

# FPGA Co-Processing Architectures for Video Compression

Alex Soohoo  
Altera Corporation  
101 Innovation Drive  
San Jose, CA 95054, USA  
(408) 544-8063  
[asooohoo@altera.com](mailto:asooohoo@altera.com)

## Overview

The push to roll out high definition video enabled video and imaging equipment is creating numerous challenges for video system architects. The increased image resolution brings with it higher performance requirements for basic video data path processing and next-generation compression standards, outstripping that which stand-alone digital signal processors (DSPs) can provide. In addition, the system specifications require designers to support a range of standard and custom video interfaces and peripherals usually not supported by off-the-shelf DSPs. While it is possible to go the route of application specific integrated circuits (ASICs) or use application specific standard products (ASSPs), these can be difficult and expensive alternatives that might require a compromised feature set. Furthermore, these choices can hasten a short product life cycle and force yet another system redesign to meet varied and quickly changing market requirements.

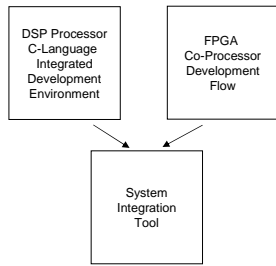
Field programmable gate arrays (FPGAs) are an option that can bridge the flexibility gap in these types of designs. Additionally, with the increasing number of embedded hard multipliers and high memory bandwidth, the latest generation of FPGAs can enable customized designs for video systems while offering a manifold performance improvement over the fastest available stand-alone DSPs. Designers now have the ability with state-of-the-art FPGA co-processor design flows to implement high-performance DSP video and image

processing applications. This new generation of tools facilitates the design of a system architecture that is more scalable and powerful than traditional DSP-only designs while at the same time taking advantage of the price and performance benefits of FPGAs.

## Design Flow

The emergence of these new DSP design flows has made the combined DSP processor and FPGA co-processor architecture an attractive option for video and image processing systems. What has made this possible is the co-processor flow that merges the traditional C-language based development environments for programmable DSPs and hardware description language (HDL) tools for FPGAs with powerful system integration capabilities (see Fig. 1). Through clever system partitioning, designers now have the ability to leverage a legacy code base for DSPs and offload the most computationally intensive blocks of an algorithm to an FPGA to create systems optimized for both price/performance and time-to-market.

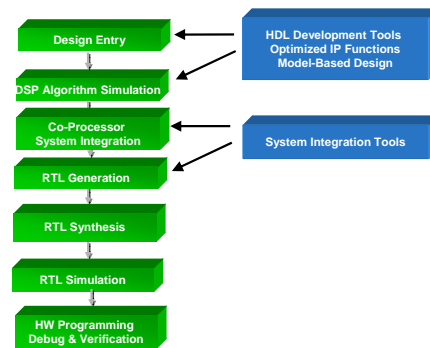
**Figure 1: Combined DSP Design Flow**



Software development environments for DSPs are quite mature, having been refined over many years to address the most common design bottlenecks. On the other hand, there are many options for designing and creating FPGA co-processors. The design of DSP systems with FPGAs can utilize both high-level algorithm and hardware description language (HDL) development tools as seen in Figure 2. The most straightforward approach is to create an entire design from scratch, writing custom DSP functions in HDL and then using standard FPGA design software. While it is possible to develop high-performance, optimized designs, it can be a time-consuming and labor intensive effort. FPGA suppliers and third-party vendors now offer highly optimized, parameterizable, off-the-shelf intellectual property (IP), typically the most common video and image processing functions and key video compression algorithm blocks. These IP cores with well defined high-speed interface wrappers can be quickly integrated into a system design enabling shorter design cycles and an accelerated time-to-market.

Model-based design environments such as The Mathworks Simulink allow designers to develop, simulate and verify a DSP processing data path for an FPGA co-processor. Models can be built using a mix of proprietary and off-the-shelf DSP building blocks. FPGA design software can integrate this environment combining its capabilities with standard FPGA HDL synthesis, simulation and customized development tools.

**Figure 2: FPGA Co-Processor Flow**

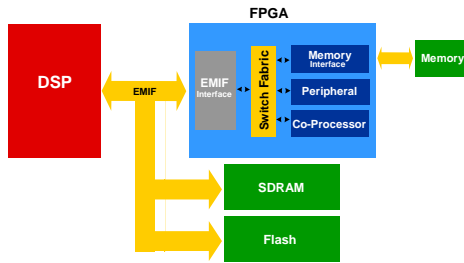


Finally, new system integration tools enable rapid development of custom FPGA co-processor solutions and the ability to leverage existing solutions to add new capabilities and improve system performance. By automating the integration phase of system components and peripherals, this design software can allow users to focus attention on system-level requirements instead of the mundane, manual task of integrating individual blocks with varying requirements. For example, the job of creating and verifying the interface between an FPGA and a DSP can be complex. The newest system integration tools allow the designer to drop in a FIFO-based IP core and interface to an external processor without having to manage or consider the specific pin-out details. This can be critically important for a DSP software engineer with limited experience in FPGA design and hardware implementation.

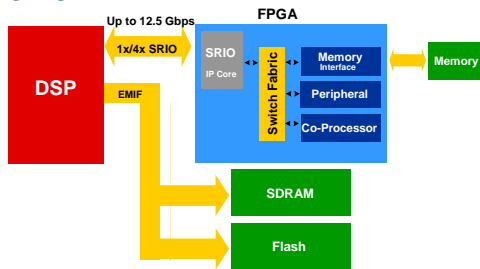
Figure 3 and Figure 4 illustrate example DSP/FPGA co-processing architectures using the Texas Instruments external memory interface (EMIF) and the industry standard Serial RapidIO (SRIO) interface. These architectures can provide memory and peripheral expansion as well as the capability for increased processing performance. The latest generation of system integration tools can automatically generate a seamless bridge between the DSP and the FPGA, making it easier to implement algorithms defined at the block or component level without having to focus

on the detailed device interface mapping.

**Figure 3: Co-Processing With DSP - EMIF**



**Figure 4: Co-Processing With DSP – SRIO**



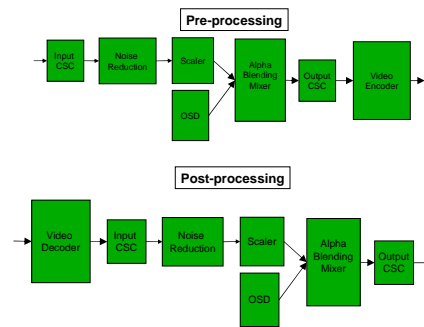
### FPGA Co-Processing for High Performance Video and Image Processing

The main justification for the FPGA co-processor design flow approach is the benefit of enhanced system price/performance. Properly architected designs can offload a DSP processor and execute computationally intensive blocks of a DSP algorithm in a more efficient parallel implementation on an FPGA. This is especially attractive for emerging video and image processing applications where DSP performance requirements are growing at the fastest rates.

Consider the typical video compression (encoding/decoding) processing chains. By taking a closer look at the pre-processing and post-processing halves, it is possible to identify the types of algorithms that might be partitioned between DSP processors and FPGAs to implement a video data path. Multiply-accumulate (MAC) intensive algorithms such as color space conversion (CSC), noise reduction filtering, scaling and

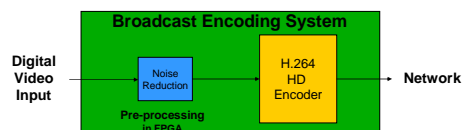
image mixing/blending have little or no control flow component. For that reason, the bulk of the video processing chain should be implemented completely on an FPGA. Video compression algorithms, which have a well defined mix of control and processing operations, might be implemented in a DSP processor or split between a DSP and FPGA depending on the system requirements. The following examples highlight the challenges and rationale for FPGA co-processor architectures.

**Figure 5: Video Processing Chains**



A simple video noise reduction filtering example seen in Figure 6 demonstrates the potential of the FPGA co-processor approach. For video pre-processing in a high definition encoding system, a 7x7 two-dimensional filter kernel is applied to broadcast HDTV 1080p video at 1920x1080 resolution, 30 frames per second, 24 bits per pixel. This operation will require over 9 gig multiply-accumulates per second (GMACs), more performance than the fastest commercially available DSP can offer. The same function can be implemented on a low-cost FPGA with headroom to spare.

**Figure 6: High Definition Encoding System**



For video compression systems, FPGA co-processing architectures can create especially cost effective solutions compared to platforms based on multiple DSPs. High-definition broadcast quality encoding utilizing video codecs MPEG2, MPEG4 and H.264 can be implemented with a single FPGA and DSP.

**Figure 7: H.264 Encoding Co-Processing Partition**

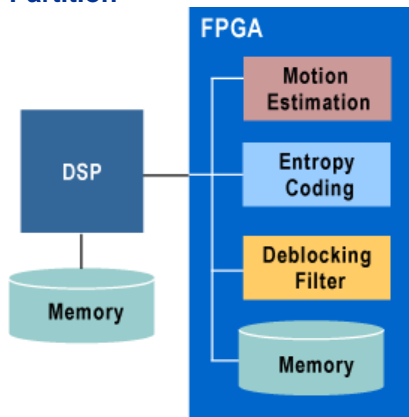


Figure 7 shows an example FPGA co-processor partition of the H.264 encoding standard. The FPGA has absorbed the sections of the algorithm that require the most cycles on the DSP, including the motion estimation block, entropy coding and the deblocking filter. The DSP can execute the remaining parts on the algorithm that are more control flow oriented and better mapped to a C-code implementation. Newer entropy coding techniques such as CAVLC and CABAC do not map well to a typical DSP instruction set and are best realized as hardware accelerated blocks on the FPGA.

In the case of the latest video compression standards, the FPGA co-processor architecture provides a number of advantages. When a standard is relatively new or in flux, many system developers prefer that some degree of flexibility be allocated into the design. When the video compression community converges on the optimal algorithmic approach to the parts of the standard that have some room for enhancement, the hardware architecture can be preserved with only modifications to the

programmable parts of the system. The motion estimation block, in particular, leaves room to incorporate a range of different techniques for motion vector search. From the equipment vendor's point of view, this flexibility allows for customization and differentiation that is not possible when the only choice is a fixed ASSP.

### Conclusion

Performance requirements for video and image processing end equipment is growing as a direct correlation to the new compression standards and higher resolution formats that are being adopted. FPGA co-processor system architectures, complemented by leading-edge design software, allow designers to implement these high performance DSP algorithms in a cost-effective, efficient manner and realize significant benefits.



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
<http://www.altera.com>

Copyright © 2005 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.