NVENC - NVIDIA KEPLER HARDWARE VIDEO ENCODER

Application Note
# DOCUMENT CHANGE HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Authors</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>January 30, 2012</td>
<td>AP/CC</td>
<td>Initial release</td>
</tr>
<tr>
<td>02</td>
<td>September 24, 2012</td>
<td>AP</td>
<td>Updated for NVENC SDK release 2.0</td>
</tr>
</tbody>
</table>
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 .................................................................................................. 11
3. Programming NVENC ................................................................................ 13
4. References .................................................................................................. 14
LIST OF FIGURES

Figure 1. NVENC block diagram .......................................................... 7
Figure 2. NVENC Performance With Single HD Encode Session (bitrate = 30 Mbps) ............... 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 ............................ 11
Figure 5. Quality Comparison Among NVENC, x264 and Intel SB Encoders ............................ 12

LIST OF TABLES

Table 1. NVENC Hardware Capabilities in Kepler ........................................... 6
Table 2. NVENC Presets Exposed Via NV Encode API ..................................... 13
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 quality and performance.

Before Kepler GPUs, the only NVIDIA solution for video encoding was via use of NVIDIA’s CUDA-based encoder, exposed through the NVCUVENC API. One of the disadvantages of the CUDA-based encoder is that it used a combination of the CPU and GPU’s 3D engine for encoding, leaving very little processing power for other tasks. This approach also increased overall system power consumption.

NVENC, being dedicated H.264 hardware, does not use 3D 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. Note, however, that, video pre-processing algorithms may require CUDA, and will result in reduced performance from the CUDA encoder.

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

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

<table>
<thead>
<tr>
<th>Feature</th>
<th>What it Provides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported codec</td>
<td>H.264</td>
</tr>
<tr>
<td>H.264 base, main, high profiles</td>
<td>Wide range of use-cases</td>
</tr>
<tr>
<td>Up to 8x HD encode</td>
<td>Faster than real-time encoding</td>
</tr>
<tr>
<td>Flexible ME, QP maps</td>
<td>Customizable quality, region of interest</td>
</tr>
<tr>
<td>YUV 4:2:0 support</td>
<td>Standard YUV encoding with chroma subsampling</td>
</tr>
<tr>
<td>YUV 4:4:4 support (planar)</td>
<td>High quality encoding without chroma subsampling</td>
</tr>
<tr>
<td>MVC</td>
<td>Full-resolution stereo encode</td>
</tr>
<tr>
<td>Up to 4096 × 4096</td>
<td>High resolution encode</td>
</tr>
<tr>
<td>API</td>
<td>NVENC Video Encoder API</td>
</tr>
<tr>
<td></td>
<td>Support for Windows &amp; Linux,</td>
</tr>
<tr>
<td></td>
<td>Easy to use, coarse and fine control</td>
</tr>
</tbody>
</table>

The NVENC hardware is designed to accept YUV 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 in the proprietary software API (NVENC Video Encoder API). See Section 3.
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. 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 later releases.

Figure 1. NVENC block diagram
2.2 Performance

The NVENC hardware is designed to support up to 8X real-time HD video encoding (1080p @ 30 fps). This means that the hardware can encode 240 frames per second of 1920 × 1080 progressive video. 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.

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 [1], using a single encode session and multiple concurrent encode sessions. Figure 2 shows the measured encoding performance of NVENC with various sample video clips using several presets.

Although the performance benchmarking results below use motion video, performance is not different with synthetic content (e.g. gameplay, desktop). However, it should be noted that the quality constraints for such synthetic content can vary significantly from application to application, and this may indirectly affect the performance.
Figure 2. NVENC Performance With Single HD Encode Session (bitrate = 30 Mbps)

Figure 3. NVENC Performance With Multiple Concurrent HD Encode Sessions (bitrate = 30 Mbps)
Due to hardware context switching penalties, there is a slight drop in the encoding performance while running more than one concurrent session (less than 10% for second session, negligible beyond the second session), as illustrated in Figure 3.
2.3 Quality

NVENC hardware has been designed to provide quality comparable to x264 [5] (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 [5]):

```
--keyint 30 --no-scencut --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 an NVIDIA proprietary application programming interface (API) named NVENC Video Encoder API. This API provides coarse control (for novice application developers) and fine control (for experienced and knowledgeable developers) over various encoding parameters. The API provides several pre-defined presets for popular encoding applications, as shown in Table 2.

**Table 2. NVENC Presets Exposed Via NV Encode API**

<table>
<thead>
<tr>
<th>Preset</th>
<th>GOP structure</th>
<th>Expected performance</th>
<th>Encode quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC</td>
<td>IPPP</td>
<td>120 fps</td>
<td>Average</td>
</tr>
<tr>
<td>WIDI</td>
<td>IPPP</td>
<td>90 fps</td>
<td>Average</td>
</tr>
<tr>
<td>HP</td>
<td>IPPP</td>
<td>240 fps</td>
<td>Low</td>
</tr>
<tr>
<td>HQ</td>
<td>IBBP</td>
<td>60 fps</td>
<td>High</td>
</tr>
<tr>
<td>CLOUD_GAMING_720p60</td>
<td>IPPPP...</td>
<td>80-90 fps</td>
<td>Very high</td>
</tr>
<tr>
<td>CLOUD_GAMING_720p30</td>
<td>IPPPP...</td>
<td>50-60 fps</td>
<td>Very high</td>
</tr>
</tbody>
</table>

Version 1.0 of the NV Encode API (available March 2012) provides Windows and DirectX compatibility, whereas version 2.0 (available in October 2012) provides Windows/DirectX/Linux/CUDA interoperability.

The API provides two modes of operation in Windows: synchronous and asynchronous, whereas it supports only synchronous mode on Linux; supports external (application-driven) or internal picture type decision; supports multiple rate control options (CBR, VBR, 2-pass, low-latency, with ability to switch RC mode) etc. For details about the capabilities of the NVENC Video Encoder API, please refer to [3][3].
4. REFERENCES

[1] NVENC Software Development Kit (SDK), available through the NVIDIA Video Codec SDK License Agreement


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.