NVIDIA VIDEO CODEC SDK - ENCODER

Application Note

Table of Contents

Chapter 1. NVIDIA Hardware Video Encoder

1.1. Introduction

NVIDIA GPUs - beginning with the Kepler generation - contain a hardware-based encoder (referred to as NVENC in this document) which provides fully accelerated hardware-based video encoding and is independent of graphics/CUDA cores. With end-to-end encoding offloaded to NVENC, the graphics/CUDA cores and the CPU cores are free for other operations. For example, in a game recording scenario, offloading the encoding to NVENC makes the graphics engine fully available for game rendering. In the video transcoding use-case, video encoding/decoding can happen on NVENC/NVDEC in parallel with other video post-/pre-processing on CUDA cores.

The hardware capabilities available in NVENC are exposed through APIs referred to as NVENCODE APIs in the document. This document provides information about the capabilities of the hardware encoder and features exposed through NVENCODE APIs.

1.2. NVENC Capabilities

NVENC can perform end-to-end encoding for H.264, HEVC 8-bit, HEVC 10-bit, AV1 8-bit and AV1 10-bit. This includes motion estimation and mode decision, motion compensation and residual coding, and entropy coding. It can also be used to generate motion vectors between two frames, which are useful for applications such as depth estimation, frame interpolation, encoding using other codecs not supported by NVENC, or hybrid encoding wherein motion estimation is performed by NVENC and the rest of the encoding is handled elsewhere in the system. These operations are hardware accelerated by a dedicated block on GPU silicon die. NVENCODE APIs provide the necessary knobs to utilize the hardware encoding capabilities.

Table 1 summarizes the capabilities of the NVENC hardware exposed through NVENCODE APIs.

Table 1. NVENC Hardware Capabilities

Feature	Description	Kepler GPUs	1st Gen Maxwell GPUs	2nd Gen Maxwell GPUs	Pascal GPUs	Volta and TU117 GPUs	Ampere and Turing GPUs (except TU117)	Ada GPUs
H.264 baseline, main and high profiles	Capability to encode YUV 4:2:0 sequence and generate a H.264-bit stream.	Y	Y	Y	Y	Y	Y	Y
H.264 4:4:4 encoding (only CAVLC)	Capability to encode YUV 4:4:4 sequence and generate a H.264-bit stream.	N	Y	Y	Y	Y	Y	Y
H.264 lossless encoding	Lossless encoding.	N	Y	Y	Y	Y	Y	Y
H.264 motion estimation (ME) only mode	Capability to provide macro-block level motion vectors and intra/inter modes.	N	Y	Y	Y	Y	Y	Y
H.264 field encoding	Capability to encode field content.	Y	Y	Y	Y	Y	N	N
H.264/HEVC weighted prediction	Support for weighted prediction.	N	N	N	Y	Y	Y	Y
Encoding support for H.264 ARGB content	Capability to encode RGB input.	Y	Y	Y	Y	Y	Y	Y
Multiple reference frames for H.264	Capability to use different reference frames	N	N	N	N	N	Y	Y
HEVC main profile	Capability to encode YUV 4:2:0 sequence and generate a HEVC bit stream.	N	N	Y	Y	Y	Y	Y
HEVC lossless encoding	Lossless encoding.	N	N	N	Y	Y	Y	Y
HEVC main10 profile	Support for encoding 10-bit content generate a HEVC bit stream.	N	N	N	Y	Y	Y	Y
HEVC 4:4:4 encoding	Capability to encode YUV 4:4:4 sequence and generate a HEVC bit stream.	N	N	N	Y	Y	Y	Y
HEVC motion estimation (ME) only mode	Capability to provide CTB level motion vectors and intra/ inter modes.	N	N	N	Y	Y	Y	Y
HEVC 8K encoding	Support for encoding 8192 × 8192 Content.	N	N	N	Y*	Y	Y	Y
HEVC sample adaptive offset (SAO)	Improves encoded video quality.	N	N	N	Y	Y	Y	Y
HEVC B frame	Improves encoded quality	N	N	N	N	N	Y	Y

Feature	Description	Kepler GPUs	1st Gen Maxwell GPUs	2nd Gen Maxwell GPUs	Pascal GPUs	Volta and TU117 GPUs	Ampere and Turing GPUs (except TU117)	Ada GPUs
Multiple reference frames for HEVC	Capability to use different reference frames	N	N	N	N	N	Y	Y
AV1 main profile	Capability to encode YUV 4:2:0 8-bit and 10-bit content up to 8192 × 8192 resolution and generate a AV1 bit stream.	N	N	N	N	N	N	Y

- **Y**: Supported, **N**: Not supported
- *Supported in select Pascal generation GPUs

1.3. NVENC Licensing Policy

As far as NVENC hardware encoding is concerned, NVIDIA GPUs are classified into two categories: "qualified" and "non-qualified". On qualified GPUs, the number of concurrent encode sessions is limited by available system resources (encoder capacity, system memory, video memory etc.). On non-qualified GPUs, the number of concurrent encode sessions is limited to 8 per system. This limit of 8 concurrent sessions per system applies to the combined number of encoding sessions executed on all non-qualified cards present in the system.

For a complete list of qualified and non-qualified GPUs, refer to https://developer.nvidia.com/nvidia-video-codec-sdk..

For example, on a system with one Quadro RTX4000 card (which is a qualified GPU) and three GeForce cards (which are non-qualified GPUs), the application can run N simultaneous encode sessions on Quadro RTX4000 card (where N is defined by the encoder/memory/hardware limitations) and 8 sessions on all the GeForce cards combined. Thus, the limit on the number of simultaneous encode sessions for such a system is $N + 8$.

1.4. NVENC Performance

With every generation of NVIDIA GPUs (Maxwell 1st/2nd gen, Pascal, Volta, Turing, Ampere and Ada), NVENC performance has increased steadily. Table 2 provides *indicative*[1] NVENC performance on Pascal, Turing, and Ada GPUs for different presets and rate control modes (these two factors play a major role in determining the performance and quality). Note that performance numbers in Table 2 are measured on GeForce hardware with assumptions listed under the table. The performance varies across GPU classes (e.g. Quadro, Tesla), and scales (almost) linearly with the clock speeds for each hardware.

[1] Encoder performance depends on many factors, including but not limited to: Encoder settings, GPU clocks, GPU type, video content type etc.

While first-generation Maxwell GPUs had one NVENC engine per chip, certain variants of the second-generation Maxwell, Pascal, Volta and Ada GPUs have two/three NVENC engines per chip. This increases the aggregate encoder performance of the GPU. NVIDIA driver takes care of load balancing among multiple NVENC engines on the chip, so that applications don't require any special code to take advantage of multiple encoders and automatically benefit from higher encoder capacity on higher-end GPU hardware. The encode performance listed in Table 2 is given *per NVENC engine*. Thus, if the GPU has 2 NVENCs (e.g. GP104, AD104), multiply the corresponding number in Table 2 by the number of NVENCs per chip to get aggregate maximum performance (applicable only when running multiple simultaneous encode sessions). Note that unless Split Frame Encoding is enabled, performance with single encoding session cannot exceed performance per NVENC, regardless of the number of NVENCs present on the GPU. Multi NVENC Split Frame Encoding is a feature introduced in SDK12.0 on Ada GPUs for HEVC and AV1. Refer to the NVENC Video Encoder API Programming Guide for more details on this feature.

NVENC hardware natively supports multiple hardware encoding contexts with negligible context-switching penalty. As a result, subject to the hardware performance limit and available memory, an application can encode multiple videos simultaneously. NVENCODE API exposes several presets, rate control modes and other parameters for programming the hardware. A combination of these parameters enables video encoding at varying quality and performance levels. In general, one can trade performance for quality and vice versa.

Table 2. NVENC encoding performance in frames/second (fps)

Preset	RC Mode	Tuning Info	H.264				HEVC				AV1
			Pascal	Turing	Ampere	Ada	Pascal	Turing	Ampere	Ada	Ada
p1	CBR	LL	681	787	810	805	537	834	875	903	930
	VBR	HQ	700	761	788	779	503	824	865	883	696
p3	CBR	LL	663	550	570	595	440	412	433	458	697
	VBR	HQ	396	547	568	589	438	494	514	639	532
p5	CBR	LL	364	243	253	278	367	271	282	326	475
	VBR	HQ	325	236	246	272	366	298	308	389	419
p7	CBR	LL	322	204	213	238	341	271	282	325	338
	VBR	HQ	244	186	194	206	262	149	156	178	312

- Resolution/Input Format/Bit depth: 1920 × 1080/YUV 4:2:0/8-bit
- Above measurements are made using the following GPUs: GTX 1060 for Pascal, RTX 8000 for Turing, RTX 3090 for Ampere, and RTX 4090 for Ada. All measurements are done at the highest video clocks as reported by nvidia-smi (i.e. 1708 MHz, 1950 MHz, 1950 MHz, 2415 MHz for GTX 1060, RTX 8000, RTX 3090, and RTX 4090 respectively). The performance should scale according to the video clocks as reported by nvidia-smi for other GPUs of every

individual family. Information on nvidia-smi can be found at https://developer.nvidia.com/nvidia-system-management-interface.

- H.264 and HEVC encoding fps for Volta GPU can be obtained by multiplying the Pascal fps in the above table by ratio of the clocks, as reported by nvidia-smi.
- Software: Windows 11, Video Codec SDK v12.2, NVIDIA display driver: 551.76
- CBR: Constant bitrate rate control mode, VBR: Variable bitrate rate control mode, LL : Low latency tuning info, HQ: High quality tuning info

1.5. Programming NVENC

Refer to the SDK release notes for information regarding the required driver version.

Refer to the documents and the sample applications included in the SDK package for details on how to program NVENC.

1.6. FFmpeg Support

FFmpeg is the most popular multimedia transcoding tool used extensively for video and audio transcoding.

The video hardware accelerators in NVIDIA GPUs can be effectively used with FFmpeg to significantly speed up the video decoding, encoding and end-to-end transcoding at very high performance. For more information on how to use NVENC or NVDEC with FFmpeg, please refer to the FFmpeg guide in the Video Codec SDK.

Note that FFmpeg is open-source project and its usage is governed by specific licenses and terms and conditions for FFmpeg.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com