NGCodec Hardware HEVC Encoding

User Guide

UG1408 (v1.0.1) October 2, 2020

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
10/02/2020 Version 1.0.1	
1	Link update.
4	Link update.
07/01/2020 Version 1.0	
Initial release	N/A

Table of Contents

Chapter 1

Introduction

System Description

Hardware acceleration allows for faster encoding with better quality for the same bitrate at lower cost and lower latency than encoding with only software. The NGCodec FPGA-based hardware HEVC encoder provides you with these benefits.

The system on which the hardware accelerator is running is shown in the following figures.

Figure 1: **FPGA on PCIe® Card System Configuration**

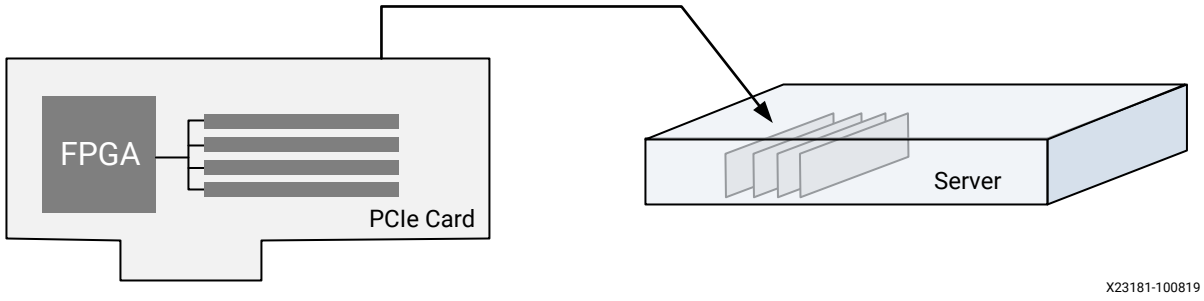
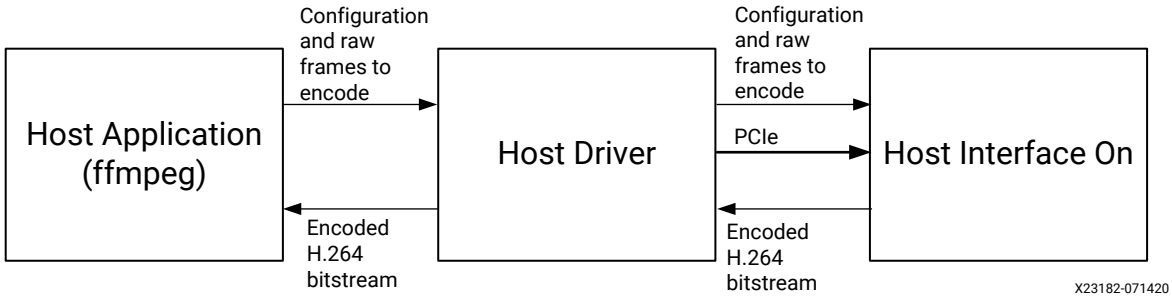


Figure 2: **Software Configuration**



The encoder hardware is programmed into an FPGA. The FPGA is located on a PCIe card. The PCIe card also includes local DDR memory that is used by the hardware encoder when it is operating. One or more PCIe cards are installed in a server in a data center. Because all encoding is done by the hardware, the host server does not do much work, and can therefore support multiple hardware encoder cards simultaneously.

An application, such as FFmpeg, is provided allowing you to configure the encoder parameters and to encode video. The FFmpeg application is pre-compiled with links to a host driver that knows how to communicate with the hardware on the PCIe card. When FFmpeg is given NGC265 as the encoder, it automatically sends the video to be HEVC-encoded to the hardware, and receives the encoded bitstream.

Encoder Throughput

Unlike software encoders, the NGCodec hardware HEVC encoder runs at the same speed at high or low bitrates, or if the input video stream is simpler or more complex to encode. The throughput of the encoder is fixed and can be subdivided across multiple streams.

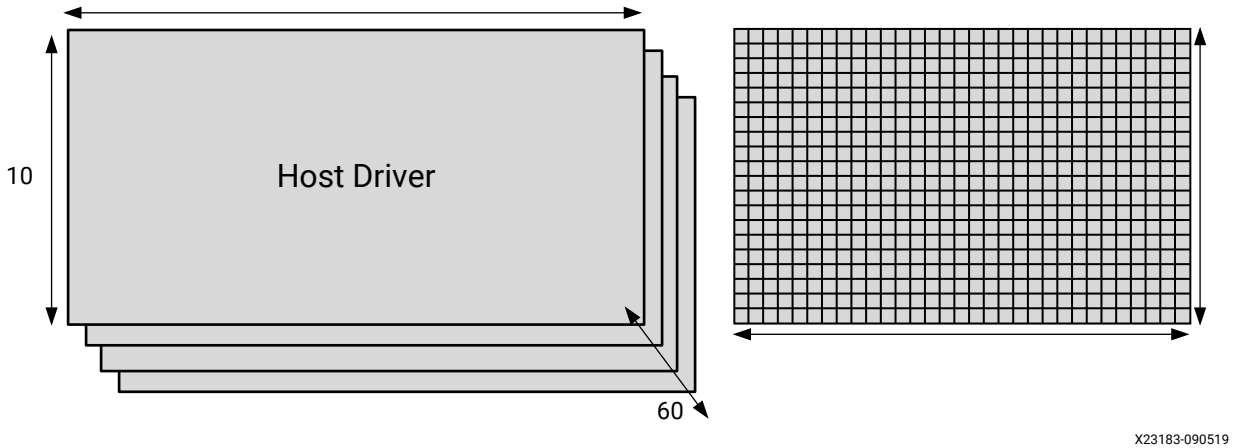
In many cases, an encoding workload requires encoding an input video at multiple resolutions and bitrates. The NGCodec encoder can support multiple streams of encoding as long as the overall throughput does not exceed the limits of 1080p120fps.

For example, suppose you want to encode three streams at the same time from the same source, one at 1080p60, 720p60 and 480p60. Can the encoder with a throughput of 1080p120 support this? The following figure shows how to make the calculation.

1. Calculate the throughput of the encoder in blocks of 64x64 pixels. This is called "CTU" in HEVC. One 1080p frame is 1920x1080 pixels, which is 30x17 CTUs.

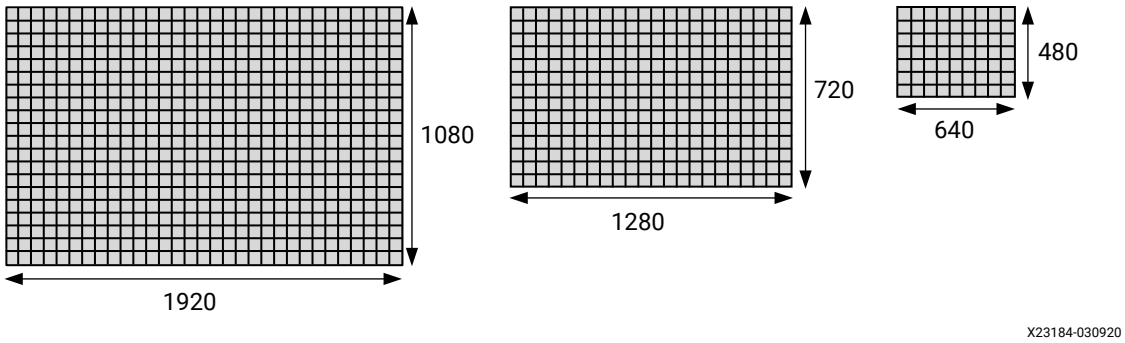
   ***Note*:** If the height or width is not evenly divisible by 64, round up to the next number. Because there are 60 frames per second, the capacity of the NGCodec encoder is 30x17x120 = 61,200 CTUs per second.

Figure 3: **Calculating Encoder Capacity for Multiple Streams**



2. Calculate the CTUs required by each stream as shown in the following figure.

Figure 4: **Stream CTUs**



3. 1080p60 = 30x17x60fps = 30,600 CTUs/sec. 720p60 = 20x12x60fps = 14,400 CTUs/sec. 480p60 = 10x8x60fps = 4800 CTUs/sec.

   Adding these numbers together gives 30,600 + 14,400 + 4800 = 49,800 CTUs/sec. This is less than the encoders throughput of 61,200 CTUs/sec so this set of resolutions can be encoded.

The NGCodec encoder is intended primarily as a real-time video encoder. It is expected that one or more sources of video input, either from files or from live video streams, are fed into the encoder. The encoder encodes one or more output streams from each input. The custom FFmpeg software supplied with the encoder will detect whether the configuration of outputs fits within the overall processing capability of the encoder as discussed. If it does not, it will gives an error.

XILINX

Chapter 2

Encoding Parameters

The NGCodec encoder can be configured according to your requirements.

Basic Parameters

The following list shows the basic parameters that can be configured in the encoder.

- **Pixel Format:** The supported pixel format for the NGCodec HEVC encoder is YUV 4:2:0 non-interlaced.
- **Width:** For Alveo™ U200 FPGA card, the maximum width supported is 3840 and for U50 FPGA card, the maximum width supported is 1920 and the minimum width is 320. Both width and height should be divisible by 4.
- **Height:** For Alveo U200 FPGA card, the maximum height supported is 2160 and for U50 FPGA card, the maximum height supported is 1080 and the minimum height is 240.
- **Frames Per Second:** The maximum supported fps varies based on the resolution and the number of channels. Maximum input bandwidth should not exceed 1080p120. See Encoder Throughput.
- **Bitrate:** The target number of bits in one second of encoded video. The range of bit rate supported is 100 Kb/s to 20 Mb/s (10 Mb/s for 1920x1080 and lower resolutions and 20 Mb/s above 1920x1080).
- **Fixed QP:** The supported Quantization Parameter (QP) values are from 0 to 51. QP regulates the quality of the bitstream. If this parameter is used it overrides the given bitrate. Usually this parameter is not used, bitrate is used instead.

Advanced Parameter

The following is an additional parameter available to configure the encoder. This parameter should be left at its default value as it is optimized for the best encoding results. Ask for guidance from Xilinx if you need to adjust this parameter for your application.

- **aq-mode:** This feature enables you to turn on/off adaptive quantization at CTU level. By default, it is enabled. Adaptive quantization allows the encoder to more effectively distribute bits where they are needed in each video frame to enhance the overall quality of the encoding.

XILINX

Chapter 3

Using FFmpeg for Encoding

FFmpeg Command Line Options

FFmpeg is an industry standard, open source, widely used utility for handling video. FFmpeg has many capabilities, including encoding and decoding almost all video compression formats, encoding and decoding audio, encapsulating and extracting audio and video from transport streams, and many more.

It is not within the scope of this document to provide a tutorial on the basic usage of FFmpeg. Various tutorials can be found online, for example:

- https://www.ffmpeg.org/documentation.html
- http://howto-pages.org/ffmpeg/

FFmpeg is the primary interface to the NGCodec HEVC hardware encoder. NGCodec supplies an enhanced version of FFmpeg that can communicate with the hardware encoder to encode video files. The following table shows the options used with FFmpeg to configure the various encoder parameters described in Chapter 2: Encoding Parameters.

Table 1: **FFmpeg Configuration Options**

Parameter	Example FFmpeg Command Line Syntax	Standard FFmpeg Command	NGCodec Custom Parameter	Notes
Pixel Format	`-pix_fmt yuv420p`	✓		Yuv 4:2:0 is the only mode supported
Width, Height	`-s:v 3840x2160`	✓		See Chapter 5: Limitations.
Frames per second	`-vf fps=60`	✓		See Chapter 5: Limitations
Fixed QP	`-qp <value>`	✓		Value from 0 to 51. See Chapter 5: Limitations.
Bitrate	`-b:v 1000K`	✓		See Chapter 5: Limitations
AQ mode	`-aq-mode 1 or 0`		✓	Enable or disable adaptive quantization

Example FFmpeg Command Lines

- **Single channel, CBR 1 Mb/s:**

```
./ffmpeg -f rawvideo -s:v 3840x2160 -pix_fmt yuv420p -i inputfile.yuv -
c:v NGC265 -b:v 1M -f rawvideo output.265
```

- **Single channel, VBR, AQ enabled:**

```
./ffmpeg -f rawvideo -s:v 1920x1080 -pix_fmt yuv420p -i sample.YUV -c:v
NGC265 -q 35 -f rawvideo output.265
```

- **Single channel configuration, CBR 1 Mb/s, AQ disabled:**

```
./ffmpeg -f rawvideo -s:v 1920x1080 -pix_fmt yuv420p -i sample.YUV -c:v
NGC265 -b:v 1M -aq-mode 0 -f rawvideo output.265
```

- **Multi-channel using software scalar:**

```
./ffmpeg -i sample.mp4 -c:v NGC265
-vf fps=30 -b:v 3M -f rawvideo ~/output_ch0.265 -c:v NGC265 -s:v
1280x720 -vf fps=30 -b:v 2M -f rawvideo ~/output_ch1.265 -c:v NGC265
-s:v 640x480 -vf fps=30 -b:v 1M -f rawvideo ~/output_ch2.265 -c:v
NGC265 -s:v 480x360 -vf fps=30 -b:v 500K -f rawvideo ~/output_ch3.265
```

Digital Rights Mangement

Subscribe and Run DRM

The HEVC encoder usage is protected and monitored through digital rights management (DRM) provided by Accelize. The DRM IP is part of the encoder binary running on the FPGA. A separate DRM application is provided with the encoder release package.

Use the following steps to subscribe and run DRM:

1. Install DRM from https://tech.accelize.com/documentation/stable/drm_library_installation.html.
2. Create an account in the DRM portal: https://xilinx-ngcodec.accelize.com/.
3. Generate and save an access key (`cred.json`) file from the portal.
4. Subscribe to the HEVC Encoder UHD Dual Density in the DRM portal.
5. Run the DRM application before running the encoder `./drmapp.exe`.

For more information on the Accelize DRM IP, contact the Xilinx support team or Accelize team.

Limitations

1. For HEVC, the total input bandwidth of all the channels or all the processes should not be more than 61,200 CTUs per second (see Encoder Throughput).
2. The maximum bit-rate supported and tested is 20 Mb/s; if more, the encoder might not work properly.

Troubleshooting

If the FFmpeg application fails, perform the following checklist:

1. Check whether the FPGA is connected and in a working state.

```
source /opt/xilinx/xrt/setup.sh
xbutil list
```

2. Load the encoder using `xbutil`.

```
xbutil program -d 0 -p <encoder xclbin>
```

3. If the programming is successful, run FFmpeg and check for any errors in the console log and in the log file generated at `/tmp/newxma.log`. Check whether the yaml file is pointing to the correct DSA, encoder `xclbin` and plugin library (`libnghevc.so`).
4. Capture “xbutil query”, console logs and xma log file (`/tmp/newxma.log`) and share it with us for further investigation.

Appendix A

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator (DocNav) provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open DocNav:

- From the Vivado® IDE, select **Help → Documentation and Tutorials**.
- On Windows, select **Start → All Programs → Xilinx Design Tools → DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In DocNav, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on DocNav, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this guide:

1. https://www.andrew.cmu.edu/user/lshea/2.Tech_PDFs/Mpeg_and-h264_compression.pdf
2. https://www.ffmpeg.org/documentation.html
3. http://howto-pages.org/ffmpeg/
4. https://tech.accelize.com/documentation/stable/drm_library_installation.html

Please Read: Important Legal Notices

Copyright