Application Description

Application: V-Nova XDE (HEVC) Single FPGA 4KP60 real-time Encoder

Hyperscale video services like social and e-sports networks need to encode and serve vast numbers of streams to their users. The server infrastructure required to satisfy this demand is often the largest cost for these businesses.
V-Nova XDE “Xilinx Dense Encoder” running on Xilinx FPGA enables any service operating in a private/public cloud to radically transform their efficiency, reducing operating costs by up to 4x whilst improving the streaming quality-of-service for their users.
Furthermore, public or private clouds can deploy V-Nova XDE on Xilinx® AlveoTM Data Center accelerator cards as an ultra-dense encoding solution to offer significant quality and cost benefits to their video delivery customers.

V-Nova XDE employs the MPEG-5 Part 2 Low-complexity enhancement video coding standard, a unique video encoding approach that significantly enhances the quality and throughput of any standard codec such as AVC/H.264, HEVC, VP9 and AV1. When combined with a Xilinx FPGA, V-Nova’s LCEVC-enhanced encoder provides the highest density encoding solution in the market enabling use cases such as live 4Kp60 encoding on a single card. Playback is supported on a broad range of devices since MPEG-5 LCEVC leverages the hardware decoding capabilities of the underlying codec already present.

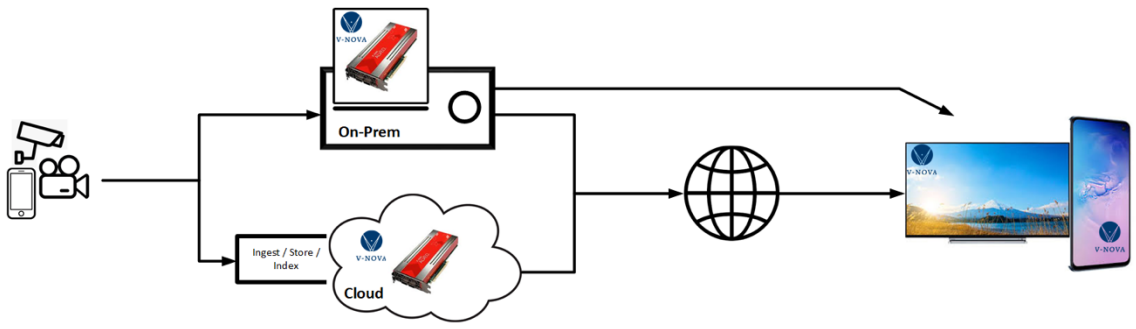

Figure 1 : End–to–End system example

How does it work?

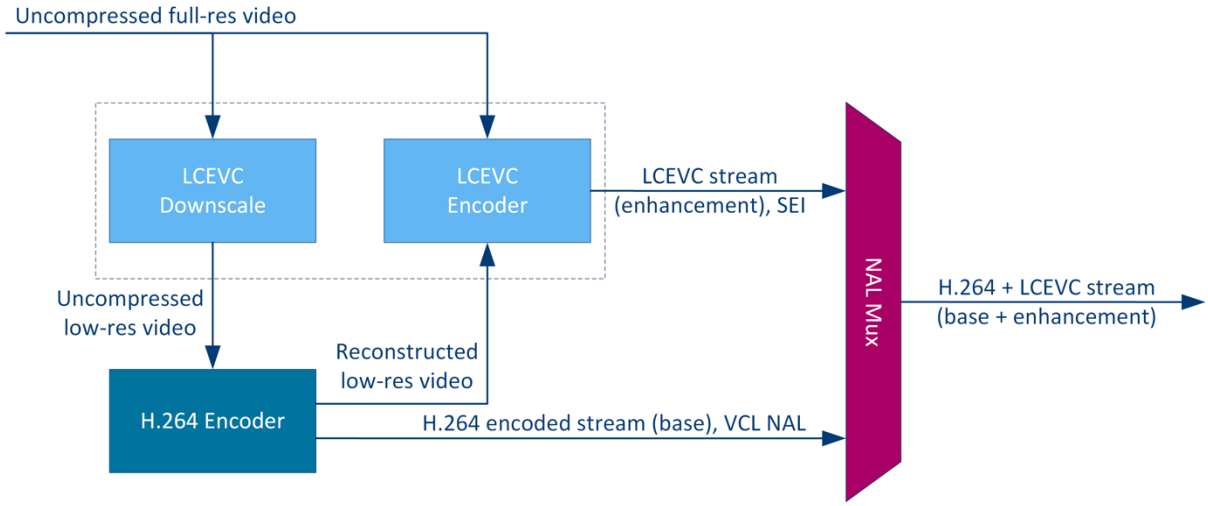

Figure 2: V-Nova LCEVC Encoder pipeline

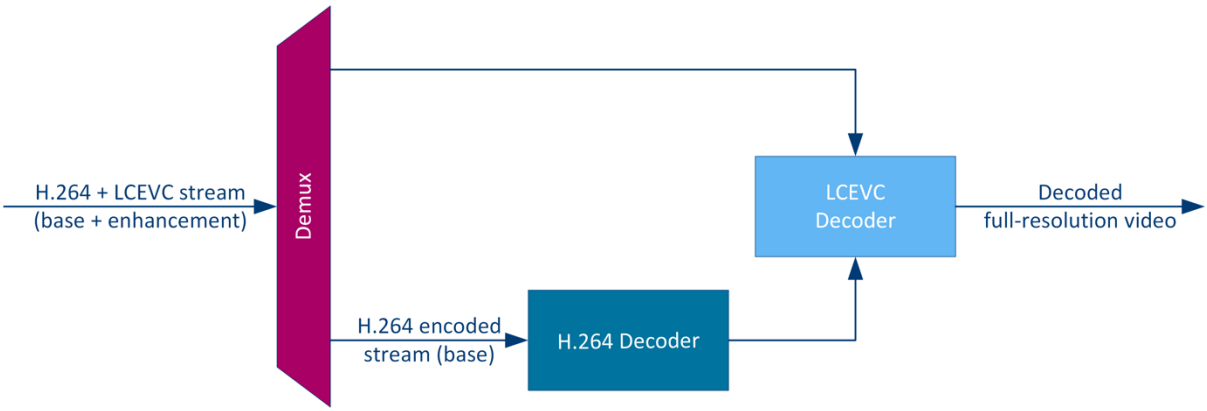

Figure 3: V-Nova LCEVC Decoder pipeline

The base encoder (e.g HEVC) is applied to a lower resolution version of the input video signal while LCEVC takes care of the additional details to reconstruct a full resolution picture.

On the consumer end, the base decoder (HEVC) can still provide a lower resolution video irrespective of whether the decoder has been LCEVC-enabeld or not. The LCEVC decoder Plugin reconstructs the additional details to provide a full resolution decoded picture, thereby providing backwards compatibility with existing infrastructure.

Demo Container Feature Set

- Live 4kp60 or multiple HD streams per card
- HEVC base codec (V-Nova XDE)
- V-Nova LCEVC codec performance
- Basic Video Quality features
- FFmpeg integration

Full Container Feature Set

- Live 4kp60 or multiple HD streams per card
- HEVC base codec (V-Nova XDE)
- ABR Ladder processing
- MPEG-5 LCEVC standard compliant
- V-Nova XSA (“Xilinx Software Accelerator) included (H.264, HEVC, VP9)
- Enhanced Video Quality features
- FFmpeg integration

Key Target Markets & Applications

- HD to UHD migration (Backward compatible base)
- OTT, Films/VOD
- Live eSport streaming
- Video-Conferencing applications
- UHD / 4K cameras / VR
- High-Res DSNG feeds
- Security applications
- UHD, 4K, HD Asset encoding
- Playback servers
- Proxy editing feeds
- In-car entertainment systems
- UHD enabler over ADSL (IPTV Headends)

Supported platforms

Supported Xilinx Cards: Alveo U200 (xdma-201830.2)
Supported OS: Centos 7.x, Ubuntu 16.04 / 18.04

Supported Cloud Providers

Nimbix (TBC)
AWS (TBC)

Demo Version information

V-Nova XDE V2.0

Contact information

For more information or enquiries please contact us using the following email addresses below;

info@v-nova.com
sales@v-nova.com

Key Labels / Tag words

V-Nova, LCEVC, MPEG-5, Perseus, P+, Mpeg, Video, XSA, XDE, OTT, Transcode, Image Processing, Jpeg, h264.

Quick Start

This Docker image contains the environment and application to run the V-Nova XDE NGC HEVC FPGA Encoder.

Minimum system requirements

-16GB RAM (32GB+ if input is placed in ramdisk)

-Intel Core i7-9700 or similar high-performance CPU

-CentOS 7.x

-Docker

-SSD with read speed of > 1GB/s for 4Kp60 encoding

-Xilinx XRT xrt.x86_64 2.2.2158-1

-Xilinx XDMA xilinx-u200-xdma.x86_64 201830.2-2580015

Important host machine considerations:

-The performance of the file-based encoder is highly dependent on the host disk read performance. Ensure the input file is on a ram-disk or SSD with a read speed of > 1GB/s

-Performing other CPU/Disk heavy tasks while encoding will impact performance

-Consider setting CPU scaling governor to performance mode for maximum encoder performance

Installation

Sign-In / create an account on docker hub

docker pull hubxilinx/vnova_pplus_alveo_u200:ppxde-demo

Licensing

A license can be obtained from https://fpga.v-nova.com/

Register and obtain a cred.json identifier. The path to this file must be passed to the encoder run command (see below).

Subscribe to a pricing plan of your choice

Running the encoder

Retrieve the xocl & xclmgmt device handles peculiar to your machine

 # sudo xbutil scan

Example output: INFO: Found total 1 card(s), 1 are usable

~~~~~~

System Configuration

Sysname: Linux

Release: 3.10.0-957.el7.x86_64

Version: #1 SMP Thu Nov 8 23:39:32 UTC 2018

Machine: x86_64

Glibc: 2.17

Distribution: CentOS Linux 7 (Core)

Now: Fri Feb 14 10:15:21 2020

~~~~~~

XRT

Version: 2.2.2158

Git Hash: ab0d3e66d0244d65b520f1abf15446739d46acdb

Git Branch: 2019.1

Build Date: 2019-05-24 18:41:46

~~~~~~

[0]mgmt:[20:00.0]:0x5000:0xe000:[xclmgmt:2.2.2158,ab0d3e66d0244d65b520f1abf15446739d46acdb:512] <---------**** **512** ****

[0]user:[20:00.1]:0x5001:0xe000:[xocl:2.2.2158,ab0d3e66d0244d65b520f1abf15446739d46acdb:129] <------------**** **129** ****
~~~~~~

Encoding command line example

 # docker run -it --rm –name demo --device=/dev/xclmgmt512:/dev/xclmgmt512 --device=/dev/dri/renderD129:/dev/dri/renderD129 -v /host/path/to/assets:/io:Z -v /host/path/to/cred.json:/vnova_pplus/cred.json:Z docker.io/hubxilinx/vnova_pplus_alveo_u200:ppxde-demo -y -s 3840x2160 -pix_fmt yuv420p -vcodec rawvideo -stream_loop 3 -r 60 -i /samples/sample_3840x2160_8bit_P420.yuv -c:v pplusenc_hevc -base_encoder ngcodec_sdx_hevc -eil_params "accel_type=xilinx_xma;encoding_transform_type=dd;temporal_enabled=0;temporal_use_refresh=0;api_mode=asynchronous;baseEncType=ngcodec;xclbinHasNGC=1;encoding_downsample_luma=area;encoding_upsample=cubic;rate_control_mode=variableratio;rc_bucket_duration_ms=2000;encoding_step_width_model=v1;encoding_step_width_min_qp=41;bitrate_max_base_prop=0.8;bitrate_base_prop=0.8;perseus_mode=robust;qp_min=30;xma_config=/vnova_pplus/SystemCfg_194.yaml" -b:v 8000k "/io/output.ts"

-This will encode the 4K video input located at **/samples/sample_3840x2160_8bit_P420.yuv** to generate an output bitstream located at **/host/path/to/assets/output.ts**

-The above command line encodes at a bitrate of 8mbps (8000kbps) @ 60 frames per second.

Note: /host/path/to/assets/ is intended to represent the location of a folder to store the input or/and output videos/bitstreams. It is recommended that this path be /dev/shm (i.e ramdisk)

Relevant encoding parameters (Demo only)

Docker specific:

docker run → standard command to run a Docker container
-it --rm → Run Docker in interactive mode and delete container after execution has completed
--device → Map a system device into the Docker container. These mappings give the Docker container access to the FPGA hardware.
-v host_path:container_path → Map a volume from the host filesystem into the Docker container (necessary for file input/output and providing the cred.json identifier for licensing)

FFmpeg Specific:

-y → Overwrite output files without asking
-s WxH → Resolution of input
-pix_fmt → Pixel format of input file (yuv420p)
-c:v codec → codec used for encoding (For this demo, *pplusenc_hevc*, which specifies LCEVC with a HEVC base codec)
-base_encoder codec → codec used for the base (*ngcodec_sdx_hevc* specifies the HEVC encoder from Xilinx-NGCodec running on the FPGA)

-stream_loop ➔ Specifies the number of times to loop over the input file during encodes, set to -1 for infinite

-i ➔ location of input file to be encoded

-b:v ➔ Specifies the encoding bit rate, e.g 4500k will encode at 4.5mbps

Installing the decoder

NOTE: The decoder is based on Microsoft UWP (Universal Windows Player) and as such will **ONLY** run on a Windows 10 PC

1. Download and install the HEVC video extension app from the Microsoft store https://www.microsoft.com/en-us/store/p/hevc-video-extension/9n4wgh0z6vhq
2. Download and install the V-Nova decoder app from https://vnovaltd.sharepoint.com/:f:/s/CustomerSuccess/CSMPublic/EgV4kmFWu7VOjltA7Eg-iPQBx4jOEyZ_Dl9IGXf6aCWf6Q?e=SLlnhb
    a. Password: Vnova@UWP
3. Install the V-Nova decoder according to the instructions below
    a. Make sure you are in the Windows “developer mode” (in Settings search for “developer mode”, it will show “Use developer features”, then select “Developer mode”);
    b. Unzip the “**VNova_VideoPlayback_3.3.1.0.zip**” archive in a directory of your choice
    c. In the “V**ideoPlayback_3.3.1.0_Test**” directory right-click on the ‘**Add-AppDevPackage.ps1’** file and select "Run with PowerShell"