A High-Performance Parallel CAVLC Encoder on a Fine-Grained Many-core System

Zhibin Xiao and Bevan M. Baas

VLSI Computation Lab, ECE Department
University of California, Davis
Outline

- Introduction to H.264 CAVLC Encoder
- Features of Target Fine-Grained Many-Core System
- The Proposed Parallel CAVLC Encoder
- Results and Performance Analysis
- Summary
Advanced Video Processing

Video applications are everywhere: High definition video, realtime video conference, portable handset
Introduction to H.264/AVC Standard

- Drafted on May 2003 from JVT formed by ITU and ISO MPEG organization
- Target from high-definition TV to low-resolution mobile video
- Huge computation complexity with more data dependency and irregular processings
Introduction of H.264 CAVLC Encoder

- Context-adaptive variable-length coding (CAVLC)
  - Adopted in H.264 baseline profile
  - Reverse zigzag scanned run-length coding and adaptive coding table selection
  - Up to 27 4x4 or 2x2 blocks within a macroblock in order

- Less processing regularity
  - Serial in pixel level
  - SIMD approach is not feasible in this case
  - Task-level parallelism is available

![16x16 Macroblock CAVLC Processing Order](image)
Introduction of H.264 CAVLC Encoder

- **CAVLC Encoding**
  - Five parameters of each 4x4 block are coded separately
    - `coeff_token`, `Sign_trail`, `Levels`, `Total_zeros`, `Run_before`

- **CAVLC data-flow graph**
  - Serial scanning phase
  - Parameter coding phase
Outline

- Introduction to H.264 CAVLC Encoder
- **Features of Target Fine-Grained Many-Core System**
- The Proposed Parallel CAVLC Encoder
- Results and Performance Analysis
- Summary
Key features

- 164 Enhanced prog. procs.
- 3 Dedicated-purpose procs.
- 3 Shared memories
- Long-distance circuit-switched communication network
- Dynamic Voltage and Frequency Scaling (DVFS)
Project motivation and mapping methodology

- Fine-grained many-core system for DSP applications
  - energy efficient
  - scalable performance
  - highly flexible

- Mapping methodology
  - Sequential C code
  - Parallel C code
  - Fine-grained assembly-level code
Outline

- Introduction to H.264 CAVLC Encoder
- Features of Target Fine-Grained Many-Core System
- The Proposed Parallel CAVLC Encoder
- Results and Performance Analysis
- Summary
Parallel CAVLC : Memory Optimization

- **Coeff_Token** table selection
  - Encode number of non-zero coefficients (nnz) in current 4x4 block
  - The table index depends on top and left 4x4 blocks
  - A row of nnz values of previous blocks has to be stored in the shared big memory

- Table elimination and compression
  - **Levels** encoded at runtime
  - Reduce more than 75% table memory for **coeff_token**, **total_zeros** and **run_before**
    - Width compression
    - Zero-value reduction

720p HDTV: 324 word memory for nnz
A 20-processor mapping

- No long-distance link
- 8 routing processors
- Every encoding blocks can fit into a fine-grained core (128 word instruction and data memory)
Mapping and Throughput Optimization

- 15-processor mapping
  - 4 long-distance link
  - Reduce 5 routing processors

- Throughput optimization
  - Readjust workload
  - Code optimization
Outline

- Introduction to H.264 CAVLC Encoder
- Features of Target Fine-Grained Many-Core System
- The Proposed Parallel CAVLC Encoder
- Results and Performance Analysis
- Summary
Parallel CAVLC Encoder Performance

- Throughput
  - Five QCIF video test sequences with varying Quantization Parameter (10-40)
  - Scaled performance can achieve 30fps 720p HDTV (1280x720) processing
Performance Comparison with General CPU

- Performance comparison
  - Intel Core 2 Duo, Intel Pentium 4 and Pentium 4 HT
  - Throughput
    - 4.86-6.83 times better
  - Scaled area
    - 20.2 times smaller

![Graph showing processing time comparison](b)
### Performance Comparison: traditional DSPs

- Performance estimation on DSPs
  - CAVLC takes 18.2% computation time for H.264 baseline encoder
  - 1.0-6.15 higher throughput and 6.2 times smaller area compared to TI C642 DSP
- Scaled to 65nm
- More demanding test for our design

<table>
<thead>
<tr>
<th>Platform</th>
<th>Target App.</th>
<th>Processor Type</th>
<th>Tech.</th>
<th>Area (mm²)</th>
<th>Freq. (MHz)</th>
<th>Scaled Area to 65nm (mm²)</th>
<th>Scaled Freq. to 65nm (MHz)</th>
<th>Test Sequence</th>
<th>CAVLC Performance (fps 720p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI C642</td>
<td>CIF 24fps</td>
<td>8-way VLIW CMOS</td>
<td>130nm</td>
<td>72</td>
<td>600</td>
<td>18</td>
<td>1200</td>
<td>50 frames IPPP...PQP=25</td>
<td>28</td>
</tr>
<tr>
<td>ADSP BF561</td>
<td>CIF 30fps</td>
<td>Dual-core DSP</td>
<td>130nm</td>
<td>N/A</td>
<td>600</td>
<td>N/A</td>
<td>1200</td>
<td>N/A</td>
<td>36</td>
</tr>
<tr>
<td>TI C641</td>
<td>QCIF 24.5fps</td>
<td>8-way VLIW CMOS</td>
<td>130nm</td>
<td>72</td>
<td>600</td>
<td>18</td>
<td>1200</td>
<td>100 frames IPPP...PQP=28</td>
<td>7.4</td>
</tr>
<tr>
<td>This work AsAP</td>
<td>720p HDTV 30fps</td>
<td>Array (15 cores) 65nm CMOS</td>
<td>2.89*</td>
<td>1070</td>
<td>2.89*</td>
<td>1070</td>
<td>2 frames IPPQP=20</td>
<td>36.0-41.3</td>
<td></td>
</tr>
</tbody>
</table>
Processor activity type
- Execution
- Stalls on input or output

Analysis
- Data receiving stall on output
- 7%-65% active time for most processors
- Bottleneck: zigzag reorder and CAVLC scanning, over 94% active time

Power estimation
- One processor
  - 59mW@1.07GHz, 1.3V, 65nm 100% active
  - Nearly zero leakage when processor is idle
- 323mW@1.07GHz, 1.3V, 15-processor + memory
Outline

- Introduction to H.264 CAVLC Encoder
- Features of Target Fine-Grained Many-Core System
- The Proposed Parallel CAVLC Encoder
- Results and Performance Analysis
- Summary
Summary

- Fine-grained many-core system
  - Energy efficient, scalable and flexible
  - Exploiting task-level parallelism

- The proposed parallel CAVLC encoder
  - 15-processor plus 324 word memory, 720p HDTV at 30 fps
  - 4.86-6.83 times higher scaled throughput than latest general-purpose processor
  - 1.0-6.15 higher scaled throughput and 6.2 times smaller area compared with traditional DSPs

- Future work
  - Further power reduction using DVFS
  - A complete parallel H.264 baseline encoder
Acknowledgments

- Intellasys Inc.
- SRC GRC Grant 1598 and CSR Grant 1659
- ST Microelectronics
- NSF Grant 0430090 and CAREER Award 0546907
- Intel and S Machines Corporation
- UC Micro
- UCD Faculty Research Grant
Thank You!