APPS008-Codensity T4xx Encoder Reconfiguration Application Note

V3.0

09/23/2019

Table of Contents

1 Table of Abbreviations

Abbreviation:	**Full Form:**
AI	Artificial Intelligence
API	Application Programming Interface
AUD AV	Access Unit Delimiters Audio Video
AV1	Alliance Open Media Video 1 Codec
CABAC	Context Adaptive Binary Arithmetic Coding
CBR	Constant Bit Rate
CPU	Central Processing Unit
CRF	Constant Rate Factor
DSP	Digital Signal Processing
EOF	End Of File
FIXQB	Fixed Quantization Parameter
FF	Form Factor
FPS	Frames Per Second
GOP	Group of pictures
HDR	High Dynamic Range
HHHL	Half Height, Half Length
HHFL	Half Height, Full Length
HLG	Hybrid Log Gamma
HRD	Hypothetical Reference Decoder
IE	Inference Engine
Mbps	Mega Bit per second
MBps	Mega Byte per second
NLP	Natural Language Processing
NPU	Neural Processing Unit
NVMe	Non-Volatile Memory express
PCIe	Peripheral Component Interconnect express
PPS	Picture Parameter Set
QP	Quoted Printable
RGB	Red Green Blue
RGBA	Red Green Blue Alpha
RDO	Optimization
ROI	Region of interest
SDK	Software Development Kit
SEI	Supplemental Enhancement Information

SRIOV	Single Root I/O Virtualization
SPS	Sequence Parameter Set
VBR	Variable Bit Rat
VCL	Video Coding Layer
VPS	Video Parameter Set
VUI	Video Usability Information

2 References

Technical Notes:

Specifications:

3 Background

This document describes the reconfiguration feature and its usage in operation/integrating with the NETINT T4XX transcoder.

3.1 Intended Audience

This document is intended to help engineers/technicians/developers operating NETINT transcoders better understand the reconfiguration feature.

3.2 Compatibility

Software Compatibility

This guide is intended to be used with NETINT Codensity T4XX Video Transcoder software Release 1.6.0 or newer.

Hardware Compatibility

Release 1.6.0 or newer supports NETINT Codensity T4XX Video Transcoder hardware.

Operation system

All OS supporting T4XX Video Transcoder.

4 Encoder reconfig

4.1 Intro

Encoder reconfig is a feature that permits encoding parameters to be changed while encoding is under way. The types of parameters that can be changed on the T4XX is currently limited. Bitrate is one of the first implemented.

The reconfig information is passed from application to encoder through a data structure below:

```
typedef struct _ni_encoder_change_params_t
{
  uint32_t enable_option;

  int32_t bitRate;                    /**< A target bitrate */

  … …
  int32_t reserved[8];
} ni_encoder_change_params_t;
```

The **enable_option** is a flag indicating which group(s) of parameters is to be changed with the info in this struct. It is constructed by **OR**ing one or more of the following reconfig group IDs:

```
/*!*
* \brief This is an enumeration for encoder parameter change.
*/
typedef enum _ni_param_change_flags
{
  // COMMON parameters which can be changed frame by frame.
  NI_SET_CHANGE_PARAM_PPS                   = (1 << 0),
  NI_SET_CHANGE_PARAM_INTRA_PARAM           = (1 << 1),
  NI_SET_CHANGE_PARAM_RC_TARGET_RATE        = (1 << 8),
  NI_SET_CHANGE_PARAM_RC                    = (1 << 9),
  NI_SET_CHANGE_PARAM_RC_MIN_MAX_QP         = (1 << 10),
  NI_SET_CHANGE_PARAM_RC_BIT_RATIO_LAYER    = (1 << 11),
  NI_SET_CHANGE_PARAM_INDEPEND_SLICE        = (1 << 16),
  NI_SET_CHANGE_PARAM_DEPEND_SLICE          = (1 << 17),
  NI_SET_CHANGE_PARAM_RDO                   = (1 << 18),
  NI_SET_CHANGE_PARAM_NR                    = (1 << 19),
  NI_SET_CHANGE_PARAM_BG                    = (1 << 20),
  NI_SET_CHANGE_PARAM_CUSTOM_MD             = (1 << 21),
  NI_SET_CHANGE_PARAM_CUSTOM_LAMBDA         = (1 << 22),
```

```
  NI_SET_CHANGE_PARAM_RC2                  = (1 << 23),
  NI_SET_CHANGE_PARAM_VUI_HRD_PARAM        = (1 << 24),

} ni_param_change_flags_t;
```

For example, if encoding target bitrate is to be changed, **enable_option** shall be set to **NI_SET_CHANGE_PARAM_RC_TARGET_RATE**, and **bitRate** is set to the desired new bitrate value.

The reconfig group IDs in **bold** are currently supported by NetInt encoder. See Application Notes for each of these groups and features for details. Note that multiple groups can be changed at the same time by **OR**ing one or more of the reconfig group IDs, although all parameters of each group involved need to be specified, and those not to be changed must be set to their current values.

The data structure and sample code for the reconfig can be found in NetInt's libxcoder library source file libxcoder/source/ni_device_api.h, and libavcodec's NetInt encoder source code libavcodec/nienc.c.

4.2 Demo mode

The reconf feature is demonstrated as follows: a libxcoder parameter **ReconfDemoMode** is used to specify the demo mode. For example, when its value is set to 1, the bitrate reconfig demo mode will take effect. Another libxcoder parameter **ReconfFile** is used to specify a file containing a list of bitrates and frame numbers where the bitrate should be changed. The resulting FFmpeg command line options look like this:

```
sudo ./ffmpeg  -i input.264 -c:v h265_ni_enc -xcoder-params
"RcEnable=1:bitrate=1000000:ReconfDemoMode=1:ReconfFile=reconf.tx
t" output.265
```

A sample reconfig file content is as follows:

```
300:5000000
600:2000000
```

On each line of the config file, the first column is the frame index and the second column the bitrate value. This sample reconfig file specifies a test that changes the bitrate to 5 Mbps at the

300th frame, then changes the bitrate to 2 Mbps at the 600th frame. Note that in order to change the bitrate, rate control needs to be enabled in the xcoder-params, and the initial bitrate set i.e., RcEnable=1:bitrate=1000000.

5 Revision History

Version	Change	Create Date	Author
1.0	Document Created	Feb 19,2020	Zhong Wang
2.0	Updated	Jan 29, 2021	Zhong Wang
3.0	Updated to new app note template	November 30, 2021	Balwinder Sidhu

6 Legal Notice

Information in this document is provided in connection with NETINT products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in NETINT’s terms and conditions of sale for such products, NETINT assumes no liability whatsoever and NETINT disclaims any express or implied warranty, relating to sale and/or use of NETINT products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right.

A "Mission Critical Application" is any application in which failure of the NETINT Product could result, directly or indirectly, in personal injury or death. Should you purchase or use NETINT’s products for any such mission critical application, you shall indemnify and hold NETINT and its subsidiaries, subcontractors and affiliates, and the directors, officers, and employees of each, harmless against all claims costs, damages, and expenses and reasonable attorney’s fees arising out of, directly or indirectly, any claim of product liability, personal injury, or death arising in any way out of such mission critical application, whether or not NETINT or its subcontractor was negligent in the design, manufacture, or warning of the NETINT product or any of its parts.

NETINT may make changes to specifications, technical documentation, and product descriptions at any time, without notice. The information here is subject to change without notice. Do not finalize a design with this information. The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications.

NETINT, Codensity, and NETINT Logo are trademarks of NETINT Technologies Inc. All other trademarks or registered trademarks are the property of their respective owners.