HEVC Decoder IP Core Reference Design

Introduction
HEVC/H.265 is the latest cutting-edge video compression standard, which is positioned as the successor to the popular H.264/MPEG-4 AVC standard. It promises a 50% bit rate reduction at the same level of video quality compared to H.264. HEVC is predicted to become the industry’s preferred video coding standard for “Ultra High Definition” (UHD/4K) video applications.

With the growing popularity of HEVC and UHD video, many companies have started developing HEVC decoder ASICs and IP cores. Due to the increased computational complexity of HEVC and the inherent high bandwidth requirement associated with UHD/4K video, it has become very challenging to implement real-time Ultra-HD HEVC decoders on hardware. In such a scenario, the ability to have a real-time 4K HEVC decoder IP core, which can be easily implemented and deployed on a standard FPGA development platform, becomes a major advantage for system developers as well as academic researchers.

This document provides a reference design for the real-time 4K (3840x2160p, 30Hz) HEVC decoder IP core developed by ParaQum Technologies. It shows how to implement a complete end to end system for evaluation of the IP core on a standard Xilinx ZC706 Zynq7000 SoC development platform.

Optionally, with a few modifications, the same system has also been implemented on a high performance Xilinx VC707 Virtex7 development platform (which offers more hardware resources for user system integration).

Hardware Requirements
The following hardware is required to implement the reference design as described here. This document will focus on deploying a simple stand-alone design with minimum hardware components. Additional components required for full evaluation of capability will be described later.

Required
- Xilinx ZC706 Development Board
- PC with Ethernet cable
- HD/UHD Display with high speed HDMI cable

Optional
- Avnet DVI/DisplayPort FMC or TED Inrevium DisplayPort FMC – Required for 4K display output
- UHD display with DisplayPort input or HDMI input with 4K Active DisplayPort to HDMI converter
Reference Design Block Diagram

Figure 1 shows the reference design block diagram, implemented on a standalone ZC706 Development Platform.

The whole system is implemented on the ZC706 development board. The HEVC decoder IP core resides in the Programmable Logic (PL) side of the Zynq SoC. The Zynq Processing Subsystem (PS), handles the header parsing. The external input comes in the form of an encoded HEVC bitstream, wrapped in Ethernet packets through the built in Ethernet interface of the ZC706 board. The built in Ethernet controller on the PS side handles the Ethernet traffic. The header parsing is done by an application running on an ARM processor core on the PS side.

The decoded video output is sent from the FPGA to the onboard HDMI transmitter chip, and then output through the HDMI interface of the ZC706 board. The on board HDMI transmitter chip does not support the bandwidth required for 4K, so only HD video can be displayed in this configuration. Additional components are needed to get a 4K video output.
Component Description
The primary components of the reference design and their core functionality is explained here.

Ethernet Controller
The built in Ethernet controller on the PS side, handles the Ethernet traffic coming from a host PC or other input source. The encoded HEVC bitstream, is wrapped in Ethernet packets by a host PC and sent through the built in Ethernet interface of the ZC706 board. The Ethernet controller receives and processes the Ethernet packets and writes the data to the PS side external DDR3 memory to be read by the header parser.

ARM Application
The ARM application handles the header parsing and transferring of data from the PS side DDR3 memory to the Decoder IP Core. This is implemented on one of the ARM cores of the Zynq processing subsystem. It is written as a processor application in C code and run on the ARM core using Xilinx SDK. The HEVC standard defines several activities that have to be carried out once per frame. These are defined in a header in the encoded bitstream. This application reads the received data from the PS side DDR3 memory, does the initial header parsing and passes it on to the HEVC Decoder. It outputs an elementary HEVC syntax stream to the HEVC Decoder IP core through an AXI interface connecting the PS to PL fabric.

HEVC Decoder IP Core
The HEVC decoder IP core is implemented on the Programmable Logic side of the Zynq SoC. It takes the input in the form of an elementary HEVC syntax stream through the AXI interface from the PS side. A high level functional block diagram of the HEVC Decoder IP is shown in Figure 2.

The HEVC Decoder IP has two AXI external memory interfaces for reference pixel buffer and display buffer. The reference pixel buffer is used to store the last decoded frames for future reference (inter prediction) as specified by the HEVC standard. The display buffer is used to store the decoded frames until they are displayed. The Decoder IP provides two separate AXI external memory interfaces so that the user has the ability to either use a single external memory (using an AXI interconnect IP) or use two separate external memories for the two buffers (as done in this reference design). The Decoder IP contains a display buffer handler, which can write and can also read the decoded frames to and from the display buffer. Once the decoded frames are written to the display buffer, they can be used for displaying by any user application by reading from the display buffer memory. In addition, the display buffer handler itself can read back the decoded frames from the display buffer and send it out for display through the “Video Out” interface of the Decoder IP.
In this reference design, the reference frame buffer is implemented using the external DDR3 memory on the PL side. The Xilinx Memory Interface Generator (MIG) is used to connect the AXI memory mapped interface to the external DDR3 SODIMM.

The display buffer is implemented on the external DDR3 component memory on the PS side. The built in memory controller along with the AXI DMA path in the PS is used to connect the AXI memory mapped interface to the external DDR3 memory on the PS side. This makes it possible for a user application running on the ARM processor(s) to read the decoded frames from the display buffer.

In this reference design, the display buffer handler itself both writes and reads to/from the display buffer. The display buffer handler reads the decoded frames from the display buffer through the AXI DMA interface and sends it out for displaying though the “Video out” interface of the Decoder IP.

**Memory Interface Generator (MIG) Core**

This is a memory controller IP core generated from the Xilinx Memory Interface Generator. It has an AXI user interface and uses the standard ZC706 board specific parameters. Refer to Xilinx documentation on MIG for more information.
HDMI Transmitter Core

The HDMI tx core is used to take the decoded raw video input from the “Video Out” interface of the Decoder IP and convert it to a digital video signal suitable for HDMI transmitter chips. The output of the HDMI tx core consists of a standard bit-parallel digital video interface (along with synchronization signals) suitable for HDMI transmitter chips such as the ADV7511 on the ZC706 board. The output of the HDMI tx core is assigned to the IO pins of the Zynq PL which are connected to the ADV7511 HDMI transmitter chip on the ZC706 board.
Running the Reference Design on the ZC706 Board

This section describes the software requirements and step by step process for running this reference design on the ZC706 development board.

Figure 3 shows the basic components and their connectivity for running the reference design.

![Figure 3 – Required components and their connectivity](image)

This reference design requires a host PC running an Ethernet packet transmission application software. The Ethernet packet transmission application software is written in C code. The host PC is connected to the ZC706 board through the Ethernet interface.

Steps to program the FPGA and run the application software of the Zynq and host PC

1. Connect the Ethernet interface of the ZC706 board to the host PC using an Ethernet cable. Also connect the HDMI interface of the ZC706 to an HD display using a high speed HDMI cable.
2. Launch the Ethernet packet transmission application on the Host PC side once the Ethernet connection is established. This application will act as a server which responds to Zynq’s request to start sending of Ethernet packet stream.
3. Download the bit stream on to the FPGA via JTAG interface.
4. Run the header parsing application on one of the ARM processor cores in the PS side. The decoding operation will commence after that.
An HD display with an HDMI input is required to view the video output. It is connected to the HDMI interface of the ZC706 board. Note that the onboard ADV7511 HDMI tx chip only supports up to HD video resolutions. Therefore it is not possible to view a 4K video output. Only an HD video output can be viewed using this configuration. When decoding 4K video streams, the HDMI tx core can be configured (using a virtual input) to down-sample the decoded 4K frames to HD resolution just before displaying (effectively dropping 3 out of 4 pixels), thus demonstrating the ability to decode 4K video streams.

Displaying the native 4K video output requires the implementation of a DisplayPort interface using external components (FMC and converters). It will be described in Annex 1.

Figure 4 shows a snapshot of the demonstration of this reference design in operation.

![Figure 4 – Demonstration of the Reference Design in Operation](image)
Resource Utilization

The resource utilization after implementing this reference design on the ZC706 board is shown below.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Utilization</th>
<th>Available</th>
<th>Utilization %</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>950/6</td>
<td>437200</td>
<td>21.75</td>
</tr>
<tr>
<td>LUT</td>
<td>17/8105</td>
<td>218600</td>
<td>81.75</td>
</tr>
<tr>
<td>Memory LUT</td>
<td>62/38</td>
<td>7/0400</td>
<td>8.55</td>
</tr>
<tr>
<td>I/O</td>
<td>3/4/6</td>
<td>362</td>
<td>40.33</td>
</tr>
<tr>
<td>BRAM</td>
<td>273.5</td>
<td>545</td>
<td>50.18</td>
</tr>
<tr>
<td>DSP48</td>
<td>37</td>
<td>900</td>
<td>4.11</td>
</tr>
<tr>
<td>BUF0</td>
<td>7</td>
<td>32</td>
<td>21.68</td>
</tr>
<tr>
<td>MCM0</td>
<td>2</td>
<td>8</td>
<td>25.00</td>
</tr>
<tr>
<td>PLL</td>
<td>1</td>
<td>8</td>
<td>12.50</td>
</tr>
</tbody>
</table>

Contact Information

For more information about the product, please contact us via email hevc@paraqum.com

Postal address:

Paraqum Technologies (Pvt) Ltd.,

106, 1st Floor, Bernards' Business Park, Dutugemunu Street, Kohuwala 10350, Sri Lanka.

Hot line: +9411-5925133, +9411-2099700
Annex 1 – Implementing a DisplayPort Interface for 4K video output

As mentioned above, due to the limitations of the ADV7511 HDMI transmitter on board the ZC706, it is not possible to get a 4K video output from the built in HDMI interface. In order to get a 4K video output, the most suitable solution is to implement a DisplayPort output interface. This section describes the implementation of a 4K capable DisplayPort output interface along with the HEVC Decoder IP core reference design on the ZC706 development board.

The following additional components are required to implement a DisplayPort output interface.

- Avnet DVI/DisplayPort FMC module (used in this reference design) OR Tokyo Electron Device Limited DisplayPort FMC module (TB-FMCH-DP2)
- Xilinx DisplayPort IP core v4.2 (Hardware evaluation license should be acquired)
- 4K display with DisplayPort input OR 4K display with HDMI input along with 4K DisplayPort to HDMI active converter

The block diagram of the Reference Design along with the DisplayPort output interface is shown in Figure A1.

Figure A1 – Reference design Block Diagram with DisplayPort output
The DisplayPort transmitter design is an adapted version of the Xilinx application note “DisplayPort Transmit Reference Design” XAPP1178. The reference design in XAPP1178 is modified appropriately to work on the ZC706 board with the Avnet DisplayPort FMC module. The DisplayPort Source Policy Maker (SPM) and Link Policy Maker (LPM) applications are implemented on the other ARM core on the Zynq PS for this implementation. This application handles the processes of configuring the DisplayPort IP core, detecting a DisplayPort sink device, establishing and maintaining the link and setting main stream video parameters. Once the link is setup and properly configured, the output video stream can be sent through the DisplayPort interface.

The DisplayPort Config+tx core contains all the AXI peripherals and video clock generator similar to that shown in XAPP1178 (without the video pattern generator). It is connected to the LPM+SPM application running on the ARM core through an AXI interface connecting the PS to PL fabric. There is an interactive UART terminal (as shown in XAPP1178) from which the user can configure and read parameters of the DP transmitter subsystem. The link can be configured through the UART interface.

Refer to Xilinx application note XAPP1178 and other Xilinx documentation on DisplayPort IP core for more information.