ABSTRACT

H.264/MPEG-4 AVC is the prevailing video compression standard [2]. H.264 is presently used for most of the streaming applications in mobile and handheld devices that run on digital signal processor (DSP) platforms. Efficient implementation of the H.264 encoder/decoder on DSP platforms has been a major area of focus in recent times. This project proposes implementation and optimization the H.264 encoder on the Texas Instruments [16] TMS320C64x+ and the ARM 9E [17] processors; the two most widely used digital signal processors.

INTRODUCTION

The proposed profile of H.264 for the project is the baseline profile since this profile provides simplicity of implementation. This profile offers a low complexity encoder and decoder. It was designed for low delay applications, as well as for applications that run on platforms with low processing power and in high packet loss environment [4]. The baseline profile finds applications in video-conferencing and mobile video streaming, where low latency, coding efficiency and low complexity of the encoder/decoder are the primary considerations.

The important features of the baseline profile are [4] –

- I- and P- slice coding
- Enhanced error resilience coding such as flexible macroblock ordering (FMO) and arbitrary slice ordering (ASO) and redundant slices (RS)
- Context adaptive variable length coding (CAVLC)

The features not included in baseline profile are [4] –

- B- slices, SI- or SP- slices
- Interlace coding tools
- Context adaptive binary arithmetic coding (CABAC)

Figure 1 illustrates the various profiles of H.264 and the features offered by each one.
The joint model (JM) [1] implementation of the H.264 encoder is used as the reference for the project.

The first digital signal processor (DSP) proposed for the project is the Texas Instruments [16] TMS320C64x+. The Texas Instruments DSPs are the first choice processors for video applications due to their high performance. The salient features of TMS320C64x+ are listed below -
• 32 bit DSP engine

• The C64x+ family uses an advanced very long instruction word (VLIW) architecture called VelociTI.2

• The architecture uses parallelism. It contains multiple execution units running in parallel, which allow them to perform multiple instructions in a single clock cycle

• High clock speeds - 1.1 GHz or more

Figure 3 shows the major functional units of the C64x+ CPU.

![Diagram of C64x CPU](image)

Fig. 3. The major functional units of the TMS320C64x+ processor [9]

The major functional units of this processor are [9]-

• Two general-purpose register files (A and B)

• Eight functional units (.L1, .L2, .S1, .S2, .M1, .M2, .D1, and .D2)
- Two load-from-memory data paths (LD1 and LD2)
- Two store-to-memory data paths (ST1 and ST2)
- Two data address paths (DA1 and DA2)
- Two register file data cross paths (1X and 2X)

The multiple functional units in the processor provide for parallelism of operations. This along with the high clock speeds enhances the overall performance. The Code Composer Studio (CCS) 4.0 [11] simulation software provided by Texas Instruments is used to simulate the TMS320C64x+ processor.

The second processor proposed for the project is ARM 9E. ARM [17] is the industry's leading provider of 32-bit embedded microprocessors. The salient features of the ARM processor are –

- 32 bit reduced instruction set computer (RISC) processor
- RISC architecture provides fixed length instructions that can each execute in a single cycle. In contrast, in complex instruction set computer (CISC) processors the instructions are often of variable size and take many cycles to execute [13]
- Uses load – store architecture. This aids the processor to operate on the same fetched data multiple number of times

Figure 4 shows the ARM processor architecture.
The ARM architecture has its own special features such as the Inline Barrel Shifter, Thumb 16 bit instruction set, and enhanced digital signal processing instructions [13]. The Real View Developer Suite 2.1 simulation environment provided by ARM is used to simulate the ARM 9E processor. Alternatively the TI Code Composer Studio can also be used for the ARM 9E simulation. The ARM 9E processor is not expected to match the TMS320C64x+ on the performance front.
PROPOSED APPROACH

The focus of the project is to study in depth each block of H.264 encoder and to understand the joint model (JM) implementation in the first phase. The second phase involves understanding the Texas Instrument’s TMSC64x+ and the ARM 9E DSP architectures and porting the H.264 encoder on the two platforms. In the third phase the performance of the encoder on the two processors is analyzed in terms of MIPS, and the most computationally expensive blocks are identified and optimized separately for both the DSP cores. The goal of the project is to optimize the existing JM implementation and achieve MIPS reduction of about 30% for both the processors. The output of the optimized implementation needs to be bit exact with the reference output generated by the JM implementation.

WEIGHTAGES OF DIFFERENT PHASES OF THE PROJECT

- Understanding the H.264 encoder blocks and the C implementation – 50%
- Porting and optimization on the TMS320C64x+ processor – 25%
- Porting and optimization on the ARM9E processor – 25%
### List of Acronyms in Alphabetical Order

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM</td>
<td>Advanced RISC Machine</td>
</tr>
<tr>
<td>ASO</td>
<td>Arbitrary slice ordering</td>
</tr>
<tr>
<td>AVC</td>
<td>Advanced video coding</td>
</tr>
<tr>
<td>CABAC</td>
<td>Context adaptive binary arithmetic coding</td>
</tr>
<tr>
<td>CAVLC</td>
<td>Context adaptive variable length coding</td>
</tr>
<tr>
<td>CCS</td>
<td>Code composer studio</td>
</tr>
<tr>
<td>CISC</td>
<td>Complex instruction set computer</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital signal processor</td>
</tr>
<tr>
<td>FMO</td>
<td>Flexible macroblock ordering</td>
</tr>
<tr>
<td>JM</td>
<td>Joint model</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>RISC</td>
<td>Reduced instruction set computer</td>
</tr>
<tr>
<td>RS</td>
<td>Redundant slices</td>
</tr>
<tr>
<td>RVDS</td>
<td>Real view developer suite</td>
</tr>
<tr>
<td>TI</td>
<td>Texas Instruments</td>
</tr>
<tr>
<td>VLIW</td>
<td>Very long instruction word</td>
</tr>
</tbody>
</table>
REFERENCES


7. An overview of the H.264 encoder: www.vcodex.com

8. TMS320C64x/C64x+ DSP CPU and instruction set reference guide: http://focus.ti.com/lit/ug/spru732h/spru732h.pdf


12. ARM optimizing C/C++ compiler v4.6 user’s guide: http://focus.ti.com/lit/ug/spnu151e/spnu151e.pdf


14. ARM assembly language tools v4.6 user’s guide: http://focus.ti.com/lit/ug/spru118h/spru118h.pdf

15. ARM 9E family overview: http://www.arm.com/products/CPUs/ARM968E-S.html