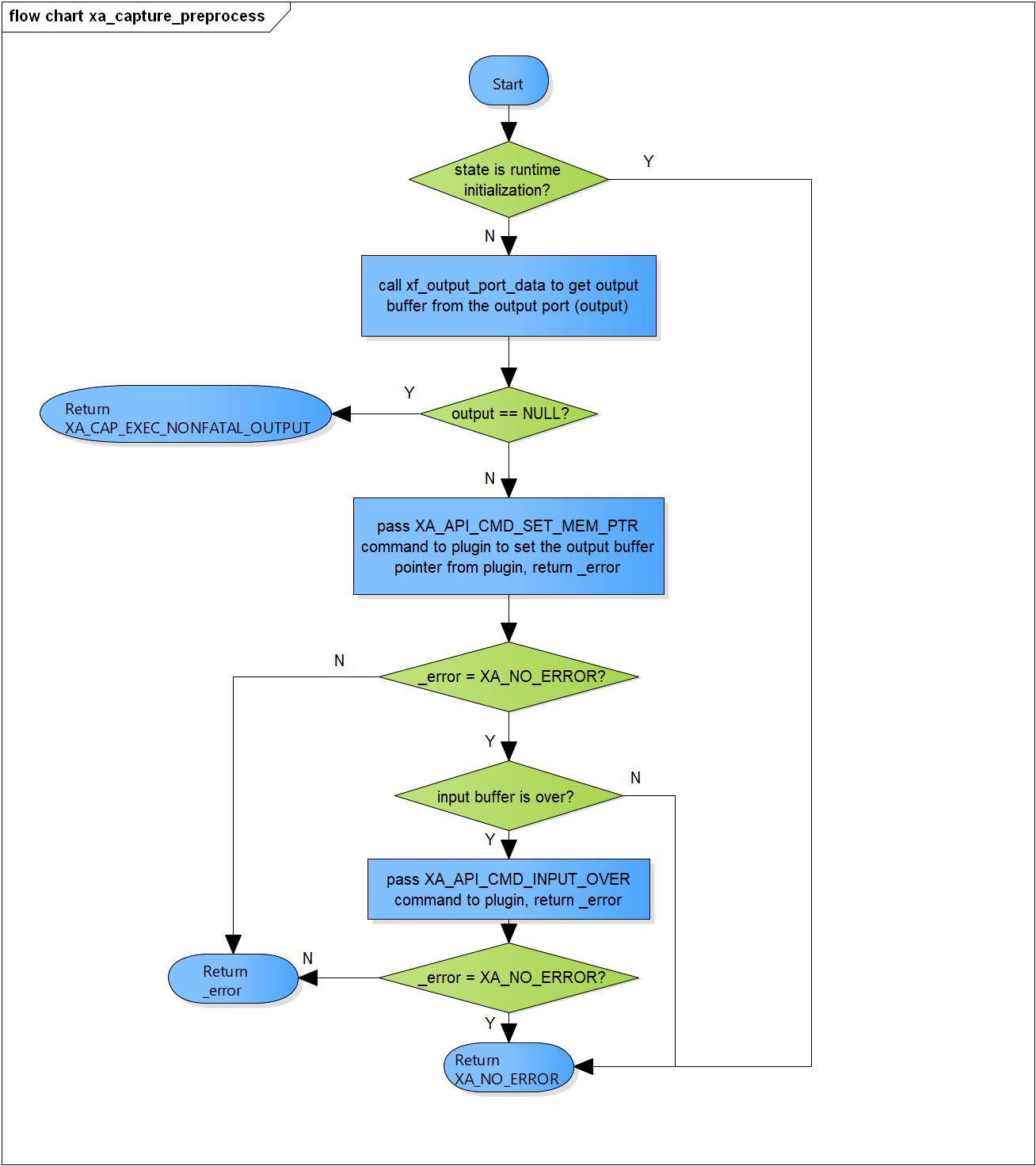


Figure 3‑7 xa\_capture\_preprocess flowchart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_capture\_postprocess(XACodecBase \*base, s32 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). |
| s32 | done | I | State of processing of plugin. |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | API structure is invalid (error from plugin). | |
| XA\_CAP\_EXEC\_FATAL\_STATE | | Capture is not in running state (error from plugin).  Ring-buffer is invalid (error from plugin). | |
| **Description** | * xa\_capture\_postprocess command processing:   - Get number of produced samples from plugin.  - Check state if process is done.  - Reschedule execution if we have pending output. | | | |

### xa\_capture\_postprocess

DD\_FWK\_RDR\_01\_008

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

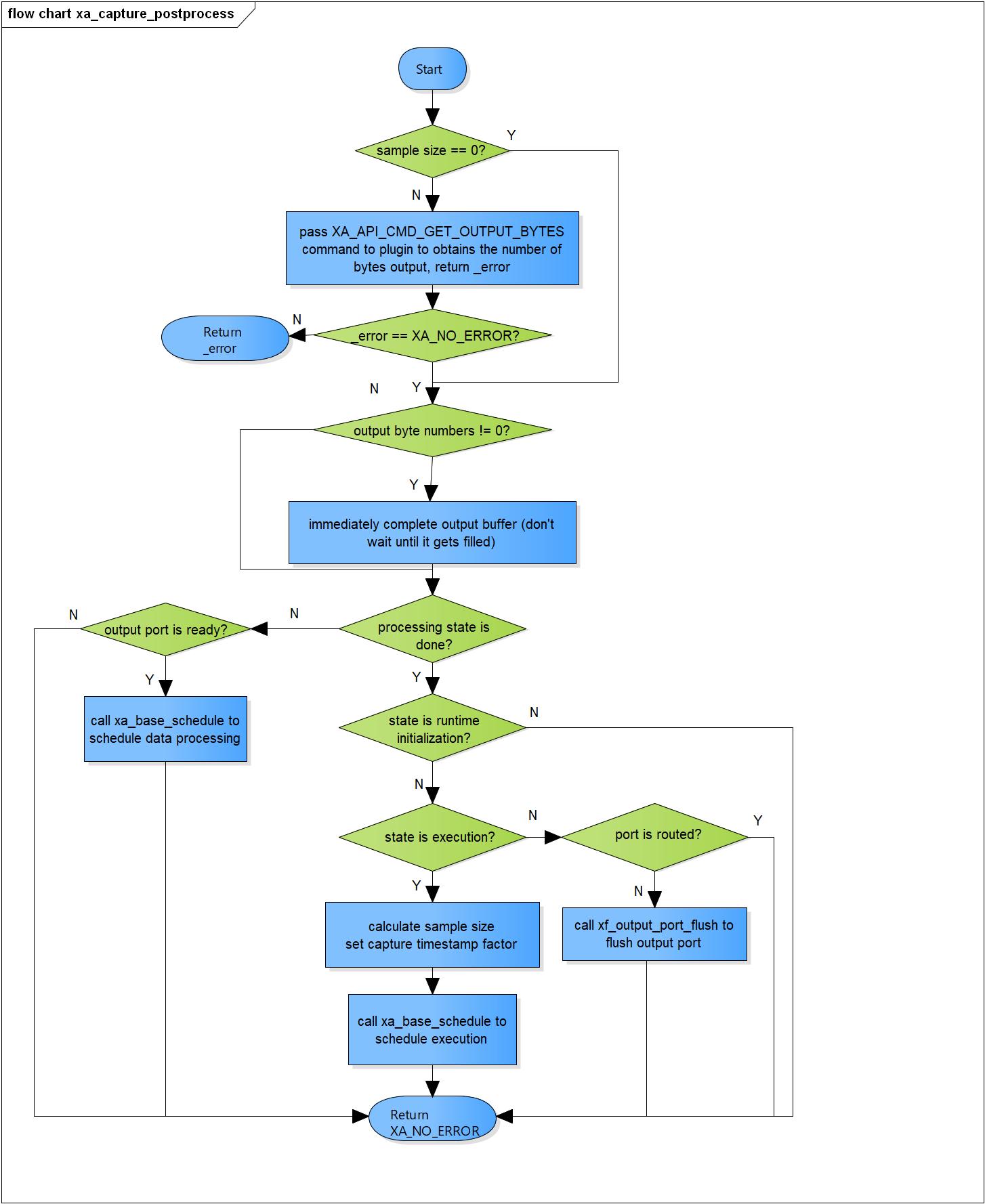


Figure 3‑8 xa\_capture\_postprocess flowchart

### xa\_capture\_flush

DD\_FWK\_RDR\_01\_009

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_capture\_flush(XACodecBase \*base, xf\_message\_t \*m) | | | |
| **Function** | This function is to purge port queue. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I | 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. | |
| **Description** | * xa\_capture\_flush command processing:   - Check if post initialization state is not completed yet.  - Ensure input parameter length is zero.  - Cancel data processing if needed.  - Output port flushing; purge content of output buffer. | | | |

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

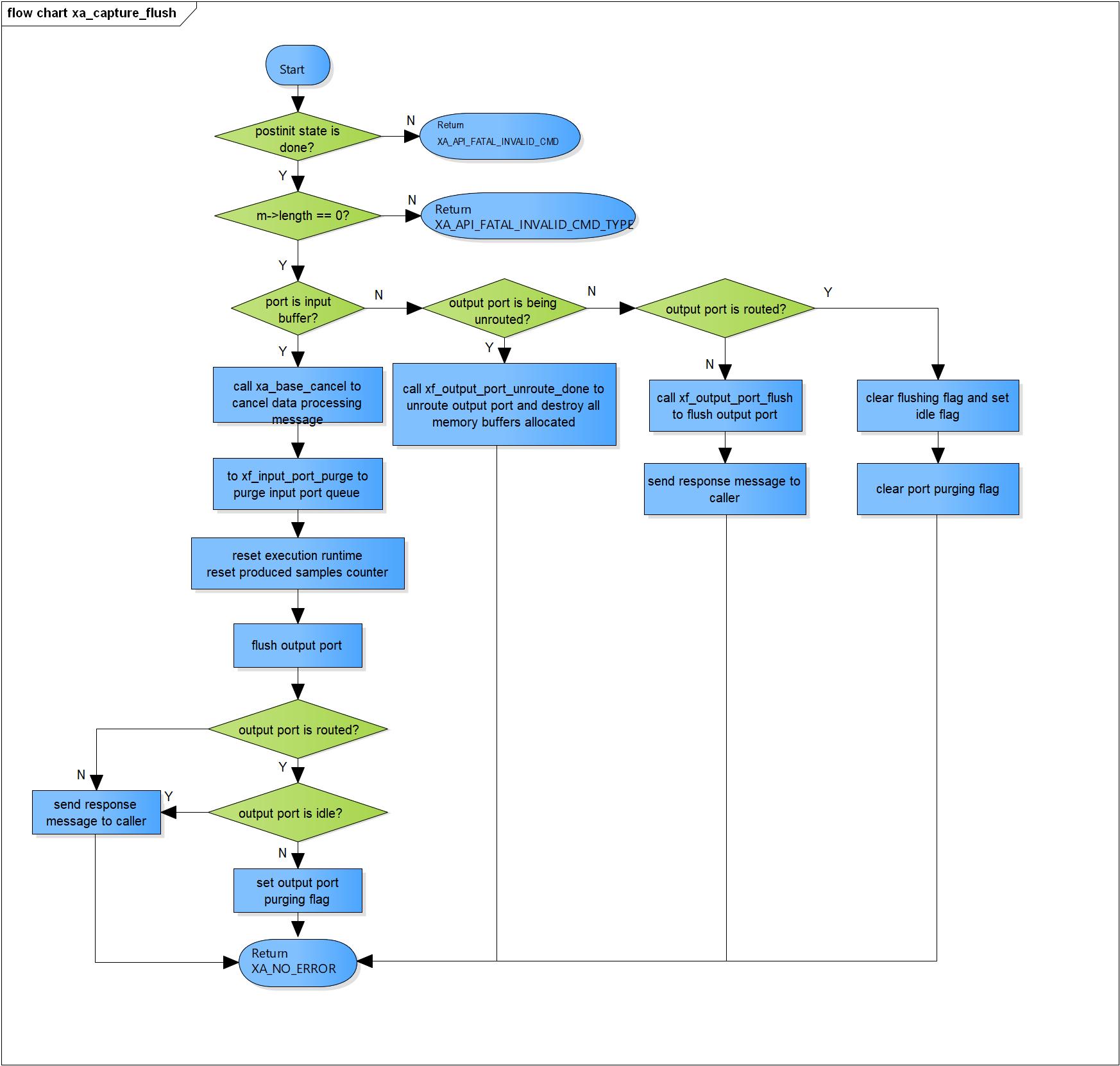


Figure 3‑9 xa\_capture\_flush flowchart

### xa\_capture\_cleanup

DD\_FWK\_RDR\_01\_010

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_capture\_cleanup(xf\_component\_t \*component, xf\_message\_t \*m) | | | |
| **Function** | This function is to purge port queue and cancel component task execution. | | | |
| **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/O | Pointer to audio message (struct xf\_message). |
| **Return value** | 1 | | Capture component cannot be destroyed. | |
| 0 | | Capture component has been destroyed totally. | |
| **Description** | * xa\_capture\_cleanup command processing:   - Complete message with error result code.  - Cancel component task execution if needed.  - Stop hardware capture if it's running.  - Destroy capture. | | | |

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

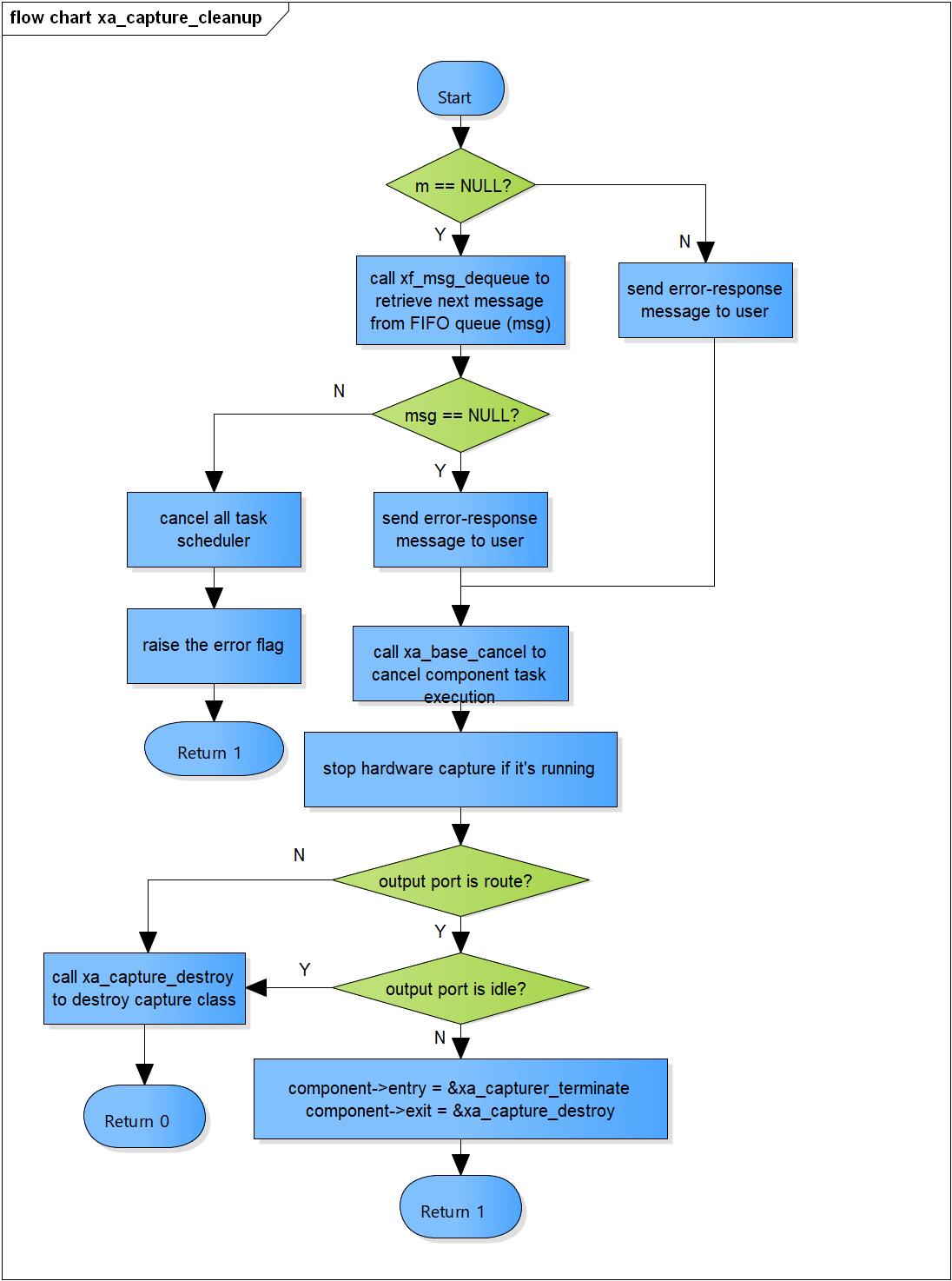


Figure 3‑10 xa\_capture\_cleanup flowchart

### xa\_capture\_terminate

DD\_FWK\_RDR\_01\_011

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_capture\_terminate(xf\_component\_t \*component, xf\_message\_t \*m) | | | |
| **Function** | Capture termination-state command processor. | | | |
| **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/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\_capture\_terminate command processing:   - Check if we received output port control message. | | | |

[Covers: FD\_FWK\_CMN\_004]

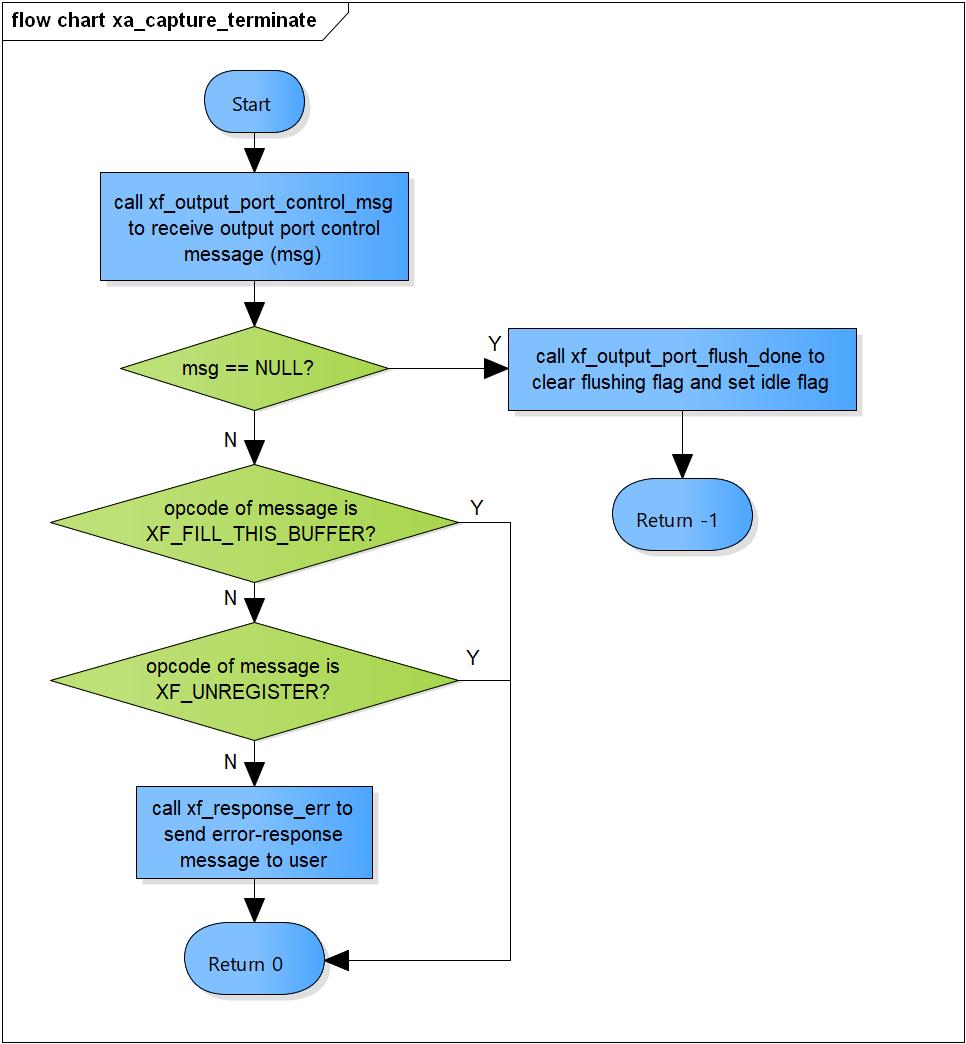


Figure 3‑11 xa\_capture\_terminate flowchart

### xa\_capture\_destroy

DD\_FWK\_RDR\_01\_012

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_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 | Pointer to codec instance structure (struct xf\_component ). |
| xf\_message\_t | m | x | Pointer to audio message (struct xf\_message). |
| **Return value** | 0 | | Indicate the component is destroyed. | |
| **Description** | * xa\_capture\_destroy command processing:   - Destroy output port.  - Destroy base object. | | | |

[Covers: FD\_FWK\_CMN\_004]

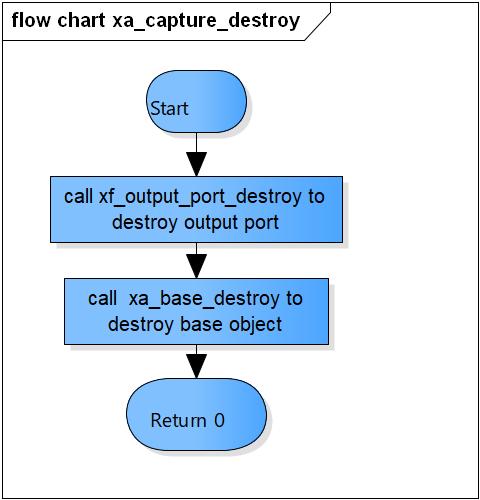


Figure 3‑12 xa\_capture\_destroy flowchart

### xa\_capture\_factory

DD\_FWK\_RDR\_01\_013

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | xf\_component\_t \* xa\_capture\_factory(u32 core, xa\_codec\_func\_t process) | | | |
| **Function** | This function is to initialize 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\_capture\_factory command processing:   - Initialize Capture component. | | | |

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

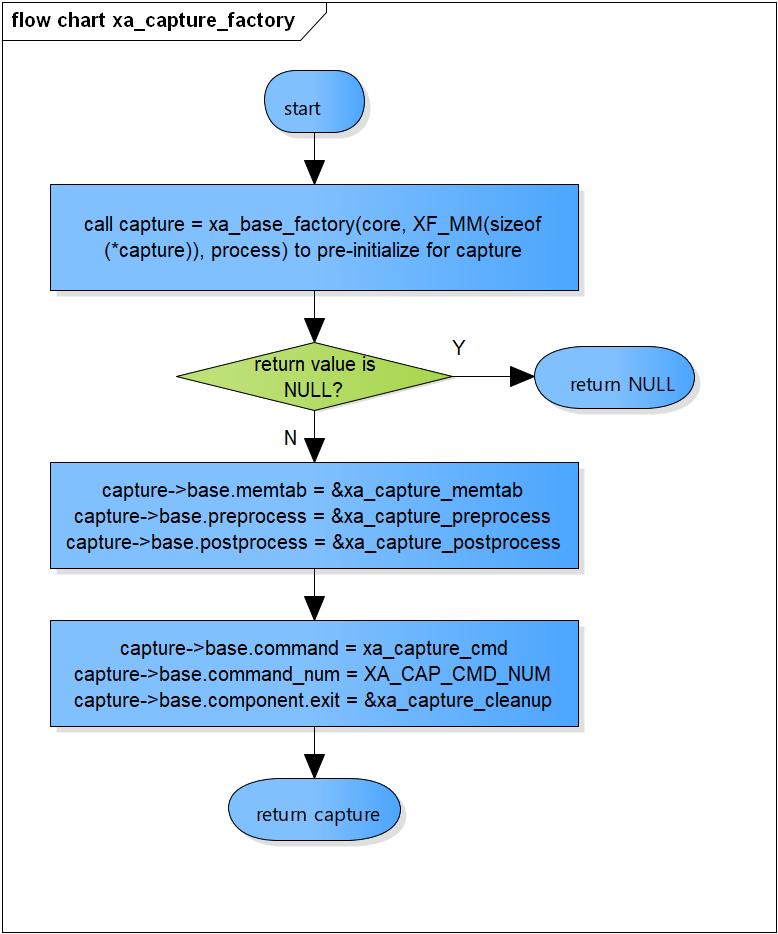


Figure 3‑13 xa\_capture\_factory flowchart

### xa\_capture\_mmap\_this\_buffer

DD\_FWK\_RDR\_01\_014

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_capture\_mmap\_this\_buffer(XACodecBase \*base, xf\_message\_t \*m) | | | |
| **Function** | This function is to map buffer address from user to output port index. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase\* | base | I | 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 | | Normal return | |
| **Description** | * xa\_capture\_mmap\_this\_buffer command processing:   - Check the message buffer and its length.  - Send buffer address for memory map process.  - Respond message back to user. | | | |

[Covers: FD\_FWK\_CMN\_013]

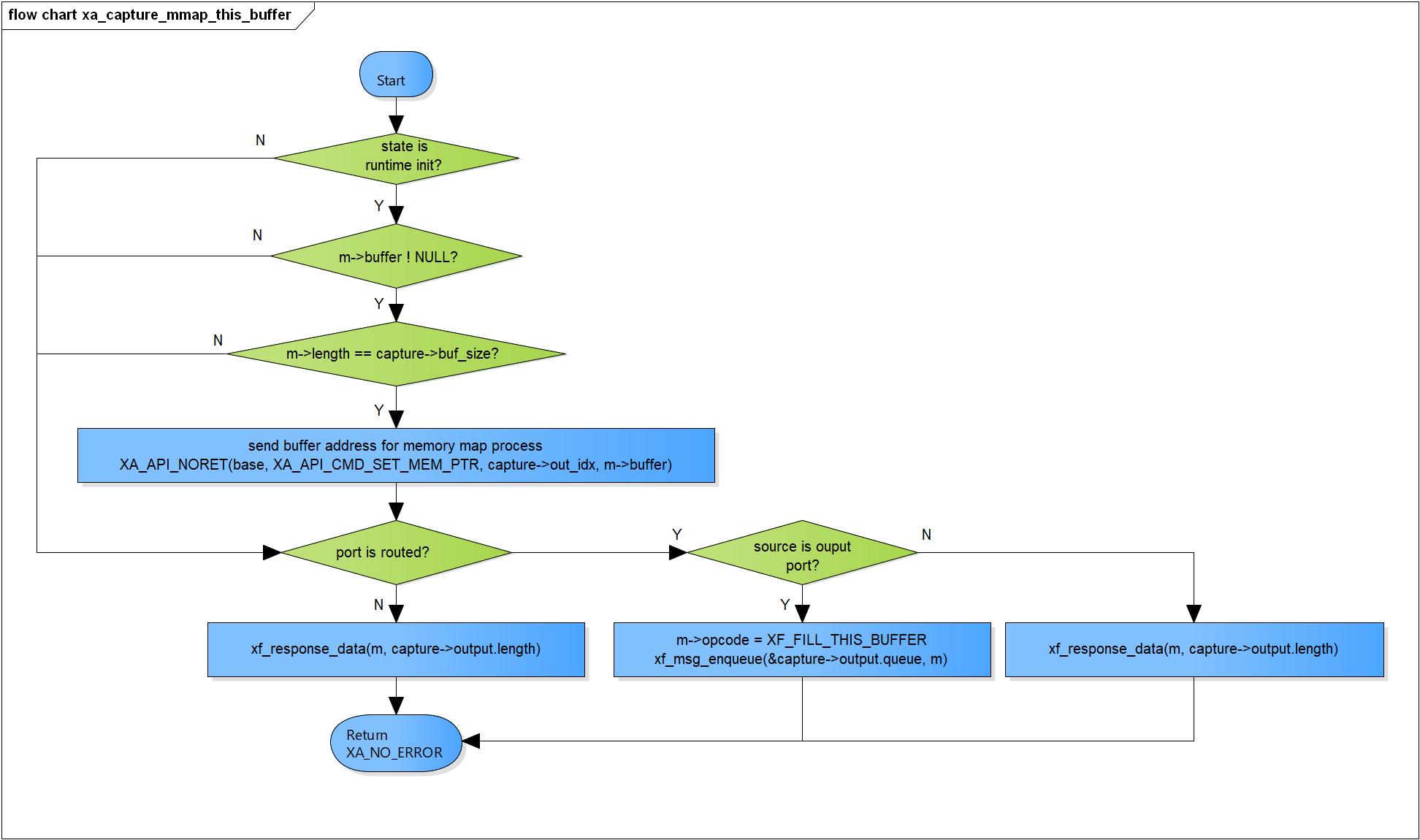


Figure 3‑14 xa\_capture\_mmap\_this\_buffer flowchart

# 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 |
| 1.3.0 | Apr 17, 2019 | - | Update for Android |  |  |