Low Power Design of the Next-Generation High Efficiency Video Coding

Authors: Muhammad Shafique, Jörg Henkel
Outline

- Introduction to the High Efficiency Video Coding (HEVC)
- HEVC Analysis
  - complexity, memory access, thermal
- Power-Efficient HEVC System Design
- Conclusion
High Efficiency Video Coding (HEVC)

- **Ultra-HD (or supervision)**
  - 7680×4320 ≈ 33 million pixels per frame
  - By 2017: 80% – 90% global internet traffic

- **New video compression standards/techniques required**

- **JCT-VC’s High Efficiency Video Coding (HEVC)**
  - ~2× compression efficiency compared to H.264

- **Full HD @ 30fps**
  - 1 second ≈ 712 Mbits
  - 1 hour ≈ 2.4 Tbits

---

**Diagram:**

(a) Time vs. Bitrate for different video scenes:
- Basketball
- Kimono
- PeopleOnStreet

(b) Memory BW. [GB/s] for different resolutions:
- HD720
- HD1080
- 2K

Normalized values for:
- HEVC
- H.264/AVC
Challenges for Developing HEVC-based Multimedia Systems

Challenges & Requirements

- **Compute Complexity**
  - Content-Awareness, HW-SW Collaboration, Many-core Systems

- **Power Efficiency**
  - Accelerator Design, Content-Awareness, Power-Gating

- **Thermal Management**
  - Thermal Analysis, Configurations, Content-Adaptive

- **Parallelization**
  - Workload Balancing, Arch.-Awareness, Power Budgeting

- **Video Memory**
  - Memory Hierarchy Design, Content-Aware
HEVC Overview: Encoding Flow

- Input Video in CTUs
- Recursive CU/PU Size Reduction
- Intra Prediction
- Inter Prediction
- Recursive TU Size Reduction
- Transform and Quantization
- Inverse Transform and Quantization
- Decoding and SAO Filter
- Output Reconstructed Video
- CABAC Entropy Coder
- Output Bitstream
HEVC Overview: Slices and Tiles

Slice 0
Slice 1
Slice 2
Slice 3

Tile 0
Tile 1
Tile 2
Tile 3
Tile 4
Tile 5

Core 0
Core 1
Core K-1

f 0
f 1
f K-1

GOP 0

F 0
F M-1

T 0
T 1
T K-1

HEVC Parallel Encoding
HEVC Overview: Tree-Block Structure

CTU_0

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>4</th>
<th>5</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

CTU_1

Example PU Configuration

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Tested TU Configurations

Example CU Configuration
CTU Distribution
HEVC Overview: Intra and Inter Prediction

**HEVC Intra Prediction**

Vertical Angular Predictors

Horizontal Angular Predictors

0: Planar
1: DC

\[ \alpha = \sum_{i=0}^{\log_2 M - 2} \left( 2^{2i} \times N_i \right) \]

**HEVC Inter Prediction**

\[ \beta = 13 \times \sum_{i=0}^{\log_2 M - 3} \left( 2^{2i} \right) \]

— HEVC-Intra: \(~2.56\times\) more mode decisions than H.264
— HEVC-Inter: \(~2.2\times\) more complex than H.264
HEVC Overview: Motion Estimation

- Block Matching (BM) or Motion Estimation (ME)
  - Compression by searching temporal neighbors
  - High energy/time, high compression efficiency (H.264-Inter, HEVC-Inter)

![Diagram showing Motion Estimation process](image)

- Reference Frame
- Current Frame
- Residue Frame

- Motion Vector
- Best Matching
- Search Window
- Previous Frame
- Current Frame
- Current Block
HEVC Overview: Search Data Fetching

- High leakage
- High dynamic
- External Memory (DRAM)

- High bus power
- External Memory Bus

- Very high leakage
- On-Chip Memory (SRAM)

A memory subsystem with low power consumption and high efficiency is required…
Outline

- Introduction to the High Efficiency Video Coding (HEVC)
- HEVC Analysis
  - complexity, memory access, thermal
- Power-Efficient HEVC System Design
- Conclusion
Early PU size prediction may provide significant reduction in computational and energy requirements.
HEVC Analysis: CTU Distribution
HEVC Analysis: Memory Accesses

- Memory Access for Motion Estimation
  - Memory accesses of HEVC ≈ 3.86× of H.264
  - Most of the on-chip memory is wasted (leakage power)

(a)

- Only a part of the full search window is utilized
- Adapting the search window size at run-time provides increased potential for leakage power savings
Using a thermal camera setup

DIAS Pyroview thermal camera operates at 50Hz with spatial resolution of 50 µm

Copyright: © Chair for Embedded Systems (CES), Karlsruhe Institute of Technology (KIT), Germany
Temperature Measurements for HEVC
[RaceHorses@37QP vs. 22QP]

Temp max.: 55.0°C  
Temp min.: 36.0 °C  
Temp avg.: 53.0 °C

Temp max.: 53.0°C  
Temp min.: 35.0 °C  
Temp avg.: 49.0 °C

hevcDTM @ DATE’14

Copyright: © Chair for Embedded Systems (CES), Karlsruhe Institute of Technology (KIT), Germany
So What is Required?

Interplay between Software and Hardware needs to be explored for power/energy optimization

1. Optimized Algorithms for Fast Intra- and Inter-Prediction
2. Energy-Efficient Hardware Accelerators
3. Energy-Efficient Video Memory Heirarchy
4. Content-Adaptive Power Management
Outline

- Introduction to the High Efficiency Video Coding (HEVC)

- HEVC Analysis
  - complexity, memory access, thermal

- Power-Efficient HEVC System Design

- Conclusion
Power Efficient HEVC Design: Hardware Architecture

HEVC Software Layer

Application Driven Adaptive Power/Thermal Manager

Video Tile Formation

Energy to Quality Tradeoff

Data Analysis and Statistics

HEVC Encoding Intra/Inter

Complexity Reduction Scheme

Adaptive Workload Budgeting

HEVC Hardware Processing Architecture

Feedback Monitors to Software

Off-chip DRAM

Battery

CT

R

CT

R

CT

R

CT

R

CT

R

CT

R

CT

R

Off-chip DRAM
Variance and Motion based Classification
Complexity Reduction: PU Size Estimation

CTU variance computation at 4×4

Recursive 4 neighbors merge

PU Map (PUM)

PU Map Above (PUMA)

HEVC CTU Compressor

\[ v = \frac{1}{n-1} \sum_{i=0}^{n-1} (x_i - \mu_x)^2 \]

\[ v_c = \text{CombineVariances}(v_{i,i\in\{1,2,3,4\}}) \]

if \( v_c < v_{Th} \) OR \( v_{i,i\in\{1,2,3,4\}} < v_{Th} \)

\( \text{MergeBlocks}(\ ) \)

\( \mu_v = \text{Mean of variance curve} \)

\( \Delta = \text{CDF threshold (0.8)} \)

\( H = \text{Size of PU to combine} \)

\( v_{Th} = \]

Rayleigh CDF Analysis

Empirical Analysis
## Time Savings and Video Quality Results

<table>
<thead>
<tr>
<th></th>
<th>Class</th>
<th>Size</th>
<th>BD-PSNR</th>
<th>BD-Rate</th>
<th>Traffic</th>
<th>BasketballDrive</th>
<th>BasketballDrill</th>
<th>BQSquare</th>
<th>RaceHorses</th>
<th>Johnny</th>
<th>BasketballDrillText</th>
</tr>
</thead>
<tbody>
<tr>
<td>RaceHorses</td>
<td>D</td>
<td>WQVGA</td>
<td>-0.03009</td>
<td>0.4482</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Johnny</td>
<td>E</td>
<td>720p</td>
<td>-0.08711</td>
<td>2.1241</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BasketballDrillText</td>
<td>F</td>
<td>WVGA</td>
<td>-0.05827</td>
<td>1.1123</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tile Mapping and Parallelization

**Workload is not equal for tiles**

- **Cores**
  - Core₀
  - Core₁

- **CPUs Max freq.** $f_{\text{max}}$
- **Frame Rate** $f_p$

- **Tile Formation and Maximum Workload Estimator**

- **Workload (per core)**
- **Frequency (per CPU)**

- **Offline Tuning**

- **Video Input**
- **Frame Rate** $f_p$
- **CPUs Max freq.** $f_{\text{max}}$

- **Core Frequency Selector**
- **Workload Allocator**
- **Monitoring Unit**
- **Threshold Generator**
- **Workload Adaptation**

- **Total Intra Angles** ($\theta$)

- **Core frequency selection**
- **Intra Mode Prediction**
- **User bit-rate tolerance** $n$

**Tile 0**

**Tile 1**

**Tile 3**

**Tile 4**

**Time [msec]**

**Frame**

- **Unit**
- **Generator**
- **Workload Adaptation**
- **Total Intra Angles** ($\theta$)
HEVC Thermal Management

Application-Driven DTM

HEVC Encoder

Extract Motion Intensity

Frequency scaling

Execute HM

Core0 Sensor

Core1 Sensor

∀n ∈ M

\[ D_n = \sqrt{v_x n^2 + v_y n^2} \]

\[ D = \sum_{\forall n \in M} D_n / \text{size}(M) \]

\[ \psi = \begin{cases} 
\text{Low Motion (LM)} & \text{if } (D \leq Th_{D1}) \\
\text{Medium Motion (MM)} & \text{if } (Th_{D1} < D \leq Th_{D2}) \\
\text{High Motion (HM)} & \text{if } (D > Th_{D2}) 
\end{cases} \]

T_{current} > T_{critical}
HEVC Thermal Management

![Temperature Distribution](image)

- Temperature (°C)
  - Max
  - Average
  - Min

![PSNR and Bit Rate](image)

- PSNR (dB)
- Bit rate (kbps)

**No DTM**
- DTM 54°C
- DTM 50°C
- DTM 46°C

Shafique @ ASPDAC, Jan. 2014
HEVC Thermal Management

Keiba

BasketballDrill
Power Efficient HEVC Design: Hardware Architecture
Hardware Accelerators

Half filter: 
\[ \text{Half filter} = -1^*s_0 + 4^*s_1 - 11^*s_2 + 40^*s_3 = s_1\ll2 + s_3\ll5 + s_3\ll3 - (s_0 + s_2\ll3 + s_2\ll1 + s_2) \]

Quarter filter: 
\[ \text{Quarter filter} = -1^*s_0 + 4^*s_1 - 10^*s_2 + 58^*s_3 = s_1\ll2 + s_3\ll6 + s_3\ll1 - (s_0 + s_2\ll1 + s_2\ll3 + s_3\ll3) \]

Quarter filter: 
\[ \text{Quarter filter} = 1^*s_0 - 5^*s_1 + 17^*s_2 = s_0 + s_3\ll1 + s_3\ll4 - (s_1 + s_1\ll2), s_3=0 \]

>>>6 operation is applied for some quarter-pel

Legend:
- Blue: Slice LUTs (luma)
- Purple: Slice LUTs (chroma)
- Red: Slice registers (luma)
- Green: Slice registers (chroma)
- Orange: Occupied Slices (luma)
- Grey: Occupied Slices (chroma)
AMBER: Memory Subsystem

- External Memory holds the current frame
  - High density, low read and write power
- On-chip SRAM memory (FIFO) holds only the current block
  - High read and write speed and low dynamic write power
  - Hides latencies from HEVC engine
AMBER: MRAM Reference Buffers

- One MRAM buffer holds a full reference frame
- Each column (sector) of reference buffer is power-gated
- Reference read and write masters read and write data to the MRAM buffer
AMBER: Reference Buffer Power Management

- Observation: Not all of the search window is used
  - Block matching algorithm accesses only a small percentage of reference buffer sectors
- Power-gate unused sectors
  - Reduce leakage

\[ S_1 = C_{UL} \]
\[ S_2 = C_{UL} \]

Prediction of Unused Sectors is based on:
1. Self-Organizing Map
2. Content Properties
— Increasing the number of reference frames improves the power consumption of the AMBER system compared to the search window approach
Conclusion

- Comprehensive analysis of HEVC
  - Architecture, power, thermal and complexity

- Challenges posed by HEVC
  - Architectural (memory, reconfiguration, accelerators)
  - Power/thermal (power-gating, configuration control)
  - Complexity (parallelization, many-core, workload balancing)

- Both Hardware and Software need to be optimized while leveraging the application-specific knowledge

- Our approach
  - Adaptive complexity management
  - Video tiling, workload budgeting, CU/PU partitioning
  - Power and thermal aware HEVC configuration
  - Hybrid video memory hierarchy with content-driven power-gating
ces265: Multi-threaded HEVC Encoder

- Open-source
- C++ based
- Multithreading via pthread API
- One thread of ces265 ≈ 13.2 × faster than HM-9.2

Web
- http://ces.itec.kit.edu/ces265/

Download
- https://sourceforge.net/projects/ces265/
Acknowledgement

Muhammad Usman Karim Khan
Daniel Palomino
Claudio M. Diniz
Felipe Sampaio
Thank you!

Questions?

Web: http://ces.itec.kit.edu/ces265/
Download: https://sourceforge.net/projects/ces265/