# A Realtime 1080P30 H.264 Encoder System on a Zynq Device

## Introduction

The Zynq all programmable System On a Chip is a recently introduced device from Xilinx which incorporates two ARM A9 CPU cores, I/O peripherals, memory controller, and programmable logic. This paper describes the implementation of a 1080P30 realtime H.264 encoder system on the device.

## System Overview

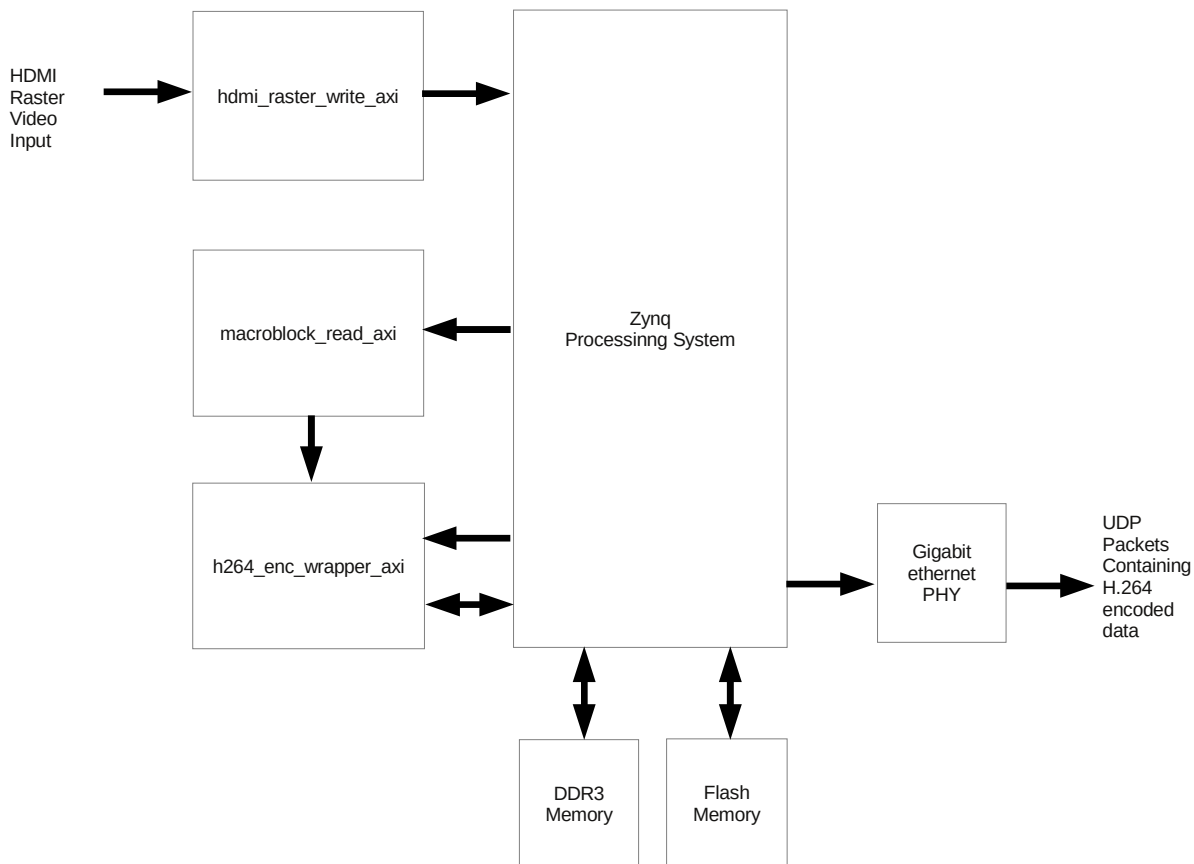Figure 1 shows the block diagram of the system.



**Figure 1 : System block diagram**

An HDMI 1080P60 raster video source provides the input video for the encoder. Every second frame of raster video is written to a frame buffer in the DDR3 memory to achieve the 2:1 frame rate conversion. This step is performed as 1080P60 is a standard HDMI video resolution and frame rate while 1080P30 is not.

16x16 macroblocks are then read from the frame buffer and fed into the H.264 encoder. The encoded output is encapsulated into UDP Ethernet packets and streamed out through the Ethernet interface for decoding by a remote device running the VLC multimedia software.

The Zedboard with its ZC7020 device was selected as the platform for this exercise as it is a cost effective evaluation platform with all the required features for the system. The HDMI input interface is provided by the AES-FMC-IMAGEON-G FMC card from Avnet.

**System Implementation**

The system was implemented using the Vivado design software from Xilinx. This software provides an integrated environment to build a system using IP blocks.
Apart from the standard IP blocks in Vivado, the system also used IP blocks created by Ocean Logic for the video input raster to macroblock conversion, and H.264 encoding.

The system uses the High-Performance as well as the General Purpose AXI buss. The High-Performance buss configured for 64-bit data is used for connecting the video input logic and also the H.264 encoder to the DDR3 memory controller via the High-Performance slave ports of the Processing System.
The 32-bit General Purpose AXI buss is used by the embedded processor to configure the IP blocks as well as by the DMA engine to collect the encoded H.264 data.

The Zedboard provides 512Mbytes of 32-bit DDR3 memory clocked at 533MHz which is controlled by the hard-IP DDR3 controller on the Zynq device. This memory is shared by the embedded processor (program and data storage), the H.264 encoder (6.3Mbytes reference frame memory), and the video input raster to macroblock conversion logic (3.13Mbytes frame buffer).

While the amount of memory used by the H.264 encoder is only a small proportion of the available memory, the bandwidth required is significant. The H.264 encoder requires 460 64-bit read/write accesses every 1024 encoder clock cycles. The encoder is in this case is clocked at 126MHz to achieve the encoding throughput for 1080P30 video so the 1024 encoder clocks equates to 8.13us. While there is theoretically 4.26Gbytes/s (4327 64-bit accesses in 8.13us) of available bandwidth, it is important to ensure that the embedded processor does not lock out the H.264 encoder's access to the DDR3 memory. In addition, the raster to macroblock conversion logic requires 384 64-bit accesses every 8.13us. There are also DDR3 memory page activations and deactivation cycles which will erode the available bandwidth.

With the current system there are no issues with available DDR3 memory bandwidth. If the available bandwidth in a system were to be exceeded, a separate soft–IP memory controller could be used to support the H.264 encoder and the raster to macroblock logic.
Table 1 shows the resource utilization of the system.

| Resource | Used | Available |
|---|---|---|
| Slice LUTs | 35362 | 53200 |
| Slice Registers | 22754 | 106400 |
| Memory | 138 | 140 |
| DSP | 11 | 220 |
| IO | 31 | 202 |
| Clocking | 7 | 32 |
| Slice Logic Utilization | 11195 | 13300 |

**Table 1 : ZC7020 resource utilization**

As mentioned above, the encoder clock frequency is 126MHz. This is not achievable in the ZC7020-1 device on the Zedboard for worst case conditions. A ZC7020-2 device will be required to achieve the encoder clock frequency for worst case conditions.

**System Software**

The embedded processor on the Zynq configures the H.264 encoder and also services the DMA engine interrupts for transferring the encoded data packets from the H.264 encoder to the Ethernet MAC. The embedded processor does not implement any of the H.264 encoding algorithm. The system software flow diagram is shown in Figure 2.

After configuring the peripherals on the Zynq device and the HDMI interface, the status of the HDMI is read to see of a 1080P60 source is present. The operation is aborted if it is not present.

The flash memory on the Zedboard is then checked to see if the CRC is valid for the set of encoding parameters. If it is not, the serial terminal connected to the USB UART can be used to enter a set of encoding parameters. Once entered, the parameters can be saved to the flash memory with a valid CRC. If SWITCH7 on the Zedboard is set to 1, the encoding parameters can also be input using the same method.

With a valid set of encoding parameters, the H.264 encoder is configured and then encoding operation started. When the buffer full interrupt from the encoder is received, the data is transferred from the buffer via DMA to the Ethernet MAC on the Zynq processing system.
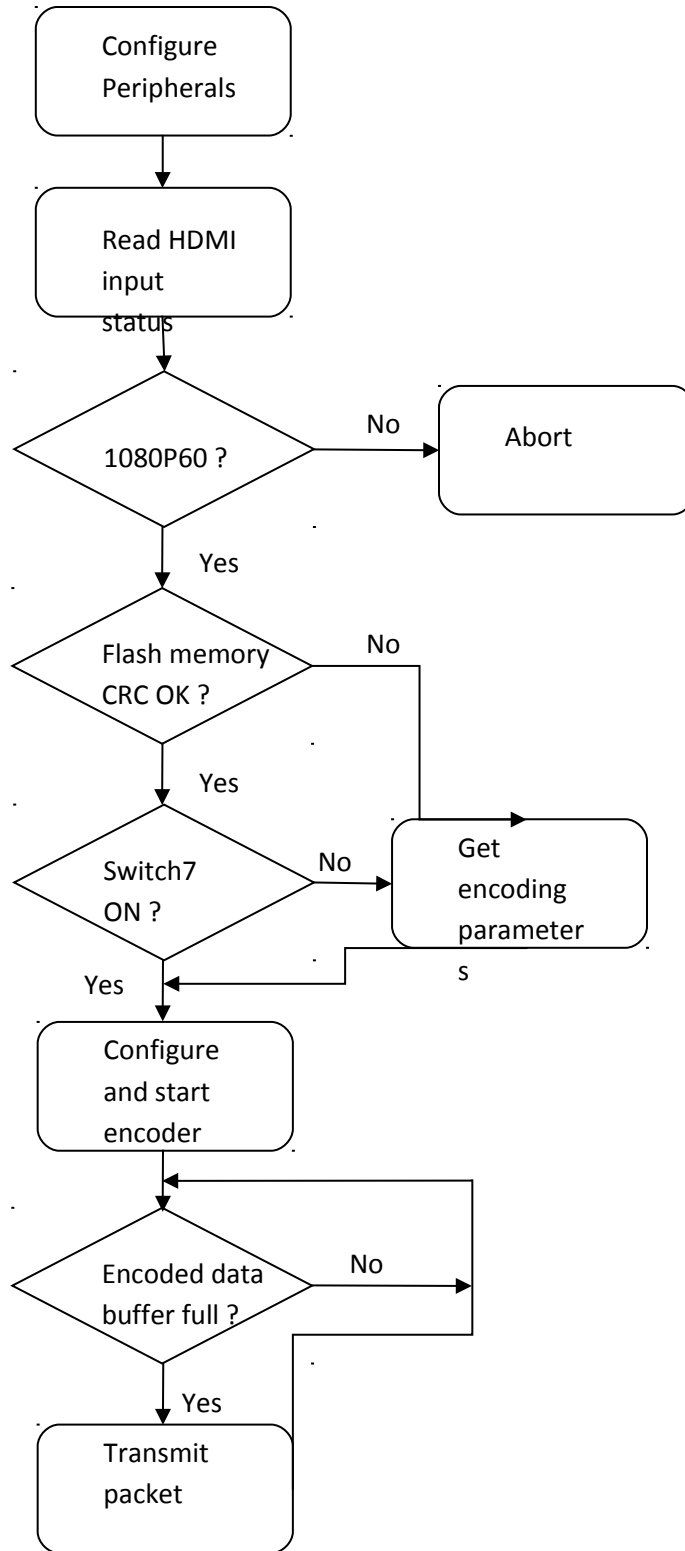
```
        ┌──────────────┐
        │  Configure   │
        │  Peripherals │
        └──────┬───────┘
               │
               ▼
        ┌──────────────┐
        │  Read HDMI   │
        │  input       │
        │  status      │
        └──────┬───────┘
               │
               ▼
           ◇─────────◇           No     ┌──────────┐
          < 1080P60 ? >─────────────────│  Abort   │
           ◇─────────◇                  └──────────┘
               │ Yes
               ▼
           ◇───────────◇         No
          < Flash memory >───────────────┐
          < CRC OK ?     >                │
           ◇───────────◇                 │
               │ Yes                      │
               ▼                          ▼
           ◇─────────◇      No     ┌──────────────┐
          < Switch7   >────────────│  Get         │
          < ON ?      >            │  encoding    │
           ◇─────────◇            │  parameter   │
               │ Yes              │  s           │
               ◄──────────────────┘
               ▼
        ┌──────────────┐
        │  Configure   │
        │  and start   │◄──────────────┐
        │  encoder     │               │
        └──────┬───────┘               │
               ▼                        │
           ◇─────────◇      No          │
          < Encoded data >──────────────┘
          < buffer full ? >
           ◇─────────◇
               │ Yes
               ▼
        ┌──────────────┐
        │  Transmit    │
        │  packet      │
        └──────────────┘
```

**Figure 2 : Software flow diagram**

This type of software is referred to by Xilinx as a "Bare Metal" application as there is no underlying operation system. It was developed in the Xilinx SDK environment using C, with drivers and examples from Xilinx and also the AES-FMC-IMAGEON-G FMC card reference design.

**Summary**

The Zynq device incorporates a lot of built-in functionally coupled with the programmable fabric which makes it a very capable platform for implementing a large variety of embedded systems. The system shown here only uses a fraction of what the device is capable of. The Vivado design software provides an environment to quickly develop a system such as this.