**Document Type: Detail Design**

**Document Name:**

**ADSP FRAMEWORK: TDM RENDERER CLASS**

Renesas Design Vietnam Co. Ltd

R-Car Software Solution 2 Group

Middleware 1 Team

Renesas Electronics Corporation

Table of Contents

[1. Overview 3](#_Toc534271012)

[2. Function list 4](#_Toc534271013)

[3. Detail information 5](#_Toc534271014)

[3.1 Data type and Macro definition 5](#_Toc534271015)

[3.2 Globle variable 6](#_Toc534271016)

[3.3 Function definition 6](#_Toc534271017)

[3.3.1 xa\_tdm\_renderer\_prepare\_runtime 6](#_Toc534271018)

[3.3.2 xa\_tdm\_renderer\_sufficient\_inport\_check 8](#_Toc534271019)

[3.3.3 xa\_tdm\_renderer\_empty\_this\_buffer 9](#_Toc534271020)

[3.3.4 xa\_tdm\_renderer\_fill\_this\_buffer 11](#_Toc534271021)

[3.3.5 xa\_tdm\_renderer\_memtab 12](#_Toc534271022)

[3.3.6 xa\_tdm\_renderer\_preprocess 14](#_Toc534271023)

[3.3.7 xa\_tdm\_renderer\_postprocess 16](#_Toc534271024)

[3.3.8 xa\_tdm\_renderer\_flush 18](#_Toc534271025)

[3.3.9 xa\_tdm\_renderer\_cleanup 19](#_Toc534271026)

[3.3.10 xa\_tdm\_renderer\_destroy 21](#_Toc534271027)

[3.3.11 xa\_tdm\_renderer\_factory 22](#_Toc534271028)

[4. Revision history 23](#_Toc534271029)

List of Figures

[Figure 1‑1 The software architecture 3](#_Toc534271030)

[Figure 3‑1 xa\_tdm\_renderer\_prepare\_runtime flowchart 7](#_Toc534271031)

[Figure 3‑2 xa\_tdm\_renderer\_sufficient\_inport\_check flowchart 8](#_Toc534271032)

[Figure 3‑3 xa\_tdm\_renderer\_empty\_this\_buffer flowchart 10](#_Toc534271033)

[Figure 3‑4 xa\_tdm\_renderer\_fill\_this\_buffer flowchart 11](#_Toc534271034)

[Figure 3‑5 xa\_tdm\_renderer\_memtab flowchart 13](#_Toc534271035)

[Figure 3‑6 xa\_tdm\_renderer\_preprocess flowchart 15](#_Toc534271036)

[Figure 3‑7 xa\_tdm\_renderer\_postprocess flowchart 17](#_Toc534271037)

[Figure 3‑8 xa\_tdm\_renderer\_flush flowchart 18](#_Toc534271038)

[Figure 3‑9 xa\_tdm\_renderer\_cleanup flowchart 20](#_Toc534271039)

[Figure 3‑10 xa\_tdm\_renderer\_destroy flowchart 21](#_Toc534271040)

[Figure 3‑11 xa\_tmd\_renderer\_factory flowchart 22](#_Toc534271041)

List of Table

[Table 2‑1 Function list 4](#_Toc534271042)

[Table 3‑1 XATDMRenderer type structure information 5](#_Toc534271043)

[Table 3‑2 Macro definitions 5](#_Toc534271044)

[Table 3‑3 Globle variable xa\_tdm\_renderer\_cmd 6](#_Toc534271045)

# 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\_renderer\_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 renderer for steady operation. |
| xa\_tdm\_capture\_sufficient\_outport\_check | This function is to check the sufficient input port. |
| xa\_tdm\_renderer\_empty\_this\_buffer | This function is to put message into all input port. |
| xa\_tdm\_renderer\_fill\_this\_buffer | This is a dummy function. |
| xa\_tdm\_renderer\_flush | This function is to purge input port queue and cancel data processing. |
| xa\_tdm\_renderer\_memtab | This function is to initialize input port structure. |
| xa\_tdm\_renderer\_preprocess | This function is to prepare data for all input buffer. |
| xa\_tdm\_renderer\_postprocess | This function is to consume bytes from all input buffer and reschedule processing. |
| xa\_tdm\_renderer\_destroy | This function is to destroy component. |
| xa\_tdm\_renderer\_cleanup | This function is to purge input port queue and cancel internal scheduling. |

# Detail information

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

## Data type and Macro definition

The XATDMRenderer type structure is the work area used by the TDM renderer 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 XATDMRenderer 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 input\_total | The number of total input 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-renderer-api.h

Table 3‑2 Macro definitions

|  |  |  |
| --- | --- | --- |
| Macro | Value | Outline |
| XA\_TDM\_RDR\_FLAG\_INPUT\_PORT\_INIT\_DONE | \_\_XA\_BASE\_FLAG(1 << 0) | Input port init done flag |
| XA\_TDM\_RDR\_FLAG\_EOS | \_\_XA\_BASE\_FLAG(1 << 1) | End of stream signal |

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

## Globle variable

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

|  |  |
| --- | --- |
| static XA\_ERRORCODE (\* const xa\_tdm\_renderer\_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\_renderer\_empty\_this\_buffer |
| XF\_OPCODE\_TYPE(XF\_FILL\_THIS\_BUFFER) | xa\_tdm\_renderer\_fill\_this\_buffer |
| XF\_OPCODE\_TYPE(XF\_FLUSH) | xa\_tdm\_renderer\_flush |

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

## Function definition

### xa\_tdm\_renderer\_prepare\_runtime

DD\_FWK\_TDM\_02\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static inline XA\_ERRORCODE xa\_tdm\_renderer\_prepare\_runtime(XATDMRenderer \*tdm\_rdr) | | | |
| **Function** | This function is to prepare TDM renderer for steady operation. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XATDMRenderer | tdm\_rdr | I/O | Pointer to codec instance structure (struct XATDMRenderer). |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_TDM\_RDR\_CONFIG\_FATAL\_CHANNEL\_MODE | | Channel mode is invalid. | |
| XA\_TDM\_RDR\_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\_RDR\_CONFIG\_FATAL\_STATE | | Plugin has not pre-init yet (error from plugin). | |
| **Description** | * xa\_tdm\_renderer\_prepare\_runtime command processing:   - Get config parameters: sample rate, pcm width, channels.  - Calculate sample size.  - Set renderer timestamp factor. | | | |

[Covers: FD\_FWK\_CMN\_002]

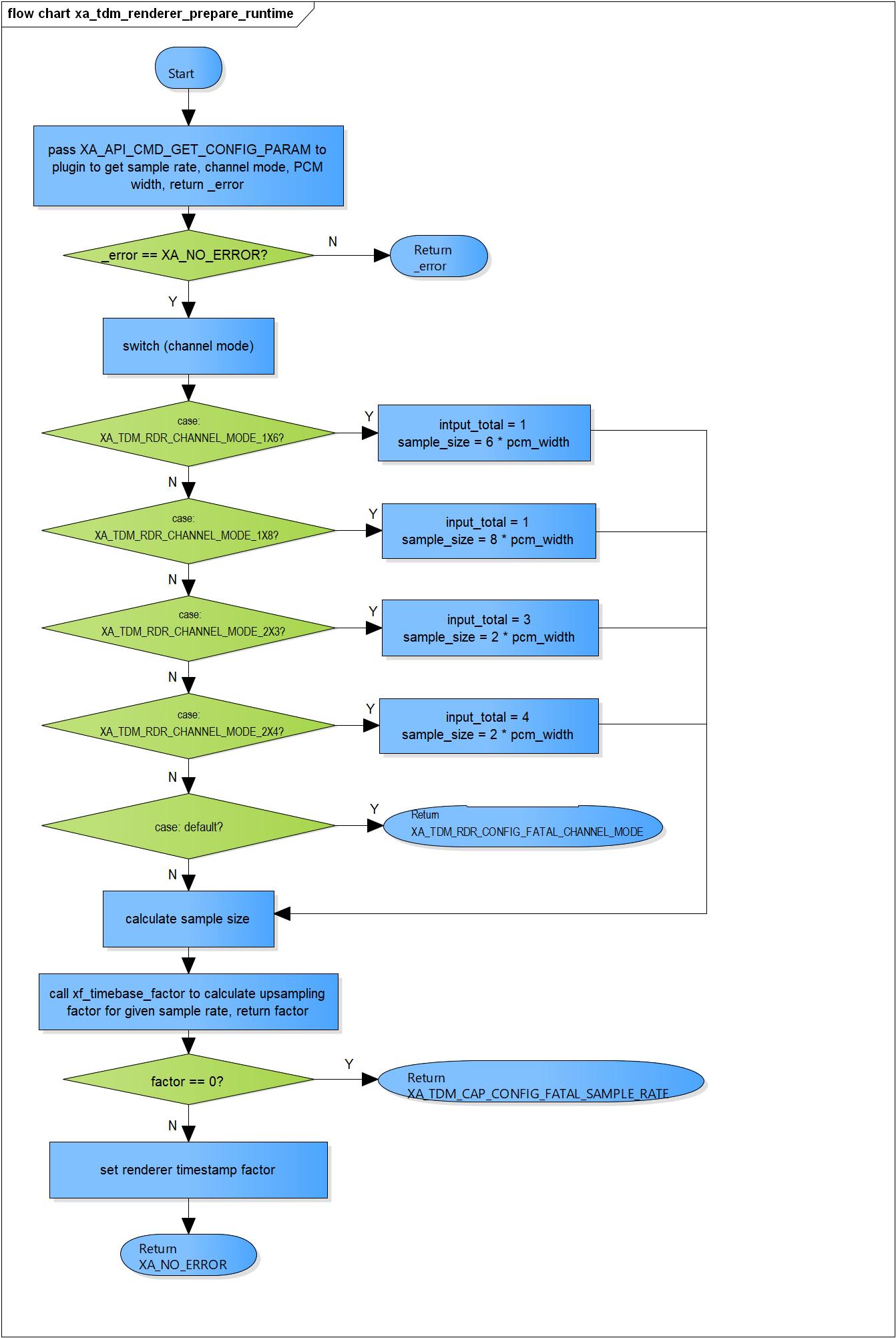


Figure 3‑1 xa\_tdm\_renderer\_prepare\_runtime flowchart

### xa\_tdm\_renderer\_sufficient\_inport\_check

DD\_FWK\_TDM\_02\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static inline s32 xa\_tdm\_renderer\_sufficient\_outport\_check(XACodecBase \*base) | | | |
| **Function** | This function is to check the sufficient input 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\_002, FD\_FWK\_CMN\_003, FD\_FWK\_CMN\_010]

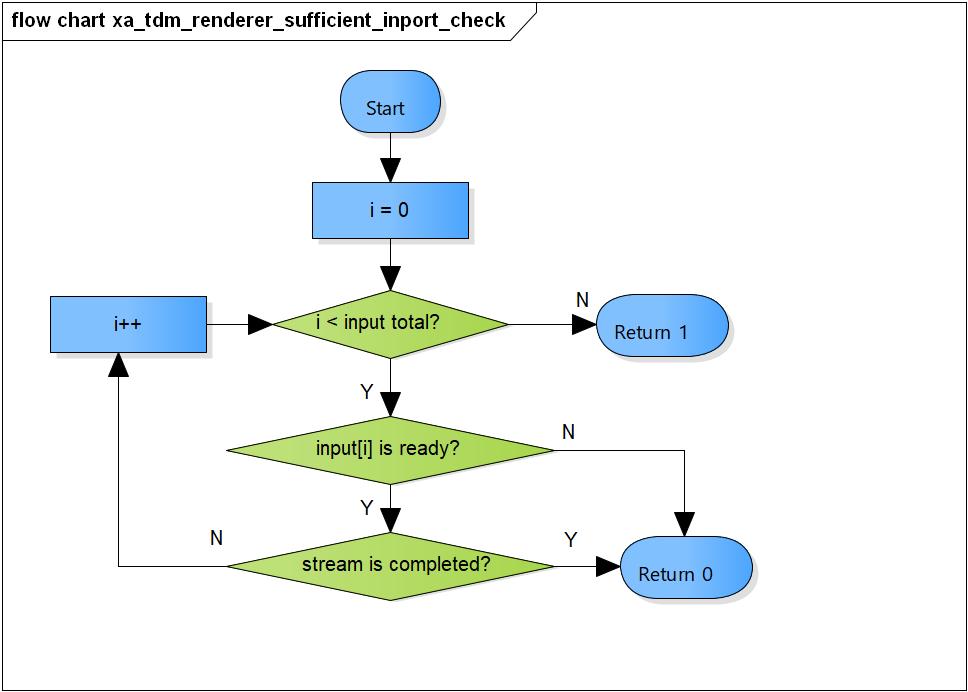


Figure 3‑2 xa\_tdm\_renderer\_sufficient\_inport\_check flowchart

### xa\_tdm\_renderer\_empty\_this\_buffer

DD\_FWK\_TDM\_02\_003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_renderer\_empty\_this\_buffer(XACodecBase \*base,  xf\_message\_t \*m) | | | |
| **Function** | This function is to put message into all 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\_API\_FATAL\_INVALID\_CMD\_TYPE | | Port is invalid. | |
| XA\_API\_FATAL\_INVALID\_CMD | | Post initialization state is not completed yet. | |
| XA\_NO\_ERROR | | Nomally end. | |
| **Description** | * xa\_tdm\_renderer\_empty\_this\_buffer command processing:   - Make sure the port is valid.  - Check if post initialization state is completed.  - Put message into the input port. | | | |

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

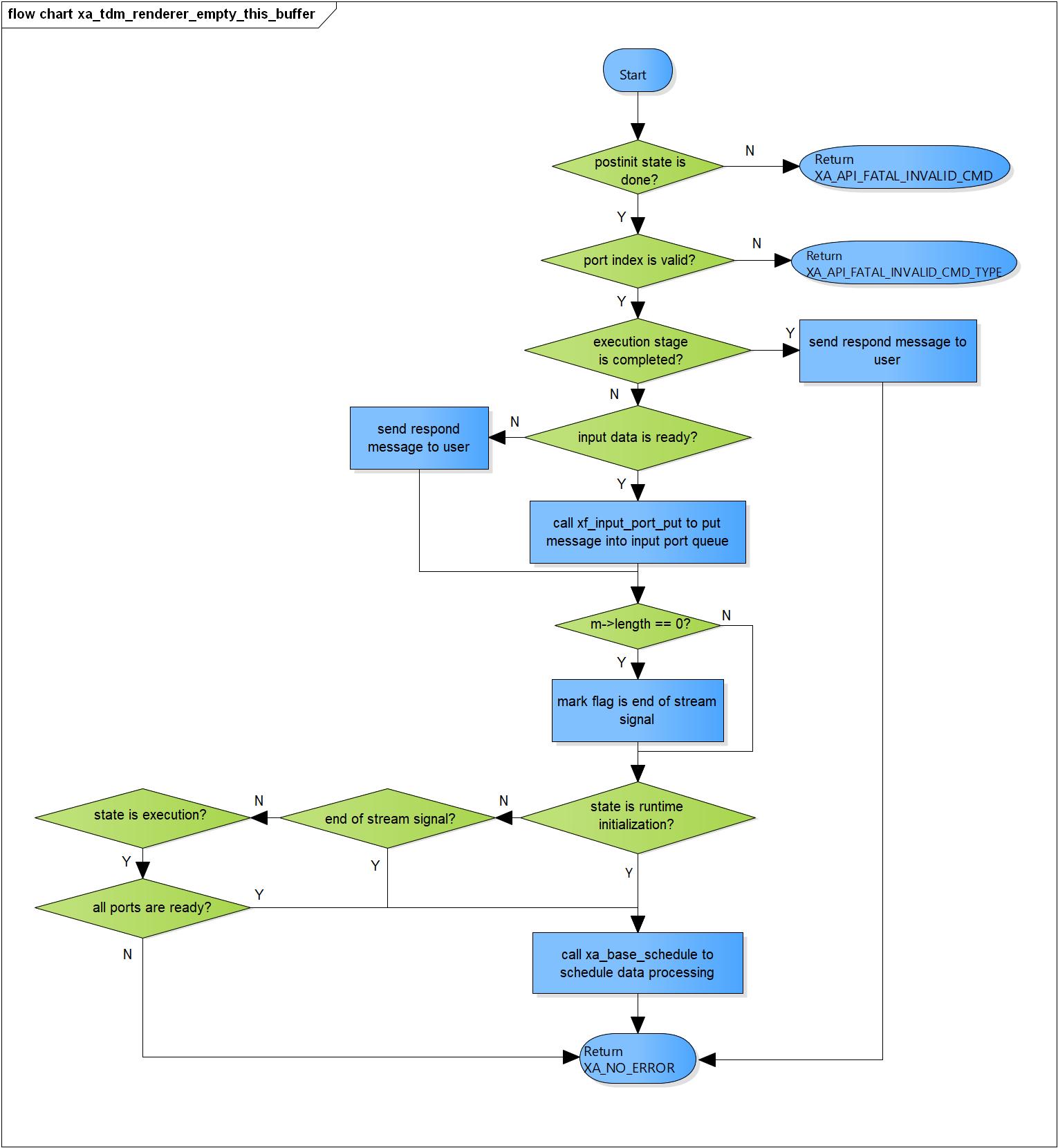


Figure 3‑3 xa\_tdm\_renderer\_empty\_this\_buffer flowchart

### xa\_tdm\_renderer\_fill\_this\_buffer

DD\_FWK\_TDM\_02\_004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_renderer\_fill\_this\_buffer(XACodecBase \*base,  xf\_message\_t \*m) | | | |
| **Function** | This is a dummy function to send message back to user. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | X | 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 | | If have any port is invalid. | |
| **Description** | * xa\_tdm\_renderer\_fill\_this\_buffer command processing:   - Send respond message back to user. | | | |

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

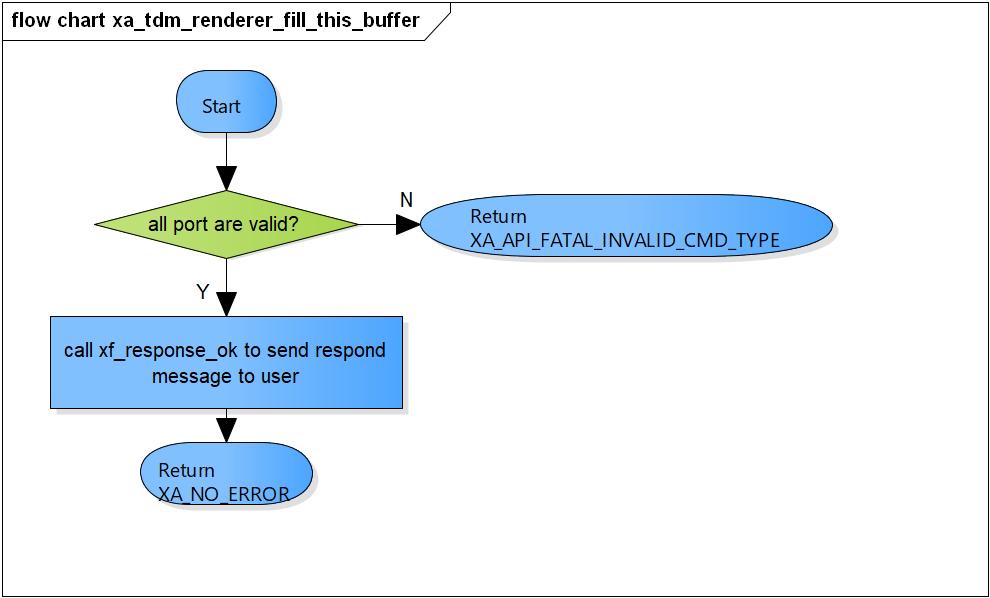


Figure 3‑4 xa\_tdm\_renderer\_fill\_this\_buffer flowchart

### xa\_tdm\_renderer\_memtab

DD\_FWK\_TDM\_02\_005

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_renderer\_memtab(XACodecBase \*base, WORD32 idx, WORD32 type, WORD32 size, WORD32 align, u32 core) | | | |
| **Function** | This function is to initialize input port structure | | | |
| **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\_INPUT |
| WORD32 | size | I | Buffer size of port |
| WORD32 | align | I | Port alignment |
| u32 | core | I | Core index of ADSP framework |
| **Return value** | XA\_NO\_ERROR | | Nomally end. | |
| XA\_API\_FATAL\_INVALID\_CMD\_TYPE | | Port type is not a input port.  Port index are invalid. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | Initialize input port is fail.  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\_RDR\_CONFIG\_FATAL\_STATE | | TDM-Renderer is not in post-init state (error from plugin). | |
| **Description** | * Memtab command processing:   - Initialize input port for a track.  - Set input port buffer. | | | |

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

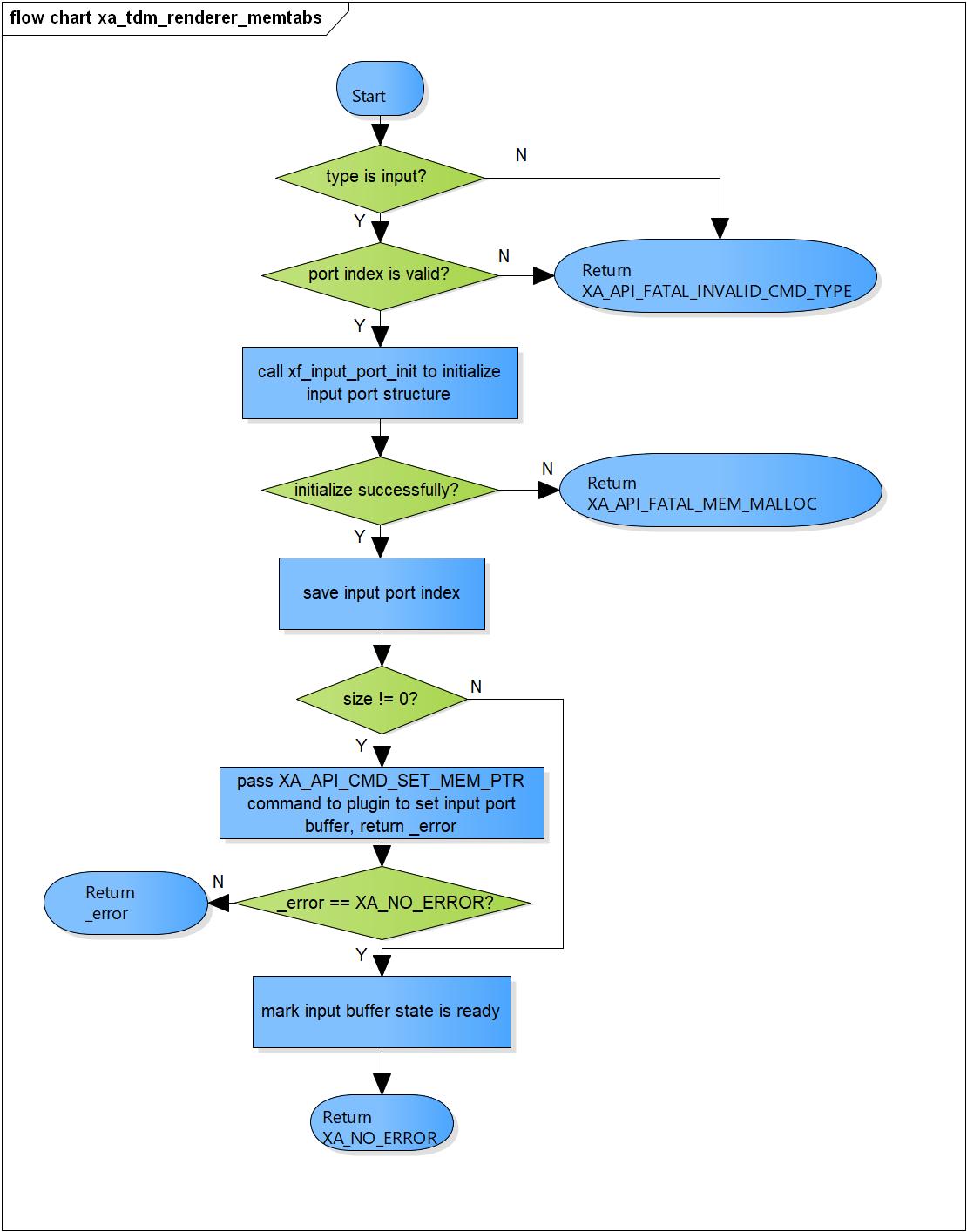


Figure 3‑5 xa\_tdm\_renderer\_memtab flowchart

### xa\_tdm\_renderer\_preprocess

DD\_FWK\_TDM\_02\_006

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_renderer\_preprocess(XACodecBase \*base) | | | |
| **Function** | This function is to prepare data for all input buffer. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase). |
| **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 structure is not aligned to 4 bytes (error from plugin). | |
| XA\_TDM\_RDR\_CONFIG\_FATAL\_STATE | | TDM-Renderer is not in init state (error from plugin).  TDM-Renderer is not in post-init state (error from plugin). | |
| XA\_TDM\_RDR\_EXEC\_FATAL\_STATE | | Input area is not ready (error from plugin). | |
| XA\_TDM\_RDR\_EXEC\_FATAL\_INPUT | | Invalid number of submitted data (error from plugin). | |
| **Description** | * Preprocess command processing:   - Submit all input buffer.  - Set total number of bytes we have in buffer. | | | |

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

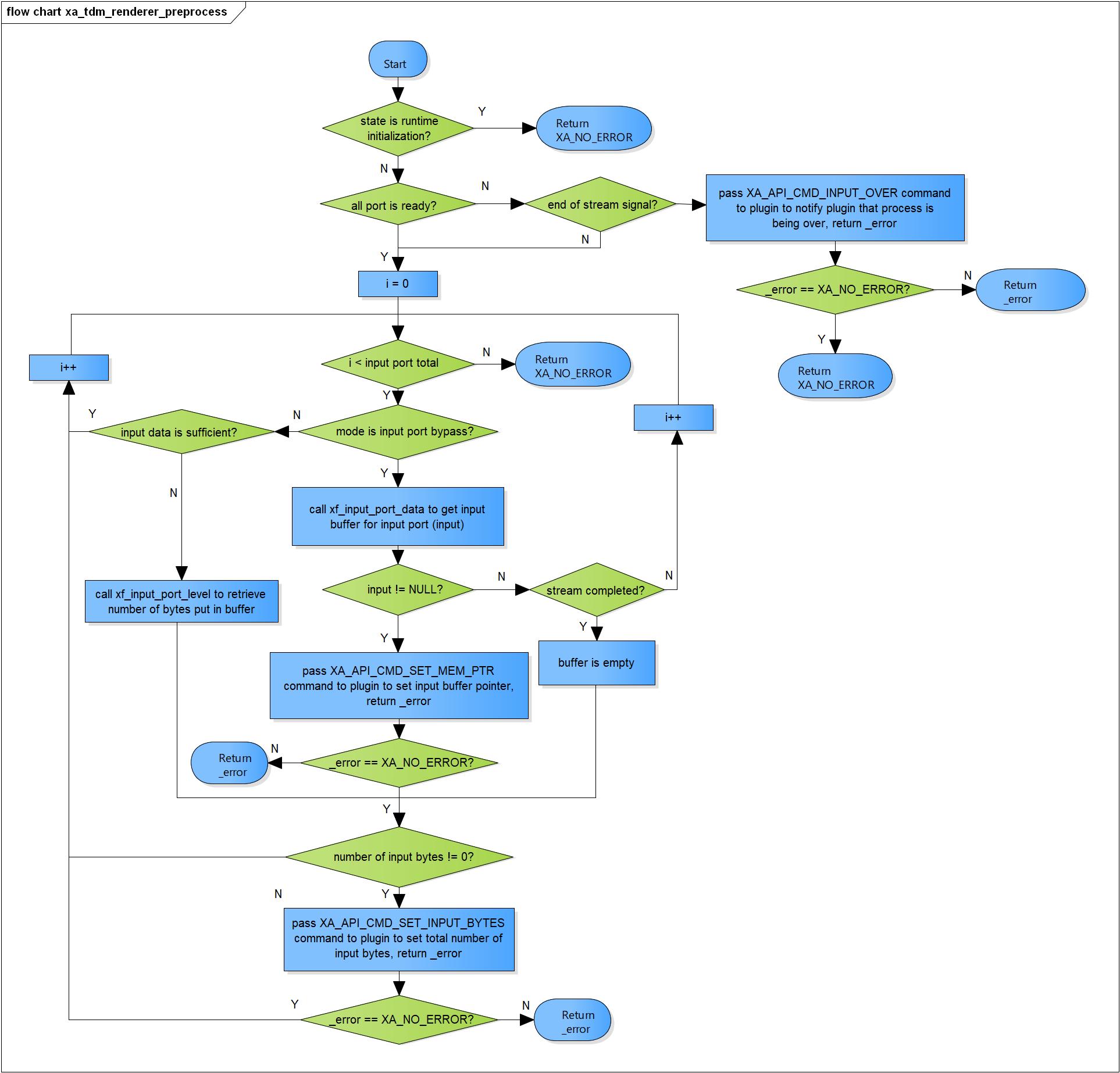


Figure 3‑6 xa\_tdm\_renderer\_preprocess flowchart

### xa\_tdm\_renderer\_postprocess

DD\_FWK\_TDM\_02\_007

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_renderer\_postprocess(XACodecBase \*base,  WORD32 done) | | | |
| **Function** | This function is to consume bytes from all input buffer 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 structure is not aligned to 4 bytes (error from plugin). | |
| XA\_TDM\_RDR\_EXEC\_FATAL\_STATE | | Input area is not ready. | |
| **Description** | * xa\_tdm\_renderer\_postprocess command processing:   - Check state if processing is done.  - Get total amount of consumed sample from plugin, consume bytes from input buffer.  - Reschedule execution if we have pending input. | | | |

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

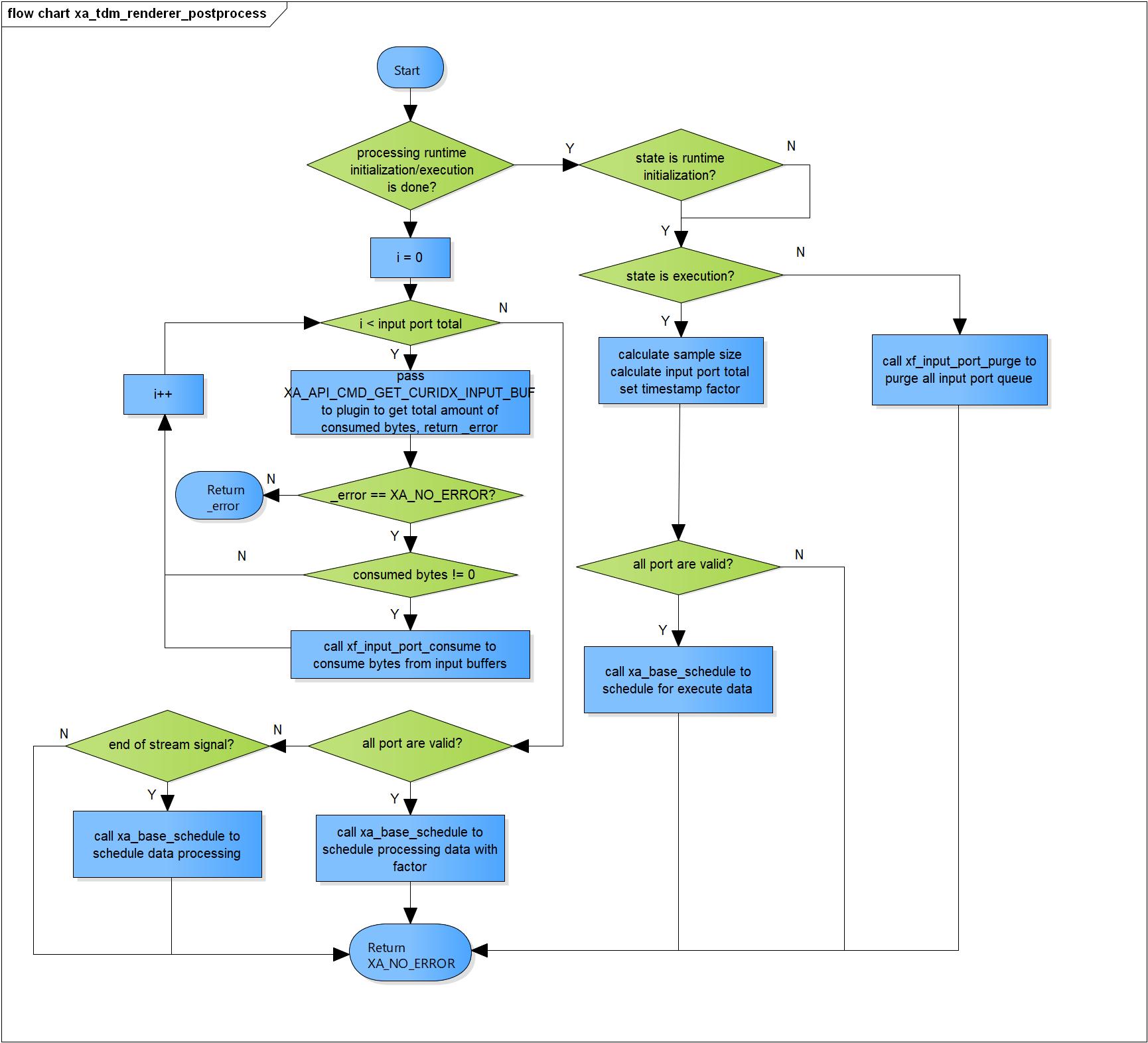


Figure 3‑7 xa\_tdm\_renderer\_postprocess flowchart

### xa\_tdm\_renderer\_flush

DD\_FWK\_TDM\_02\_008

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_tdm\_renderer\_flush(XACodecBase \*base, xf\_message\_t \*m) | | | |
| **Function** | This function is to purge input port queue and cancel data processing. | | | |
| **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 | | Message length is not zero. | |
| **Description** | * xa\_tdm\_renderer\_flush command processing:   - Ensure input parameter length is zero.  - Cancel data processing.  - Purge all input port queue.  - Send respond message to user | | | |

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

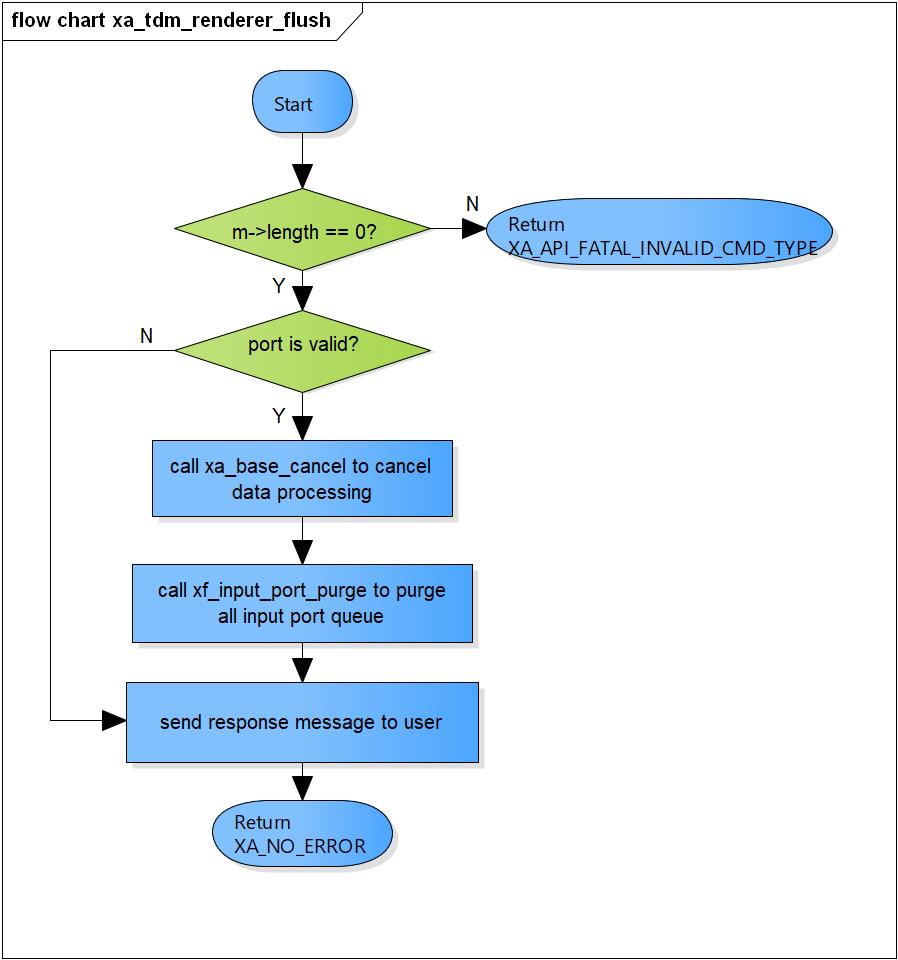


Figure 3‑8 xa\_tdm\_renderer\_flush flowchart

### xa\_tdm\_renderer\_cleanup

DD\_FWK\_TDM\_02\_009

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_tdm\_renderer\_cleanup(xf\_component\_t \*component, xf\_message\_t \*m) | | | |
| **Function** | This function is to purge input 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/O | Pointer to audio message (struct xf\_message). |
| **Return value** | 1 | TDM renderer component cannot be destroyed. | | |
| 0 | TDM renderre component has been destroyed totally. | | |
| 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\_RDR\_CONFIG\_FATAL\_STATE | TDM-Renderer is not init yet (error from plugin). | | |
| **Description** | * xa\_tdm\_renderer\_cleanup command processing:   - Complete message with error response.  - Cancel internal scheduling message if needed.  - Purge all input port. | | | |

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

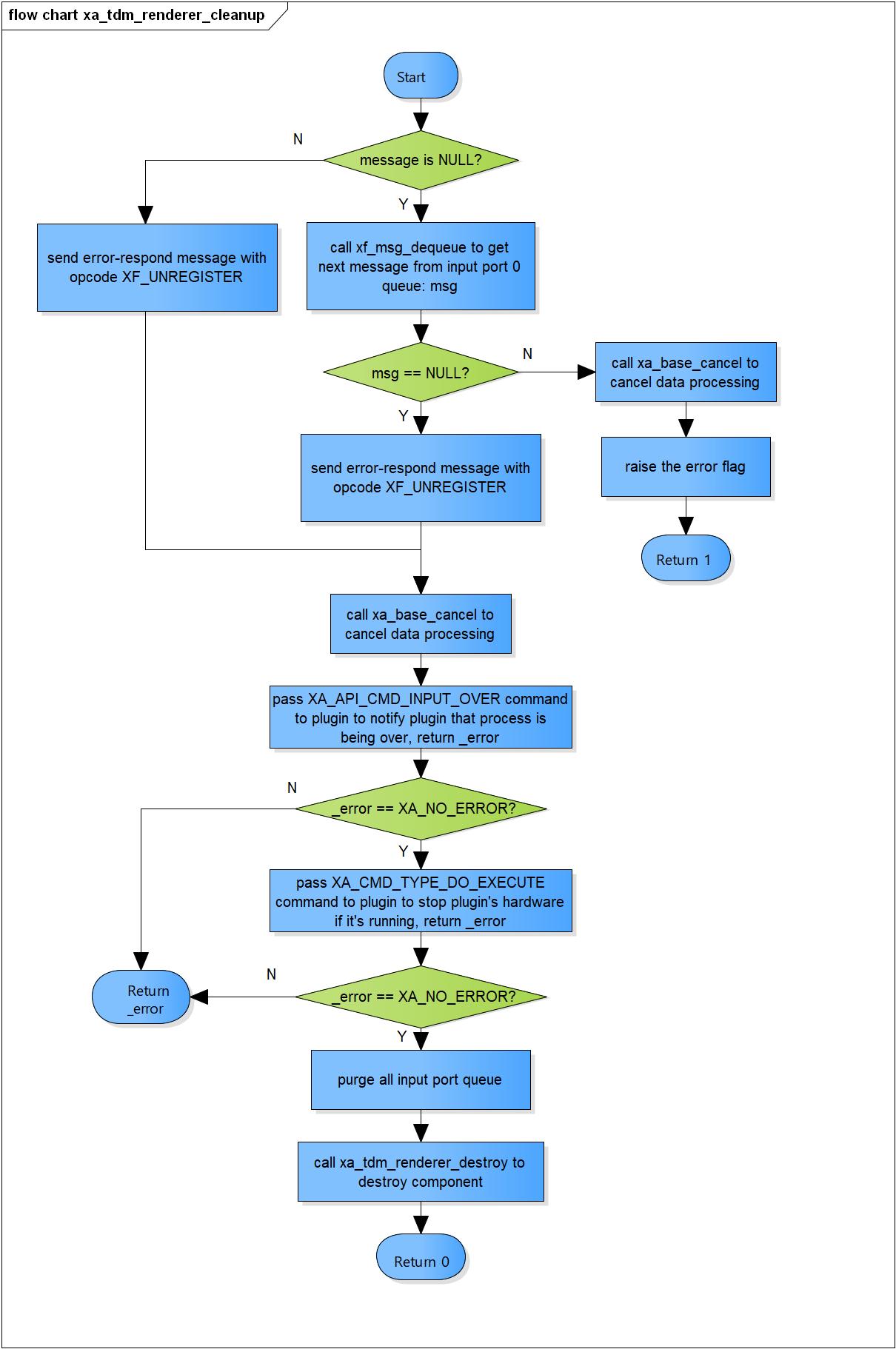


Figure 3‑9 xa\_tdm\_renderer\_cleanup flowchart

### xa\_tdm\_renderer\_destroy

DD\_FWK\_TDM\_02\_010

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_tdm\_renderer\_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 | I | Pointer to audio message (struct xf\_message). |
| 0 | | Indicate the component is destroyed. | |
| **Description** | * xa\_tdm\_renderer\_destroy command processing:   - Destroy all input port.  - Destroy base object. | | | |

[Covers: FD\_FWK\_CMN\_004]

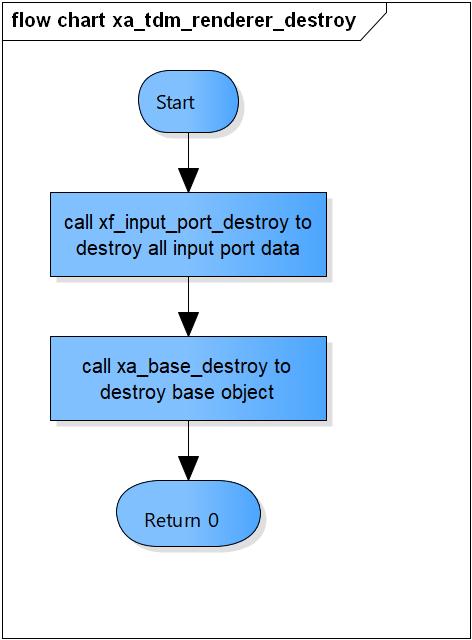


Figure 3‑10 xa\_tdm\_renderer\_destroy flowchart

### xa\_tdm\_renderer\_factory

DD\_FWK\_TDM\_02\_011

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

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

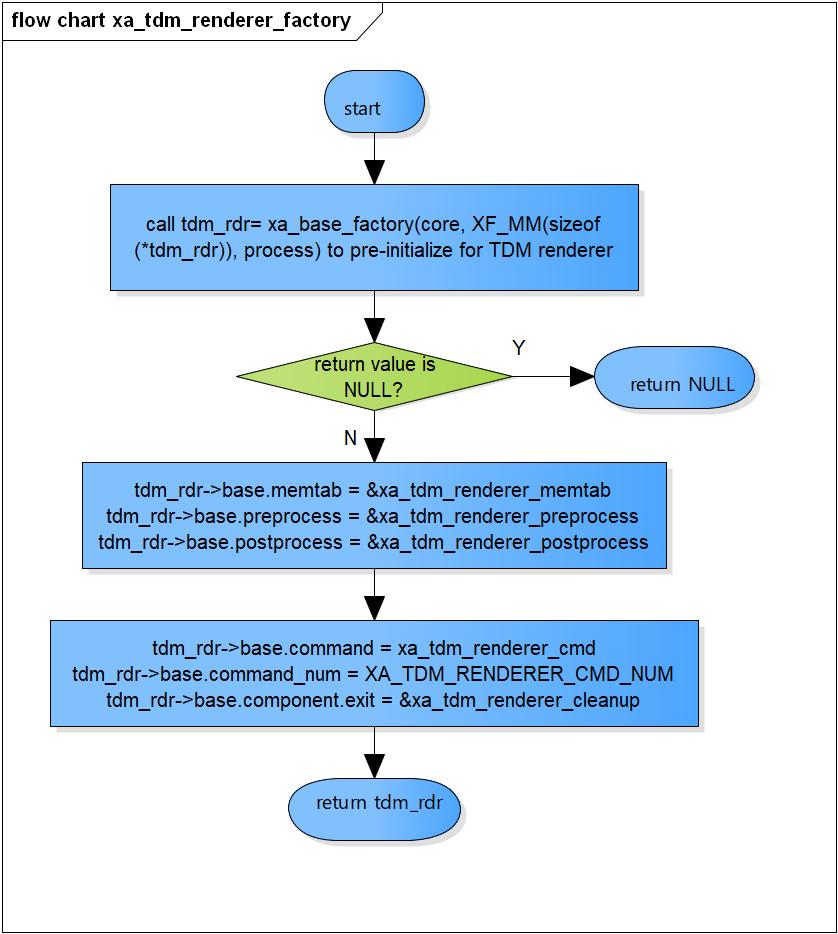


Figure 3‑11 xa\_tmd\_renderer\_factory flowchart

# Revision history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Page** | **Content** | **Approved** | **Changed** |
| 1.0.0 | - | - | 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 |