June, 2010

Designing and Implementing H.264/SVC on the Multicore MSC815x and MSC825x StarCore DSPs

FTF-NET-F0559

Yaniv Klein
Team Leader, Video Software
Introduction

► The computational effort of DSP applications is constantly increasing due to increase in bandwidth and data rates.

► On the other hand processors are reaching frequency limitations due to stringent power constraints.

► Multi processing is one approach that enables high complexity applications while keeping power requirements relatively low.

► We will show H.264/SVC video Codec as an example of such a high complexity application.
Agenda

► H.264/SVC Overview
► Challenges
► MSC8256 StarCore DSP overview
► MSC8256 StarCore DSP advantages
► Conclusions
H.264/SVC Overview
H.264 Overview

► H.264 is a video compression standard defined by ITU intended to provide good quality with substantially lower bit rates than previous standards (H.263, MPEG2, MPEG4).

► The complexity of H.264 is higher due to introduction of new coding tools such as
  • In-loop filter (deblocking).
  • 6-Tap quarter-pixel interpolation.
  • Enhanced intra prediction modes.
  • Motion vector per 4x4 pixel blocks.

► Moreover, H.264 supports HD resolutions such as 720p (1280x720) and 1080i/p (1920x1080)
H.264/SVC Overview

- H.264 Scalable Video Coding (SVC) is an extension of H.264/AVC that provides 3 scalability options:
  - Temporal scalability – different frame-rates.
  - SNR/Quality/Fidelity scalability – different video quality.
  - Spatial scalability – different video resolutions.

- The ITU standard is in-force since November 2007.

- Advantages:
  - Encode multiple “streams” in a single stream, can serve different consumers with not additional effort.
  - More efficient than multi cast (encoding each stream separately).
  - Error resilient – since there is much redundant information, lost data has almost no effect on quality.
Normal Stream
Scalable Stream

quality Enhancement

temporal Enhancement

spatial Enhancement

Base layer

Time
SVC scalability types

- Temporal scalability
- Spatial scalability
- Quality scalability
Temporal scalability

- The referencing structure allows for complete frame to be discarded without harming decoding.

- Degrades reference frame quality since time difference is bigger. Also - bit allocation between temporal layers is not trivial.

- Requires adaptation of motion estimation.
Spatial scalability

► Spatial enhancement layers are coded as the base layer but have additional prediction options

► Spatial enhancement layers use temporal prediction from different reference frames than those of the base layer
Quality scalability is usually achieved by two methods:

- Re-quantization – in this method the coefficients are quantized for each quality layer and the residual between layers is being transmitted.
- Scan partitioning – in this method the coefficients are divided into groups and each group is transmitted in a different quality layer, thus enhancing picture quality.

Ex. : Scan partitioning
SVC Encoder Block Diagram

* Per dependency layer
Challenges & Possible Solutions
Challenges

► Complex software Codecs will currently not fit on any single core DSP.
► Thereby software solutions must span on multiple cores or multiple devices.
► Software implementation is simpler on a single device solution with multiple cores, rather than multiple single-core devices.
► Therefore, we will focus on the challenges that arise in multiple cores solutions.
Challenges

The implementation of the codec holds great challenges, such as:

► How to partition the codec?

► How to implement a multi-core Rate Control?

► How to parallelize Deblocking?

► How to manage task allocation?
How to partition the codec?

Two approaches can be considered:

► Functional partitioning

► Slice based partitioning
Functional Partitioning

Functional partitioning (pipelining)

► Breaking the processing based on functional stages of encoding.

► A possible partitioning is illustrated below.

Stage 1:
- Motion Estimation
- Intra Prediction

Stage 2:
- Diff
- Transform + Quantize
- Inverse Transform quantize
- Add

Stage 3:
- Entropy coding
- Deblocking Filter
Frame partitioning (Slicing)
► Breaking each video frame into slices, each slice is allocated to an available resource.
### Functional vs. Slice Based Partitioning to Tasks

#### Pros and Cons

<table>
<thead>
<tr>
<th></th>
<th>Slice Based</th>
<th>Functional Based</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Implementation</strong></td>
<td>Some stages require the entire video frame (deblocking)</td>
<td>Simple, maintains the normal macroblocks processing order</td>
</tr>
<tr>
<td><strong>Timing considerations</strong></td>
<td>Almost no time dependency between processing units. State variables need to be synchronized (RC, Quantizers)</td>
<td>Requires synchronization mechanism between stages</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Scalable – a frame can be divided into many slices</td>
<td>Not scalable - functionality division is naturally limited</td>
</tr>
<tr>
<td><strong>Load Balancing</strong></td>
<td>Easier to balance, requires dynamic balancing mechanism between slices</td>
<td>Hard to balance. Each stage’s MIPS can vary significantly based on the input stream</td>
</tr>
</tbody>
</table>
Challenges - Rate Control

► Rate-control - Controlling the output bit rate by adjusting the quantization factor

- How does each core perform rate-control?
  - Rate-control can be done independently, budget is divided between cores and each core handles rate-control locally.
  - A master core handles rate-control by collecting data from all slaves and updating all the slaves with the rate-control changes.

- Adaptive algorithm must be used because input video may vary and the budget needed for each slice might change between frames

- If a frame is sliced, it is important not to have very big difference in quantization factor on the edges of the slice or the edge may be visible.

- Rate Control has to take into account the conflict of constraints between the different layers.
Challenges – Deblocking

Deblocking filter is applied to blocks in decoded video to improve visual quality by smoothing the sharp edges between blocks

- Deblocking is done in raster order, top-down from left to right.
- Deblocking has a very serialized nature because its MB processing is dependent on the MB above it and to the left of it.
- Moreover, each MB deblocking also affects the MBs to the left and top.
- It is a major challenge to parallelize it on several cores.

<table>
<thead>
<tr>
<th>Partition options on one device with multiple cores</th>
<th>Partition options on Multi device</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Deblocking on a single core</td>
<td>• Task allocation is critical in order to reduce data traffic - best if each device will do most of the processing on its own and transfer minimal data to other devices</td>
</tr>
<tr>
<td>• Balance load by deblocking for luma and chroma separately</td>
<td>• Each device does its own deblocking. need to synchronize between devices for the blocks on the device partition border</td>
</tr>
<tr>
<td>• Deblocking on a partial reconstructed frame on one core, while the remainder of the frame is being encoded on the other cores</td>
<td></td>
</tr>
</tbody>
</table>
The distribution and control of the different tasks can be done in several ways. Example:

- **Master/Slave**
  - One task acts as master task for all slave tasks
  - The master task provides a well defined activity to the other tasks and controls the load balancing

- **Fully distributed system**
  - Each task is completely independent
  - Each task must make sure that it’s work is not done by any other task
  - Each task is responsible to update the work status when it ends
  - Task code is basically identical
MSC8256 Overview
MSC8256 Block Diagram

- 6x SC3850 Cores Subsystems (6GHz/48GMACS) each with:
  - SC3850 DSP core at up to 1GHz (8GMACs 16b or 8b)
  - 512 Kbyte unified L2 cache / M2 memory.
  - 32 Kbyte I-cache, 32Kbyte D-cache, WBB, WTB, MMU, PIC
  - Fully Programmable

- Internal/External Memories/Caches
  - 1056 KByte M3 shared memory (SRAM)
  - Two DDR 2/3 64-bit SDRAM interfaces at up to 800 MHz
  - Internal/External Memories/Caches

CLASS – Chip-Level Arbitration & Switching Fabric
- Non-Blocking, fully pipelined, low latency
- Full fabric 12 masters to 8 slaves, up to 512 Gbps throughput

- High Speed Interconnects
  - Dual 4x/1x Serial RapidIO at 1.25/2.5/3.125 Gbaud
  - PCI-e 4x/1x
  - Dual RISC QUICC Engine® supporting
  - Dual SGMII/RGMII Gigabit Ethernet ports
  - Eth. Protocols, Talitos control and sRIO offload

- Ethernet
  - Dual Gigabit Ethernet ports (SGMII/RGMII)

- TDM Highway
  - 1024 ch., 400Mbps, divided into 4 ports of 256

- DMA Engine 16 bi-directional channels

- Other Peripheral Interfaces
  - SPI, UART, I2C, 32 GPIO, 16 Timers, 96KB boot ROM, JTAG/SAP, 8WDT

- Technology
  - Process: 45nm SOI
  - Voltage: 1V core, 2.5, 1.8/1.5V I/O
  - Package: FCBPGA (29x29) 1mm pitch, RoHS

-Freescale, the Freescale logo, AliVec, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKIT, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MXC, Platform in a Package, Processor Expert, CorIQ, QUICC Engine, SMARTMOS, TurboLink and VorIQs are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
MSC8256 StarCore Multicore Advantages
MSC8256 StarCore DSP Advantages for Multicore

► 6 cores DSP provides best in class performance.
► M3 and DDR are fully accessible by all 6 cores.
► DMA transfer supports
  • Up to 4 dimensions
  • Freeze capability after each dimension
  • Up to 16 channels of DMA
► DMA makes it possible to easily transfer a full frame MB by MB with a one time programming in the beginning of each frame processing.
► Large L2 Cache with L2 pre-fetch capabilities can replace DMA traffic in simple cases.
► Dynamic partitioning of M2/L2.
► MMU translation and virtual addressing can help abstract private memory of each core, same virtual address mapped to different physical address makes code simpler.
► Easy to communicate between cores in a non-cacheable area in M3
Additional MSC8256 StarCore DSP Advantages

- SRIO provides ~20Gb/second throughput required for moving uncompressed video between devices
- SRIO supports one dimensional DMA transfers that are done in parallel to device processing
- SRIO support of doorbell implementation can act as an interrupt between devices for possible synchronization mechanism
- SRIO broadcasting capability can help distribution of data to more than one device
- PCI Express provides ~8Gb/second throughput if SRIO is not fully utilized.
- 2 ports of Gigabit Ethernet.
Conclusions
Conclusions

► DSP Processors are steadily moving towards a multicore architecture due to power constraints and increased computational effort

► HD Video Codecs processing requirements are constantly increasing and need a multi-task approach to be fully supported

► Today’s presentation has shown that implementing a HD Video solution on a multi-core device requires:
  • Smart partitioning and management of tasks between cores
  • A powerful device to support all application needs

► The MSC8256 StarCore DSP combined with FSL’s knowledge in multi-core applications has proven to be a compelling solution to the challenges of high performance video processing