User's Manual

32

# OMX Media Component

## User's Manual: H.264 Encoder Part

— Preliminary —



Renesas Electronics
www.renesas.com

Rev.1.01 Dec. 15, 2014

# Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

## Table of Contents

## Figures

## Tables

# 1. Overview

## 1.1. About This Document

This document is the User's Manual for OMX Media Component. It describes the specifications of H.264 Encoder Media Component. For the specifications that are common to OMX video encoders, see related documents [1] and [2].

## 1.2. H.264 Encoder Media Component Overview and Scope

Figure 1-1 illustrates the software stacks for the H.264 Encoder Media Component and shows the scope of this document. OMX Media Component H.264 Encoder Library is a library that provides H.264 Video encoding functions. It requires OMX Media Component Video Encoder Common Library and OMX Media Component Common Library.

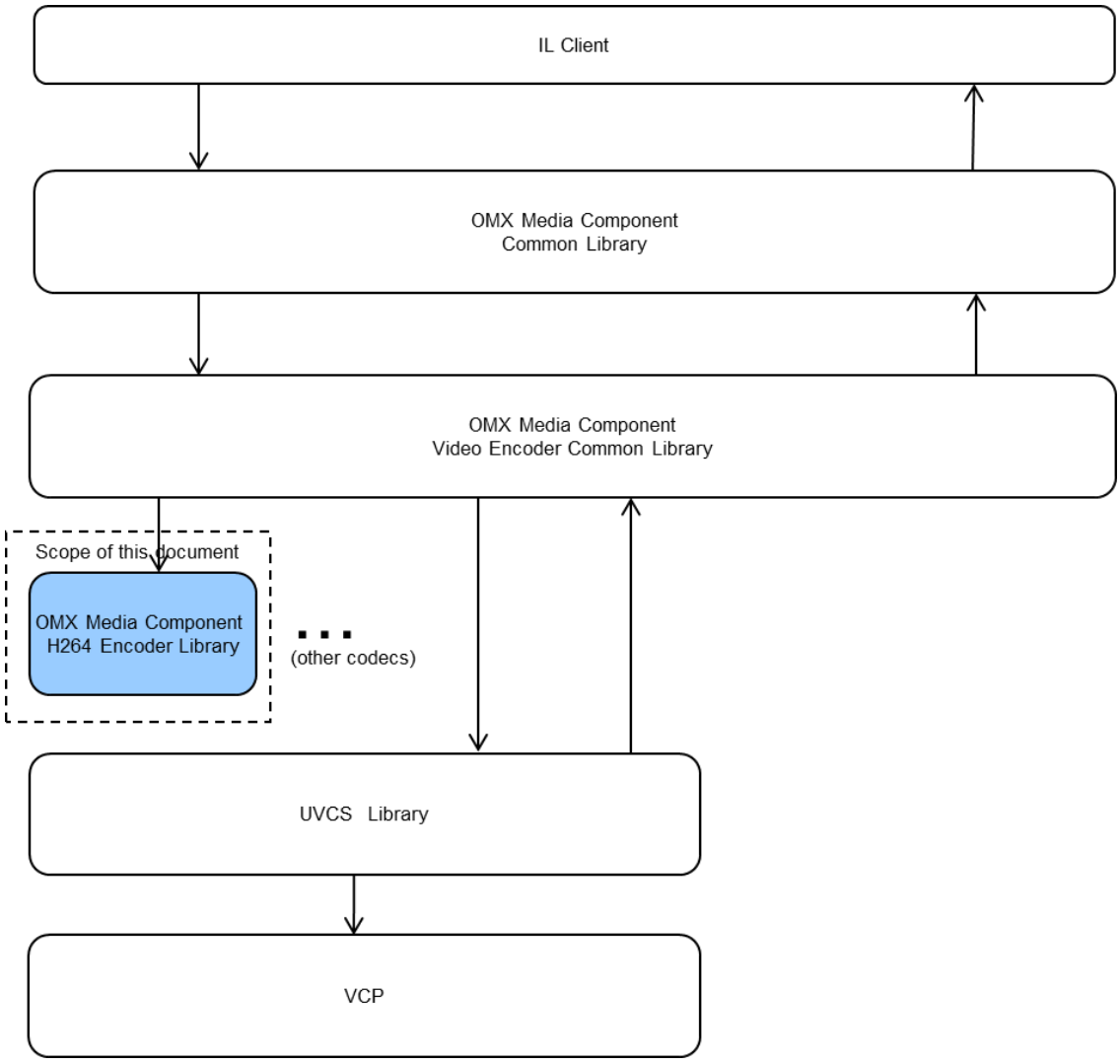


**Figure 1-1 Software Stacks and Scope**

This document describes the specifications of OMX Media Component H.264 Encoder library part. For the specifications of OMX Media Component Video Encoder Common Library and OMX Media Component Common Library, see related documents [1] and [2] respectively.

### 1.3. Required Header Files

Table 1-1 lists the header files that are required to use the OMX extended indexes and structures that are described in this document. Regarding the other header files, see related documents [1] and [2].

**Table 1-1 Required Header Files**

| File name | Remarks |
|---|---|
| OMXR_Extension_h264.h | - |
| OMXR_Extension_h264e.h | - |

### 1.4. Role Name and Component Name

Table 1-2 shows the role name and the component name for H.264 Encoder Media Component.

**Table 1-2 Role Name and Component Name**

| Role name | Component name |
|---|---|
| video_encoder.avc | OMX.RENESAS.VIDEO.ENCODER.H264 |

### 1.5. Related Documents

Table 1-3 lists the related documents.

**Table 1-3 List of Related Documents**

| No. | Document Name | Remarks |
|---|---|---|
| [1] | OMX Media Component User's Manual Common Part | The common specifications for OMX Media Component |
| [2] | OMX Media Component User's Manual Video Encoder Common Part | The common specifications for OMX Video Encoder Media Component |
| [3] | OpenMAX Integration Layer Application Programming Interface Specification Version 1.1.2, September 1, 2008 | http://www.khronos.org/registry/omxil/specs/OpenMAX_IL_1_1_2_Specification.pdf |

## 1.6. Terminology

Table 1-4 lists the terms that are used in this document.

**Table 1-4 Terminology**

| Term | Abbreviation | Description |
|---|---|---|
| Video Port Base | VPB | The base value of the port index of the Video Media Component. The port index values of the input and output ports are obtained by adding offset values to this base value. |
| UVCS | - | Renesas proprietary video codec software module that provides multi-processing function for video decoding and encoding. OMX Video Codec products contain UVCS library. |
| VCP | - | The abbreviation for Video Coding Processor. The VCP is the Renesas Hardware IP and provides video decoding and encoding functions for compressed video stream. |

# 2. Functions

H.264 Encoder Media Component is a media component which provides functions to encode raw video to compressed stream according to H.264 codec standard. H.264 Encoder Media Component receives raw video data on the input port and emits the encoded video data on the output ports.
For the specifications that are common to OMX video encoders, see related document [2].

## 2.1. Function Details

### 2.1.1. Encode Functions

Table 2-1 shows the codec standard and functions that H.264 Encoder Media Component supports.

**Table 2-1 Supported Codec Standard and Functions**

| | |
|---|---|
| Codec standard | ITU-T Rec. H.264 ¦ ISO/IEC 14496 Part-10 (H.264/MPEG-4 AVC) |
| Profile | Baseline Profile / Constrained Baseline Profile Note1 / Main Profile / High Profile / Progressive High Profile Note2 / Constrained High Profile Note2 |
| Level | 1 / 1b / 1.1 / 1.2 / 1.3 / 2 / 2.1 / 2.2 / 3 / 3.1 / 3.2 / 4 / 4.1 / 4.2 |
| Unsupported tools | - ASO ( Arbitrary Slice Order )<br>- FMO ( Flexible Macroblock Ordering )<br>- RS ( Redundant Slice )<br>- PCAFF ( Picture-Adaptive Frame-Field Coding)<br>- MBAFF ( Macroblock-Adaptive Frame-Field Coding )<br>- Weighted Prediction |
| Picture size | <Progressive><br>- Width : 80 - 1920 (multiple of 8)<br>- Height : 80 - 1080 (multiple of 8)<br><Interlace><br>- Width : 80 - 1920 (multiple of 8)<br>- Height : 80 - 1080 (multiple of 8) |
| Bit rate | Maximum 40Mbits/s Note3 |
| Frame rate | Maximum 60p / 60i Note3 |
| Input format | YUV420 Semi-Planar format<br>YUV420 Planar format |
| Output format | H.264 Elementary Stream (Annex B Byte stream format) |

Note1: H.264 Encoder Media Component includes Constrained Baseline Profile in Baseline Profile.

Note2: H.264 Encoder Media Component includes Progressive High Profile and Constrained High Profile in High Profile.

Note3: Regarding the throughput, the following description should be noticed:
- The maximum throughput is different for each LSI. For the detail, see the LSI hardware manual.
- The throughput may fall depends on CPU load and bus traffic caused by the modules except OMX Media Component.

# 3. I/O Data Format

## 3.1. Buffer Payload

### 3.1.1. Input Buffer Payload

See related document [2].

### 3.1.2. Output Buffer Payload

H.264 Encoder Media Component outputs buffer as follows:
- One encoded frame data is stored into a single output buffer and not divided into multiple buffers.
- Multiple frame data are not stored into the same output buffer.
- In the case of the interlace encoding, a pair of field data is stored in a single buffer.
- In the case of the multiple slices, all NAL units are stored in a single buffer.
- The top buffer contains only SPS NAL and PPS NAL and does not contain Data NAL.
- The last buffer contains only End Of Sequence NAL and the End Of Stream NAL and does not contain Data NALs.

Figure 3-1 illustrates the output buffer sequence.

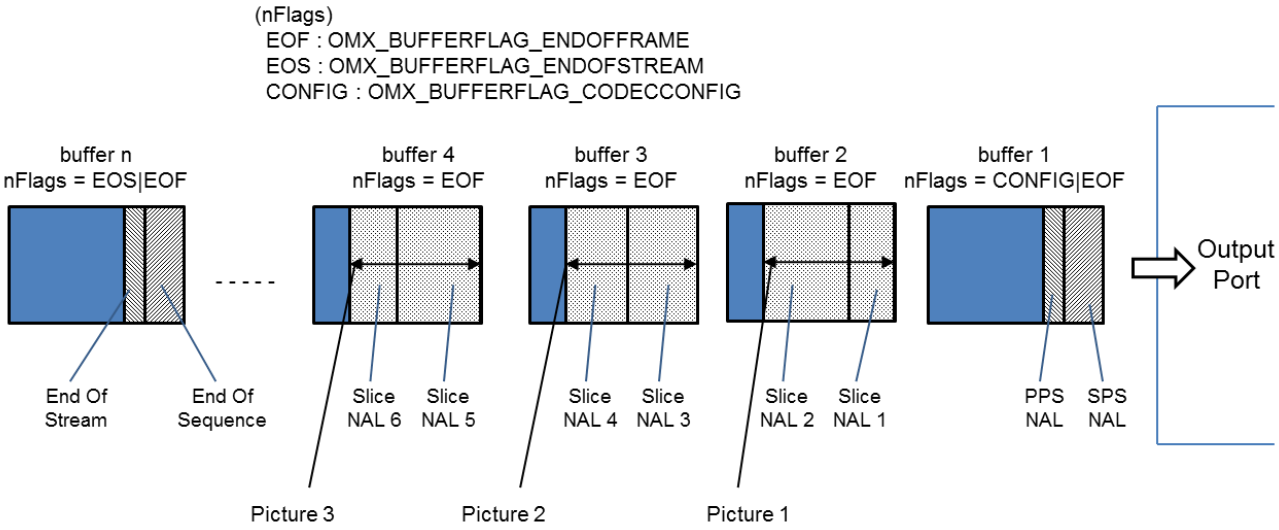


**Figure 3-1 Example of Output Buffer Sequence**

Figure 3-2 illustrates the output buffer sequence when the *bSequencHeaderWithIDR* member of the OMXR_MC_VIDEO_PARAM_SEQUENCE_HEADER_WITH_IDRTYPE structure (see section 6.4) is set to OMX_TRUE. In this case, SPS NAL and PPS NAL are prepended to every IDR frame except the top IDR frame.

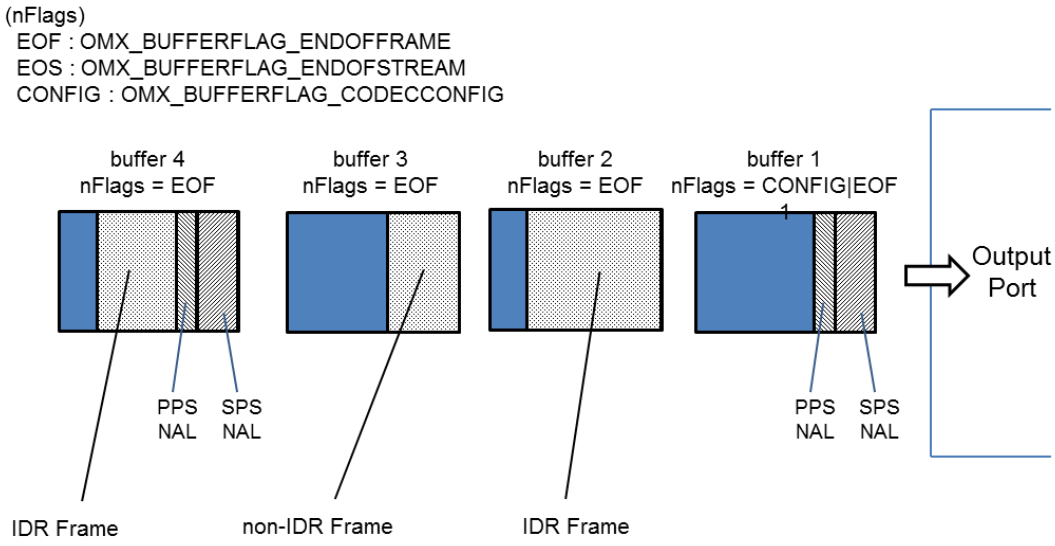


**Figure 3-2 Example of Output Buffer Sequence (bSequencHeaderWithIDR=OMX_TRUE)**

## 3.2. Input Picture Data Format

See related document [2].

### 3.3. Output Stream Data Format

The output stream data format of H.264 Encoder Media Component is H.264 Annex B Byte stream format. Figure 3-3 illustrates the output stream data format.

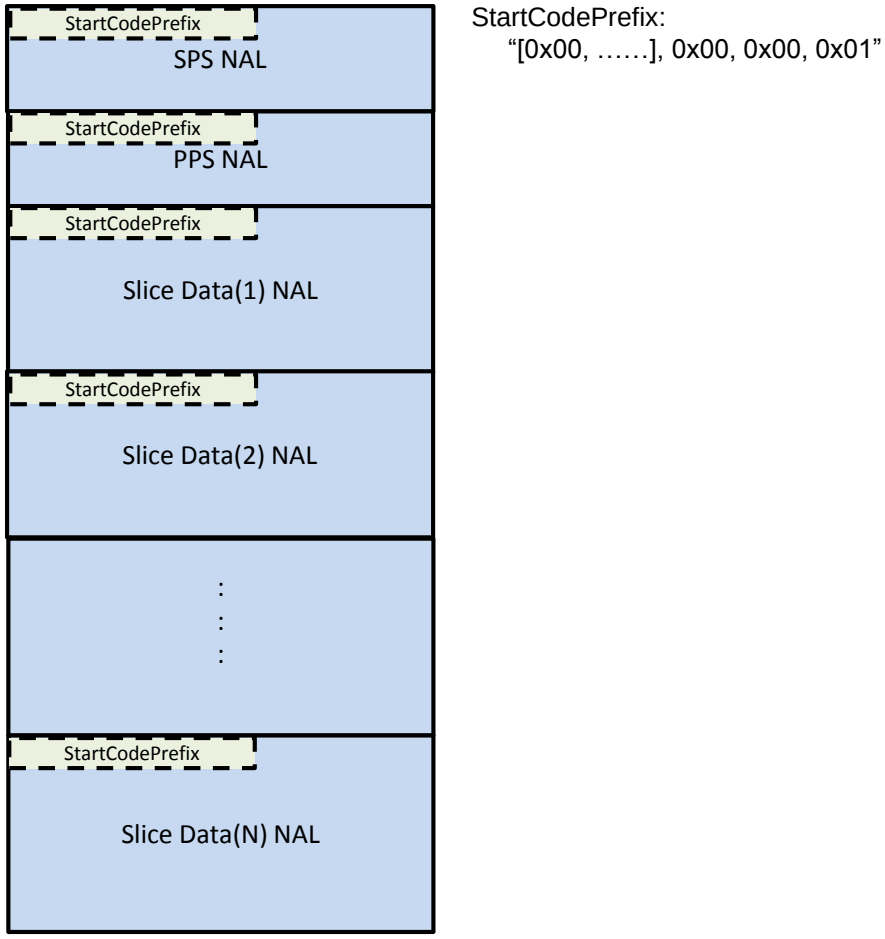


**Figure 3-3 Output Stream Data Format**

# 4. API Reference

See related document [2].

# 5. Indexes

## 5.1. Standard Indexes of H.264 Encoder Media Component

Table 5-1 lists the OpenMAX IL standard indexes that are available for H.264 Encoder Media Component.

**Table 5-1 Available Standard Indexes for H.264 Encoder Media Component**

| Index | Description |
|---|---|
| OMX_IndexParamPortDefinition | See related document [2] |
| OMX_IndexParamVideoPortFormat | |
| OMX_IndexParamVideoBitrate | |
| OMX_IndexParamVideoProfileLevelQuerySupported | |
| OMX_IndexParamVideoProfileLevelCurrent | |
| OMX_IndexConfigVideoBitrate | |
| OMX_IndexConfigVideoFramerate | |
| OMX_IndexConfigVideoIntraVOPRefresh | |
| OMX_IndexParamVideoAvc | See section 5.1.1 |
| OMX_IndexParamVideoQuantization | See section 5.1.2 |
| OMX_IndexConfigVideoAVCIntraPeriod | See section 5.1.3 |

### 5.1.1. OMX_IndexParamVideoAvc

[Description] The index to access H.264 Video codec related parameters.

[Corresponding Structure] OMX_VIDEO_PARAM_AVCTYPE structure

[Notes] None

### 5.1.2. OMX_IndexParamVideoQuantization

[Description] The index to adjust the initial quantization parameters

[Corresponding Structure] OMX_VIDEO_PARAM_QUANTIZATIONTYPE structure

[Notes] None

### 5.1.3. OMX_IndexConfigVideoAVCIntraPeriod

[Description] The index to enable and configure the IDR and Intra periodicity.

[Corresponding Structure] OMX_VIDEO_CONFIG_AVCINTRAPERIOD structure

[Notes] None

## 5.2. Extended Indexes of H.264 Encoder Media Component

Table 5-2 lists the OMX extended indexes that are available for H.264 Encoder Media Component.

**Table 5-2 Available extended indexes for H.264 Encoder Media Component**

| Index | Description |
|---|---|
| OMXR_MC_IndexParamVideoPictureMemoryAlloc | See related document [2] |
| OMXR_MC_IndexParamVideoSequenceHeaderWithIDR | See section 5.2.1 |
| OMXR_MC_IndexParamVideoAVCSyntaxOption | See section 5.2.2 |

### 5.2.1. OMXR_MC_IndexParamVideoSequenceHeaderWithIDR

[Description] The index to enable H.264 Encoder Media Component to prepend SPS and PPS to every IDR frame.

[String] “OMX.RENESAS.INDEX.PARAM.VIDEO.SEQUENCE.HEADER.WITH.IDR”

[Corresponding Structure] OMXR_MC_VIDEO_PARAM_SEQUENCE_HEADER_WITH_IDRTYPE structure

[Notes] None

### 5.2.2. OMXR_MC_IndexParamVideoAVCSyntaxOption

[Description] The index to enable H.264 Encoder Media Component to prepend option syntax.

[String] “OMX.RENESAS.INDEX.PARAM.VIDEO.AVC.SYNTAX.OPTION”

[Corresponding Structure] OMXR_MC_VIDEO_PARAM_AVC_SYNTAX_OPTION structure

[Notes] None

## 5.3. Valid Indexes for OpenMAX IL Macro Functions

Table 5-3 shows which index is available for each port and which OpenMAX IL Macro function can be called to access the index.

**Table 5-3 Valid Indexes and OpenMAX IL Macro Function**

| PortIndex | Index | Get/Set Parameter | | Get/SetConfig | |
|---|---|---|---|---|---|
| | | Get | Set | Get | Set |
| VPB+0 | OMX_IndexParamPortDefinition | See related document [2] | | | |
| | OMX_IndexParamVideoPortFormat | | | | |
| | OMX_IndexConfigVideoFramerate | | | | |
| | OMXR_MC_IndexParamVideoPictureMemoryAlloc | | | | |
| VPB+1 | OMX_IndexParamPortDefinition | See related document [2] | | | |
| | OMX_IndexParamVideoPortFormat | | | | |
| | OMX_IndexParamVideoBitrate | | | | |
| | OMX_IndexParamVideoProfileLevelQuerySupported | | | | |
| | OMX_IndexParamVideoProfileLevelCurrent | | | | |
| | OMX_IndexConfigVideoBitrate | | | | |
| | OMX_IndexConfigVideoFramerate | | | | |
| | OMX_IndexConfigVideoIntraVOPRefresh | | | | |
| | OMX_IndexParamVideoAvc | X | X | - | - |
| | OMX_IndexParamVideoQuantization | X | X | - | - |
| | OMX_IndexConfigVideoAVCIntraPeriod | - | - | X | X |
| | OMXR_MC_IndexParamVideoSequenceHeaderWithIDR | X | X | - | - |
| | OMXR_MC_IndexParamVideoAVCSyntaxOption | X | X | - | - |

X : Valid
- : Invalid

# 6. Structures

Table 6-1 lists H.264 Encoder Media Component specific structures.

**Table 6-1 H.264 Encoder Media Component Specific Structures**

| Structure Name | Description |
|---|---|
| OMX_VIDEO_PARAM_AVCTYPE | See section 6.1 |
| OMX_VIDEO_PARAM_QUANTIZATIONTYPE | See section 6.2 |
| OMX_VIDEO_CONFIG_AVCINTRAPERIOD | See section 6.3 |
| OMXR_MC_VIDEO_PARAM_SEQUENCE_HEADER_WITH_IDRTYPE | See section 6.4 |
| OMXR_MC_VIDEO_PARAM_AVC_SYNTAX_OPTION | See section 6.5 |

Table 6-2 shows the notation for the access attribute of a structure member described in this section.

**Table 6-2 Notation for the access attribute of a structure member**

| Member Name | Get | Set |
|---|---|---|
| Indicates the member name | Indicates the access attribute of the member in the OMX_GetParameter() or OMX_GetConfig().<br><br>“R” means IL client can get a value from the member.<br><br>“W” means IL client must specify a value for the member. | Indicates the access attribute of the member in the OMX_SetParameter() or OMX_SetConfig().<br><br>“W” means IL client must/can specify a value for the member.<br><br>“-“ means a specified value is ignored and not reflected. |

## 6.1. OMX_VIDEO_PARAM_AVCTYPE

[Definition] See related document [3] section 4.3.18.

[Index] OMX_IndexParamVideoAvc

[Member]

| Member Name | Get | Set |
|---|---|---|
| *nSize* | W | W |
| *nVersion* | W | W |
| *nPortIndex* | W | W |
| *nSliceHeaderSpacing* | R | W |
| *nPFrames* | R | W |
| *nBFrames* | R | W |
| *bUseHadamard* | R | - |
| *nRefFrames* | R | W |
| *nRefIdx10ActiveMinus1* | R | - |
| *nRefIdx11ActiveMinus1* | R | - |
| *bEnableUEP* | R | - |
| *bEnableFMO* | R | - |
| *bEnableASO* | R | - |
| *bEnableRS* | R | - |
| *eProfile* | R | W |
| *eLevel* | R | W |
| *nAllowedPictureTypes* | R | - |
| *bFrameMBsOnly* | R | - |
| *bMBAFF* | R | - |
| *bEntropyCodingCABAC* | R | - |
| *bWeightedPPrediction* | R | - |
| *nWeightedBipredicitonMode* | R | - |
| *bconstIpred* | R | - |
| *bDirect8x8Inference* | R | - |
| *bDirectSpatialTemporal* | R | - |
| *nCabacInitIdc* | R | - |
| *eLoopFilterMode* | R | W |

[Details]

***nSize***

| | |
|---|---|
| **Write Value** | The size of the structure in bytes. |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nVersion***

| | |
|---|---|
| **Write Value** | The version number of OpenMAX IL specifications 1.1.2 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nPortIndex***

| | |
|---|---|
| **Write Value** | VPB + 1 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nSliceHeaderSpacing***

| | |
|---|---|
| **Write Value** | 0, 5 – 8160 |
| **Read Value** | (Current setting) |
| **Initial Value** | 0 |
| **Notes** | − If this member is 0, the slice partitioning is disabled.<br>− If this member is set to non-zero value, the value is rounded up to the nearest multiple of the number of the horizontal macroblocks.<br>− If the specified value exceeds the number of the supported total macroblocks, the value automatically calculated as following: nSliceHeaderSpacing = horizontal macroblocks x vertical macroblocks |

***nPFrames***

| | |
|---|---|
| **Write Value** | 0 – 119 |
| **Read Value** | (Current setting) |
| **Initial Value** | 29 |
| **Notes** | − If 0 is specified, all the frames are encoded with IDR frame.<br>− The value larger than the maximum is rounded down to the maximum.<br>− The setting of the *nPFrames* and the *nBFrames* member must fulfill the following condition:<br>1 + nPFrames + (nPFrames * nBFrames) <= 120<br>If not fulfilled, the value of the *nPFrames* member is automatically reduced to fulfill the condition.<br>− The available setting depends on the other parameters. For the combination of the available settings, see Table 6-3. |

***nBFrames***

| | |
|---|---|
| **Write Value** | 0 – 3 |
| **Read Value** | (Current setting) |
| **Initial Value** | 0 |
| **Notes** | − The value larger than the maximum is rounded down to the maximum.<br>− The setting of the nPFrames and the nBFrames member must fulfill the following condition:<br>1 + nPFrames + (nPFrames * nBFrames) <= 120<br>If not fulfilled, the value of the nPFrames member is automatically reduced to fulfill the condition.<br>− The available setting depends on the other parameters. For the combination of the available settings, see Table 6-3.<br>− The value of the nBFrames must be less than or equal to *nBufferCountActual* - 2, which is member of the OMX_VIDEO_PORTDEFINITIONTYPE structure for the output port. Otherwise the encode operation results in an error. |

***bUseHadamard***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_TRUE |
| **Initial Value** | OMX_TURE |
| **Notes** | This member is not supported. |

***nRefFrames***

| | |
|---|---|
| **Write Value** | 0 – 2 |
| **Read Value** | (Current setting) |
| **Initial Value** | 1 |
| **Notes** | − The value larger than the maximum is rounded down to the maximum.<br>− The available setting depends on the other parameters. For the combination of the available settings, see Table 6-3. |

***nRefIdx10ActiveMinus1***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | 0 – 1 |
| **Initial Value** | 0 |
| **Notes** | - |

***nRefIdx11ActiveMinus1***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | 0 – 1 |
| **Initial Value** | 0 |
| **Notes** | - |

***bEnableUEP***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE |
| **Initial Value** | OMX_FALSE |
| **Notes** | This member is not supported. |

***bEnableFMO***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE |
| **Initial Value** | OMX_FALSE |
| **Notes** | This member is not supported. |

***bEnableASO***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE |
| **Initial Value** | OMX_FALSE |
| **Notes** | This member is not supported. |

***bEnableRS***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE |
| **Initial Value** | OMX_FALSE |
| **Notes** | This member is not supported. |

***eProfile***

| | |
|---|---|
| **Write Value** | OMX_VIDEO_AVCProfileBaseline<br>OMX_VIDEO_AVCProfileMain<br>OMX_VIDEO_AVCProfileHigh |
| **Read Value** | (Current setting) |
| **Initial Value** | OMX_VIDEO_AVCProfileBaseline |
| **Notes** | − The available setting depends on the other parameters. For the combination of the available settings, see Table 6-3. |

***eLevel***

| | |
|---|---|
| **Write Value** | OMX_VIDEO_AVCLevel1<br>OMX_VIDEO_AVCLevel1b<br>OMX_VIDEO_AVCLevel11<br>OMX_VIDEO_AVCLevel12<br>OMX_VIDEO_AVCLevel13<br>OMX_VIDEO_AVCLevel2<br>OMX_VIDEO_AVCLevel21<br>OMX_VIDEO_AVCLevel22<br>OMX_VIDEO_AVCLevel3<br>OMX_VIDEO_AVCLevel31<br>OMX_VIDEO_AVCLevel32<br>OMX_VIDEO_AVCLevel4<br>OMX_VIDEO_AVCLevel41<br>OMX_VIDEO_AVCLevel42 |
| **Read Value** | (Current setting) |
| **Initial Value** | OMX_VIDEO_AVCLevel1 |
| **Notes** | - |

***nAllowedPictureTypes***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_VIDEO_PictureTypeI \| OMX_VIDEO_PictureTypeP<br>or<br>OMX_VIDEO_PictureTypeI \| OMX_VIDEO_PictureTypeP \|<br>OMX_VIDEO_PictureTypeB |
| **Initial Value** | OMX_VIDEO_PictureTypeI \| OMX_VIDEO_PictureTypeP |
| **Notes** | Read value depends on the value of *eProfile* member. |

***bFrameMBsOnly***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE<br>OMX_TRUE |
| **Initial Value** | OMX_TRUE |
| **Notes** | − It is set to OMX_TRUE if the progressive encode mode is selected. Otherwise it is set to OMX_FALSE. For the detail of the encode mode, see related document [2]. |

***bMBAFF***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE |
| **Initial Value** | OMX_FALSE |
| **Notes** | This member is not supported. |

***bEntropyCodingCABAC***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE<br>OMX_TRUE |
| **Initial Value** | OMX_FALSE |
| **Notes** | - |

***bWeightedPPrediction***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE |
| **Initial Value** | OMX_FALSE |
| **Notes** | This member is not supported. |

***nWeightedBipredicitonMode***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | 0 |
| **Initial Value** | 0 |
| **Notes** | This member is not supported. |

***bconstIpred***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE |
| **Initial Value** | OMX_FALSE |
| **Notes** | - |

***bDirect8x8Inference***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_TRUE |
| **Initial Value** | OMX_TRUE |
| **Notes** | - |

***bDirectSpatialTemporal***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_TRUE |
| **Initial Value** | OMX_TRUE |
| **Notes** | – H.264 Encoder Media Component operates with the spatial direct mode. |

***nCabacInitIdc***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | 0 |
| **Initial Value** | 0 |
| **Notes** | - |

***eLoopFilterMode***

| | |
|---|---|
| **Write Value** | OMX_VIDEO_AVCLoopFilterEnable<br>OMX_VIDEO_AVCLoopFilterDisable<br>OMX_VIDEO_AVCLoopFilterDisableSliceBoundary |
| **Read Value** | (Current setting) |
| **Initial Value** | OMX_VIDEO_AVCLoopFilterEnable |
| **Notes** | - |

**Table 6-3 Available Combination of the Encoding Parameters**

| Frame/Field | *nPFrames* | *nBFrame* | *nRefFrame* | *eProfile* |
|---|---|---|---|---|
| Frame Encoding (Progressive) | 0 | 0 | 0 | Baseline / Main / High |
| | Except 0 | 0 | 1 - 2 | Baseline / Main / High |
| | | 1 – 3 | 2 | Main / High |
| Field Encoding (Interlace) | 0 | 0 | 0 - 1 | Main / High |
| | Except 0 | 0 – 3 | 2 | Main / High |

Note:
For the detail of the frame encoding and the field encoding, see related document [2].
If the combination of the specified parameters is not allowed in Table 6-3, the parameters are automatically adjusted within the range of the members when the encoding starts. The following is the order of priority and parameter shall be adjusted in the order of ascending priorities:

*eProfile* < *nRefFrame* < *nBFrame* < *nPFrame*

## 6.2. OMX_VIDEO_PARAM_QUANTIZATIONTYPE

[Definition] See related document [3] section 4.3.6.

[Index] OMX_IndexParamVideoQuantization

[Member]

| Member Name | Get | Set |
|---|---|---|
| *nSize* | W | W |
| *nVersion* | W | W |
| *nPortIndex* | W | W |
| *nQpI* | R | W |
| *nQpP* | R | W |
| *nQpB* | R | W |

[Notes] According to OpenMAX IL standard, it is defined that this structure is not applicable to variable bit rate encoding or constant rate encoding. However, Qp value in this implementation is used only on the initial value of quantization. Therefore, unlike OpenMAX standard, it is available by variable bit rate encoding or constant rate encoding.

[Details]

***nSize***

| | |
|---|---|
| **Write Value** | The size of the structure in bytes. |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nVersion***

| | |
|---|---|
| **Write Value** | The version number of OpenMAX IL specifications 1.1.2 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nPortIndex***

| | |
|---|---|
| **Write Value** | VPB + 1 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nQpI***

| | |
|---|---|
| **Write Value** | 0 - 51, 255 |
| **Read Value** | (Current setting) |
| **Initial Value** | 255 |
| **Notes** | − The *nQpI* is the initial value of the quantization parameter for I frame.<br>− The value 255 denotes the ‘Auto Setting’ that the initial value is automatically determined. Auto setting is recommended.<br>− If the specified value is out of range value, Auto setting is selected. |

***nQpP***

| | |
|---|---|
| **Write Value** | 0 - 51, 255 |
| **Read Value** | (Current setting) |
| **Initial Value** | 255 |
| **Notes** | − The *nQpP* is the initial value of the quantization parameter for P |

| | |
|---|---|
| | frame.<br>− The value 255 denotes the ‘Auto Setting’ that the initial value is automatically determined. Auto setting is recommended.<br>− If the specified value is out of range value, Auto setting is selected. |

***nQpB***

| | |
|---|---|
| **Write Value** | 0 - 51, 255 |
| **Read Value** | (Current setting) |
| **Initial Value** | 255 |
| **Notes** | − The *nQpB* is the initial value of the quantization parameter for B frame.<br>− The value 255 denotes the ‘Auto Setting’ that the initial value is automatically determined. Auto setting is recommended.<br>− If the specified value is out of range value, Auto setting is selected. |

### 6.3. OMX_VIDEO_CONFIG_AVCINTRAPERIOD

[Definition] See related document [3] section 4.3.27.

[Index] OMX_IndexConfigAvcIntraPeriod

[Member]

| Member Name | Get | Set |
|---|---|---|
| *nSize* | W | W |
| *nVersion* | W | W |
| *nPortIndex* | W | W |
| *nIDRPeriod* | R | W |
| *nPFrames* | R | W |

[Notes] If the OMX_CONFIG_AVCINTRAPERIOD structure is updated during OMX_StateExecuting state or OMX_StatePause state via OMX_SetConfig function, the updated setting is not applied until the H.264 Encoder Media Component transits to OMX_StateIdle state.

[Details]

***nSize***

| | |
|---|---|
| **Write Value** | The size of the structure in bytes. |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nVersion***

| | |
|---|---|
| **Write Value** | The version number of OpenMAX IL specifications 1.1.2 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nPortIndex***

| | |
|---|---|
| **Write Value** | VPB + 1 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nIDRPeriod***

| | |
|---|---|
| **Write Value** | 0 – 120 |
| **Read Value** | (Current setting) |
| **Initial Value** | 1 |
| **Notes** | − If the member is set to 0, only the top frame of the stream is encoded with IDR frame.<br>− The value larger than the maximum is rounded down to the maximum.<br>− If the *nPFrames* member is set to 0, this member is updated to 1 when the encoding starts. |

***nPFrames***

| | |
|---|---|
| **Write Value** | 0 – 119 |
| **Read Value** | (Current setting) |
| **Initial Value** | 29 |
| **Notes** | − This parameter is the same as the *nPFrames* member of the OMX_VIDEO_PARAM_AVCTYPE structure. See 6.1. |

## 6.4. OMXR_MC_VIDEO_PARAM_SEQUENCE_HEADER_WITH_IDRTYPE

[Definition]
```
typedef struct tagOMXR_MC_VIDEO_PARAM_SEQUENCE_HEADER_WITH_IDRTYPE
{
    OMX_U32          nSize;
    OMX_VERSIONTYPE  nVersion;
    OMX_U32          nPortIndex;
    OMX_BOOL         bSequenceHeaderWithIDR;
} OMXR_MC_VIDEO_PARAM_SEQUENCE_HEADER_WITH_IDRTYPE;
```

[Index] OMXR_MC_ IndexParamVideoSequenceHeaderWithIDR

[Member]

| Member Name | Get | Set |
|---|---|---|
| *nSize* | W | W |
| *nVersion* | W | W |
| *nPortIndex* | W | W |
| *bSequenceHeaderWithIDR* | R | W |

[Notes] The index to enable H.264 Encoder Media Component to prepend SPS and PPS to every IDR frame.

[Details]

***nSize***

| | |
|---|---|
| **Write Value** | The size of the structure in bytes. |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nVersion***

| | |
|---|---|
| **Write Value** | The version number of OpenMAX IL specifications 1.1.2 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nPortIndex***

| | |
|---|---|
| **Write Value** | VPB + 1 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***bSequenceHeaderWithIDR***

| | |
|---|---|
| **Write Value** | OMX_TRUE<br>OMX_FALSE |
| **Read Value** | (Current setting) |
| **Initial Value** | OMX_FALSE |
| **Notes** | – If set to OMX_TRUE, SPS and PPS are prepended to IDR frames. For the details of the output buffer sequence, see Figure 3-2. |

## 6.5. OMXR_MC_VIDEO_PARAM_AVC_SYNTAX_OPTION

[Definition]
```
typedef struct tagOMXR_MC_VIDEO_PARAM_AVC_SYNTAX_OPTION {
    OMX_U32         nSize;
    OMX_VERSIONTYPE nVersion;
    OMX_U32         nPortIndex;
    OMX_BOOL        bAUDEnable;
    OMX_BOOL        bVUIEnable;
    OMX_U32         u32PPSNum;
} OMXR_MC_VIDEO_PARAM_AVC_SYNTAX_OPTION;
```

[Index] OMXR_MC_IndexParamVideoAVCSyntaxOption

[Member]

| Member Name | Get | Set |
|---|---|---|
| *nSize* | W | W |
| *nVersion* | W | W |
| *nPortIndex* | W | W |
| *bAUDEnable* | R | W |
| *bVUIEnable* | R | W |
| *u32PPSNum* | R | W |

[Notes] The index to enable H.264 Encoder Media Component to prepend option syntax.

[Details]

***nSize***

| | |
|---|---|
| **Write Value** | The size of the structure in bytes. |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nVersion***

| | |
|---|---|
| **Write Value** | The version number of OpenMAX IL specifications 1.1.2 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***nPortIndex***

| | |
|---|---|
| **Write Value** | VPB + 1 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***bAUDEnable***

| | |
|---|---|
| **Write Value** | OMX_TRUE<br>OMX_FALSE |
| **Read Value** | (Current setting) |
| **Initial Value** | OMX_FALSE |
| **Notes** | − If set to OMX_TRUE, Access unit delimiter is prepended to frame. |

***bVUIEnable***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_FALSE |
| **Initial Value** | OMX_FALSE |
| **Notes** | − This member is not supported. |

***u32PPSNum***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | 1 |
| **Initial Value** | 1 |
| **Notes** | − This member is not supported. |

## 6.6. Specific Usage on Common Structure Members

This section describes H.264 Encoder Media Component specific usage of the structures that are described in related document [2].

### 6.6.1. OMX_VIDEO_PORTDEFINITIONTYPE (Input Port)

[Index] OMX_IndexParamPortDefinition

[Details]

***xFramerate***

| | |
|---|---|
| **Write Value** | See Table 6-4 |
| **Read Value** | (Current setting) |
| **Initial Value** | 0x000F0000 |
| **Notes** | - |

### 6.6.2. OMX_VIDEO_PORTDEFINITIONTYPE (Output Port)

[Index] OMX_IndexParamPortDefinition

[Details]

***xBitrate***

| | |
|---|---|
| **Write Value** | See Table 6-5 |
| **Read Value** | (Current setting) |
| **Initial Value** | 64000 |
| **Notes** | - |

***nFramerate***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | See Table 6-4 |
| **Initial Value** | 0x000F0000 |
| **Notes** | - |

***eCompressionFormat***

| | |
|---|---|
| **Write Value** | OMX_VIDEO_CodingAVC |
| **Read Value** | (Current setting) |
| **Initial Value** | OMX_VIDEO_CodingAVC |
| **Notes** | - |

### 6.6.3. OMX_VIDEO_PARAM_PORTFORMATTYPE (Input Port)

[Index] OMX_IndexParamVideoPortFormat

[Details]

***xFramerate***

| | |
|---|---|
| **Write Value** | See Table 6-4 |
| **Read Value** | 0x003C0000 (nIndex = any value) |
| **Initial Value** | |
| **Notes** | - |

### 6.6.4. OMX_VIDEO_PARAM_PORTFORMATTYPE (Output Port)

[Index] OMX_IndexParamVideoPortFormat

[Details]

***nIndex***

| | |
|---|---|
| **Write Value** | 0 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

***eCompressionFormat***

| | |
|---|---|
| **Write Value** | OMX_VIDEO_CodingAVC |
| **Read Value** | OMX_VIDEO_CodingAVC (nIndex = 0) |
| **Initial Value** | - |
| **Notes** | - |

***xFramerate***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | 0x003C0000 (nIndex = 0) |
| **Initial Value** | - |
| **Notes** | - |

### 6.6.5. OMX_VIDEO_PARAM_BITRATETYPE

[Index] OMX_IndexParamVideoBitrate

[Details]

***eControlRate***

| | |
|---|---|
| **Write Value** | OMX_Video_ControlRateConstant<br>OMX_Video_ControlRateVariable<br>OMX_Video_ControlRateConstantSkipFrames |
| **Read Value** | (Current setting) |
| **Initial Value** | OMX_Video_ControlRateConstant |
| **Notes** | - |

***nTargetBitrate***

| | |
|---|---|
| **Write Value** | See Table 6-5 |
| **Read Value** | (Current setting) |
| **Initial Value** | 64000 |
| **Notes** | - |

### 6.6.6. OMX_VIDEO_PARAM_PROFILELEVELTYPE (ProfileLevelQuerySupport)

[Index] OMX_IndexParamVideoProfileLevelQuerySupported

[Details]

***eProfile***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_VIDEO_AVCProfileBaseline (nProfileIndex=0)<br>OMX_VIDEO_AVCProfileMain (nProfileIndex=1)<br>OMX_VIDEO_AVCProfileHigh (nProfileIndex=2) |
| **Initial Value** | OMX_VIDEO_AVCProfileBaseline |
| **Notes** | - |

***eLevel***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | OMX_VIDEO_AVCLevel42 |
| **Initial Value** | OMX_VIDEO_AVCLevel42 |
| **Notes** | - |

***nProfileIndex***

| | |
|---|---|
| **Write Value** | 0, 1, 2 |
| **Read Value** | - |
| **Initial Value** | - |
| **Notes** | - |

### 6.6.7. OMX_VIDEO_PARAM_PROFILELEVELTYPE (ProfileLevelCurrent)

[Index] OMX_IndexParamVideoProfileLevelCurrent

[Details]

***eProfile***

| | |
|---|---|
| **Write Value** | OMX_VIDEO_AVCProfileBaseline,<br>OMX_VIDEO_AVCProfileMain,<br>OMX_VIDEO_AVCProfileHigh |
| **Read Value** | (Current setting) |
| **Initial Value** | OMX_VIDEO_AVCProfileBaseline |
| **Notes** | − This member is the same as the *eProfile* member of OMX_VIDEO_PARAM_AVCTYPE structure. For details, see 6.1. |

***eLevel***

| | |
|---|---|
| **Write Value** | OMX_VIDEO_AVCLevel1,<br>OMX_VIDEO_AVCLevel1b,<br>OMX_VIDEO_AVCLevel11,<br>OMX_VIDEO_AVCLevel12,<br>OMX_VIDEO_AVCLevel13,<br>OMX_VIDEO_AVCLeve2,<br>OMX_VIDEO_AVCLevel21,<br>OMX_VIDEO_AVCLevel22,<br>OMX_VIDEO_AVCLevel3,<br>OMX_VIDEO_AVCLevel31,<br>OMX_VIDEO_AVCLevel32,<br>OMX_VIDEO_AVCLevel4,<br>OMX_VIDEO_AVCLevel41,<br>OMX_VIDEO_AVCLevel42 |
| **Read Value** | (Current setting) |
| **Initial Value** | OMX_VIDEO_AVCLevel1 |
| **Notes** | − This member is the same as the *eLevel* member of OMX_VIDEO_PARAM_AVCTYPE structure.For details, see 6.1. |

***nProfileIndex***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | 0 |
| **Initial Value** | 0 |
| **Notes** | - |

### 6.6.8. OMX_VIDEO_CONFIG_BITRATETYPE

[Index] OMX_IndexConfigVideoBitrate

[Details]

***eEncodeBitrate***

| | |
|---|---|
| **Write Value** | See Table 6-5. |
| **Read Value** | (Current setting) |
| **Initial Value** | 64000 |
| **Notes** | - |

### 6.6.9. OMX_VIDEO_CONFIG_FRAMERATETYPE (Input Port)

[Index] OMX_IndexConfigVideoFramerate

[Details]

***xEncodeFramerate***

| | |
|---|---|
| **Write Value** | See Table 6-4. |
| **Read Value** | (Current setting) |
| **Initial Value** | 0x000F0000 |
| **Notes** | - |

### 6.6.10. OMX_VIDEO_CONFIG_FRAMERATETYPE (Output Port)

[Index] OMX_IndexConfigVideoFramerate

[Details]

***xEncodeFramerate***

| | |
|---|---|
| **Write Value** | - |
| **Read Value** | See Table 6-4. |
| **Initial Value** | 0x000F0000 |
| **Notes** | - |

### 6.6.11. Frame Rate

Table 6-4 shows the relation between the parameter setting and the frame rate. The frame rate setting is represented in Q16 format. When a setting value that is not described in Table 6-4 is specified, the frame rate is calculated with some loss of precision.

**Table 6-4 Frame Rate Settings**

| Setting | Frame Rate [fps] |
|---|---|
| 0x00010000 - 0x00090000 | 1 - 9 (not supported) |
| 0x000A0000 | 10 |
| 0x000B0000 | 11 |
| 0x000C8000 | 12.5 |
| 0x000D0000 | 13 |
| 0x000EFC28 | 14.985 |
| 0x000F0000 | 15 |
| 0x00100000 | 16 |
| 0x00110000 | 17 |
| 0x00120000 | 18 |
| 0x00130000 | 19 |
| 0x00140000 | 20 |
| 0x00150000 | 21 |
| 0x00160000 | 22 |
| 0x00170000 | 23 |
| 0x00180000 | 24 |
| 0x00190000 | 25 |
| 0x001A0000 | 26 |
| 0x001B0000 | 27 |
| 0x001C0000 | 28 |
| 0x001DF851 | 29.97 |
| 0x001E0000 | 30 |
| 0x003BF0A3 | 59.94 |
| 0x003C0000 - 0xFFFFFFFF | 60 |

### 6.6.12. Bit Rate

Table 6-5 shows the valid settings of the bit rate for the encoding. The bits per a frame (bit rate / frame rate) should be less than or equal to 4 Mbits. When the bits per a frame exceeds 4 Mbits, the bit rate is automatically calculated so that it falls within 4 Mbits.

**Table 6-5 Bit Rate Settings**

| Setting (in decimal) | Encoded Bit Rate [bps] |
|---|---|
| 0 - 12,000 | 12,000 |
| 12,001 - 39,999,999 | 12,001 - 39,999,999 |
| 40,000,000 - 4,294,967,295 | 40,000,000 |

# 7. Memory Requirement

Table 7-1 describes the types of the memory that H.264 Encoder Media Component requires.

**Table 7-1 Required Memory Types**

| Memory Type | Accessible from | Description |
|---|---|---|
| input buffer | Hardware and CPU | Buffers for the input port.<br><br>The required memory size is (*nStride* x *nSliceHeight* x 3 / 2) x *nBufferCountActual.*<br>For details of the *nBufferCountActual* member, see related document [2]. |
| output buffer | Hardware and CPU | Buffers for the output port.<br><br>The required memory size is 3,110,400 x *nBufferCountActual.*<br>For details of the *nBufferCountActual* member, see related document [2].. |
| stream_work_0 | Hardware | Work buffers for encoding. |
| stream_work_1 | Hardware and CPU | |
| stream_work_2 | Hardware | |
| stream_work_4 | Hardware | |
| stream_work_5 | Hardware and CPU | |

Table 7-2 shows the memory requirement in the case of 1920x1080 stream encoding.

**Table 7-2 Memory Requirement for 1920x1080 Stream Encoding**

| Memory Type | Size | Notes |
|---|---|---|
| input buffer | 15 [Mbyte] | In the case where the *nBufferCountActual* for the input port is set to 5. |
| output buffer | 6 [Mbyte] | In the case where the *nBufferCountActual* for the output port is set to 2. |
| stream_work_0 | 20 [Mbyte] | - |
| stream_work_1 | 18 [Kbyte] | Fixed size |
| stream_work_2 | 277 [Kbyte] | - |
| stream_work_4 | 10 [Mbyte] | - |
| stream_work_5 | 9 [Kbyte] | - |

# 8. Appendix

## 8.1. Examples of typical configuration

**Table 8-1 Examples of setting 720x480 without Bframe**

| PortIndex | Index Structure | Structure member | Value |
|---|---|---|---|
| VPB+0 | OMX_VIDEO_PORTDEFINITIONTYPE | nFrameWidth | 720 |
| | | nFrameHeight | 480 |
| | | nStride | 736 |
| | | nSliceHeight | 480 |
| | | xFrameRate | 0x001E0000 |
| | | eColorFormat | OMX_COLOR_FormatYUV420SemiPlanar |
| | OMX_VIDEO_PARAM_PORTFORMATTYPE | eColorFormat | OMX_COLOR_FormatYUV420SemiPlanar |
| | | xFrameRate | 0x001E0000 |
| | OMXR_MC_VIDEO_PARAM_PICTURE_MEMORY_ALLOCTYPE | eMemoryAlloc | OMXR_MC_VIDEO_MemAllocFrame |
| VPB+1 | OMX_VIDEO_PORTDEFINITIONTYPE | nBitRate | 10000000 |
| | | eCompressionFormat | OMX_VIDEO_CodingAVC |
| | OMX_VIDEO_PARAM_BITRATETYPE | eControlRate | OMX_Video_ControlRateConstant |
| | | nTargetBitrate | 10000000 |
| | OMX_VIDEO_PARAM_AVCTYPE | nSliceHeaderSpacing | 0 |
| | | nPFrames | 29 |
| | | nBFrames | 0 |
| | | nRefFrames | 1 |
| | | eProfile | OMX_VIDEO_AVCProfileBaseline |
| | | eLevel | OMX_VIDEO_AVCLevel32 |
| | | eLoopFilterMode | OMX_VIDEO_AVCLoopFilterEnable |
| | OMX_VIDEO_PARAM_PROFILELEVELTYPE | eProfile | OMX_VIDEO_AVCProfileBaseline |
| | | eLevel | OMX_VIDEO_AVCLevel32 |
| | OMX_VIDEO_PARAM_QUANTIZATIONTYPE | nQpI | 255 |
| | | nQpP | 255 |
| | | nQpB | 255 |

**Table 8-2 Examples of setting 1920x1080 with Bframe**

| PortIndex | Index Structure | Structure member | Value |
|---|---|---|---|
| VPB+0 | OMX_PARAM_PORTDEFINITIONTYPE | nBufferCoutActual | 5 |
| | OMX_VIDEO_PORTDEFINITIONTYPE | nFrameWidth | 1920 |
| | | nFrameHeight | 1080 |
| | | nStride | 1920 |
| | | nSliceHeight | 1080 |
| | | xFrameRate | 0x001E0000 |
| | | eColorFormat | OMX_COLOR_FormatYUV420SemiPlanar |
| | OMX_VIDEO_PARAM_PORTFORMATTYPE | eColorFormat | OMX_COLOR_FormatYUV420SemiPlanar |
| | | xFrameRate | 0x001E0000 |
| | OMXR_MC_VIDEO_PARAM_PICTURE_MEMORY_ALLOCTYPE | eMemoryAlloc | OMXR_MC_VIDEO_MemAllocFrame |
| VPB+1 | OMX_VIDEO_PORTDEFINITIONTYPE | nBitRate | 20000000 |
| | | eCompressionFormat | OMX_VIDEO_CodingAVC |
| | OMX_VIDEO_PARAM_BITRATETYPE | eControlRate | OMX_Video_ControlRateConstant |
| | | nTargetBitrate | 20000000 |
| | OMX_VIDEO_PARAM_AVCTYPE | nSliceHeaderSpacing | 0 |
| | | nPFrames | 10 |
| | | nBFrames | 2 |
| | | nRefFrames | 2 |
| | | eProfile | OMX_VIDEO_AVCProfileMain |
| | | eLevel | OMX_VIDEO_AVCLevel41 |
| | | eLoopFilterMode | OMX_VIDEO_AVCLoopFilterEnable |
| | OMX_VIDEO_PARAM_PROFILELEVELTYPE | eProfile | OMX_VIDEO_AVCProfileMain |
| | | eLevel | OMX_VIDEO_AVCLevel41 |
| | OMX_VIDEO_PARAM_QUANTIZATIONTYPE | nQpI | 255 |
| | | nQpP | 255 |
| | | nQpB | 255 |

## 8.2. Examples of the GOP structure by setting

The following figures describe an example of GOP structure encoded by setting nPFrames, nBFrames and nIDRPeriod.

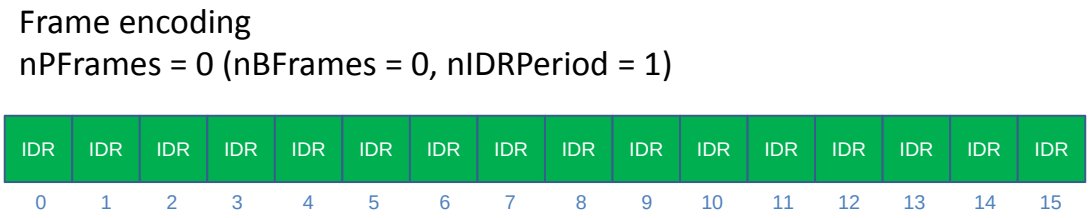


**Figure 8-1 Example 1 of GOP structure (frame encoding, nPFrames=0, nBFrames=0)**

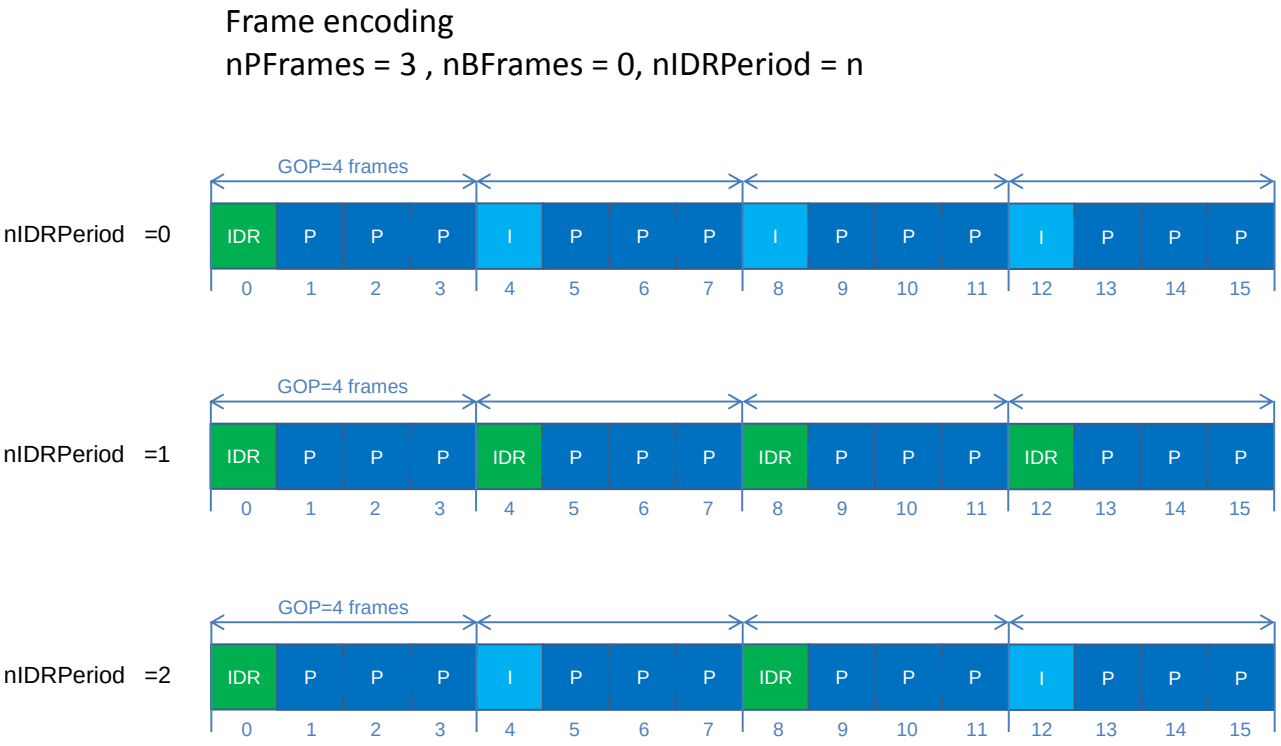


**Figure 8-2 Example 2 of GOP structure (frame encoding, nPFrames=3, nBFrames=0)**

Frame encoding
nPFrames = 2 , nBFrames = 1, nIDRPeriod = n

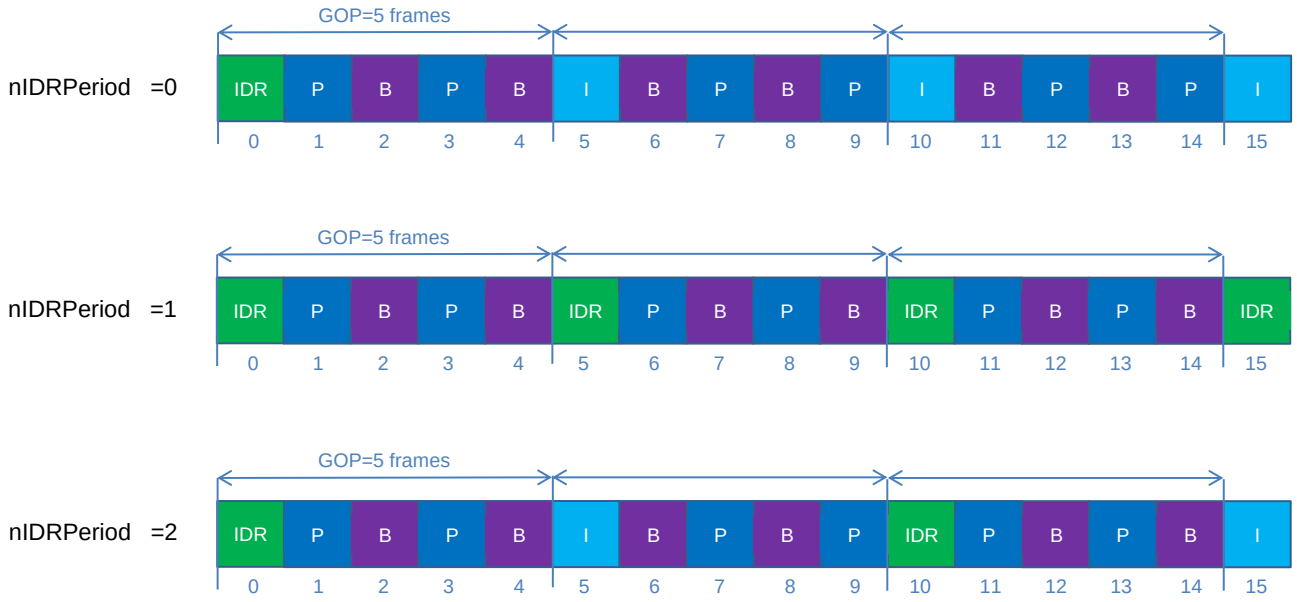


**Figure 8-3 Example 3 of GOP structure (frame encoding, nPFrames=2, nBFrames=1)**

Frame encoding
nPFrames = 2 , nBFrames = 2, nIDRPeriod = n

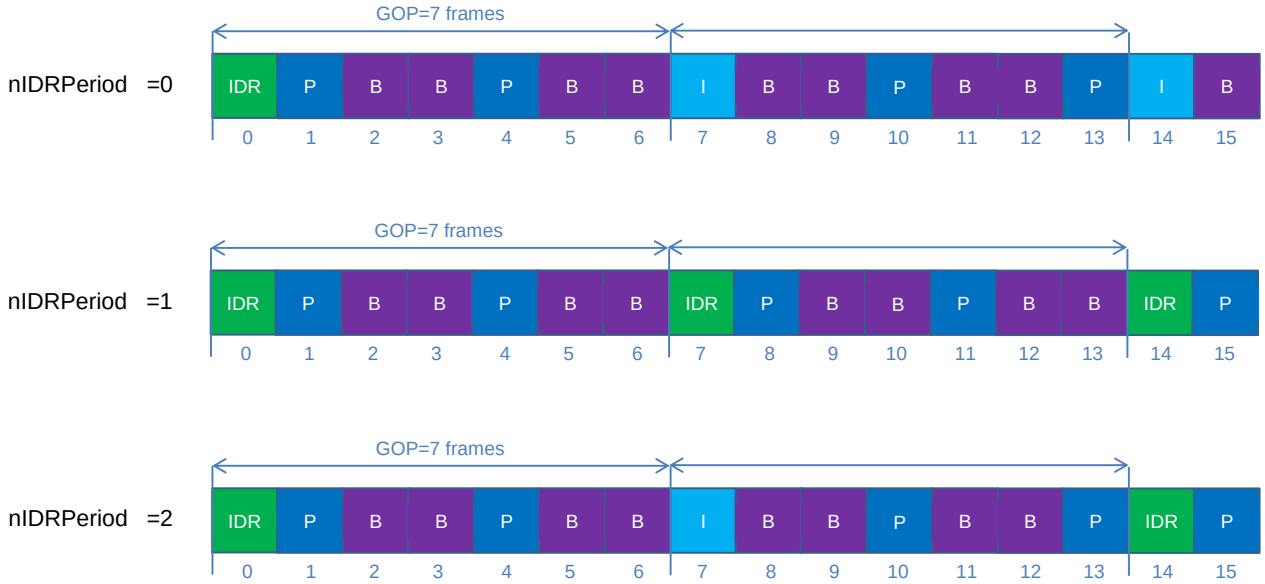


**Figure 8-4 Example 4 of GOP structure (frame encoding, nPFrames=2, nBFrames=2)**

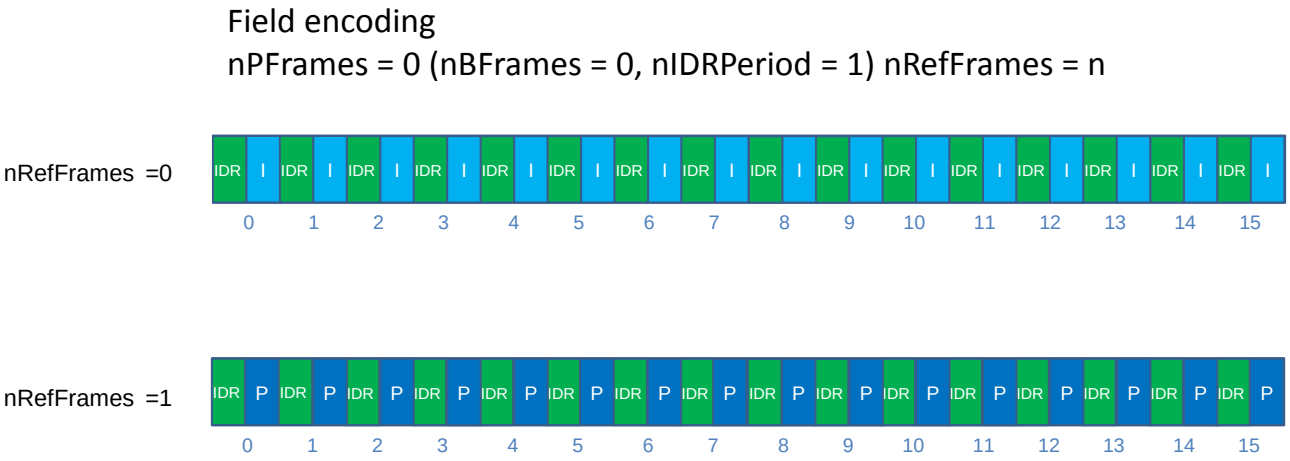


**Figure 8-5 Example 5 of GOP structure (field encoding, nPFrames=0, nBFrames=0)**

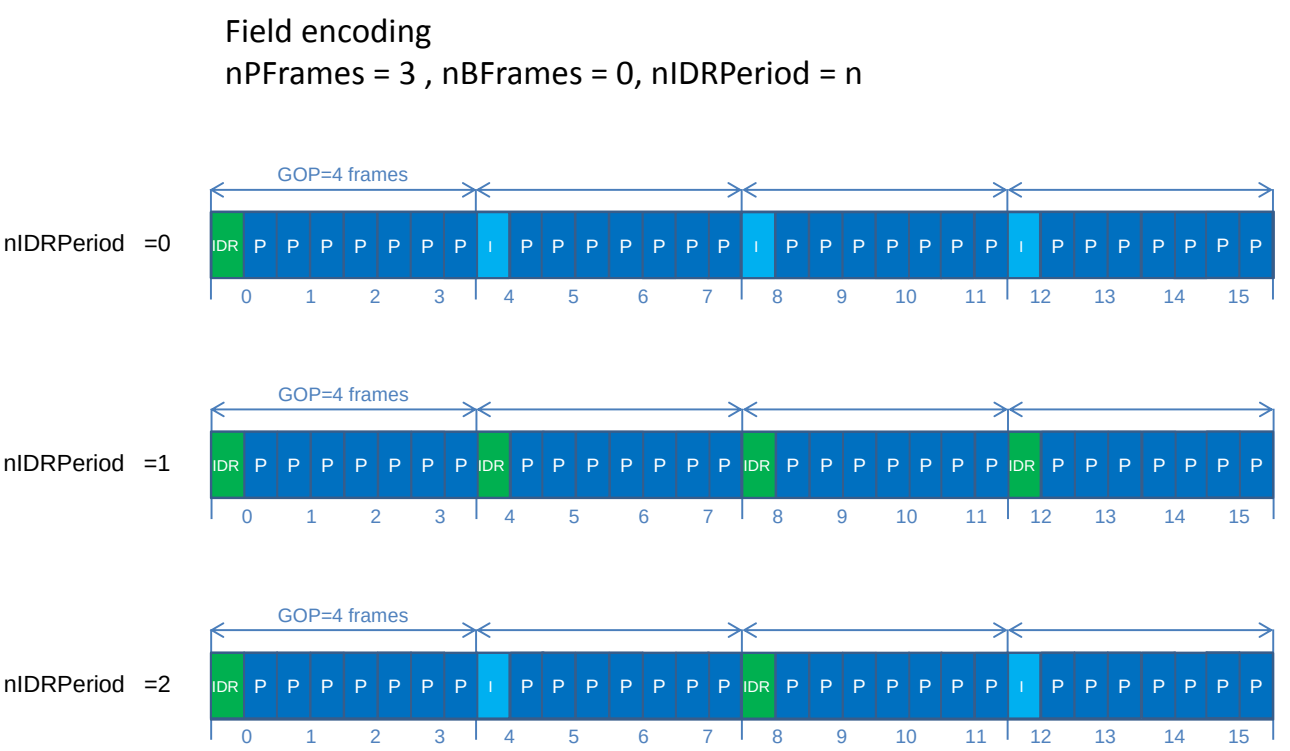


**Figure 8-6 Example 6 of GOP structure (field encoding, nPFrames=3, nBFrames=0)**

Field encoding
nPFrames = 2 , nBFrames = 1, nIDRPeriod = n


**Figure 8-7 Example 7 of GOP structure (field encoding, nPFrames=2, nBFrames=1)**

Field encoding
nPFrames = 2 , nBFrames = 2, nIDRPeriod = n

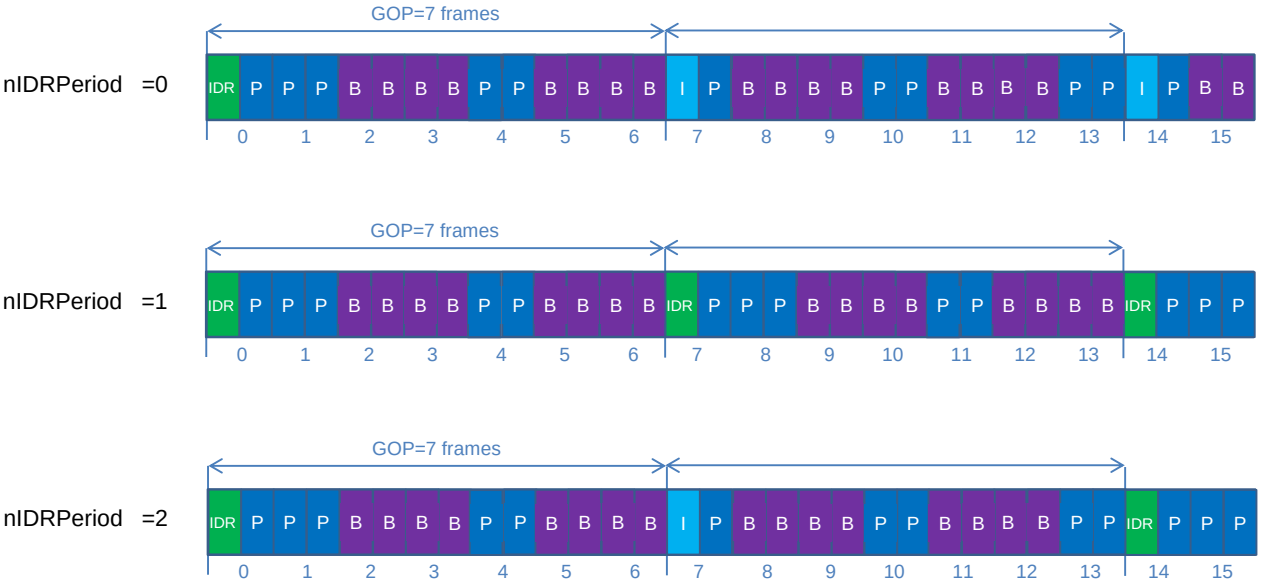


**Figure 8-8 Example 8 of GOP structure (field encoding, nPFrames=2, nBFrames=2)**

| REVISION HISTORY | OMX Media Component<br>User's Manual : H.264 Encoder Part |
|---|---|

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 0.02 | Jan. 31, 2014 | – | Draft revision based on Japanese User's Manual Rev.0.02. |
| 0.03 | Mar. 26, 2014 | – | Fixed typos<br>- The *xBitrate* member of 6.5.2.OMX_VIDEO_PORTDEFINITIONTYPE (Output Port)<br><br>Add description about the combination of the parameters and the range of the value:<br>- 6.1. OMX_VIDEO_PARAM_AVCTYPE<br>- Table 6 - 3 Available Combination of the Encoding Parameters<br>- 6.2.OMX_VIDEO_PARAM_QUANTIZATIONTYPE<br>- 6.3.OMX_VIDEO_CONFIG_AVCINTRAPERIOD<br><br>Add section:<br>- 6.5.OMXR_MC_VIDEO_PARAM_AVC_SYNTAX_OPTION.<br><br>Add detail description about the common member of the different structures:<br>- 6.6.7.OMX_VIDEO_PARAM_PROFILELEVELTYPE (ProfileLevelCurrent)<br><br>Add memory requirement information in 7. Memory Requirement<br><br>Add examples of typical configuration in 8. Appendix |
| 0.0.4 | May. 22.2014 | – | - Correct the description about the minimum value of the Encoded Bit Rate.<br>- Correct the description about the default value of the bUseHadamard member. |
| | May. 30, 2014 | – | - Correct the description for stream_work_x and lib_work_mem in Table 7-1 |
| | June. 4, 2014 | – | - Correct the value for stream_work_2 size in Table 7-2 |
| | Jul. 4, 2014 | 38 | Updated Notes and size in Table7-2 |
| 1.00 | Aug. 20 2014 | 38 | Fixed Table 7-1 |
| | Aug. 26, 2014 | 18 | Updated parameter of Set in OMX_VIDEO_PARAM_AVCTYPE. |
| | | 19 | Change "IDR frame or I frame" to "IDR frame". |
| | Sep. 9, 2014 | 6 | Fixed Table 0-1 Role Name and Component Name. |
| 1.0.1 | Oct. 14, 2014 | 38 | Updated Size in Table7-2 |
| | Oct. 23, 2014 | 41 | Add examples of the GOP structure. |
| | Nov. 06, 2014 | 24 | Add a note about OMX_VIDEO_PARAM_QUANTIZATIONTYPE |
| | Dec. 15 2014 | 38 | Remove "lib_work_mem", "stream_work_3" and "tmp_work_mem" from Table7-1 and Table7-2. |
| | | | |

**SALES OFFICES** Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.


**Renesas Electronics America Inc.**
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

**Renesas Electronics Canada Limited**
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

**Renesas Electronics Europe Limited**
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

**Renesas Electronics Europe GmbH**
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

**Renesas Electronics (China) Co., Ltd.**
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

**Renesas Electronics (Shanghai) Co., Ltd.**
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

**Renesas Electronics Hong Kong Limited**
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

**Renesas Electronics Taiwan Co., Ltd.**
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

**Renesas Electronics Singapore Pte. Ltd.**
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

**Renesas Electronics Malaysia Sdn.Bhd.**
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

**Renesas Electronics Korea Co., Ltd.**
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141



Colophon 1.3

OMX Media Component
User's Manual: H.264 Encoder Part

Publication Date: Rev. 1.01 Dec. 15, 2014

Published by: Renesas Electronics Corporation

# OMX Media Component

## User's Manual: H.264 Encoder Part

Renesas Electronics Corporation