

# A Network-on-Chip Based H.264 Video Decoder Prototype Implemented on FPGAs

Ian J. Barge and Cristinel Ababei  
Dept. of Electrical and Computer Engineering  
Marquette University, Milwaukee WI, USA  
Email: {ian.barge,cristinel.ababei}@marquette.edu

## Abstract

We present a field programmable gate array (FPGA) based implementation of the H.264 video decoder algorithm. The novelty of our design is that the communication between the decoder modules is done using a network-on-chip (NoC). This makes our design scalable and easily integrated within larger future NoC based systems, where the same hardware platform can host other algorithms such as compression, filtering, etc. Our primary objective is to study the achievable performance with a NoC based H.264 decoder solution. The design process involves primarily three main steps.

First, the H.264 algorithm is split into eight different partitions, which are implemented as individual processing elements (PEs). These processing elements are attached to the routers of the regular mesh NoC and include: network abstraction layer (NAL) parser and entropy decoder, frame buffer and integer motion, inverse quantization inverse transform, intra prediction, luma sub-pixel motion, chroma sub-pixel motion, deblocking filter, and display driver. These PEs are described in VHDL with the first two being executed on Nios II softcores. The network-on-chip was generated with the Connect tool from Carnegie Mellon University and integrated within the top level design entity.

Second, we specify the location of each of the PEs inside the regular mesh NoC. Because we use eight PEs, the NoC architecture needs to be a 3x3 regular mesh topology. When we specify the location of the PEs inside the mesh topology (i.e., specify the router to which a particular PE is attached), we effectively solve what is called the NoC mapping problem. To do that, we use manual mapping, which is done intelligently based on information about the internal structure of the decoding algorithm. This helps to reduce the number of routers that packets must travel through the network.

Finally, the entire project is synthesized, placed, and routed with Quartus Prime Standard Edition 16.1 tool. The final design is tested and verified on the DE4 development board, which uses Altera's Stratix IV GX FPGA chip. The performance of the implementation at the time of the submission is that to decode 100 frames takes 33 seconds for a frame size of 192x144 pixels and to decode 100 frames takes 56 seconds for a resolution of 320x240 pixels per frame.

Documentation and source codes of the entire project will be released to the public domain. We hope that this will enable other researchers to easily replicate and compare results to ours and that it will encourage and facilitate further research in the areas of image processing, computer vision, and advanced VHDL design and FPGAs.