mediaDSP™: A Platform for Building Programmable Multicore Video Processors

Richard Selvaggi and Larry Pearlstein
AMD
Hot Chips 20, August 2008
Recent Trends

Big-screen TV sets have been selling in ever-increasing numbers in the past 5 to 10 years.

LCDs have recently overtaken CRTs in unit shipments.

LCD display technology has advanced in many ways over these years.
- Wider viewing angle
- Higher contrast ratio
- Wider color gamut
- Higher resolution

But...**motion blur** is one of the last remaining challenges for LCD display technology.

- **This is the problem our chip is addressing.**
Perception of Motion Blur on LCD Panels

**60 frames/sec**
- Eye moves continuously, smearing the still image across the retina for 16ms.

**120 frames/sec**
- Eye moves a shorter arc in 8ms, thus reducing the amount of blur across the retina.

**Interpolated frames**
- Reduced blur

**LCD** is a hold-type display (not impulsive, like CRT).
Projected object is smeared on the retina by the amount that the eye moves in one frame-time (16 vs. 8ms).
AMD Xilleon 420 (X420) performs motion compensated frame rate conversion (MC-FRC) and doubles the frame rate. Samsung Timing Controller (TCON) for LCD panel is integrated inside X420.
mediaDSP Introduction

mediaDSP is a flexible media processing platform
- Well suited to address a range of media applications, with emphasis on video processing
- Enables guaranteed real-time performance

Has been used in two product families
- MPEG A/V encoding
- Motion compensated frame rate conversion (MC-FRC)

Characteristics of MC-FRC:
- Is an ill-posed problem – no single, optimal solution; customer preferences vary
- Extreme computational load
Classes of Video Processing Operations

Highly parallelizable operations (linear and nonlinear)
Data can be pixels, motion vectors, picture statistics, etc.
  - Programmable SIMD (single instruction, multiple data), or specialized engine

Ad-hoc computation and decision-making
  - General-purpose processor

Data movement/formatting on multiple planes
  - Multi-dimensional DMA engine

Bit-serial processing (entropy encode/decode)
  - Programmable bit-stream processor

Leads to a multicore approach with application-specific processing elements.
Video and image processing algorithm developers are accustomed to working with task flow graphs.
mediaDSP architecture is driven by this type of programming model.
mediaDSP accelerates a flexible pipeline of video tasks.
Example Task Flow Graph: MPEG-2 Video Macroblock Encoder (Unmapped)

Read from FB to refresh Cylindrical buffer (4 macroblocks)

Read downsized current MB (4 macroblocks)

Read current MB

Coarse motion search +/-128H x +/-64V effective range (2 macroblocks)

Address calc

Read half-pel reference data
24 x 10 x 4 lum
32 x 6 x 4 chr

Half-pel motion search
-4/+3.5H x +/-1V search range

Address calc

Motion compensated prediction

Average predictions

Only for bidirectional macroblocks

Not done in B frames

Not done in I macroblocks

Previous macroblock rate control info

Subtract prediction
DCT quantize

CBP skip detect

Write macroblock header to stream

DC prediction
DC differential coefficient scan
VL encode 1-6 blocks of coeffs

Write result to FB
16 x 16 lum
16 x 4 lum
16 x 8 chr

Rate control end of macroblock
Example Task Flow Graph:
MPEG-2 Video Macroblock Encoder (Mapped)

EMDMA
Read from FB
to refresh
Cylindrical buffer
(4 macroblocks)

MEE
Coarse motion search
+/-128H x +/-64V
effective range
(2 macroblocks)

RISC
Address calc

EMDMA
Read half-pel reference data
24 x 10 x 4 lum
32 x 6 x 4 chr

CRUNCH
Half-pel motion search
-4/+3.5H x +/-1V
search range

RISC
Address calc

CRUNCH
Motion compensated prediction

EMDMA
Read downsampled current MB
(4 macroblocks)

CRUNCH
Subtract prediction
DCT quantize

CRUNCH
Dequantize IDCT add prediction
downsampling

EMDMA
Write result to FB
16 x 16 lum
16 x 4 lum
16 x 8 chr

Not done in I macroblocks

Not done in B frames

EMDMA
Read current MB

RISC
Previous macroblock rate control info

RISC
CBP skip detect

RISC
Write macroblock header to stream

CRUNCH
DC prediction
DC differential coefficient scan
VL encode 1-6 blocks of coeffs

VLCE
Rate control end of macroblock

EMDMA
Read from FB
to refresh
Cylindrical buffer
(4 macroblocks)
A Task-Oriented Approach

**Task**: A well defined piece of work that must be done. A task has a definite initiation time, and continues uninterrupted until it halts.

**Task Oriented Engine (TOE)**: A processing engine, either programmable or fixed function, that executes a task, and then halts.

**Task Control Unit (TCU)**: A specialized control unit that resides “in front of” a TOE, closely coupled with the TOE.
- Queues tasks which are to be issued to the TOE
- Synchronizes with other TCUs and Control Engines in the system

**Control Engine**: General-purpose RISC processor.
- Orchestrates the tasks for maximum parallelism
- Functions as a TOE for ad hoc tasks

**Shared Memory**: A wide on-chip memory that is accessible to all of the TOEs and other elements in the mediaDSP.
- Holds inter-task data buffers

**Communication Fabric**: The interconnect network among the TOEs and other elements of the mediaDSP.
Generic mediaDSP Topology

mediaDSP

Task-Oriented Engine #1
Task Control Unit

Task-Oriented Engine #2
Task Control Unit

... Task Control Unit

Task-Oriented Engine #N
Task Control Unit

Communication Fabric

Shared Memory

Task Control Unit

External Memory DMA Engine

Task Control Unit

Control Engine

Register Slave Interface Unit

Chip-Level Memory Backbone

Chip-Level Register Backbone
**X420 mediaDSP Topology**

![Diagram of X420 mediaDSP Topology]

- **FRC-Specific Engine #1**, **FRC-Specific Engine #2**, **FRC-Specific Engine #3**, **FRC-Specific Engine #4**, **FRC-Specific Engine #5**, **FRC-Specific Engine #6**, and **Crunch Engines #1-10**
- **Control Engine #1**, **Control Engine #2**, **Control Engine #3**, and **Control Engine #4**
- **Register Slave Interface Unit**
- **Chip-Level Stream Manager**
- **Chip-Level Register Backbone**
- **Chip-Level Memory Backbone**

**Key Components:**
- **Communication Fabric**
- **TCU**
- **Stream Capture Engine**
- **Shared Memory**
- **External Memory DMA Engine**
- **Control Engine**
- **Register Slave Interface Unit**
# Re-Use, Extending, Scaling

<table>
<thead>
<tr>
<th>Module/TOE</th>
<th>SD Enc1</th>
<th>SD Enc2</th>
<th>HD FRC</th>
<th>FHD FRC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Infra</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control Engine</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Task Control Unit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced DMA Engine</td>
<td>3</td>
<td>3</td>
<td>30</td>
<td>64</td>
</tr>
<tr>
<td><strong>TOEs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crunch Engine (SIMD DSP)</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>Variable Length CODEC Engine</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motion Estimation Engine</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub-pixel Motion Estimation Engine</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRC-Specific Engine #1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>FRC-Specific Engine #2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>FRC-Specific Engine #3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>FRC-Specific Engine #4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>FRC-Specific Engine #5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>FRC-Specific Engine #6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Stream Capture Engine</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- **Re-Use** indicates components that are directly reused.
- **Scaling** indicates components that are scaled up.
- **Extending** indicates components that are extended for different TOEs.

*August 2008*  
*Hot Chips 20: mediaDSP*
Template for TOE

TOE heavily leverages infrastructure technology from the platform:

- Re-use of DMA Engine, TCU and Power Manager
- Auto-generated routers/arbiters
- Allows heterogeneous TOEs to have similar ‘look-and-feel’ (API) to the programmer.

Building a new TOE amounts to just designing the processor core.

E.g., 3D Median Filter Engine took less than 2 months from concept to RTL complete
**Task Control Unit**

**Command FIFO** decouples Control Engine writes from the activity of the TOE. FIFO holds:

- Register writes to TOE (task-setup commands)
- Commands to the TCP

**Task Command Processor** off-loads low-level task management from the Control Engine:

- Pass-through of reg writes to TOE
- Local status monitoring of TOE
- Wait for a local semaphore
- Set a remote semaphore
TCUs In Action

Task Flow Graph:
- DMA In
- Number Crunch
- DMA Out

Communication Fabric
- TCU
- EMDMA Core
- Semaphores
- Task Flow Graph:
  - DMA In
  - Number Crunch
  - DMA Out

TCU
- Crunch Engine 1
- Crunch Engine 2
- Semaphores
- Task Flow Graph:
  - DMA In
  - Number Crunch
  - DMA Out

TCU
- Crunch Engine 1
- Crunch Engine 2
- Semaphores
- Task Flow Graph:
  - DMA In
  - Number Crunch
  - DMA Out

TCU
- Crunch Engine 1
- Crunch Engine 2
- Semaphores
- Task Flow Graph:
  - DMA In
  - Number Crunch
  - DMA Out

TCU
- Crunch Engine 1
- Crunch Engine 2
- Semaphores
- Task Flow Graph:
  - DMA In
  - Number Crunch
  - DMA Out

TCU
- Crunch Engine 1
- Crunch Engine 2
- Semaphores
- Task Flow Graph:
  - DMA In
  - Number Crunch
  - DMA Out
- Multi-level memory hierarchy with DMA’s among the levels.
- Deterministic execution time is important for real-time.
- Latency tolerance is built into the application.
Crunch Engine

“Make Every Cycle Count!”

General-purpose DSP engine tailored to video and image processing.
- Wide SIMD/Reduction Processor (128-bit, 16-way)

Keep data processing path **busy**.
- Zero-overhead loops
- Predication (avoid control/decision operations)
- Zero-overhead for unaligned operand access (avoid shifts/masks)
- HW support for **streaming data types** (address processing is buried)
- Example: 16-tap FIR filter, at 1 sample/clock
  (*i.e.*, 400 Msamples/sec @ 400 MHz)

Customized instruction set: greater than 300 instructions.
- Multiply/accumulate (16 MACs/clk → 6.4 G MACs/sec)
- Sum of absolute differences (192 ops/clk → 77 GOPs)
- Find location of MIN/MAX
- Convolution-style instructions (bicubic filter @ 144 ops/clk → 58 GOPs)

Deterministic execution time.
- Dedicated instruction store (no cache)
- HW pipeline is exposed to SW-level (no dynamic stalling)
- Task-oriented API (non-interruptible)
Crunch Engine Block Diagram

- **Request from Backbone or TCU**
- **Request to Backbone**

- **Backbone Slave Interface Unit**
- **Backbone Master Interface Unit**
- **Memory Read DMA**
- **Memory Write DMA**

**Crunch Engine Core**

- **Branch Unit**
- **Instruction Memory**
- **Address Generator Unit**
- **Local RAM A**
- **Local RAM C**
- **Operand Selection**
- **Data Processing Unit**
- **Control Pipeline**

**Crunch Engine**

- **Slave Access Decoder/Router**

**Input/Output DMA engines, with streaming feature**

- **Program sequencer with zero-overhead looping**

- **8KB instruction store**

- **Triple address generator with array and stream addressing model**

- **4 KB local store, 64 B output store**

- **Zero-overhead unaligned operand-read and result-write**

- **128-bit SIMD/Reduction data processing pipeline**

- **8KB instruction store**

**August 2008**  
**Hot Chips 20: mediaDSP**
Application Profiling (in silicon)

Hardware profiling is an important part of developing parallel programs.

Flexible profiling scheme can trace both hardware and software-based events.

Trace Buffer converted to waveform file for easy visualization.
X420 mediaDSP Core Physical Specs

• Process: TSMC 65nm CMOS
• Clock Rate: 400 MHz
• 22 independent processing elements per core
• mediaDSP core power (typ):
  ➢ 1.3 Watt (@1V)
• mediaDSP core area:
  ➢ 106 million transistors
• mediaDSP core performance (peak integer ops):
  ➢ 819 GOPs [programmable]
  ➢ 1,052 GOPs [total]
• Greater than 2 TeraOPs (peak) for two cores

X420 Die Plot
Summary/Key Points

• A platform-based approach enables rapid development of video processing solutions.

• Programmability is important for addressing problems that have no unique solution and demand customer-tailored solutions.

• Programming of heterogeneous cores can be simplified by promoting a uniform infrastructure for the cores.

• Heterogeneous approach provides:
  – Programmability where flexibility is required
  – Fixed function to save cost and power
Thank You!

Questions?
Abstract: (1 of 2)

AMD mediaDSP: A Platform for Building Programmable Multicore Video Processors

Large screen LCD TV sets have been selling in higher numbers every year since the arrival of high-definition television (HDTV) content. Recent doubling of the refresh rate of LCD panels has achieved a major improvement in picture quality. Doubling the refresh rate reduces motion blur caused by the LCD being a hold-type display.

When a human observer watches an object move across a display screen, the eye naturally tracks the motion of that object. On a hold-type display, the object is displayed as stationary for 16 ms at a time; the inevitable eye-tracking causes the displayed (still) image to be smeared across the observer's retina. This smearing is perceived as motion blur. Cutting the hold time in half dramatically reduces the smearing effect.

Doubling the LCD panel's refresh rate is, by itself, not sufficient to solve the motion blur problem. Also required is a video processor capable of generating the "in-between" frame in which the object has moved half the distance between where it was in the first frame and where it is going to be in the next frame. The algorithm for generating these intermediate frames is known as "frame rate conversion", and the highest quality algorithms employ motion estimation and compensation techniques.

Frame rate conversion is computationally intensive and is an ill-posed problem, having no globally optimal solution. There is no published algorithm that achieves optimal picture quality across all input sequences, even without regard to implementation complexity. The ability to tweak the algorithmic approach based on the viewing of hours of video is needed to attain maximum picture quality. Thus, an extremely powerful programmable solution is especially attractive.

In January 2008, AMD announced a single-chip solution for Full-HD (1920x1080p) motion-compensated frame rate conversion, which is the result of a joint development effort with Samsung Electronics Company. The AMD Xilleon(tm) X420 chip integrates a Samsung LCD Timing Controller (TCON) with a video processor from AMD to deliver a cost-effective solution for digital TV sets using large-format LCD screens. We believe it is the only device that integrates these two components. Other modules found on the SoC include LVDS input/output interfaces, a stream manager to control capture and display of video frames, and a highly efficient memory controller supporting GDDR3 SDRAMs. The X421 device has also been announced, which has identical capabilities to the X420 but does not include the Samsung TCON.

The video processor of the X420 is constructed using an AMD platform technology known as mediaDSP(tm). MediaDSP is a multi-core platform utilizing application-specific processing elements. To date, this platform has been deployed in products that address two diverse applications: digital audio/video encoding (seven cores) and high-quality motion-compensated frame rate conversion at Full-HD resolution (44 cores; in X420/421).

The modular multi-core architecture of the mediaDSP allows easy customization for specific workloads, and the programmability of the platform enables tuning and algorithmic enhancement after silicon is frozen.

Some of the applications for which the mediaDSP is well suited are:
- motion-compensated frame rate conversion, deinterlacing
- multi-standard audio/video encoding/decoding and transcoding/transrating
- advanced image/video processing, deblocking, noise reduction
- super-resolution
- 2D-to-3D conversion
- gesture recognition, real-time object tracking

(cont.)
Central to the concept of mediaDSP is the notion of exploiting task level parallelism. An algorithm is partitioned into a set of parallelizable tasks and then efficiently mapped to the massively parallel architecture. Data-level parallelism is also exploited heavily inside the individual processors of the system.

The main elements of the mediaDSP platform are the Control Engines, Task-Oriented Engines (TOEs), and on-chip Shared Memory. The Control Engines, which are RISC processors executing the main application, orchestrate the distribution of tasks among the set of TOEs. The set of TOEs is heterogeneous, including both programmable engines and different types of fixed-function engines. The Shared Memory stores data passed between tasks and other miscellaneous program variables.

Incorporated inside each TOE is a task control unit (TCU). The TCU maintains a queue of tasks to be executed by the TOE, and synchronizes with other TCUs or Control Engines in the system via hardware semaphore registers. The TCU contains a “nano-processor” (called a Task Control Processor, or TCP) that off-loads low-level tracking of the TOE from the main control engine microprocessor.

In total, there are 36 TOEs and 8 RISC Control Engines in the X420 chip. Twenty of the TOEs are instances of a general-purpose SIMD DSP engine known as the “Crunch Engine”; the remaining TOEs are application-specific processing elements that execute specialized tasks in the frame rate conversion application. Together, these multiple cores represent an aggregate computational capability in the range of 1 trillion operations per second.

The mediaDSP employs a multi-level memory hierarchy, with software-managed data movement using DMA engines. This memory hierarchy serves two purposes:

- Deterministic execution time, and
- Latency tolerance.

MediaDSP achieves deterministic execution time by explicitly managing data locality rather than relying on hardware caches that can have unpredictable access latencies. The multi-level nature of the memory hierarchy provides the software with buffering space that it can allocate to reduce the effect of the high latency associated with access to an SoC’s DRAM.

One of the key TOEs is the Crunch Engine. The Crunch Engine is a SIMD processor with a 128-bit datapath and a specialized architecture and instruction set for the execution of DSP algorithms on video and image data. It has some novel features for operating on streams of data and for hiding latency. The microarchitecture of this processor will be presented.

The programming of large-scale chip multiprocessors is a difficult and daunting task. The mediaDSP’s extensive re-use of building blocks such as DMA engines and TCUs forms a framework that lends uniformity and structure to the system, thereby simplifying the programmer’s view. Additionally, the deterministic execution time of the mediaDSP’s TOEs and the latency-tolerant multi-level memory hierarchy aid programmers in developing and analyzing applications that must meet real-time execution constraints.

The ability to profile applications while running on silicon is especially important when developing real-time software on a multiprocessing system. The mediaDSP has hardware-supported profiling capabilities that allow programmers to visualize and analyze the level of parallelism that is being achieved. This feedback is essential for tuning applications to attain maximum parallelism.
Trademark Attribution

AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names used in this presentation are for identification purposes only and may be trademarks of their respective owners.

©2008 Advanced Micro Devices, Inc. All rights reserved.