A Near Optimal Debloating Filter for H.264 Advanced Video Coding

S-Y. Shih  C-R. Chang  Y-L. Lin
Department of Computer Science
National Tsing Hua University
Hsin-Chu, TAIWAN 300

2006 ASP-DAC, Yokohama, JAPAN
Team of Students

- H.264 decoding prototype in IP-based fashion

<table>
<thead>
<tr>
<th>Team Member</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>張正儒</td>
<td>System Integration</td>
</tr>
<tr>
<td>陳建文</td>
<td>Context-Based Adaptive Binary Arithmetic Decoding</td>
</tr>
<tr>
<td>曾煥鈞</td>
<td>Motion Compensation</td>
</tr>
<tr>
<td>邱俊霖</td>
<td>Inverse Quantization &amp; Inverse Discrete Cosine Transform</td>
</tr>
<tr>
<td>石勝宇</td>
<td>Deblocking Filter</td>
</tr>
<tr>
<td>張元鴻</td>
<td>Intra-Frame Prediction</td>
</tr>
<tr>
<td>駱子仁</td>
<td>Slice-Level Entropy Decoding</td>
</tr>
</tbody>
</table>
Outline

- Introduction
- Previous Work
- Proposed Architecture
- Experimental Results
- Conclusion
Introduction

- Deblocking filter is used to eliminate blocking artifact
- It achieves up to 0.5db PSNR improvement
Profiling Result

Foreman.cif 300 frames
JM 8.3 on P4 2.8G
512 MB DDR SDRAM
GOP: IBBPBBBP
Reference frame: 5
Search range: 16
QP: 28
Deblocking Filter I/O

Unfiltered Macroblock

Unfiltered 8 pixels

Unfiltered 8 pixels

Reconstructed picture

Coding Information

Get Boundary Strength

Boundary Strength

QP

Get Threshold

α, β, and indexA

Deblocking Filter

Filtered 8 pixels (middle 6 pixels changed)
Filtering Order (Raster Scan)
Filter Order : Luma Vertical Edges

\[ p_3 \quad p_2 \quad p_1 \quad p_0 \quad q_0 \quad q_1 \quad q_2 \quad q_3 \]
Filter Order : Luma Horizontal Edges
Filter Order : Chroma Edges

8 Pixels

Horizontal filtering across vertical edges

Vertical filtering across horizontal edges

Edge 0

Edge 1
Deblocking Filter I/O

Unfiltered Macroblock

Unfiltered 8 pixels

Unfiltered Macroblock

Unfiltered 8 pixels

Boundary Strength

Boundary Strength

Coding Information

QP

Get Boundary Strength

Get Threshold

\( \gamma, \beta \), and indexA

Deblocking Filter

Reconstructed picture

Filtered 8 pixels (middle 6 pixels changed)
Boundary Strength (bS) Determination

start

One of the blocks is Intra && the edge is MB edge

No

One of the blocks is Intra

Yes

One of the blocks has non-zero trans. coeff.

Yes

No

4x4 block

\[
\begin{array}{c}
p \\
q
\end{array}
\]

Current MB

Yes

No

Diff. frames or Diff. of MVs >= 4

Yes

bS = 1

No

bS = 0

end

\[
\begin{array}{c}
bS = 4 \\
bS = 3 \\
bS = 2 \\
bS = 1 \\
bS = 0
\end{array}
\]
Boundary Strength (1/3)

- Boundary strength is used to determined the strength of filtering process

- Boundary strength (bS) ranges between 0 (without filtering) and 4 (strongest filtering)
Boundary Strength (2/3)

- bS value for the process of horizontal filtering across vertical edges
Boundary Strength (3/3)

- bS value for the process of vertical filtering across horizontal edges

```
bs0' bs1' bs2' bs3'
bs4' bs5' bs6' bs7'
bs8' bs9' bs10' bs11'
bs12' bs13' bs14' bs15'
```

```
bs0' bs1' bs2' bs3'
bs4' bs5' bs6' bs7'
bs8' bs9' bs10' bs11'
bs12' bs13' bs14' bs15'
```

Edge 0
Edge 1
Edge 2
Edge 3

Edge 0
Edge 1
Edge 0
Edge 1
Deblocking Filter I/O

Unfiltered Macroblock

Unfiltered 8 pixels

Boundary Strength

Get Boundary Strength

Get Threshold

Coding Information

QP

Deblocking Filter

Filtered 8 pixels (middle 6 pixels changed)
Threshold

- \( q_{Pav} = (q_{Pp} + q_{Pq} + 1) >> 1 \)
- indexA = Clip3(0, 51, q_{Pav} + FilterOffsetA)
- indexB = Clip3(0, 51, q_{Pav} + FilterOffsetB)
- \( \alpha = \text{ThresholdTable}(\text{indexA}) \)
- \( \beta = \text{ThresholdTable}(\text{indexB}) \)

**filterSamplesFlag** = 1 if the following conditions all hold
- \( bS \neq 0 \)
- \(|p0-q0| < \) □
- \(|p1-p0| < \) □
- \(|q1-q0| < \) □
Deblocking Filter I/O

Unfiltered Macroblock

Coding Information
Get Boundary Strength

Unfiltered 8 pixels
Boundary Strength

QP
Get Threshold
α, β, and indexA

Deblocking Filter

Reconstructed picture

Filtered 8 pixels (middle 6 pixels changed)
Filter Flow Chart

start

Get pixels \((p3, p2, p1, p0, q0, q1, q2, q3)\)
Get bS, \(\theta\), \(\phi\), and indexA

N

filterSamplesFlag==1?

Y

bS==4?

Y

\(p'0 = p0bSLT4(p1, p0, q0, q1)\)
\(q'0 = q0bSLT4(p1, p0, q0, q1)\)
\(p'1 = (\text{chromaEdgeFlag}==0 && |p2-p0|<\theta)?\)
\(p1bSLT4(p2, p1, p0, q0) : p1\)
\(q'1 = (\text{chromaEdgeFlag}==0 && |q2-q0|<\theta)?\)
\(q1bSLT4(p0, q0, q1, q2) : q1\)

To next page

From next page
Filter Flow Chart (Cont.)

From previous page

N

chromaEdgeFlag==0 and
(|p2-p0|< Δ && |p0-q0|<(Δ/4+2))

Y

p’0 = p0bSEQ4HL(p2, p1, p0, q0, q1)
p’1 = p1bSEQ4HL (p2, p1, p0, q0)
p’2 = p2bSEQ4HL (p3, p2, p1, p0, q0)

p’0 = p0bSEQ4NHL(p1, p0, q1)

N

chromaEdgeFlag==0 and
(|q2-q0|< Δ && |q0-p0|<(Δ/4+2))

Y

q’0 = q0bSEQ4HL(p1, p0, q0, q1, q2)
q’1 = q1bSEQ4HL (p0, q0, q1, q2)
q’2 = q2bSEQ4HL (p0, q0, q1, q2, q3)

q’0 = q0bSEQ4NHL(p1, q0, q1)

To previous page
Equations for bS < 4

- \( tc0 = \text{ClipVarTable}(bS, \text{indexA}) \)
- \( tc = (\text{chromaEdgeFlag}==1) \) ?
  (\( tc0+1 \)) : \( tc + ((|p2-p0|<\Box)|1:0) + ((|q2-q0|<\Box)|1:0) \)
- \( \text{delta} = \text{Clip3(tc,-tc,(((q0-p0)<<2)+(p1-q1)+4)>>3)) \)
- \( \text{p0bSLT4(p1, p0, q0, q1)} = \text{Clip1(p0+delta)} \)
- \( \text{q0bSLT4(p1, p0, q0, q1)} = \text{Clip1(q0-delta)} \)
- \( \text{p1bSLT4(p2, p1, p0, q0)} = \)
  \( p1 + \text{Clip3(}-tc0,tc0,(p2+((p0+q0+1)>>1)-(p1<<1))>>1) \)
- \( \text{q1bSLT4(p0, q0, q1, q2)} = \)
  \( q1 + \text{Clip3(}-tc0,tc0,(q2+((p0+q0+1)>>1)-(q1<<1))>>1) \)
Equations for bS == 4

- \( p0bSEQ4HL(p2, p1, p0, q0, q1) = (p2+2*p1+2*p0+2*q0+q1+4) >> 3 \)
- \( q0bSEQ4HL(p2, p1, p0, q0, q1) = (p1+2*p0+2*q0+2*q1+q2+4) >> 3 \)
- \( p1bSEQ4HL(p2, p1, p0, q0) = (p2+p1+p0+q0+2) >> 2 \)
- \( q1bSEQ4HL(p0, q0, q1, q2) = (p0+q0+q1+q2+2) >> 2 \)
- \( p2bSEQ4HL(p3, p2, p1, p0, q0) = (2*p3+3*p2+p1+p0+q0+4) >> 3 \)
- \( q2bSEQ4HL(p0, q0, q1, q2, q3) = (2*q3+3*q2+q1+q0+p0+4) >> 3 \)
- \( p0bSEQ4NHL(p1, p0, q1) = (2*p1+p0+q1+2) >> 2 \)
- \( q0bSEQ4NHL(p1, q0, q1) = (2*q1+q0+p1+2) >> 2 \)
Outline

- Introduction
- Previous Work
- Proposed Architecture
- Experimental Results
- Conclusion
Previous Work (1/3)

- Two main streams
  - Platform-based
    - Load pixels of current macroblocks from the SDRAM
  - Hard wired
    - Load pixels of current macroblocks from modules or buffers prior to the deblocking filter
Previous Work (2/3)

- Platform-based designs

<table>
<thead>
<tr>
<th></th>
<th>Chen’s ‘03</th>
<th>Zhang’s ‘04</th>
<th>Ikenaga’s ‘05</th>
<th>Ling’s ‘04</th>
<th>Chang’s ‘05</th>
<th>Chiang’s ‘05</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles/MB</td>
<td>614</td>
<td>584</td>
<td>566</td>
<td>510</td>
<td>386</td>
<td>Max 374, Min 50, Avg. 86-244</td>
</tr>
<tr>
<td>Filter Cycles/MB</td>
<td>240</td>
<td>214</td>
<td>192</td>
<td>136</td>
<td>336</td>
<td>0 - 374</td>
</tr>
<tr>
<td>SRAM for Pixels</td>
<td>Dual 96x32</td>
<td>Dual 16x32</td>
<td>8 Dual 80x8</td>
<td>Dual 88x32</td>
<td>Single 80x32</td>
<td>Single 96x32</td>
</tr>
<tr>
<td></td>
<td>Dual 64x32</td>
<td></td>
<td></td>
<td>Dual 72x32</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Single 32x32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># 4x4 Registers</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>11</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td># of Edge Filter</td>
<td>1</td>
<td>1</td>
<td>1 Pipelined</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Capability 1280x720p</td>
<td>45.2 fps</td>
<td>47.5 fps</td>
<td>49.1 fps</td>
<td>54.46 fps</td>
<td>71.9 fps</td>
<td>113.8 – 322.9 fps</td>
</tr>
<tr>
<td>@100MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process (um)</td>
<td>.25</td>
<td>N/A (FPGA)</td>
<td>.35</td>
<td>N/A (FPGA)</td>
<td>.18</td>
<td>.18</td>
</tr>
<tr>
<td>Gate Count</td>
<td>20.66K</td>
<td>N/A (FPGA)</td>
<td>9.35K</td>
<td>N/A (FPGA)</td>
<td>19.2K</td>
<td>21.8K</td>
</tr>
</tbody>
</table>

*RAM is not included in Gate Count
Previous Work (3/3)

- Hard wired designs

<table>
<thead>
<tr>
<th></th>
<th>Wu’s ‘04</th>
<th>Yu’s ‘04</th>
<th>Lee’s ‘05</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles/MB</td>
<td>446</td>
<td>288</td>
<td>250</td>
</tr>
<tr>
<td>Filter Cycles/MB</td>
<td>192</td>
<td>236</td>
<td>250</td>
</tr>
<tr>
<td>SRAM for Pixels</td>
<td>Dual 64x32 2 Two 96x32</td>
<td>Dual 96x32 Dual 64x32 Single (2xFrameWidth)x32</td>
<td>2 Single 96x32 Single (2xFrameWidth+20)x32</td>
</tr>
<tr>
<td># 4x4 Registers</td>
<td>8</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td># of Edge Filter</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Capability 1280x720p @100MHz</td>
<td>62.3 fps</td>
<td>97.1 fps</td>
<td>111.1 fps</td>
</tr>
<tr>
<td>Process (um)</td>
<td>.25</td>
<td>.18</td>
<td>.18</td>
</tr>
<tr>
<td>Gate Count</td>
<td>24K</td>
<td>19.5K</td>
<td>19.6K</td>
</tr>
</tbody>
</table>

*RAM is not included in Gate Count
Outline

- Introduction
- Previous Work
- Proposed Architecture
- Experimental Results
- Conclusion
Proposed Architecture
Memory Organization

FrameWidth

FrameHeight

reconstruct mem
Single-port 96 x 32 bits

local mem 0
Two-port 32 x 32 bits

local mem 1 Single-port
(1.5\times\text{FrameWidth}) \times 32 \text{ bits}

FrameWidth/2

FrameHeight/2

FrameHeight/2

FrameHeight/2

ASPDAC 2006/01/25

NTHU-CS-YLLIN

30
Proposed Filter Order

- Keep right most column
- Reduce local memory usage
Proposed Pipelined Filter

Stage 1
Read pixels

Stage 2
Calculate parameters

Stage 3
bS=4 filtering or calculate delta

Stage 4
bS<4 filtering

Stage 5
Pipeline Hazard

Blocks for next filtering

Blocks in pipeline
Resolving Hazard by Forwarding

Stage 1: Read pixels
- r0p3, r0p2, r0p1, r0p0
- r0q0, r0q1, r0q2, r0q3

Stage 2: Calculate parameters
- p3fwd, r1p3, r1p2, r1p1, r1p0
- r1q0, r1q1, r1q2, r1q3
- bS=4 filtering or calculate delta

Stage 3: bS=4 filtering or calculate delta
- r2p3, r2p2, r2p1, r2p0
- r2q0, r2q1, r2q2, r2q3
- bS<4 filtering

Stage 4: bS<4 filtering
- r3p3, r3p2, r3p1, r3p0
- r3q0, r3q1, r3q2, r3q3

Stage 5:
Forwarding for Filtering Step 6 (1/4)
Forwarding for Filtering Step 6 (2/4)
Forwarding for Filtering Step 6 (3/4)
Forwarding for Filtering Step 6 (4/4)
Filtering Speed (Near Optimal)

Read Coding Information
&
Generate bS

Filter
&
Write Back

Write Col.

14
192
200
32
Outline

- Introduction
- Previous Work
- Proposed Architecture
- Experimental Results
- Conclusion
UniChip Multimedia SOC Platform

AHB

NTHU-CS-YLLIN

ASPDAC 2006/01/25


H.264/AVC Decoder IP Design Flow

Design Specification

Profiling

IP design

IP RTL Verification
IP FPGA Emulation
IP Platform Prototyping
IP Deliverables

System Design

IP Integration
Sys RTL Verification
Sys FPGA Emulation
Sys Platform Prototyping

ASPDAC 2006/01/25
NTHU-CS-YLLIN
SOC Platform Design Flow

- **AMBA wrapper**
  - FPGA synthesis
    - Netlist
    - FPGA P&R
      - Bit file
        - Downloader
        - Circuit in FPGA
  - AMBA

- **HW design**
  - Driver
    - Assembler
    - User constraint

- **C library**
  - Compiler
    - Object file
      - Linker
        - Axf file
        - AXD
        - Program in core
IP Qualification

- Code coverage (Verification Navigator)

<table>
<thead>
<tr>
<th></th>
<th>Our Design</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement Coverage</td>
<td>99.8%</td>
<td>95%</td>
</tr>
<tr>
<td>Branch Coverage</td>
<td>99.1%</td>
<td>95%</td>
</tr>
<tr>
<td>Toggle Coverage</td>
<td>99.0%</td>
<td>95%</td>
</tr>
<tr>
<td>Path Coverage</td>
<td>74.0%</td>
<td>50%</td>
</tr>
</tbody>
</table>

- DFT (TetraMax)

<table>
<thead>
<tr>
<th></th>
<th>Test Patterns</th>
<th>Fault Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Faults</td>
<td>69,272</td>
<td>376</td>
</tr>
<tr>
<td>Test Patterns</td>
<td>376</td>
<td>99.91%</td>
</tr>
</tbody>
</table>
### Comparison with Platform-based Designs

<table>
<thead>
<tr>
<th></th>
<th>Chen’s ‘03</th>
<th>Zhang’s ‘04</th>
<th>Ikenaga’s ‘05</th>
<th>Ling’s ‘04</th>
<th>Chang’s ‘05</th>
<th>Chiang’s ‘05</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cycles/MB</strong></td>
<td>614</td>
<td>584</td>
<td>566</td>
<td>510</td>
<td>386</td>
<td>Max 374</td>
<td>214</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Min 50</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. 86-244</td>
<td></td>
</tr>
<tr>
<td><strong>Filter Cycles/MB</strong></td>
<td>240</td>
<td>214</td>
<td>192</td>
<td>136</td>
<td>336</td>
<td>0 - 374</td>
<td>200</td>
</tr>
<tr>
<td><strong>SRAM for Pixels</strong></td>
<td>Dual 96x32, Dual 64x32</td>
<td>Dual 96x32, Dual 64x32</td>
<td>8 Dual 80x8, Dual 72x32</td>
<td>Single 80x32, Single 32x32</td>
<td>Single 96x32</td>
<td>Single 96x32, Two 32x32, Single (1.5xFrameWidth) x32</td>
<td></td>
</tr>
<tr>
<td><strong># 4x4 Registers</strong></td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong># of Edge Filters</strong></td>
<td>1</td>
<td>1</td>
<td>1 Pipeline</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1 Pipeline</td>
</tr>
<tr>
<td><strong>Capability</strong></td>
<td>45.2 fps</td>
<td>47.5 fps</td>
<td>49.1 fps</td>
<td>54.46 fps</td>
<td>71.9 fps</td>
<td>113.8 – 322.9 fps</td>
<td>128.6 fps</td>
</tr>
<tr>
<td><strong>1280x720p @100MHz</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Process (um)</strong></td>
<td>.25</td>
<td>N/A (FPGA)</td>
<td>.35</td>
<td>N/A (FPGA)</td>
<td>.18</td>
<td>.18</td>
<td>.18</td>
</tr>
<tr>
<td><strong>Gate Count</strong></td>
<td>20.66K</td>
<td>N/A (FPGA)</td>
<td>9.35K</td>
<td>N/A (FPGA)</td>
<td>19.2K</td>
<td>21.8K</td>
<td>20.9K</td>
</tr>
</tbody>
</table>

*RAM is not included in Gate Count

ASPDAC 2006/01/25  
NTHU-CS-YLLIN
# Comparison with Hard Wired Designs

<table>
<thead>
<tr>
<th></th>
<th>Wu’s ‘04</th>
<th>Yu’s ‘04</th>
<th>Lee’s ‘05</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles/MB</td>
<td>446</td>
<td>288</td>
<td>250</td>
<td>214</td>
</tr>
<tr>
<td>Filter Cycles/MB</td>
<td>192</td>
<td>236</td>
<td>250</td>
<td>200</td>
</tr>
<tr>
<td>SRAM for Pixels</td>
<td>Dual 64x32</td>
<td>Dual 96x32</td>
<td>2 Single 96x32</td>
<td>Single 96x32</td>
</tr>
<tr>
<td></td>
<td>2 Two 96x32</td>
<td>Single (2xFrameWidth)x32</td>
<td>Single (2xFrameWidth +20)x32</td>
<td>Two 32x32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Single (1.5xFrameWidth)x32</td>
</tr>
<tr>
<td># 4x4 Registers</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td># of Edge Filter</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1 Pipelined</td>
</tr>
<tr>
<td>Capability</td>
<td>62.3 fps</td>
<td>97.1 fps</td>
<td>111.1 fps</td>
<td>128.6 fps</td>
</tr>
<tr>
<td>1280x720p @100MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process (um)</td>
<td>.25</td>
<td>.18</td>
<td>.18</td>
<td>.18</td>
</tr>
<tr>
<td>Gate Count</td>
<td>24K</td>
<td>19.5K</td>
<td>19.6K</td>
<td>20.9K</td>
</tr>
</tbody>
</table>

*RAM is not included in Gate Count
Outline

- Introduction
- Previous Work
- Proposed Architecture
- Experimental Results
- Conclusion
Summary

- Improve memory organization for saving local memory usage
  - Store half of top chroma 4x4 blocks
- New filter order for saving local memory usage and data loading time
  - Keep right most column of current MB
- Pipelined filter
  - Increase clock frequency
Future Work

- Integrate the deblocking filter into our H.264 encoder and CODEC

- Reduce power consumption of the proposed architecture

- Decoder for Super HDTV (7680x4320)
Thank You!!