

National Conference on Advances in Engineering and Applied Science (NCAEAS) 16<sup>th</sup> February 2017 **In association with International Journal of Scientific Research in Science and Technology** 



# FPGA Implementation of High Performance Entropy Encoder for H.264 Video CODEC

Prajakta Bhagde, Prof. Anil Bavaskar

Department of VLSI Engineering J IT College of Engineering Nagpur, RTMNU University, India

# ABSTRACT

In information theory an entropy encoding is a lossless data compression scheme that is independent of the specific characteristics of the medium. One of the main types of entropy coding creates and assigns a unique prefix-free code to each unique symbol that occurs in the input. These entropy encoders then compress data by replacing each fixedlength input symbol with the corresponding variable-length prefix-free output codeword. The length of each codeword is approximately proportional to the negative logarithm of the probability. Therefore, the most common symbols use the shortest codes. Context-based adaptive variable-length coding (CAVLC) is an important feature of the latest video coding standard H.264/AVC. Context-adaptive variable-length coding (CAVLC) is a form of entropy coding used in H.264/MPEG-4 AVC video encoding. It is an inherently lossless compression technique, like almost all entropy-coders. In H.264/MPEG-4 AVC, it is used to encode residual, zig-zag order, blocks of transform coefficients. It is an alternative to context-based adaptive binary arithmetic coding (CABAC). CAVLC requires considerably less processing to decode than CABAC, although it does not compress the data quite as effectively. CAVLC is supported in all H.264 profiles, unlike CABAC which is not supported in Baseline and Extended profiles. The coding technique using conventional CAVLC based on area efficient design, the second is on low power design architecture will lead to low throughput. In this project, an efficient CAVLC design is proposed. The main concept is the FPGA based pipelining scheme for parallel processing of two 4x4 blocks. When one block is processed by the scanning engine to collect the required symbols, its previous block is handled by the coding engine to translate symbols into bit stream. Our block based pipelined architecture doubles the throughput of CAVLC at high bit rates. Keywords : CAVLC, H.264/AVC, Entropy, MPEG.

# I. INTRODUCTION

Video compression uses modern coding techniques to reduce redundancy in video data. Most video compression algorithms and <u>codecs</u> combine spatial image compression and temporal motion compensation. In practice, most video codecs also use audio compression techniques in parallel to compress the separate, but combined data streams as one package. The majority of video compression algorithms use lossy compression. Uncompressed video requires a very high data rate. Although lossless video compression codecs perform at a compression factor of 5-12, a typical MPEG-4 lossy compression video has a compression factor between 20 and 200. As in all lossy compression, there is a trade-off between video quality, cost of processing the compression and decompression, and system requirements. Some video compression schemes operate on square-shaped typically groups of neighboring pixels, often called macro blocks. These pixel groups or comblocks of pixels are compared from one frame to the next, and the video compression codec sends only the differences within those blocks. Commonly during explosions, flames, flocks of animals, and in some panning shots, the high-frequency detail leads to quality decreases or to increases in the variable bit rate.

#### II. H.264/MPEG-4 AVC

H.264 or MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC) is a block-oriented motion compensation-based video compression standard that is currently one of the most commonly used formats for therecording, compression, and distribution of video content. The intent of the H.264/AVC project was to create a standard capable of providing good video quality at substantially lower bit rates than previous standards (i.e., half or less the bit rate of MPEG-2, H.263, or MPEG-4 Part 2), without increasing the complexity of design so much that it would be impractical or excessively expensive to implement. An additional goal was to provide enough flexibility to allow the standard to be applied to a wide variety of applications on a wide variety of networks and systems, including low and high bit rates, low and high resolution video, broadcast, DVD storage, RTP/IP packet networks, and ITU-T multimedia telephony systems. The H.264 standard can be viewed as a "family of standards" composed of a number of different profiles. A specific decoder decodes at least one, but not necessarily all profiles. The decoder specification describes which profiles can be decoded. H.264 is typically used for lossy compression, although it is also possible to create truly lossless-coded regions within lossy-coded pictures or to support rare use cases for which the entire encoding is lossless.

#### 1. Video compression

- Video compression is about reducing and removing redundant video data so that a digital video file can be effectively sent and stored. The process involves applying an algorithm to the source video to create a compressed file that is ready for transmission or storage.
- To play the compressed file, an inverse algorithm is applied to produce a video that shows virtually the same content as the original source video.
- The time it takes to compress, send, decompress and display a file is called latency.
- A pair of algorithms that works together is called a video codec (encoder/decoder).
- Video codec that implement different standards are normally not compatible with each other.

• For instance, an MPEG-4 Part 2 decoder will not work with an H.264 encoder.

# Alternative description of data requiring less storage and bandwidth.



Figure 1. Uncompressed 1Kb



Figure 2. Compressed 50Mb

#### **III.** Entropy Encoder

An entropy encoding is a lossless data compression scheme that is independent of the specific characteristics of the medium. One of the main types of entropy coding creates and assigns a unique prefix-free code to each unique symbol that occurs in the input. These entropy encoders then compress data by replacing each fixed-length input symbol with the corresponding variable-length prefix-free output codeword. The length of each codeword is approximately proportional to the negative logarithm of the probability. Therefore, the most common symbols use the shortest codes. .According to Shannon's source coding theorem, the optimal code length for a symbol is  $-\log_b P$ , where b is the number of symbols used to make output codes and Pis the probability of the input symbol. Two of the most common entropy encoding techniques are Huffman coding and arithmetic coding.

For the encoding of the video data, two different lossless encoding techniques are available:

- 1) Context Adaptive Variable Length Coding (CAVLC)
- 2) Context Adaptive Binary Arithmetic Coding (CABAC)

# IV. CAVLC

Context-adaptive variable-length coding (CAVLC) is a form of entropy coding used in H.264/MPEG-4 AVC video encoding. It is an inherently lossless compression like almost all entropy-coders. technique, In H.264/MPEG-4 AVC, it is used to encode residual, zigzag order, blocks of transform coefficients. It is an alternative to context-based adaptive binary arithmetic coding (CABAC). CAVLC requires considerably less processing to decode than CABAC, although it does not compress the data quite as effectively. CAVLC is supported in all H.264 profiles, unlike CABAC which is not supported in Baseline and Extended profiles. CAVLC is used to encode residual, zig-zag ordered 4x4 (and 2x2) blocks of transform coefficients. CAVLC is designed to take advantage of several characteristics of quantized 4x4 blocks:

- After prediction, transformation and quantization, blocks are typically sparse (containing mostly zeros).
- The highest non-zero coefficients after zig-zag scan are often sequences of +/- 1. CAVLC signals the number of high-frequency +/-1 coefficients in a compact way.
- The number of non-zero coefficients in neighbouring blocks is correlated. The number of coefficients is encoded using a look-up table; the choice of look-up table depends on the number of non-zero coefficients in neighbouring blocks.
- The number of non-zero coefficients in neighbouring blocks is correlated. The number of coefficients is encoded using a look-up table; the choice of look-up table depends on the number of non-zero coefficients in neighbouring blocks.

# V. FPGA(Field programmable gate array)

A field Programmable Gate array is a digital integrated circuit that can be programmed to do any type of digital function. There are two main advantages of an FPGA over a microprocessor chip for controller:

- 1. PGA has the ability to operate faster than a microprocessor chip.
- 2. The new FPGAs that are on the market will support hardware that is upwards of one million gates.

FPGAs are programmed using support software and a download cable connected to a computer. Once they are programmed, they can be disconnected from the computer and will retain their functionality until the power is removed from the chip. The FPGA consists of three major configurable elements:

- 1. Configurable Logic Blocks (CLBs) arranged in an array that provides the functional elements and implements most of the logic in an FPGA.
- 2. Input-output blocks (IOBs) that provide the interface between the package pins and internal signals lines.
- 3. Programmable Interconnect resources that provide routing path to connect inputs and outputs of CLBs and IOBs onto the appropriate network.

Many manufacturers deliver FPGAs such as Quicklogic, Altera, Atmel, xilinx, etc. In this paper the architectural design of Xilinx FPGAs is studied. In 1985, a company called Xilinx introduced a completely new idea. The concept was to combine the user control and time to market of PLDs with the densities and cost benefits of gate arrays. A lot of customers liked it and the FPGA was born. Today Xilinx is still the number one FPGA vendor in the world [10]. An FPGA is a regular structure of logic cells or modules and interconnect which is under the designer's complete control. This means the user can design, program and make changes to his circuit whenever he wants. And with FPGAs now exceeding the 10 million gate limit (Xilinx Virtex II is the current record holder), the designer can dream big.

# (A)Architectures

Generally the FPGA architecture contains configurable logic block, input output block and Programmable interconnect resources. The Architectures provides the following features.

Channel Based Routing

- Tools more complex than CPLDs
- Fine Grained
- Fast register pipelining
- Post layout timing



Figure 3. FPGA Architecture

With the introduction of the Spartan range of FPGAs we can now compete with Gate Arrays on all aspects - price, gate and I/O count, performance and cost. The new Spartan IIE will provide up to 300k gates at a price point that enables Application Specific Standard Product (ASSP) replacement [23]. There are 2 basic types of FPGAs:

- SRAM-based reprogrammable
- One-time programmable (OTP).

These two types of FPGAs differ in the implementation of the logic cell, and the mechanism used to make connections in the device. The dominant type of FPGA is SRAM-based and can be reprogrammed by the user as often as the user chooses. One-time programmable (OTP) FPGAs use anti-fuses (contrary to fuses, connections are made not "blown" during programming) to make permanent connections in the chip.[5]



Figure 4.Digital Logic History

#### (C)XILINX SRAM based FPGA

The basic structure of Xilinx FPGAs is array\_based, meaning that each chip comprises a two dimensional array of logic blocks that can be interconnected via horizontal and vertical routing channels. An illustration of this type of architecture was shown in Figure 3.4. Xilinx introduced the first FPGA family, called the XC2000 series, in about 1985 and now offers three more generations: XC3000, XC4000, and XC5000. Although the XC3000 devices are still widely used, we will focus on the more recent and more popular XC4000 family. We note that XC5000 is similar to XC4000, but has been engineered to offer similar features at a more attractive price. We should also note that Xilinx has recently introduced an FPGA family based on anti-fuses, called the XC8100. The XC8100 has many interesting features, but since it is not yet in widespread use, we will not discuss it here. The Xilinx 4000 family devices range in capacity from about 2000 to more than 15,000 equivalent gates. The XC4000 features a logic block (called a Configurable Logic Block (CLB) by Xilinx) that is based on look-up tables (LUTs). A LUT is a small one bit wide memory array, where the address lines for the memory are inputs of the logic block and the one bit output from the memory is the LUT output. A LUT with K inputs would then correspond to a 2K x 1 bit memory, and can realize any logic function of its K inputs by programming the logic function's truth table directly into the memory. The XC4000 CLB contains three separate LUTs, in the configuration shown in Figure 3.6. There are two 4-input LUTS that are fed by CLB inputs, and the third LUT can be used in combination with the other two. This arrangement allows the CLB to implement a wide range of logic functions of up to nine inputs, two separate functions of four inputs or other possibilities. Each CLB also contains two flip-flops.



Figure 5.

# (A) Platform FPGA

**Xilinx Devices** 

The Virtex-II solution is the first embodiment of the Platform FPGA, once again setting a new benchmark in performance, and offering a feature set that is unparalleled in the industry. With densities ranging from 40,000 up to 10 million system gates. Virtex-II solutions are empowered by advanced design tools that drive time to market advantages through fast design, powerful synthesis, smart implementation algorithms, and efficient verification capabilities. [10]

#### (B)Vertex FPGA

The Xilinx Virtex<sup>™</sup> series was the first line of FPGAs to offer one million system gates.

#### (C) Spartan FPGA

Xilinx Spartan<sup>TM</sup> FPGAs are ideal for low-cost, high volume applications and are targeted as replacements for fixed-logic gate arrays and for application specific standard products (ASSP) products such as bus interface chip sets. The are four members of the family Spartan IIE (1.8V), Spartan II (2.5V), Spartan XL (3.3V) and Spartan (5V) devices. The Spartan-IIE (1.8V core) family offers some of the most advanced FPGA technologies available today, including programmable support for multiple I/O standards, on-chip block RAM.[10] All Xilinx FPGA contain the same basic resources.

Slices (grouped into CLBs) contain combinational logic and register resources IOBs

Interface between the FPGA and the outside world Programmable interconnect

- Other resources
  - Memory
  - Multipliers

#### (D) Slices and CLB

Each Virtex-II CLB contains four slices. Local routing provides feedback between slices in the same CLB, and it provides routing to neighboring CLBs. A switch matrix provides access to general routing resources.

Introduced in 1998. the Virtex product line fundamentally redefined programmable logic by expanding the traditional capabilities of field programmable gate arrays (FPGAs) to include a powerful set of features that address board level problems for high performance system designs. The latest devices in the Virtex-E series, unveiledin 1999, offer more than three million system gates. The Virtex-EM devices, introduced in 2000 and the first FPGAs to be manufactured using an advanced copper process, offer additional on chip memory for network switch applications.

#### VHDL

VHDL is a language for describing digital electronic systems. It arose out of the United States Government's Very High Speed Integrated Circuits (VHSIC) program, initiated in 1980[3]. It became clear that there was a need for a standard language for describing the structure and function of integrated circuits (ICs). Hence the VHSIC Hardware Description Language (VHDL) was developed, and subsequently adopted as a standard by the Institute of Electrical and Electronic Engineers (IEEE) in the US. VHDL is designed to fill a number of needs in the design process. Firstly, it allows description of the structure of a design that is how it is decomposed into sub-designs, and how those sub-designs are interconnected. Secondly, it allows the specification of the function of designs using familiar programming language forms. Thirdly, as a result, it allows a design to be simulated before being manufactured, so that designers can quickly compare alternatives and test for correctness without the delay and expense of hardware prototyping.

#### **ISE Design Suite: Logic Edition**

The ISE Design Suite: Logic Edition allows you to go from design entry, through implementation and verification, to device programming from within the unified environment of the ISE Project Navigator or from the command line. This edition includes exclusive tools and technologies to help achieve optimal design results, including the following: **Xilinx Synthesis Technology (XST)** - synthesizes VHDL, Verilog, or mixed language designs.

**ISim** - enables you to perform functional and timing simulations for VHDL, Verilog and mixed VHDL/Verilog designs.

**PlanAhead<sup>™</sup> software** - enables you to do advanced FPGA floorplanning. The PlanAhead software includes I/O Planner, an environment designed to help you to import or create the initial I/O Port list, group the related ports into separate folders called "Interfaces" and assign them to package pins. I/O Planner supports fully automatic pin placement or semi-automated interactive modes to allow controlled I/O Port assignment.

**CORE Generator<sup>TM</sup> software** - provides an extensive library of Xilinx LogiCORE<sup>TM</sup> IP from basic elements to complex, system-level IP cores.

**SmartGuide<sup>™</sup> technology** - enables you to use results from a previous implementation to guide the next implementation for faster incremental implementation.

**Design Preservation -** enables you to use placement and routing for unchanged blocks from a previous implementation to reduce iterations in the timing closure phase.

**Team Design** - enables multiple engineers to synthesize and implement portions of a design independently.

**Partial Reconfiguration** - enables dynamic design modification of a configured FPGA. The ISE software uses Partition technology to define and implement static and reconfigurable regions of the device. This feature requires an additional license code.

**XPower Analyzer** - enables you to analyze power consumption for Xilinx FPGA and CPLD devices.

**Power Optimization** - minimizes logic toggling to reduce dynamic power consumption for Spartan®-6, Virtex®-6, and 7 series devices.

**iMPACT** - enables you to directly configure Xilinx FPGAs or program Xilinx CPLDs and PROMs with the Xilinx cables. It also enables you to create programming files, readback and verify design configuration data, debug configuration problems, and execute SVF and XSVF files.

ChipScope<sup>™</sup> Pro tool - assists with in-circuit verification.

## Simulation and result

h264cavlc Project Status (07/10/2016 - 14:09:04)									
Project File:	cavlc_encoder.xise	Parser Errors:				No Errors			
Module Name: h264cavlc Impl		plementation State:			Placed and Routed				
Target Device:	xc6slx4-3tqg144	•Errors:							
Product Version:	ISE 14.5	•Warnings:							
Design Goal:	Balanced	Routing Results:		All Signals Completely Routed					
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:		All Constraints Met					
Environment:	System Settings	Final Timing Score:			0 (Timing Report)				
Device Utilization Summary [-									
Slice Logic Utilization			Used	Available	Uti	lization	Note(s)		
Number of Slice Registers			205	4,800		4%			
Number used as Flip Flops			205						
Number used as Latches			0						
Number used as Latch-thrus			0						
Number used as AND/OR logics			0						
Number of Slice LUTs			1,702	2,400		70%			
Number used as logic			1,677	2,400		69%			
Number using O6 output only			1,565						
Number using O5 output only			0						
Number using O5 and O6			112						



Figure 6.RTL Schematics



Name	Value	.  80 ns .  100 ns .  120 ns .  140 ns .  160 ns .  180 ns .  200 ns .
15 dk	0	
16 012	0	
1% enable	1	
14 ready	0	
vin[11:0]	-1	
► 56 nin(4:0)	0	
► 6 ve[24:0]	0	
vi(4:0)	0	
16 valid	0	
State[2:0]	0	• • • • • • • • • • • • • • • • • • •
State[2:0]	0	
► 🛯 🎽 nout(4:0)	2	
Vin1[11:0]	-1	
vin2[11:0]	-1	
la en1	1	
la en2	1	
t@ sin	° I	
100 VS	1	
		X1: 120.330 ns
		2004 Series
Name	Value	150 ns 200 ns 250 ns 300 ns 350 ns 400 ns 450 ns 500 ns
1₿ ek	•	
lig ette2	•	
enable	1	
le ready	•	
vin[11:0]	00000	· <u>CXX</u> • • • • • • • • • • • • • • • • • •
inin[4:0]	00000	
● (@ ve(24:0)	00000	
The second	00000	
Tele valid	000	
TX state(20)	000	
Till noutlife	00000	
► 2% vio1(11:0)	00000	
► 2% vin2[11:0]	00000	
16 ent	1	
16 en2	1	
16 sin	0	
Tés ve	0	
		V1: 200.430 pr
		A 31 400 Y 20 13
		owr_think
		ard(283)
		ase((-4)8.0)
		out_SR0
		0.01(_100)

Figure 7. Simulation

# **VI. REFERENCES**

- [1]. I.E.G. Richardson, "H.264 and MPEG-4 Video Compression", Wiley 2003.
- [2]. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)", JVT-G050r1,May 2003.
- [3]. Loredana Freda Albanese, Gian Domenico Licciardo, "An Area Reduced Design of the Context-Adaptive Variable Length Encoder Suitable for Embedded Systems", ©2010 IEEE
- [4]. C.A. Rahman and W. Badawy, "CAVLC Encoder Design for Real-Time Mobile Video Applications", IEEE Trans. on Circuits and Systems II Expr. Briefs, vol. 54, pp. 873-877, 2007.
- [5]. C.A. Rahman and W. Badawy, "CAVLC Encoder Design for Real-Time Mobile Video Applications", IEEE

Work. On Signal Processing Systems (SIPS) , pp. 368-371, 2006.

- [6]. James Au, "Context Adaptive Variable Length Decoding System and Method", United States System and Method, November 2003.
- [7]. Iain Richardson, White paper: "H264/AVC Context Adaptive Variable Length Coding", VCODEX, conference 2002-2011.
- [8]. T. Silva, L. Agostini, S. Bampi and A. Susin, "FPGA Based of CAVLC and Exp-Golomb Coders for H.264/AVC Baseline Entropy Coding", in Proc. of Southern Programmable Logic (SPL), pp.161–166, 2007