Application Report
SPRAB66–September 2009

Performance Benchmarking of Video Software on TMS320DM6467

Anirban Sengupta — *Multimedia Codecs, SDO*

ABSTRACT

The performance and complexity of video codecs is driven by several factors. Algorithm design/flow is just one of these factors. Other important factors are device architecture, software architecture, resource availability and usage, etc., to name a few. However, one of the most important factors that are often ignored in the codec development cycle is the complete system architecture and the targeted use case.

In this application report, a case study of the H264 Transrater on the DM6467 is presented and the method adopted to benchmark video software in a complete application context that models the final use case of the codecs is demonstrated.

Contents

List of Figures

List of Tables

1 Background

The H264 Transrater changes the bit rate of an H.264 high-definition (HD) bit-stream. This is one of the most computations intensive processing since a *universal* H.264 bit-stream has to be decoded and encoded again in H.264 with a different (lower) bit rate (see Figure 1). It requires more DSP computation and DDR bandwidth than the MPEG-2 to H.264 transcoder and the H.264 to MPEG-2 transcoder, which has already been developed as key DM6467 applications. Since the input is a *universal* H.264 bit-stream, no assumptions can be made with respect to the properties of the incoming bit-stream (such as limited sets of macroblock partitions, limited usage of multiple reference frames, etc.). To overcome these difficulties, an MB level tightly coupled transrater algorithm has been implemented. The subsequent sections discuss how different aspects of the implementation impact overall system performance and how the H264 Transrater software was benchmarked on DM6467.

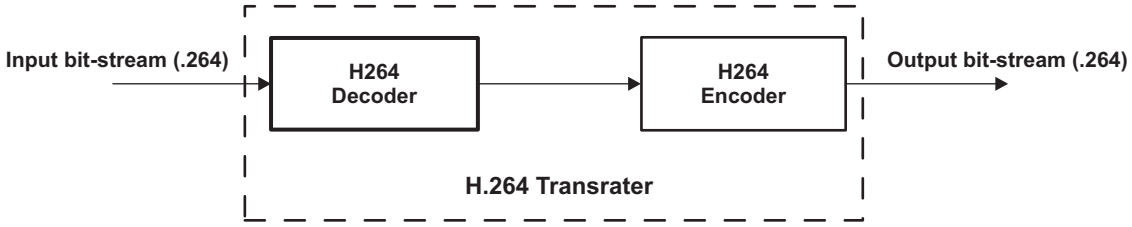

Figure 1. H264 Transrater – Conceptual Block Diagram

2 Brief Design Overview

The input to the H264 Transrater is H264 bit-stream. Therefore, Figure 1 illustrates that the decoder forms the front end of this codec. The decoding pipeline is extended using the encoder blocks to form the transrater pipeline. DM6467 has two HDVICPs: HDVICP_0 meant for encoding and decoding and HDVICP_1 only capable of decoding. Each of these HDVICPs has an ARM968 processor that takes care of scheduling and triggering DMAs and individual coprocessors of the corresponding HDVICP. However, ARM968 coprocessor may not have the bandwidth to perform certain cycle intensive operations that are required for every macroblock pair that is processed for which it needs the DSP. As a result of this design, the ARM968 schedules these macroblock pair level tasks on the DSP by implementing an appropriate signaling and messaging scheme.

In the context of the H264 Transrater, we have two HDVICPs: HDVICP_1 performs the decoder tasks and HDVICP_0 performs the encoder tasks. However, both need the DSP to perform certain macroblock pair level functions. Figure 2 shows the task partitioning and signaling scheme and the control flow between DSP, ARM968_0 and ARM968_1 to implement the transrater.

The light gray shaded area shows the decoding stages of the transrater pipeline that forms the front end. The darker gray shaded area shows how the encoding stages are plugged into the decoding stages to form the complete transrater pipeline.

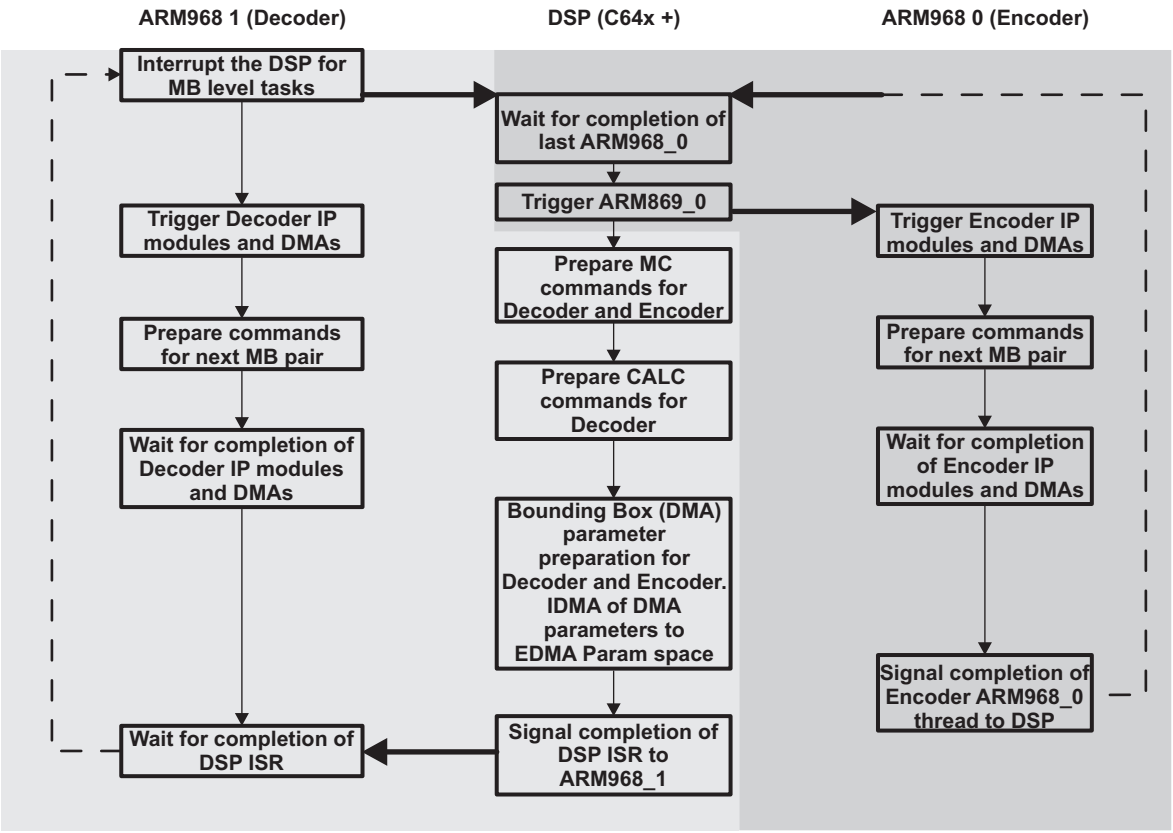

Figure 2. Task Partitioning, Signaling and Control Flow Between DSP, ARM968_0 and ARM968_1

3 Use Case Overview

The H264 Transrater is just one of several components in the targeted use case. The following is a list of all components that will be running simultaneously in the final application scenario.

- Input: H264 AC3 (5.1 channel) broadcast stream
- Output: H264 (lower bit rate) AAC stereo stream
- Host ARM
  - Linux Kernel
  - File reader/writer
  - Event logger
  - TS demux
  - Audio multichannel AC3 decoder
- DSP
  - BIOS
  - H264 to H264 Transrater: 1920x1080i@30fps
  - Audio encoder AAC-LC stereo
  - Other infrastructure software

The device is configured as follows:

DSP memory Configuration

- DSP L2
  - 32 KB Cache
  - 96 KB SRAM
    - 32 KB for critical H.264 transrater code executed per MB pair
    - 64 KB used as scratch for video
- TI EVM configured to:
  - 4-bank DDR
  - ARM frequency 337 MHz
  - DSP frequency 675 MHz
  - DDR frequency 310 MHz

4 Performance Benchmarking

4.1 The Problem

Figure 2 shows that the DSP services tasks required for both the decoder and encoder within the transrater. Also, the DSP function is invoked by an interrupt from HDVICP_1. In other words, the DSP performs its tasks in an ISR context, i.e., HDVICP_1 sends an interrupt to the DSP for every macroblock pair. As a result, the DSP can potentially have some idle cycles after completing one ISR and before the next interrupt is triggered. In the case of the transrater, the idle duration turns out to be a significant value since the HDVICP takes longer to complete one macroblock pair processing. To efficiently use the DSP, these idle cycles (holes) are used to run other functions on the DSP like the audio encoder (see Figure 3). The HDVICP_1 sends an interrupt to the DSP, which in turn kicks off HDVICP_0 and its own functions. The bold lines represent the corresponding processor threads. Figure 3 shows that the DSP thread will finish faster and provide significant amount of idle cycles spanning over multiple macroblock pairs as shown by the dotted lines.

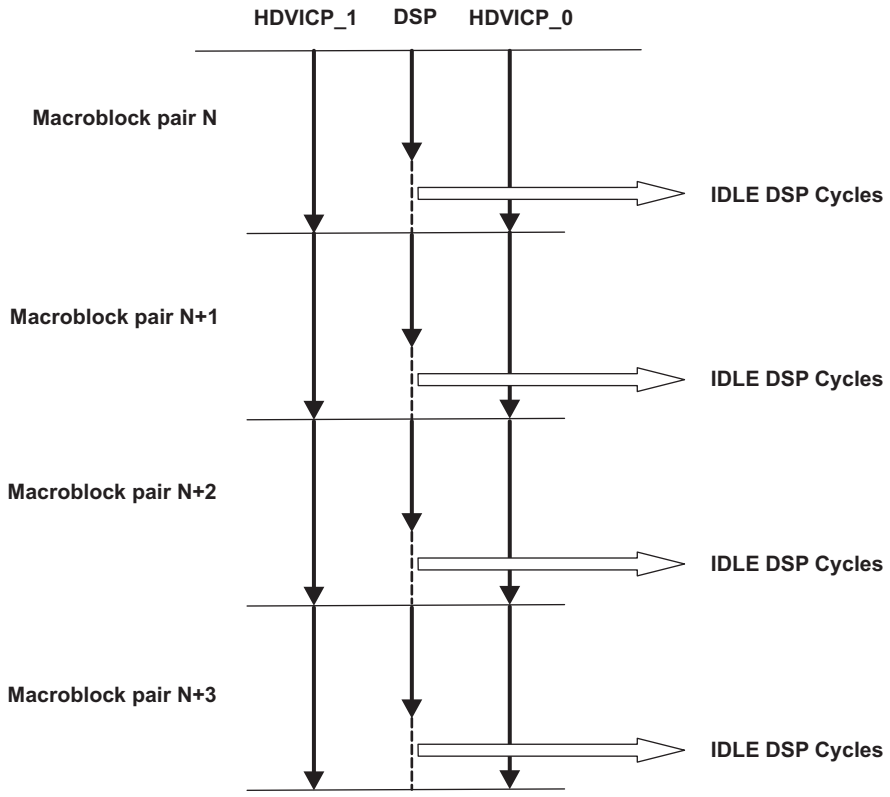

Figure 3. Representation of Available Idle Cycles on DSP

However, running the audio encoder within these idle cycles is not straightforward because:

- The audio encoder code thrashes out the transrater DSP ISR for every macroblock pair resulting in the transrater encountering additional cache penalty
- The audio encoder cycles also increase as it has to run in these *idle* slots, therefore, there will be additional context save and restore overheads apart from the obvious cache penalties

For the reasons highlighted above, measuring transrater cycles in a standalone Code Composer Studio™ software based setup is insufficient as the performance is bound to degrade in the integrated application scenario.

4.2 Non-Blocking and DSP MHz

It is extremely critical to understand the two terms: non-blocking MHz and DSP load as codec component data sheets use these interchangeably. This section elaborates on these terms in the overall context of the system DSP load.

As shown in Figure 3, the transrater utilizes the DSP through an interrupt service routine (ISR) for execution of MB pair level tasks by raising an interrupt. Apart from this, the DSP is used by the transrater for certain initialization and post-processing tasks. The accumulated cycles for performing all these tasks is known as non-blocking cycles, i.e., if there are N interrupts needed to process a given picture bit-stream data, then –

$$\text{Non-blocking cycles} = \text{Initialization Cycles} + \text{ISR}_1 \text{ Cycles} + \text{ISR}_2 \text{ Cycles} + \text{ISR}_3 \text{ Cycles} + \ldots + \text{ISR}_N \text{ Cycles} + \text{Post-processing Cycles}$$

This gives the non-blocking cycles for a given picture. Accumulating this over 1 second gives the non-blocking MHz of the given component (in this case, the transrater). As can be seen, non-blocking MHz can also be viewed as the DSP load due to the transrater component.

However, when this component is integrated into an application, the DSP load needs to be re-measured.

As explained in the previous section, the audio encoder runs using the idle DSP cycles where the DSP is waiting for the next interrupt to be raised. Let us assume that the audio encoder DSP load is 'A' MHz and the transrater DSP load (non-blocking MHz) is 'B' MHz. Therefore, when put together in the same system, what will be the total DSP load? Will it be 'A+ B' MHz? Although the idea of the DSP load just summing is simplistic, unfortunately, that is not the case. Why?

- When the audio encoder is running and the interrupt is raised, the DSP context has to be saved before the ISR execution can start.
- Since the ISR and data accessed by it has to be *cached in* for execution as the audio encoder was running on the DSP, there will be a penalty due to cache misses. Therefore, the ISR takes more cycles as compared to standalone execution (without audio)
- On completion of the ISR, the DSP context needs to be restored before the audio encoder execution can resume.
- Since the audio encoder code and data was thrashed out of the cache by the ISR, the audio encoder cycles will also increase due to cache penalty

Therefore, it is imperative that the DSP load be re-measured in a system with audio running simultaneously to evaluate the *true performance* of the system. As highlighted, apart from measuring the DSP load due to the transrater, the audio encoder cycles (or any other DSP component like encryption, etc.) and the ISR save and context cycles should also be measured. The DSP load thus measured may also be correlated with the actual idle DSP MHz and this should add up to the available DSP MHz (say, 675 MHz according to this use case)

Note that an ideal system design would minimize the increase in the transrater DSP load arising out of running other DSP components (audio encoder, encryption, etc.) simultaneously. This can be achieved by optimal placement of the audio encoder code in DDR and also by placing critical transrater ISR code in internal L2 SRAM. As mentioned in Section 3, 32 KB of L2 SRAM has been reserved for this purpose.

4.3 Optimal DMA Load and TC Balancing

In the earlier section, we discussed how the DSP load increases when different components are run simultaneously. We also discussed how to minimize this increase. Basically, the DSP can be viewed as a resource which when accessed simultaneously, results in the increase in cycles of the individual components using the DSP as compared to their standalone numbers.

Similarly, the DMA is a resource that is used by different components. The traditional component optimization approach has been to distribute the DMA load evenly across the four TCs (TC 0, TC 1, TC 2 and TC 3). This approach works as far as standalone component cycles are concerned. The factors that are generally taken into account to optimally balance DMA load across TCs are:

- Number of bytes to be transferred per MB pair
- Source and destination memory regions of the DMA transfers

However, when different components that use the DMA are put together in a system, the overall DMA load per TC may become *unbalanced*.

Typically, the ARM968 within each HDVICP triggers and waits for the DMAs to be completed. Therefore, any imbalance arising out of the DMA load will result in increase of HDVICP cycles, which results in reduced frequency of interrupts. This can cause the system to become *non-real-time*.

It is, therefore, essential to balance the DMA load across TCs by looking across components. Needless to say that it also becomes imperative to benchmark individual components with all components running simultaneously in a system.

4.4 DDR Usage

Similar to the DSP and DMA, the DDR is also a common resource accessed by the DSP, host ARM (ARM926), the DMA engine, drivers, etc. Therefore, running different components simultaneously causes increased DDR conflicts and as a result increases the individual component and overall system MHz. Note that as part of transrater optimization, DDR usage is made as less as possible by utilizing internal L2SRAM memory or HDVICP memory wherever possible.

As can be appreciated from Section 4.2 and Section 4.3, this further underlines the need to benchmark performance in a system with different components running simultaneously to simulate the actual use case and measure the *true performance* of the system.

4.5 The Traditional Approach

Figure 4 depicts the traditional approach of measuring video performance. The demerits of such a setup have already been highlighted; it is mentioned here for the sake of completion.

Figure 4. The Traditional Approach – Standalone Codec Performance Measurement

As mentioned earlier, the major drawback of this approach is that the setup does not model the use case system/application that has many other components running simultaneously.

4.6 The New Approach

Figure 5 depicts the new approach adopted in the case of H264 Transrater.

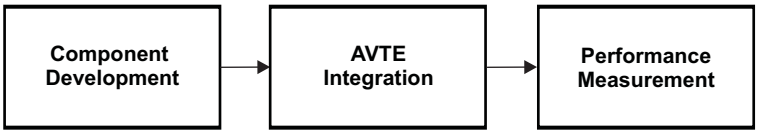

Figure 5. The New Approach – System Integrated Codec Performance Measurement

The Audio Video Transcoding Engine (AVTE) provides realistic transrater results when operating in the real system with audio and video running simultaneously. In addition to the transrater and audio, the AVTE has other software components listed in Section 3 running concurrently.

This approach has the following advantages:

- It models the customer's application and provides the component development team with a platform to reliably benchmark the transrater with respect to performance and stability
- Early detection and resolution of performance and stability issues, therefore, enabling faster integration in the customer's application
- Credible demonstration of the feasibility of the solution
- Provides a platform to correlate the results from Standalone codec environment with system environment

4.7 Results

Using the section mentioned in Figure 5, performance was measured for the transrating system (and not just the transrater) with all components listed in Section 3 running simultaneously. An example of the performance measured is given in Table 1.

Table 1. Transrater System Performance Measured Using AVTE on DM6467

Input Content Type	Output Content Type	Input Bit Rate	Output Bit Rate	Resolution	DSP Load of the System (including transrater ISR, audio, context save and restore overheads)	Transrater Load (including HDVICP threads)
Interlaced @ 60 fields per second	Interlaced @ 60 fields per second	14 Mbps	8 Mbps	1920 x 1088	600 MHz	635 MHz

5 References

- TMS320DM6467 Device Specifications can be downloaded from http://www.ti.com/corp/docs/landing/davinci/dm6467.html
- *TMS320DM6467 SoC Architecture and Throughput Overview* (SPRAAW4)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265