



NVENC - NVIDIA KEPLER HARDWARE VIDEO ENCODER

NVENC_DA-06209-001_v04 | August 2013

Application Note



DOCUMENT CHANGE HISTORY

NVENC_DA-06209-001_v03

Version	Date	Authors	Description of Change
01	January 30, 2012	AP/CC	Initial release
02	September 24, 2012	AP	Updated for NVENC SDK release 2.0
03	April 10, 2013	AP	Updated for Monterey SDK 2.0.0 update
04	August 4, 2013	AP	Updated for NVENC SDK release 3.0

TABLE OF CONTENTS

- Kepler Hardware Video Encoder (NVENC) 5**
 - 1. Introduction..... 5
 - 2. NVENC Capabilities 6
 - 2.1 Block Diagram..... 7
 - 2.2 Performance..... 8
 - 2.3 Quality.....10
 - 3. Programming NVENC.....12

LIST OF FIGURES

Figure 1. NVENC hardware block diagram.....	7
Figure 2. NVENC Performance With Single HD Encode Session (bitrate = 10 Mbps for 1080p and 5 Mbps for 720p)	9
Figure 3. NVENC Performance With Multiple Concurrent HD Encode Sessions (bitrate = 30 Mbps)9	
Figure 4. Quality Comparison Among NVENC, x264 and Intel SB Encoders	10
Figure 5. Quality Comparison Among NVENC, x264 and Intel SB Encoders	11

LIST OF TABLES

Table 1.	NVENC Hardware Capabilities in Kepler	6
Table 2	Comparison Between NVENC SDK and GRID SDK Capabilites	12

KEPLER HARDWARE VIDEO ENCODER (NVENC)

1. INTRODUCTION

NVIDIA's latest generation of GPUs based on the Kepler architecture, contain a hardware-based H.264 video encoder (henceforth referred to as NVENC). This document provides information about the capabilities of the hardware encoder, along with some relevant data about encoding quality and performance.

Before Kepler GPUs, the only NVIDIA solution for video encoding was the CUDA-based encoder. This encoder could be used via the NVCUVENC API in NVIDIA CUDA SDK. One of the disadvantages of the CUDA-based encoder is that it uses a combination of the CPU and GPU's graphics engine for encoding, taking away processing power from other tasks that can be performed on the CPU and GPU's graphics engine. This approach also increases overall system power consumption.

NVENC, being dedicated H.264 hardware on the GPU chip, does not use the GPU's graphics engine and hence uses much less power compared to the CUDA-based encoder. It also leaves the CPU to perform other tasks. The hardware is optimized to provide excellent quality at high performance, enabling a wide range of applications that require video encoding capabilities. The NVENC hardware encoder improves encoding performance by almost a factor of 4, compared to the CUDA encoder¹ at equivalent quality.

It is important to note that an application can choose to encode using both NVENC hardware and NVIDIA's legacy CUDA encoder in parallel, without affecting each other.

¹ CUDA encoder profiled with Core 2 Duo (2.6 GHz) + Tesla C2050 with GF100

Note, however, that, video pre-processing algorithms may require CUDA, and may result in reduced performance from the CUDA encoder.

As explained in Section 3, NVENC hardware's encoding capabilities can be accessed via two API's: NVENC SDK and GRID SDK. Although there is some overlap in the functionality provided by these two SDK's, they are designed for slightly different use-cases (explained further in Section 3).

2. NVENC CAPABILITIES

At a high level, capabilities of NVENC hardware in Kepler are summarized in Table 1.

Table 1. NVENC Hardware Capabilities in Kepler GPUs

Feature	What it Provides
Supported codec	H.264
H.264 base, main, high profiles	Wide range of use-cases
Up to 8x HD encode (1080p @ 240 fps)	Faster than real-time encoding
Flexible ME, QP maps	Customizable quality, region of interest encoding
YUV 4:2:0 and planar 4:4:4 support	High-quality encoding with and without chroma subsampling
MVC	Full-resolution stereo encode
Up to 4096 × 4096 in hardware	High resolution encode
API	NVENC SDK (Flexible API, Windows/Linux) GRID SDK (Capture + encode, Windows now, Linux - future)
NVENC and CUDA parallelism	Simultaneous and parallel HW and CUDA encoding for increased performance

The NVENC hardware is designed to accept YUV (NV12) picture data and output H.264 elementary encoded bit-stream, as per the specified settings. The hardware itself provides the ability to control the range of encoding parameters from software, some of which are exposed via the software API's in NVENC SDK and GRID SDK. See Section 3.

Every GPU from NVIDIA's Kepler family contains 1 NVENC engine, which is independent of the graphics engine. The NVENC engine runs at the same clock speed, and its performance is independent of the graphics' performance.

2.1 Block Diagram

Figure 1 shows the block diagram of NVENC. Apart from the rate control and picture type decision, NVENC can perform all tasks that are a critical part of the end-to-end H.264 encoding. The rate control algorithm is implemented in GPU's firmware and controlled via the driver. From the application's perspective, rate control is a hardware function controlled via the parameters exposed in the NVENC/GRID APIs. The hardware also provides capability to use external motion estimation engine (for higher quality) and custom quantization parameter maps (for "region of interest" encoding). These features, however, are currently not exposed in the software APIs and will be available in future releases of the SDK.

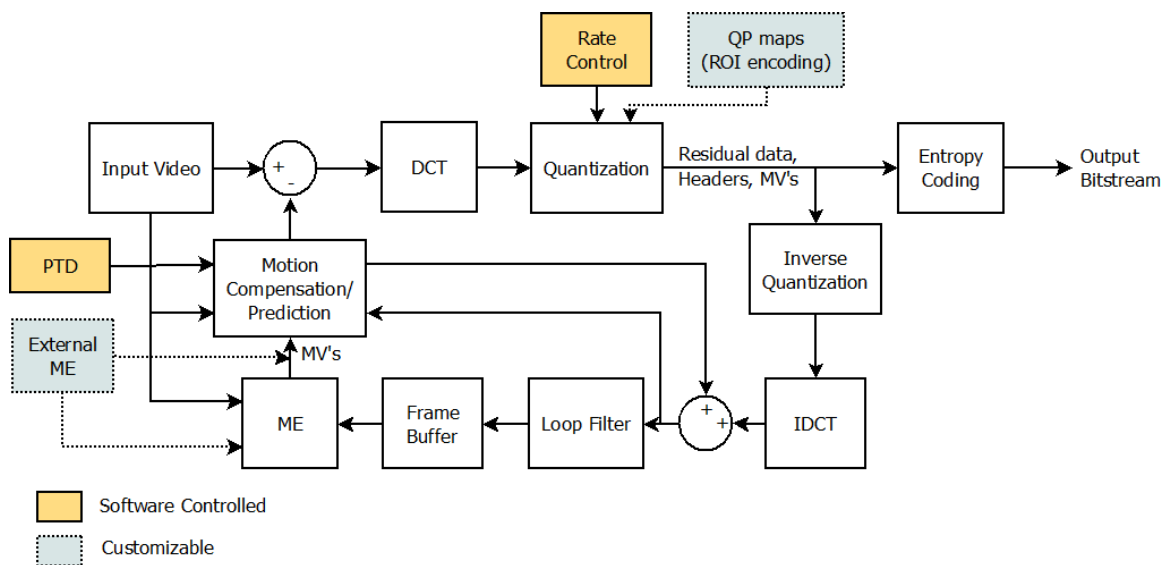


Figure 1. NVENC hardware block diagram

2.2 Performance

The NVENC hardware is designed to support up to 8X real-time HD video encoding (1 HD = 1080p @ 30 fps). This means that the hardware can encode up to 240 frames per second of 1920 × 1080 progressive video in high-performance (HP preset) mode. The application can trade performance for encoded picture quality.

A more common setting of the encoder (internally referred to as HQ – High Quality) results in a very good quality encoded bit-stream. At this setting, NVENC can encode 1080p video at 4X real-time; i.e. at 120 fps (no B frames).

With the inclusion of B frames in the encoding, the performance is lower and depends on the exact GOP structure used.

The encoding latency is currently 1 frame (without B-frames), but the software supports slice-based encoding and subsequent software API releases will expose this feature to the applications.

The hardware has been extensively tested and verified to yield the advertised performance at all settings. The performance has been measured using the sample application provided with the NVENC SDK, using a single encode session and multiple concurrent encode sessions. Figure 2 shows the measured encoding performance of NVENC with a few video and gaming clips using 2 encoding presets.

The performance does not vary if using motion video or synthetically-generated content (e.g. gameplay, desktop), although the quality and latency requirements for different types of content may be significantly different, thereby affecting the encoding performance positively or negatively.

NVENC hardware natively supports multiple hardware encoding contexts with negligible context-switching penalty. Figure 3 shows the NVENC hardware's encoding performance using 2 video clips at 2 presets, running multiple simultaneous sessions. Due to hardware context switching penalties, there is a slight drop in the encoding performance while running more than one concurrent session (less than 5-10% for second session, negligible beyond the second session), as illustrated in Figure 3.

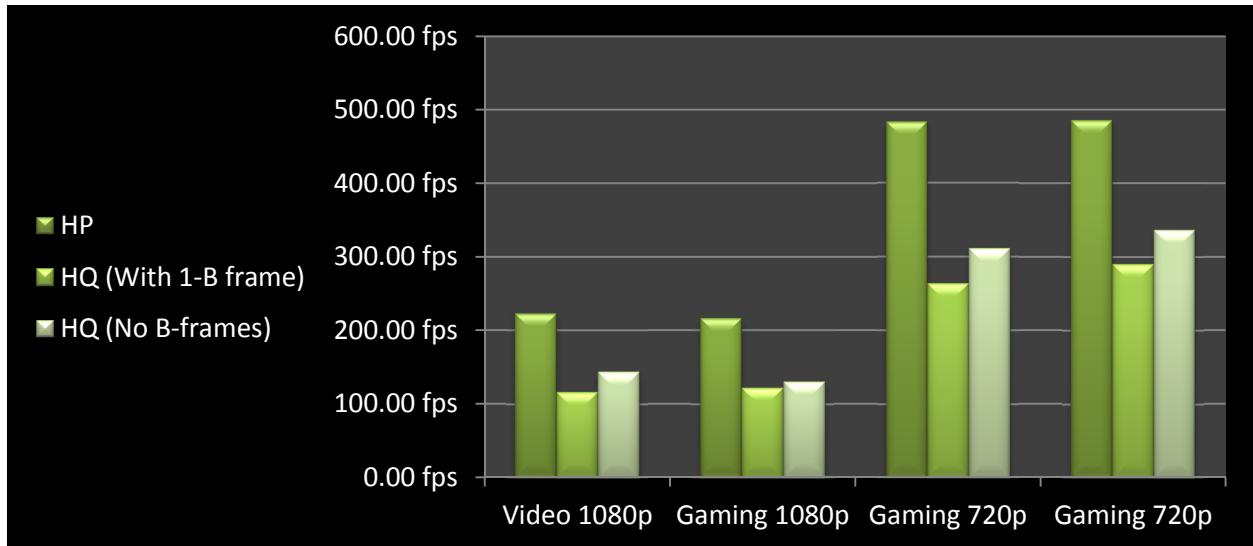


Figure 2. NVENC Performance With Single HD Encode Session (bitrate = 10 Mbps for 1080p and 5 Mbps for 720p)

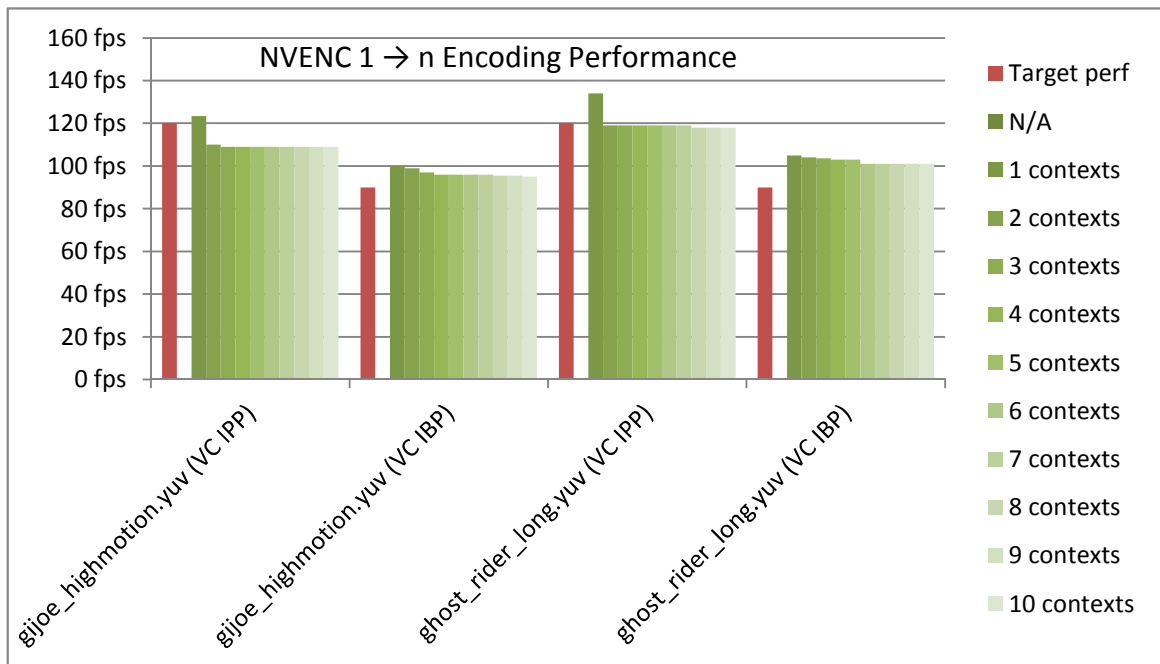


Figure 3. NVENC Performance With Multiple Concurrent HD Encode Sessions (bitrate = 30 Mbps)

2.3 Quality

NVENC hardware has been designed to provide quality comparable to x264 (an open source H.264 encode library) with much higher performance. The comparable x264 preset used for quality comparison between NVENC, x264 and other competitive solutions is as follows (refer to x264 documentation):

```
--keyint 30 --no-scenecut --no-chroma-me --fps 30 --preset faster --ref 1 --trellis 0 --weightp 0 --b-pyramid normal --bframe n
```

Figure 4 and Figure 5 show some representative rate distortion curves, representing the encode quality obtained via NVENC hardware encoder, x264 software encoder and Intel Sandy Bridge hardware encoder.

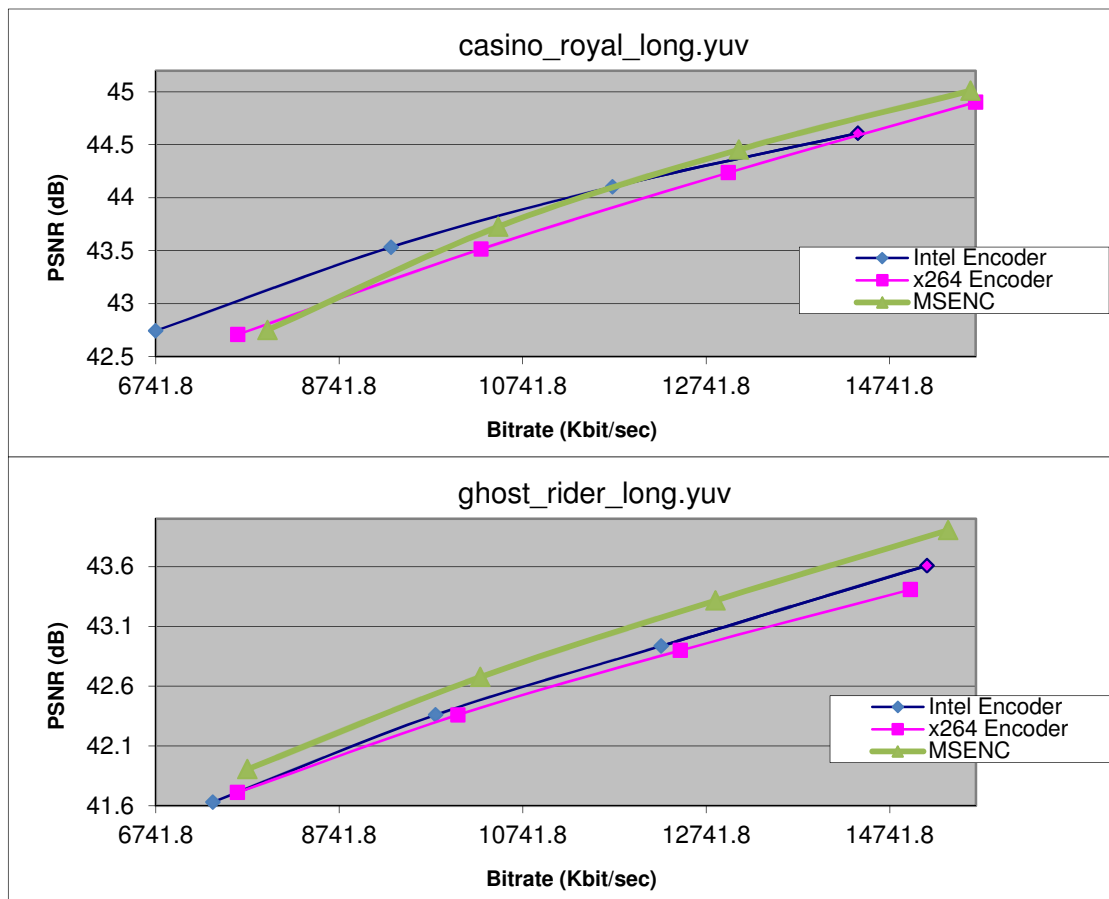


Figure 4. Quality Comparison Among NVENC, x264 and Intel SB Encoders

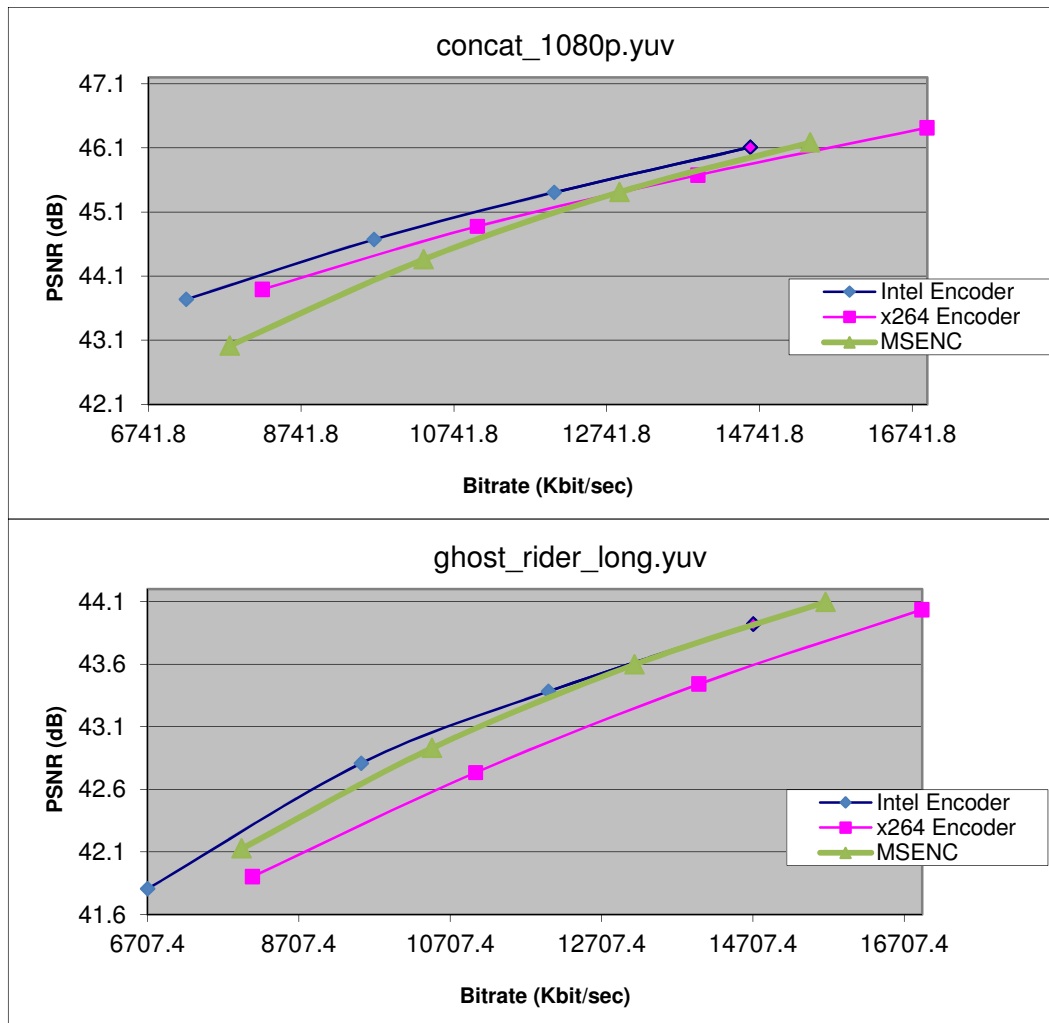


Figure 5. Quality Comparison Among NVENC, x264 and Intel SB Encoders

3. PROGRAMMING NVENC

Various capabilities of NVENC are exposed to the application software via the NVIDIA proprietary application programming interface (API). There are two API's available to use NVENC encoding capabilities:

1. NVENC SDK – Useful for direct encoding applications such as video conferencing, transcoding, video editing, archiving etc.
2. GRID SDK – Useful for screen capture + encoding use-cases such as cloud gaming, streaming etc.

Table 2 Comparison between NVENC SDK and GRID SDK Capabilities

Direct Encode - NVENC SDK	Capture + Encode - GRID SDK
No capture - H.264 encode only	Capture + H.264 encode
Use cases: Transcoding, archiving, video conferencing, video editing, camera capture and encoding	Use cases: Low-latency applications such as cloud gaming, streaming where a single API performs screen capture + encode in most optimized manner
Linux and Windows	Linux and Windows
Access to exhaustive encoder settings and fine-grained control	Limited encoder settings, applicable to only low-latency streaming use-cases
Available via NVIDIA developer zone at https://developer.nvidia.com/nvidia-video-codec-sdk	Available under license from NVIDIA
Works on Quadro, Tesla, and GRID boards	Works on Quadro and GRID boards

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

ROVI Compliance Statement

NVIDIA Products that support Rovi Corporation's Revision 7.1.L1 Anti-Copy Process (ACP) encoding technology can only be sold or distributed to buyers with a valid and existing authorization from ROVI to purchase and incorporate the device into buyer's products.

This device is protected by U.S. patent numbers 6,516,132; 5,583,936; 6,836,549; 7,050,698; and 7,492,896 and other intellectual property rights. The use of ROVI Corporation's copy protection technology in the device must be authorized by ROVI Corporation and is intended for home and other limited pay-per-view uses only, unless otherwise authorized in writing by ROVI Corporation. Reverse engineering or disassembly is prohibited.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Trademarks

NVIDIA, the NVIDIA logo, and <add all the other product names listed in this document> are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2011-2013 NVIDIA Corporation. All rights reserved.