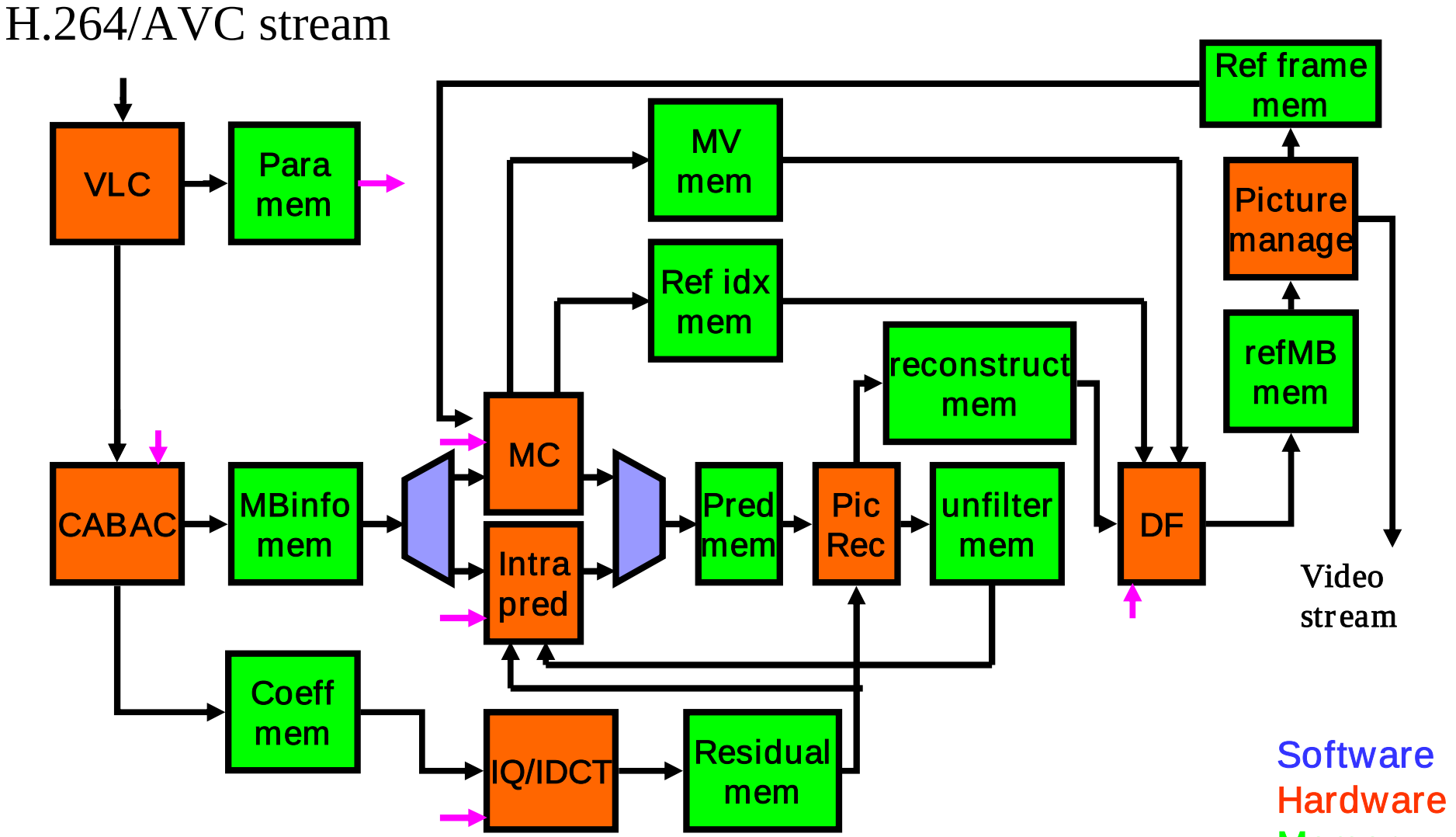
A Near Optimal Deblocking Filter for H.264 Advanced Video Coding

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

NTHU H.264/AVC Decoder



Team of Students

- H.264 decoding prototype in IP-based fashion

Team Member	Task
張正儒	System Integration
陳建文	Context-Based Adaptive Binary Arithmetic Decoding
曾煥鈞	Motion Compensation
邱俊霖	Inverse Quantization & Inverse Discrete Cosine Transform
石勝宇	Deblocking Filter
張元鴻	Intra-Frame Prediction
駱子仁	Slice-Level Entropy Decoding

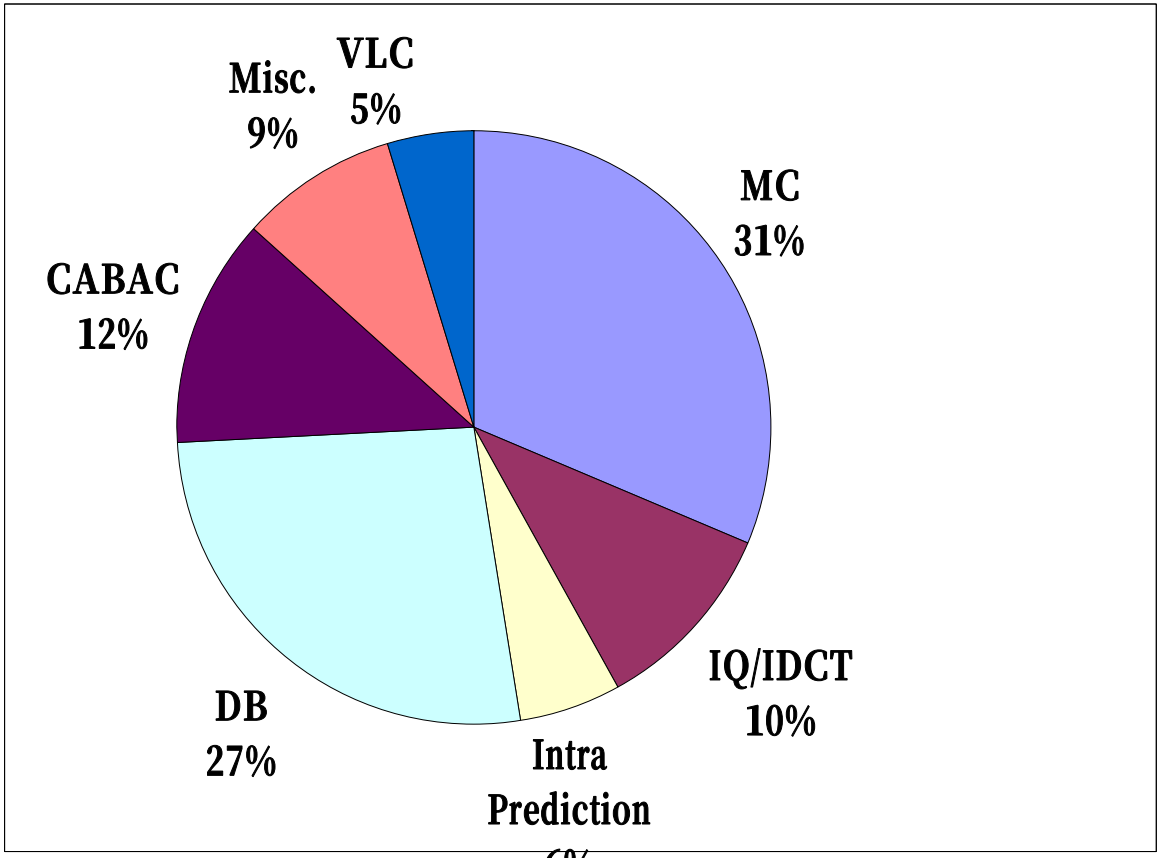
Outline

- 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

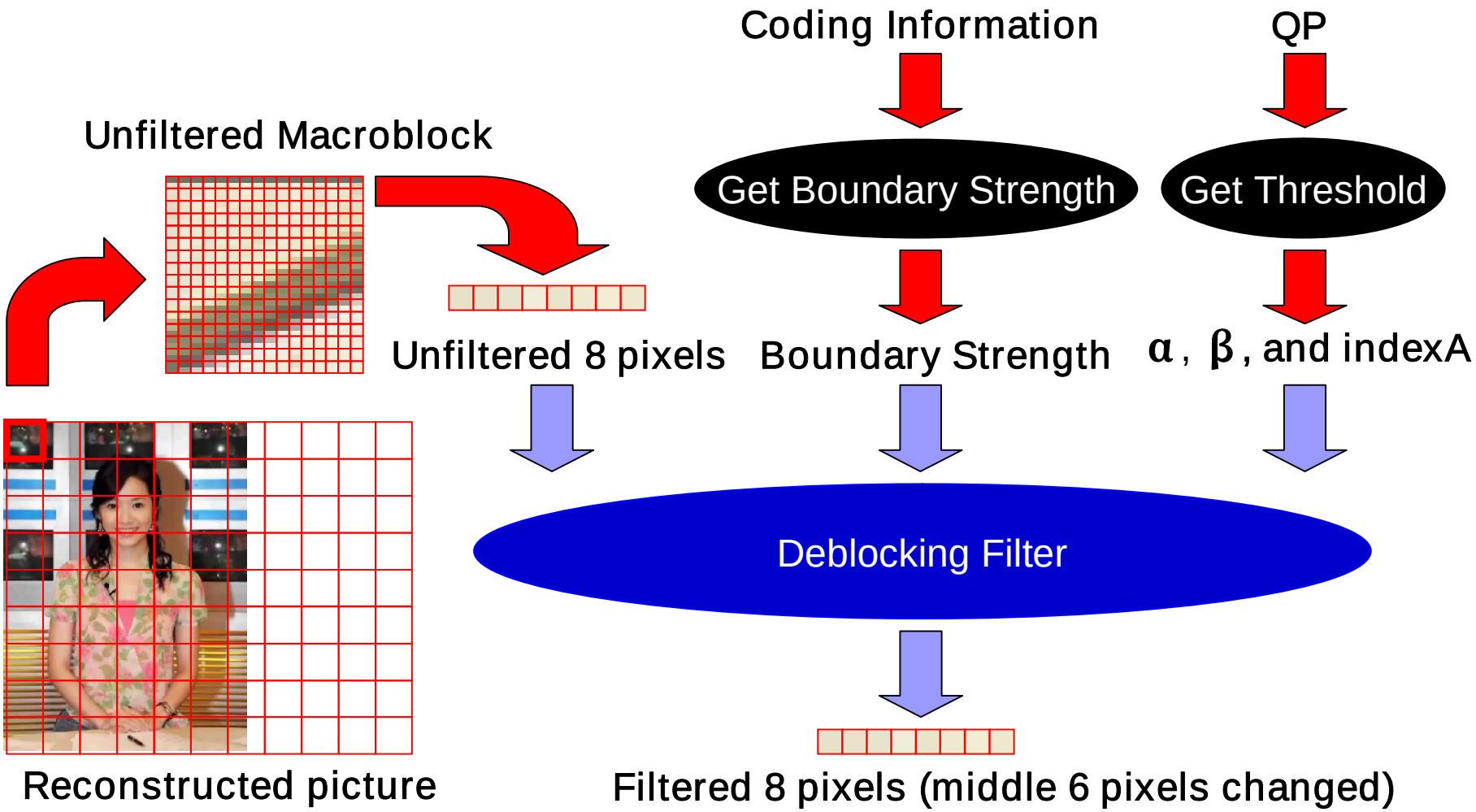
GOP: IBBPBBP

Reference frame: 5

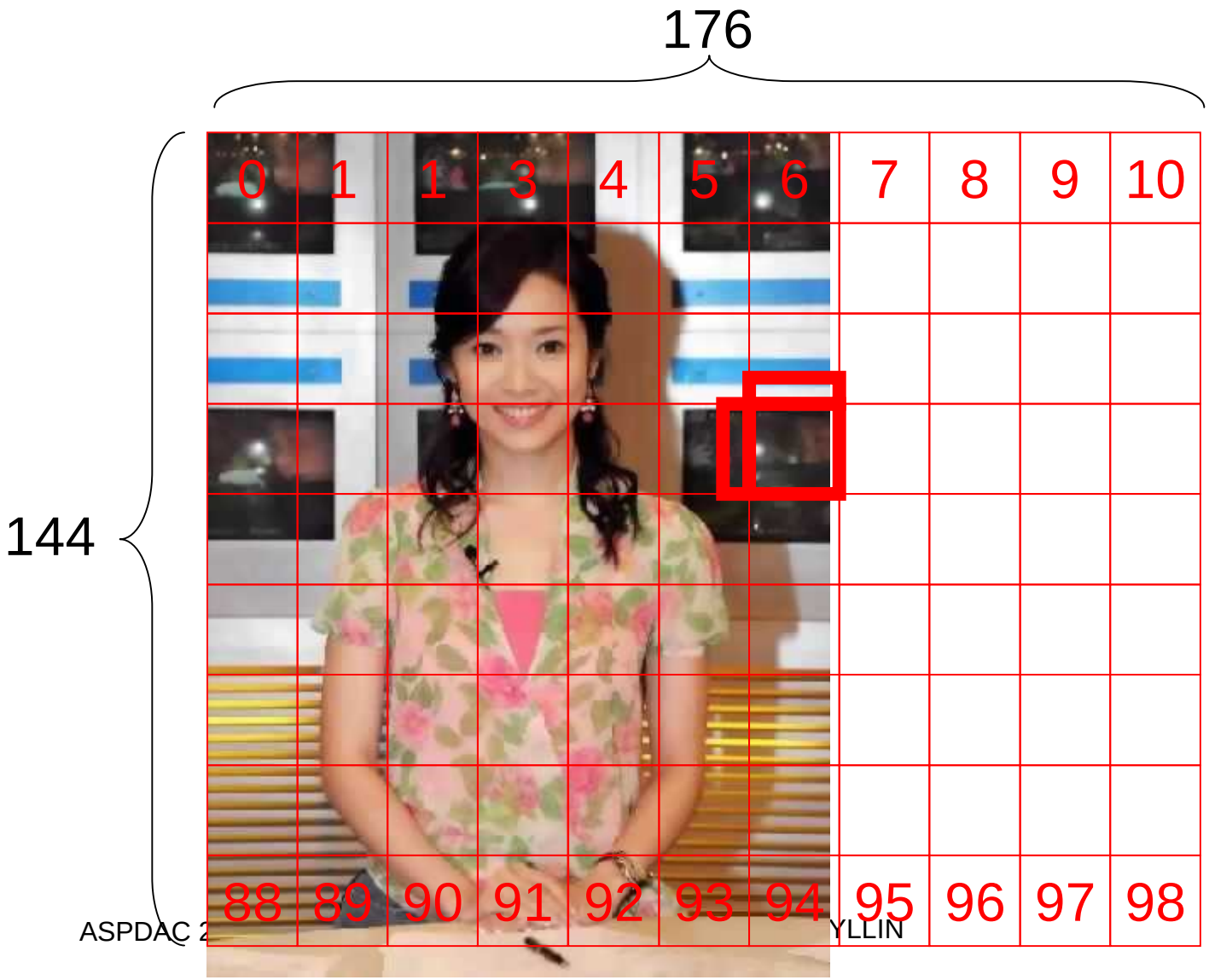
Search range: 16

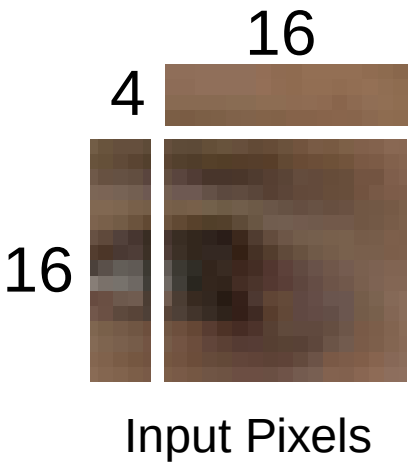
QP: 28

Deblocking Filter I/O

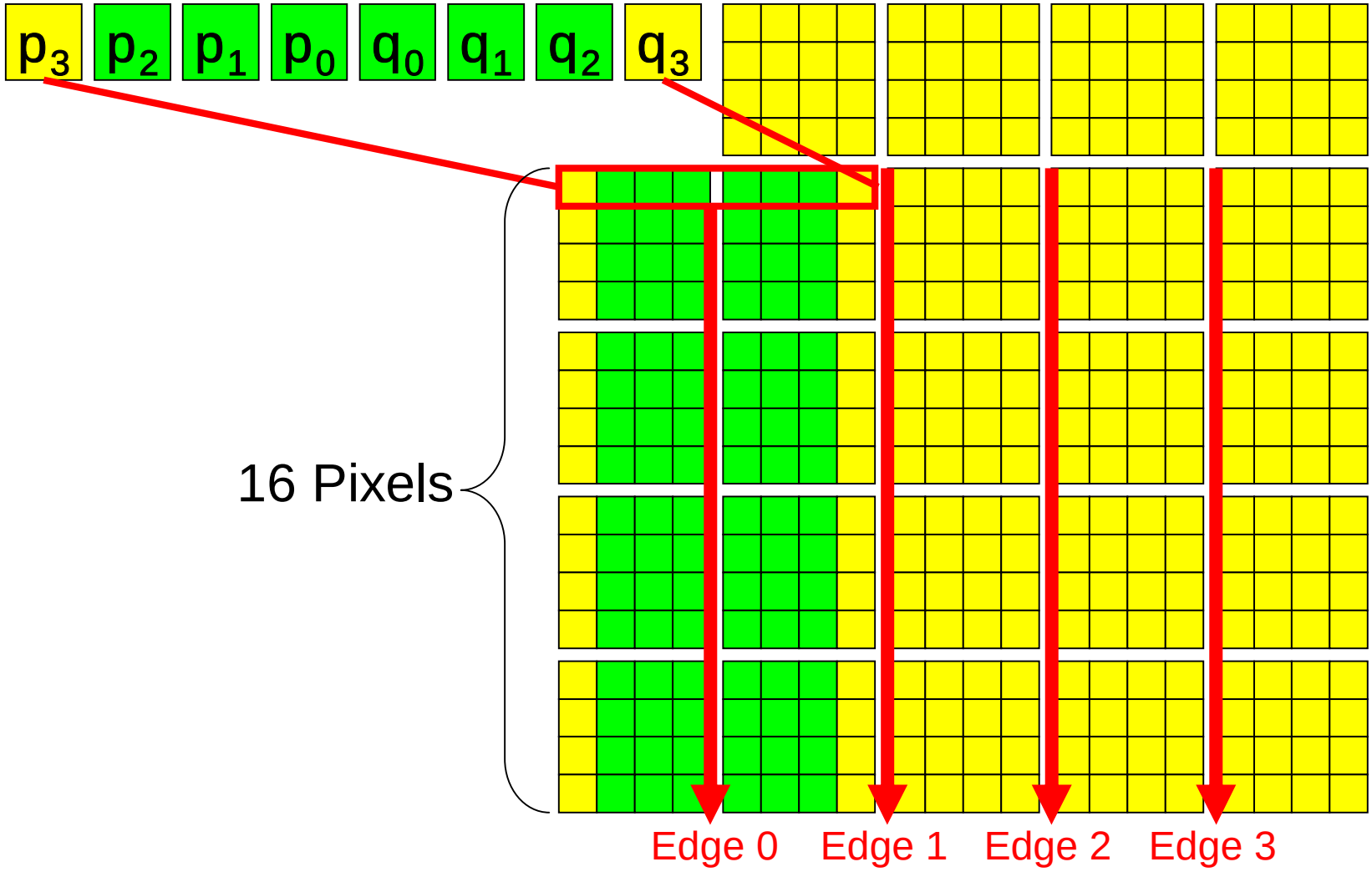


Filtering Order (Raster Scan)

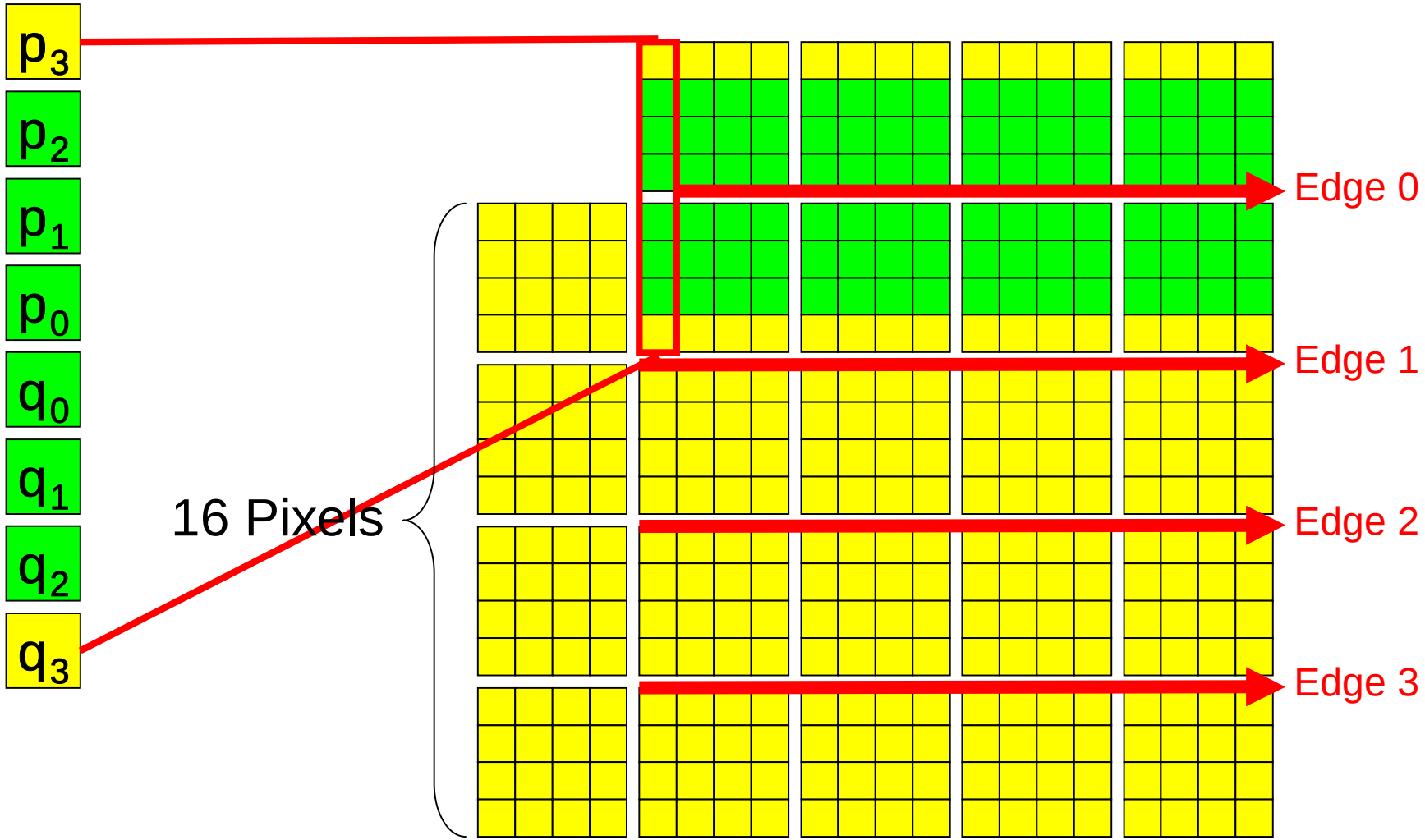




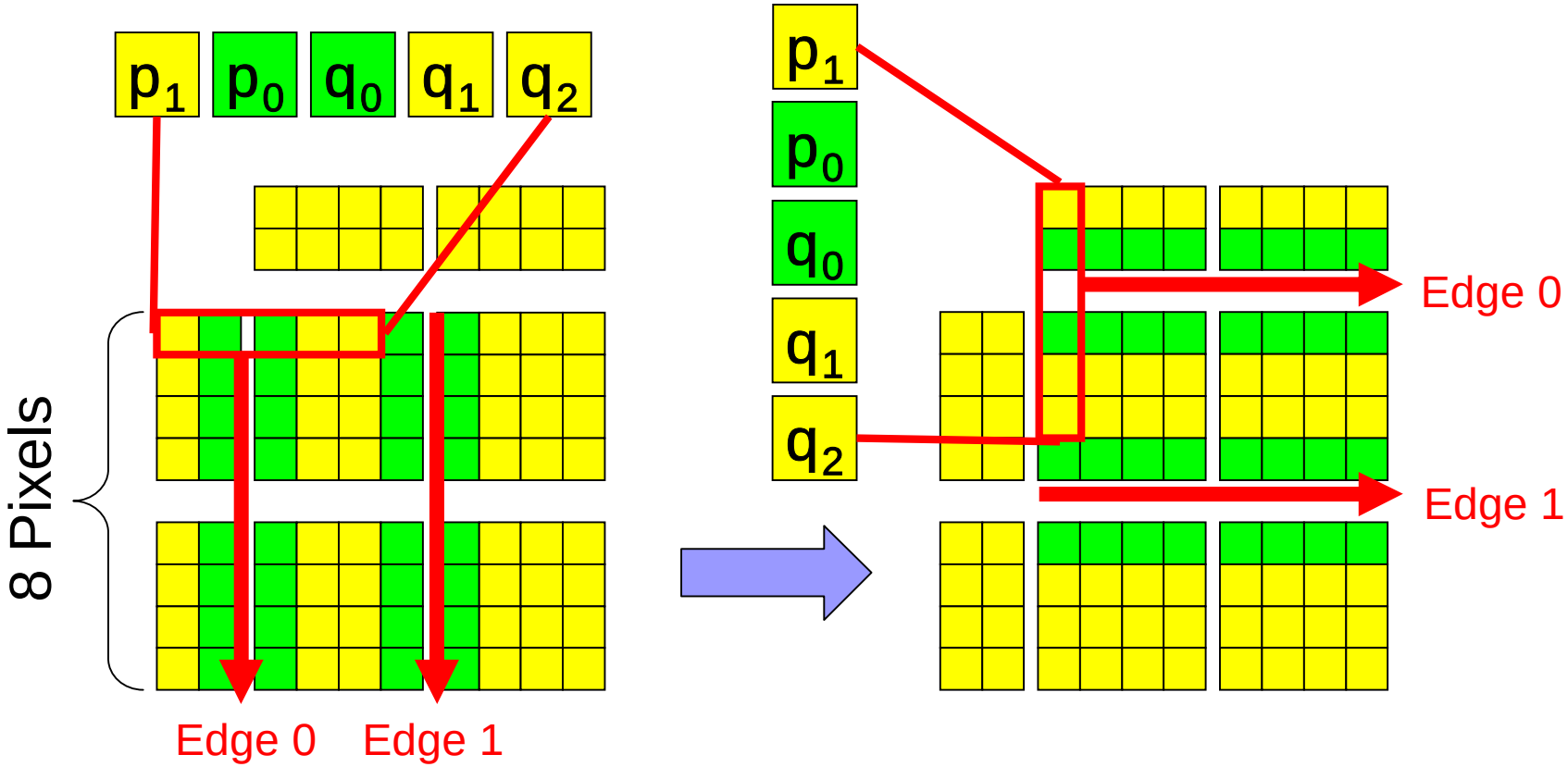
Filter Order : Luma Vertical Edges



Filter Order : Luma Horizontal Edges



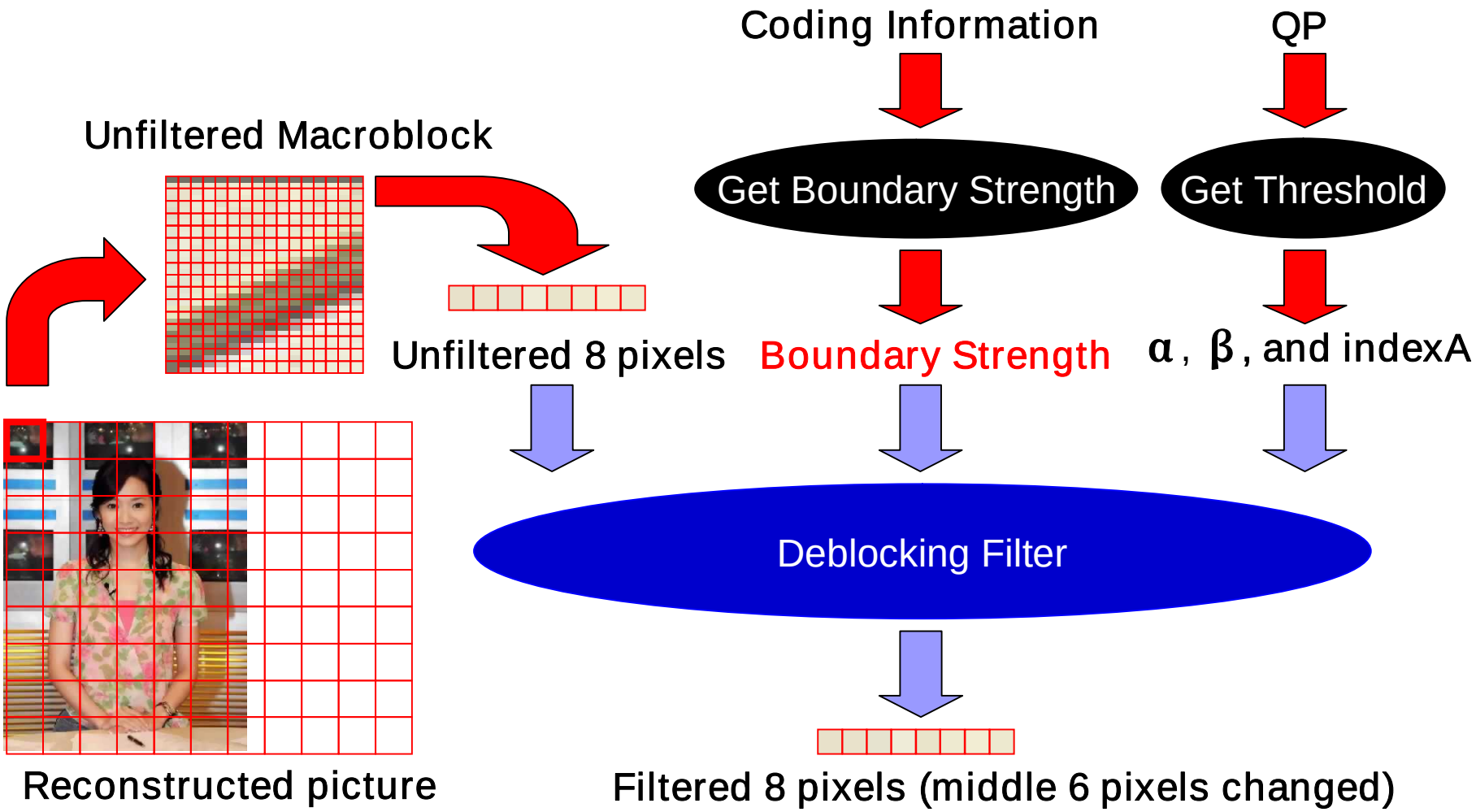
Filter Order : Chroma Edges



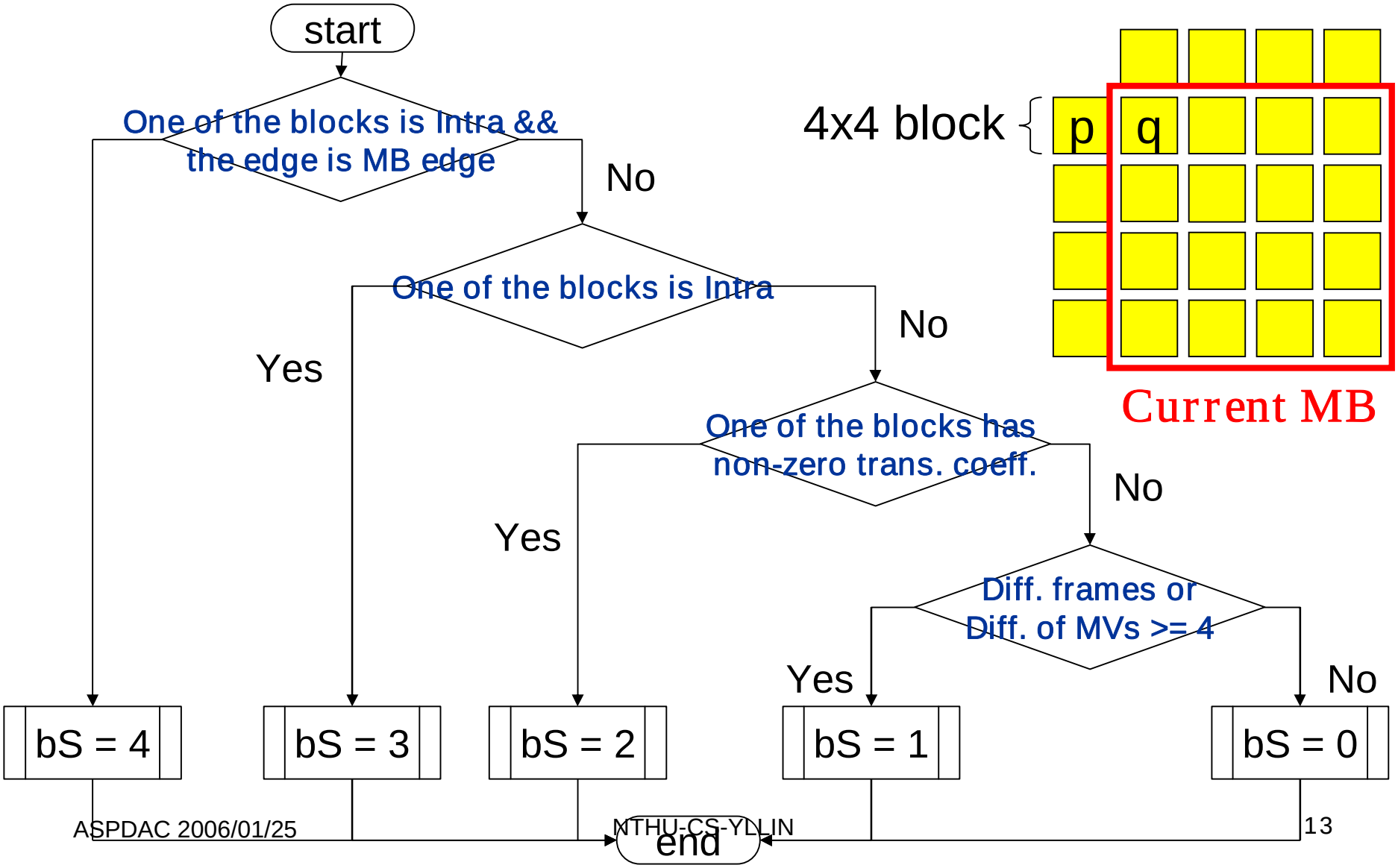
Horizontal filtering
across vertical edges

Vertical filtering
across horizontal edges

Deblocking Filter I/O



Boundary Strength (bS) Determination



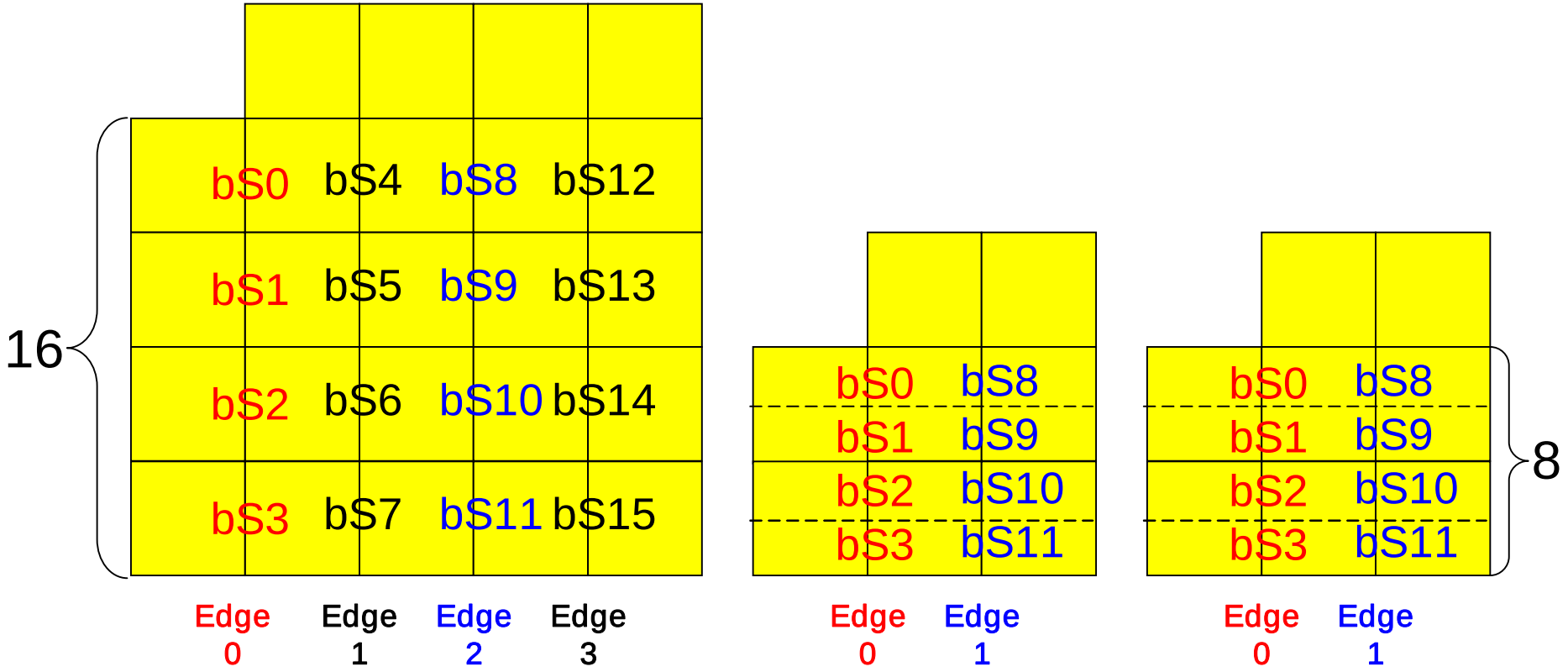
Boundary Strength (1/3)

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

- Boundary strength (bS) ranges between 0 (without filtering) and 4 (strongest filtering)

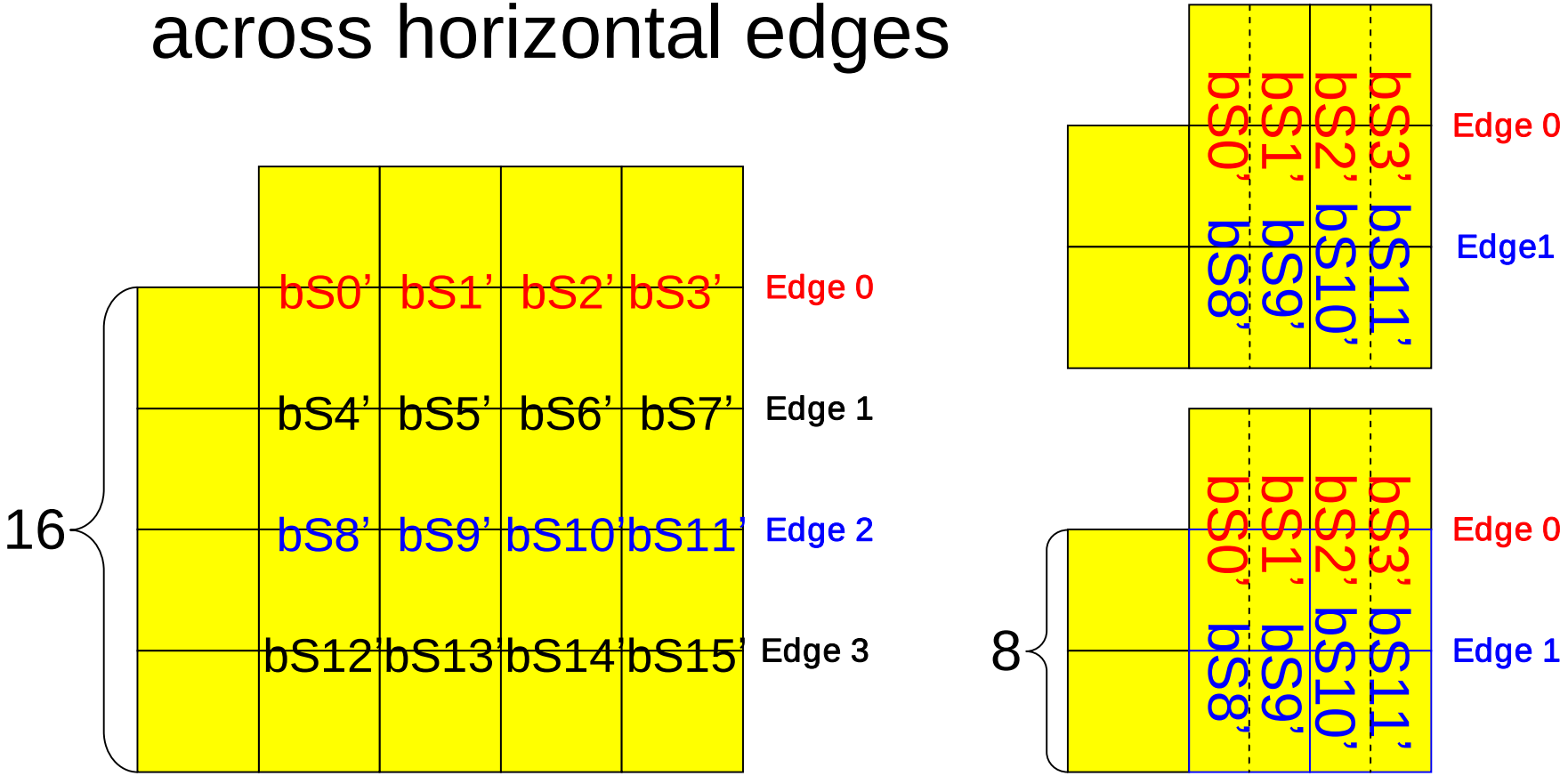
Boundary Strength (2/3)

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

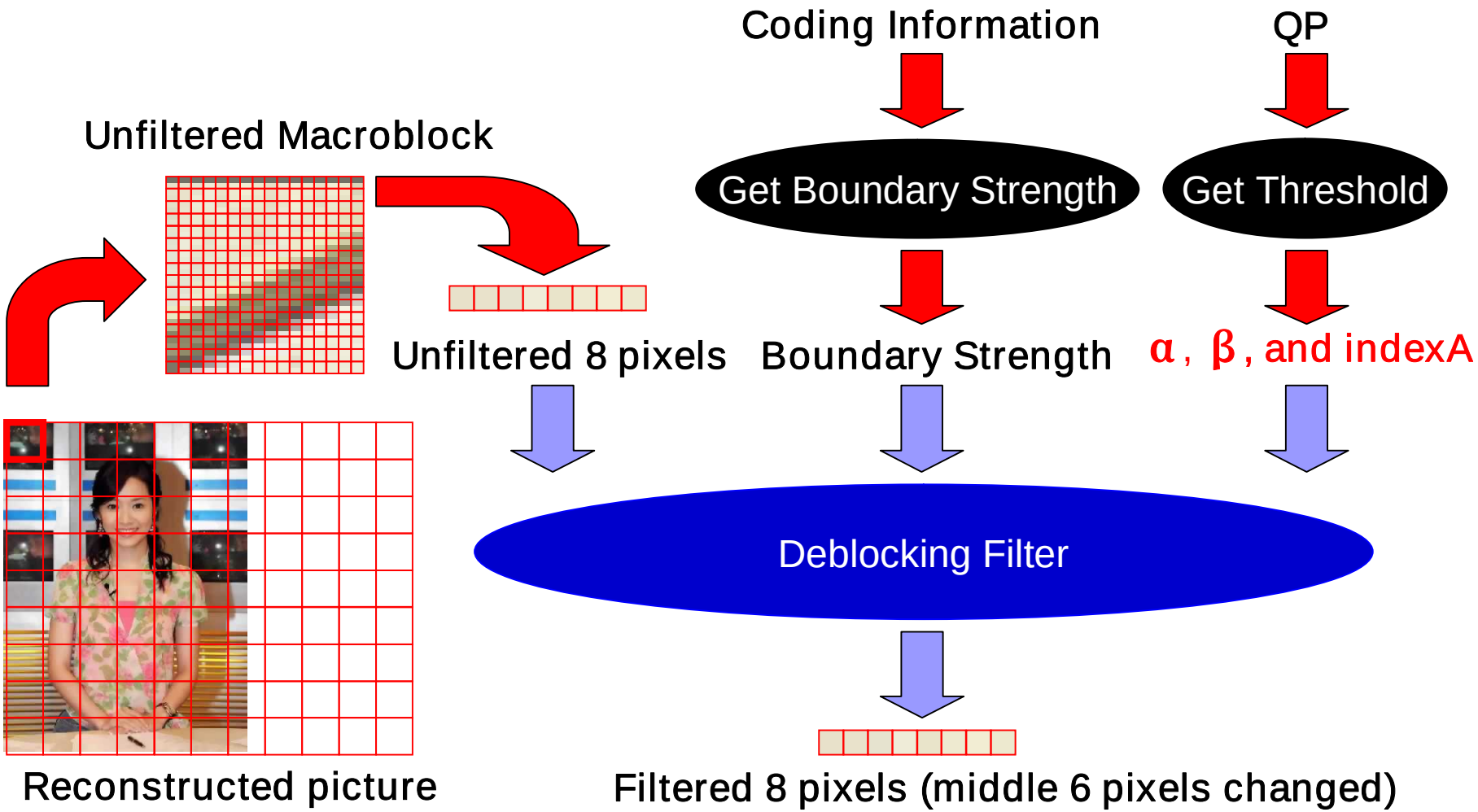


Boundary Strength (3/3)

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



Deblocking Filter I/O

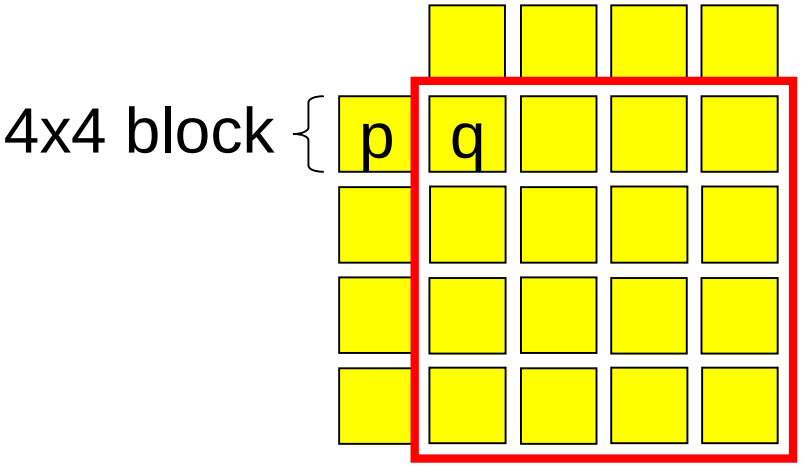


Threshold

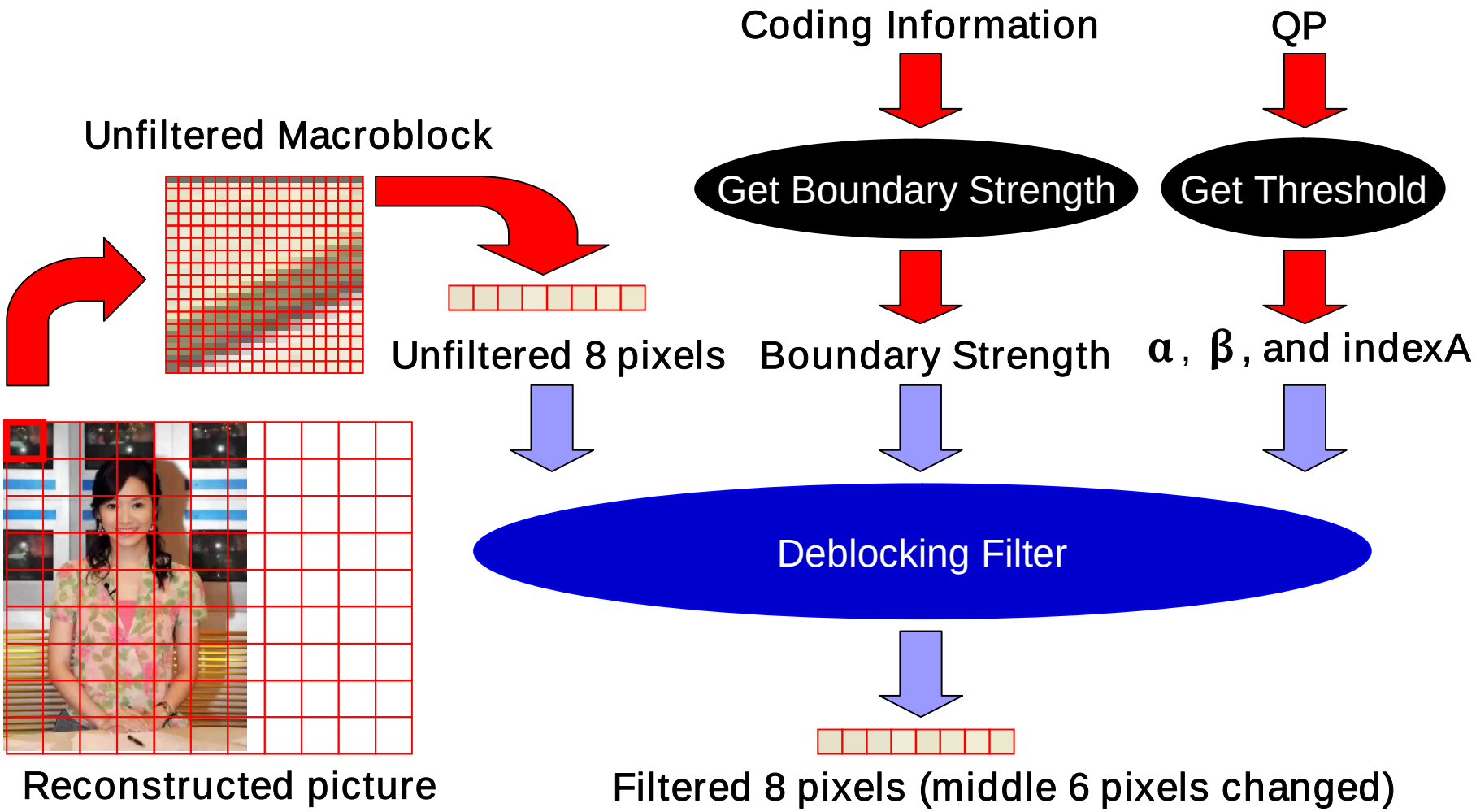
- qPav = (qPp + qPq + 1)>>1
- indexA = Clip3(0, 51, qPav+FilterOffsetA)
- indexB = Clip3(0, 51, qPav+FilterOffsetB)
- α = ThresholdTable(indexA)
- β = ThresholdTable(indexB)

filterSamplesFlag = 1 if the following conditions all hold

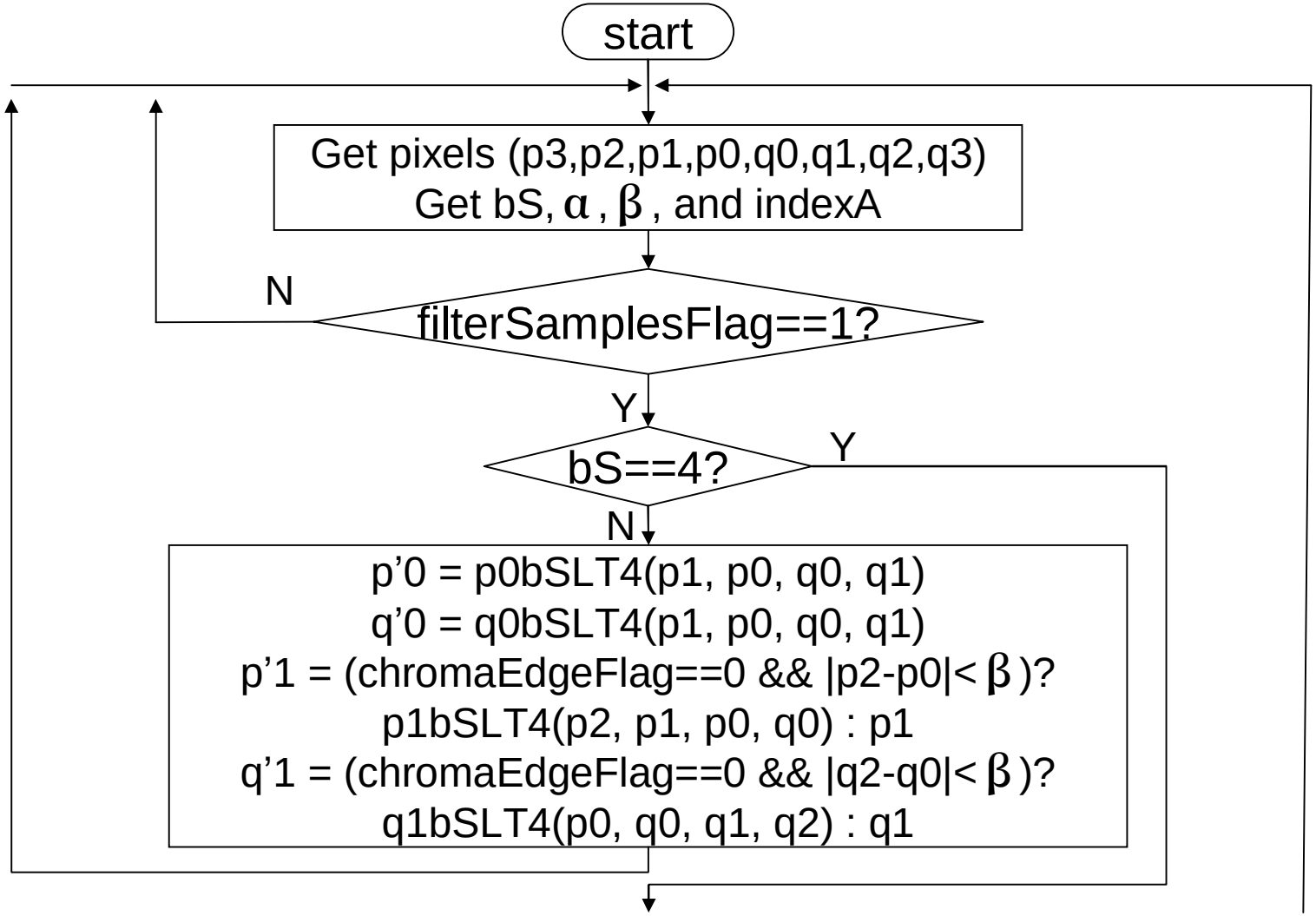
- bS != 0
- |p0-q0| < α
- |p1-p0| < β
- |q1-q0| < β



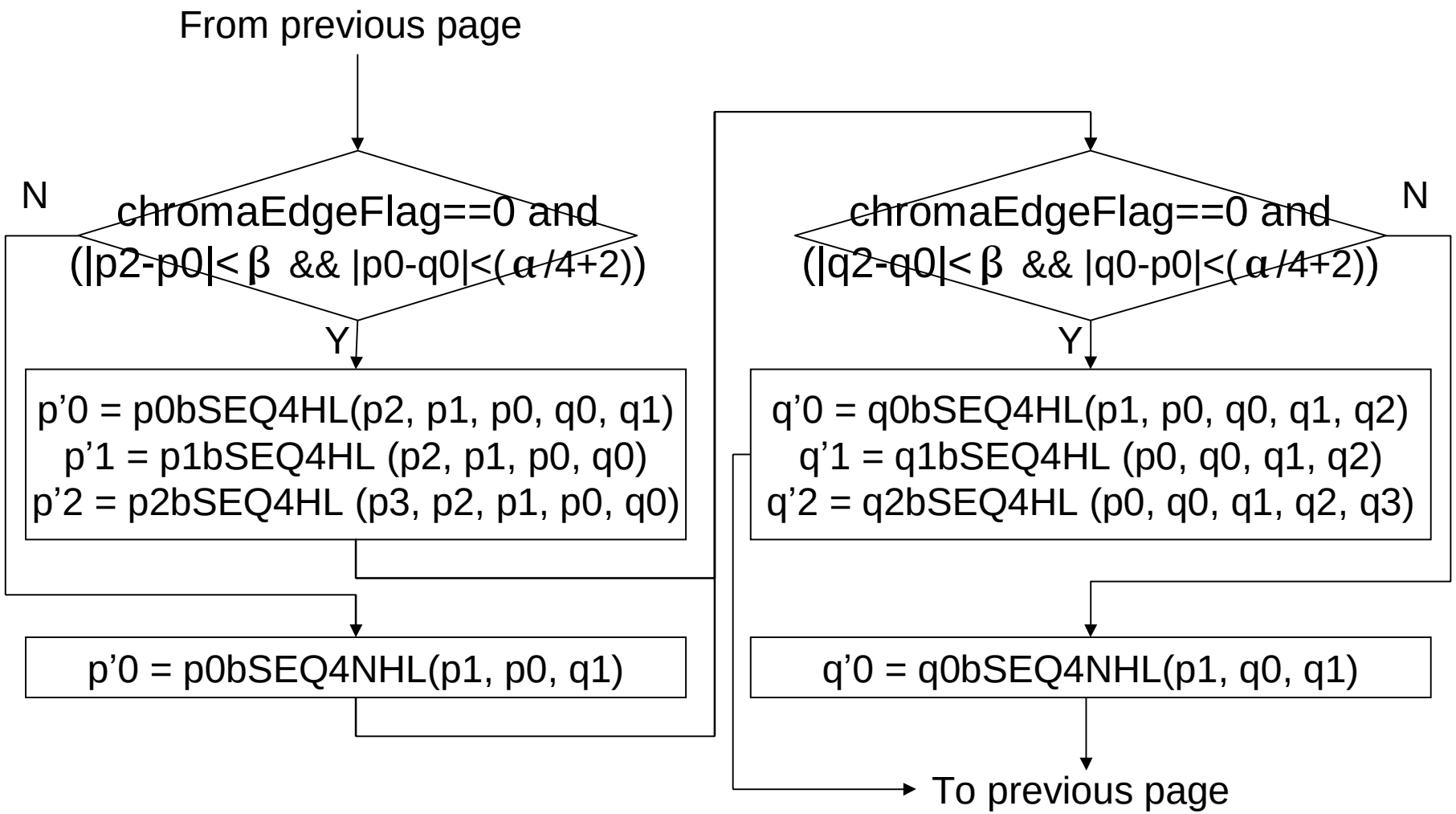
Deblocking Filter I/O



Filter Flow Chart



Filter Flow Chart(Cont.)



Equations for bS < 4

- tc0 = ClipVarTable(bS, indexA)
- tc = (chromaEdgeFlag==1) ?
  (tc0+1) : tc0 + ((|p2-p0|< β)?1:0) + ((|q2-q0|< β)?1:0)
- delta = Clip3(tc,-tc,((((q0-p0)<<2)+(p1-q1)+4)>>3))
- p0bSLT4(p1, p0, q0, q1) = Clip1(p0+delta)
- q0bSLT4(p1, p0, q0, q1) = Clip1(q0-delta)
- p1bSLT4(p2, p1, p0, q0) =
  p1 + Clip3(–tc0,tc0, (p2+((p0+q0+1)>>1)–(p1<<1))>>1)
- q1bSLT4(p0, q0, q1, q2) =
  q1 + 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

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

*RAM is not included in Gate Count

Previous Work (3/3)

- Hard wired designs

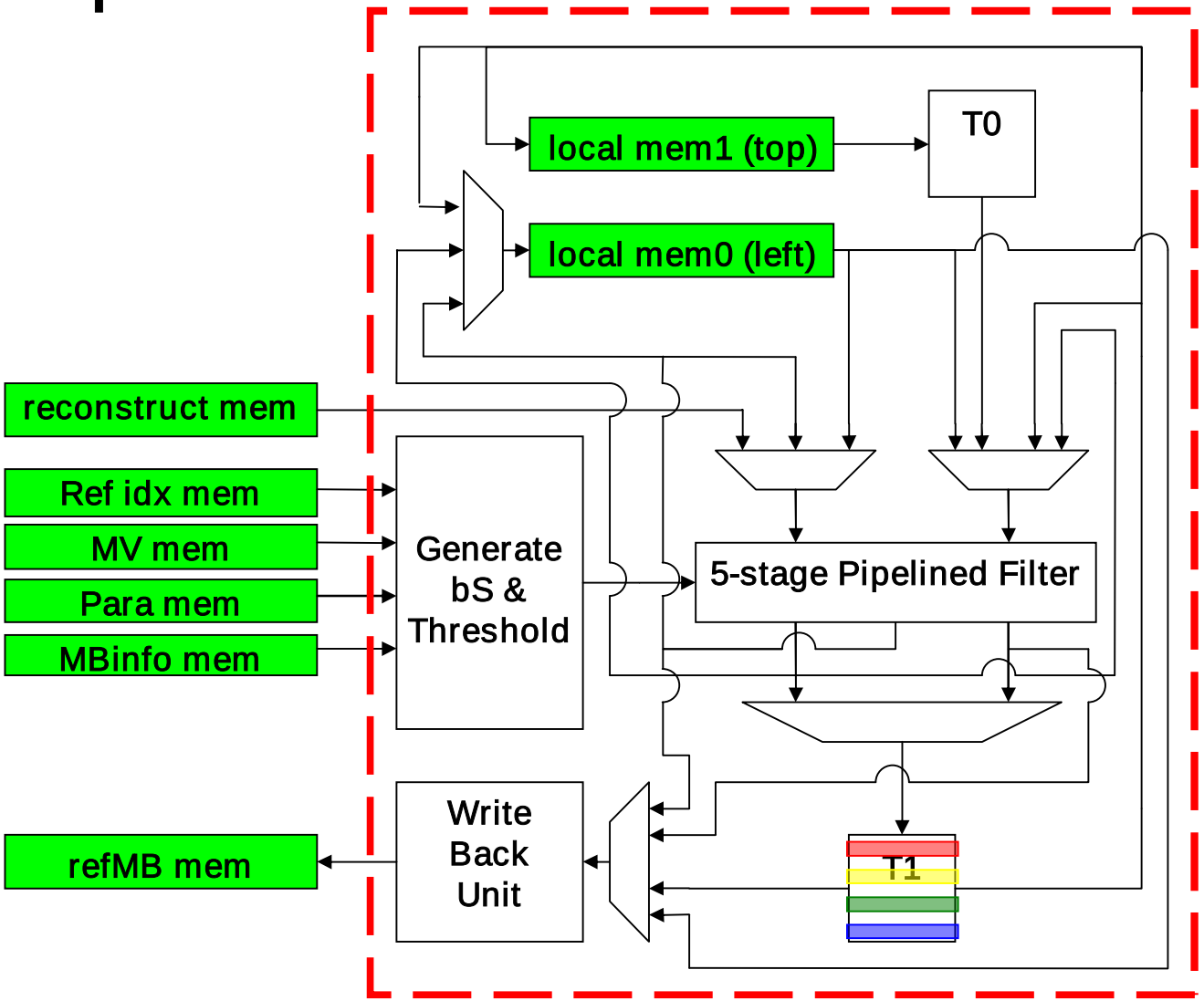
	Wu's '04	Yu's '04	Lee's '05
Cycles/MB	446	288	250
Filter Cycles/MB	192	236	250
SRAM for Pixels	Dual 64x32 2 Two 96x32	Dual 96x32 Dual 64x32 Single (2xFrameWidth)x32	2 Single 96x32 Single (2xFrameWidth+20)x32
# 4x4 Registers	8	9	4
# of Edge Filter	1	1	1
Capability 1280x720p @100MHz	62.3 fps	97.1 fps	111.1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

*RAM is not included in Gate Count

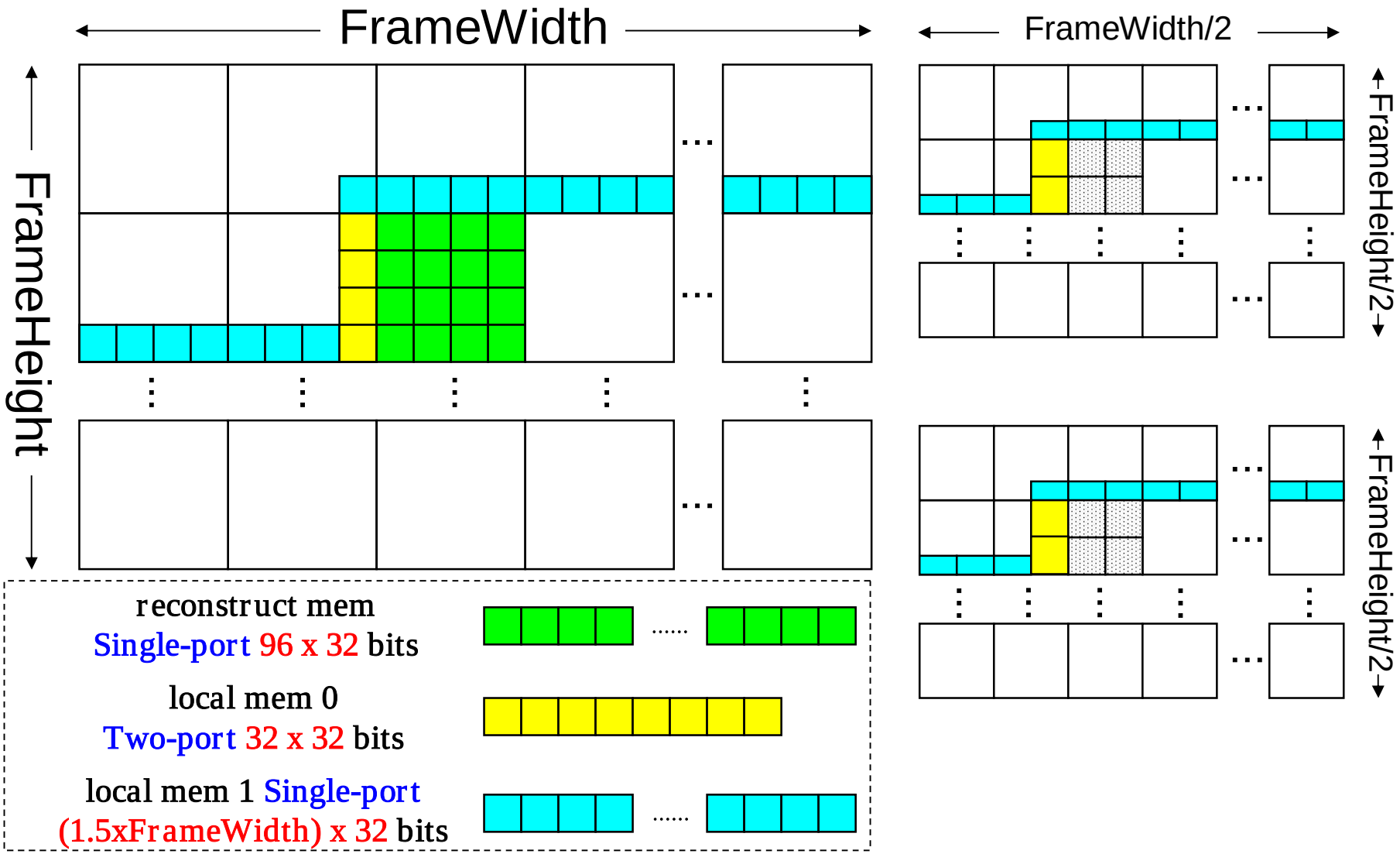
Outline

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

Proposed Architecture

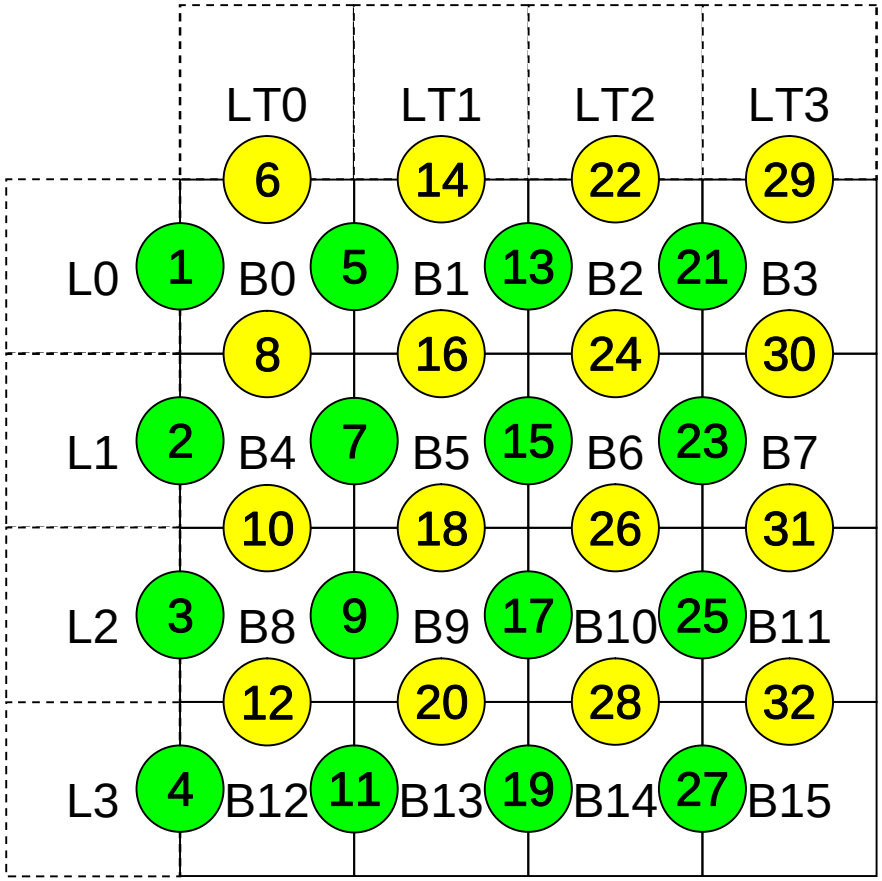


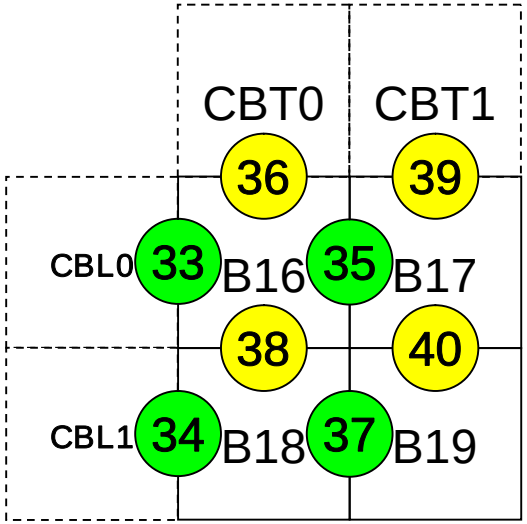
Memory Organization

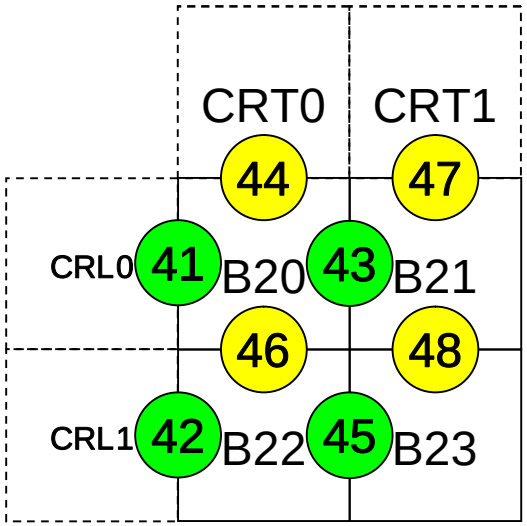


Proposed Filter Order

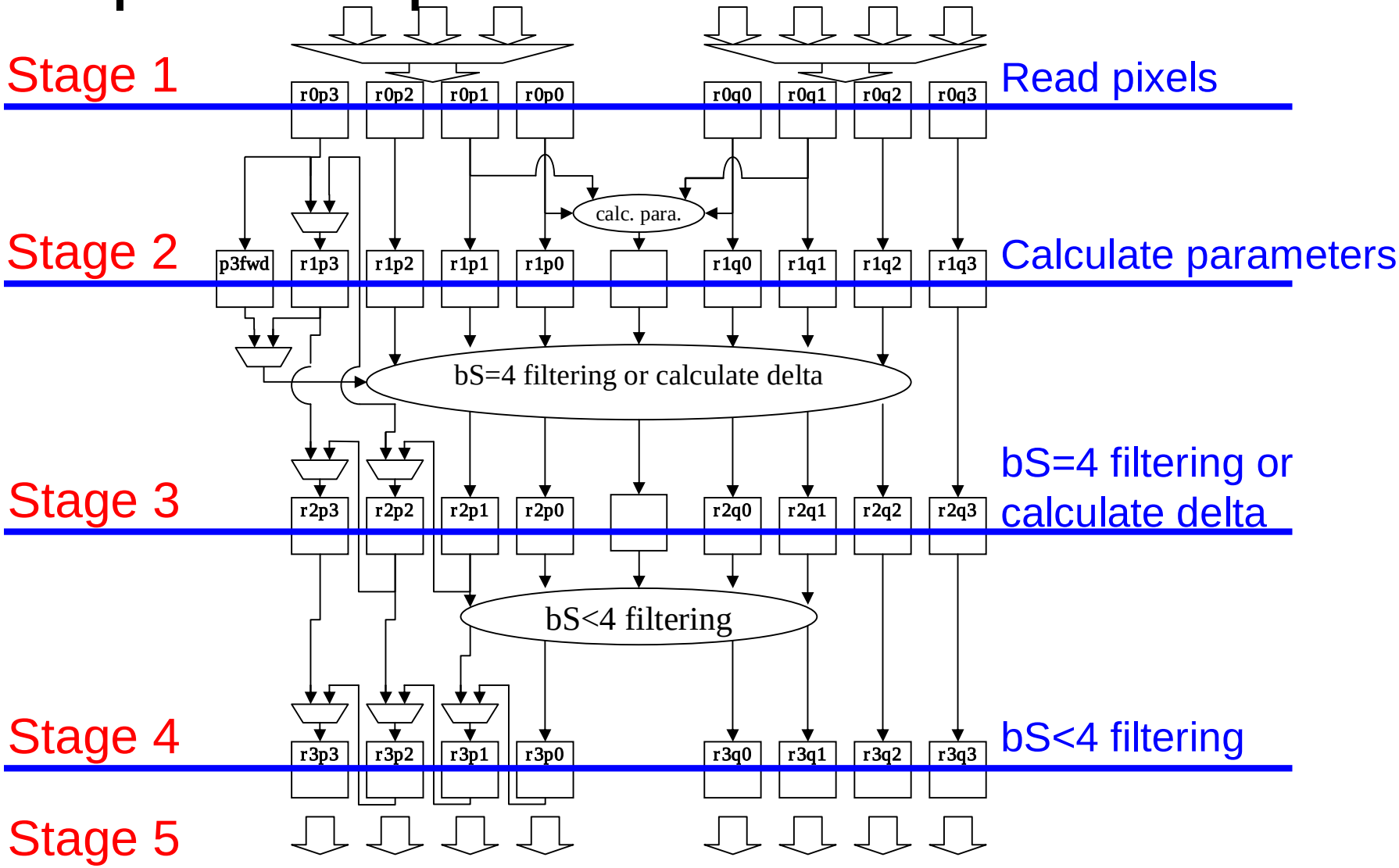
- Keep right most column
- Reduce local memory usage



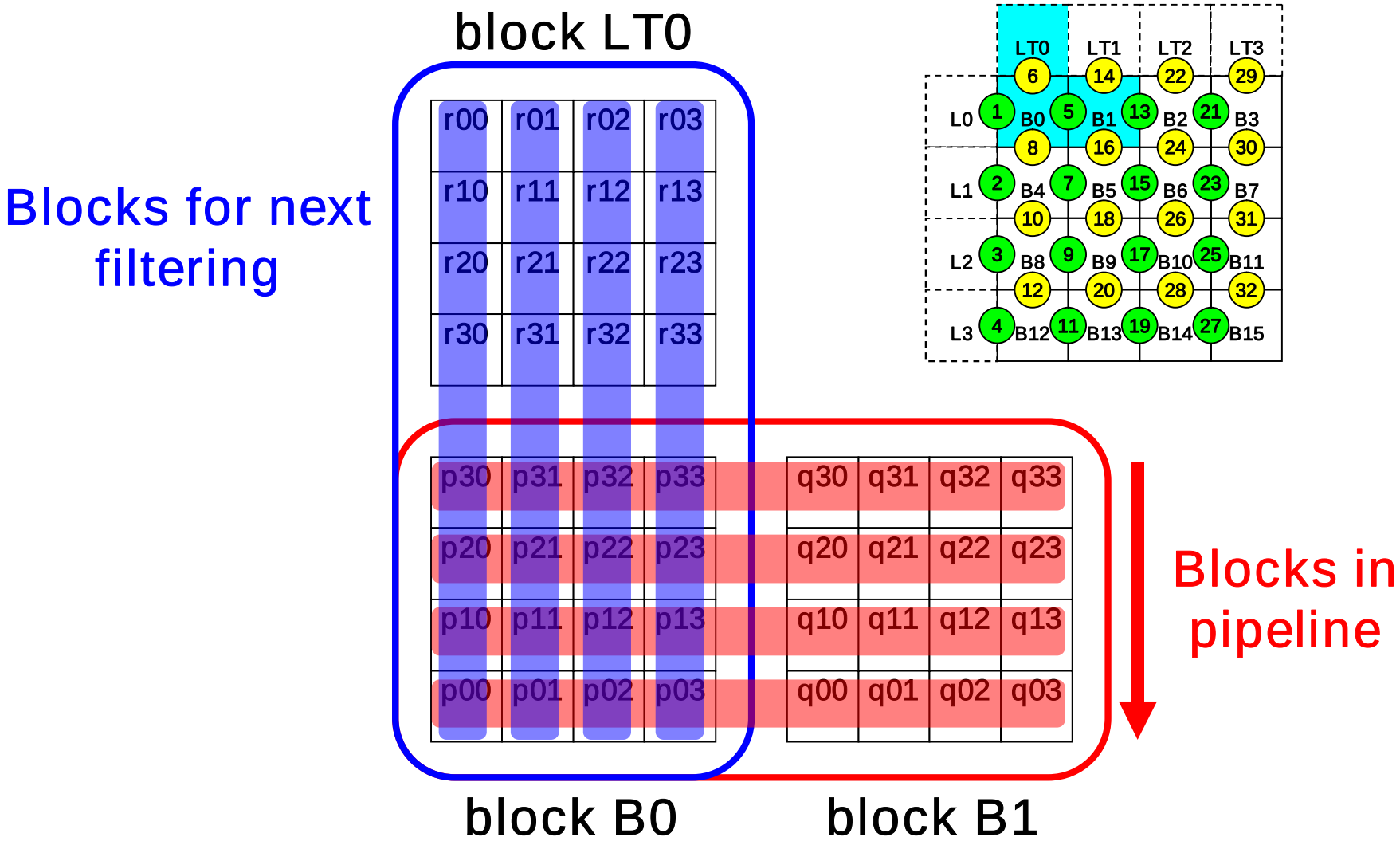




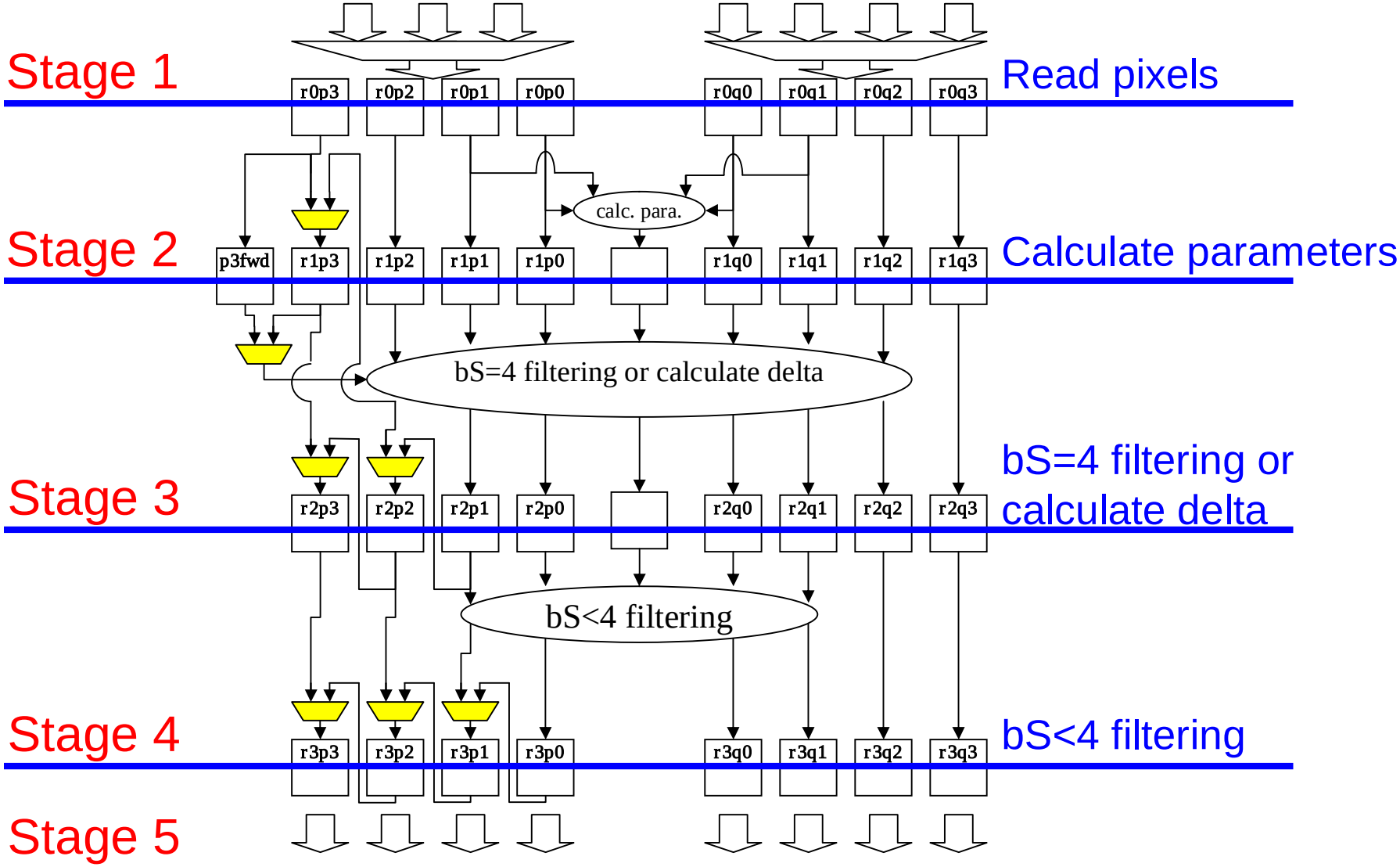
Proposed Pipelined Filter



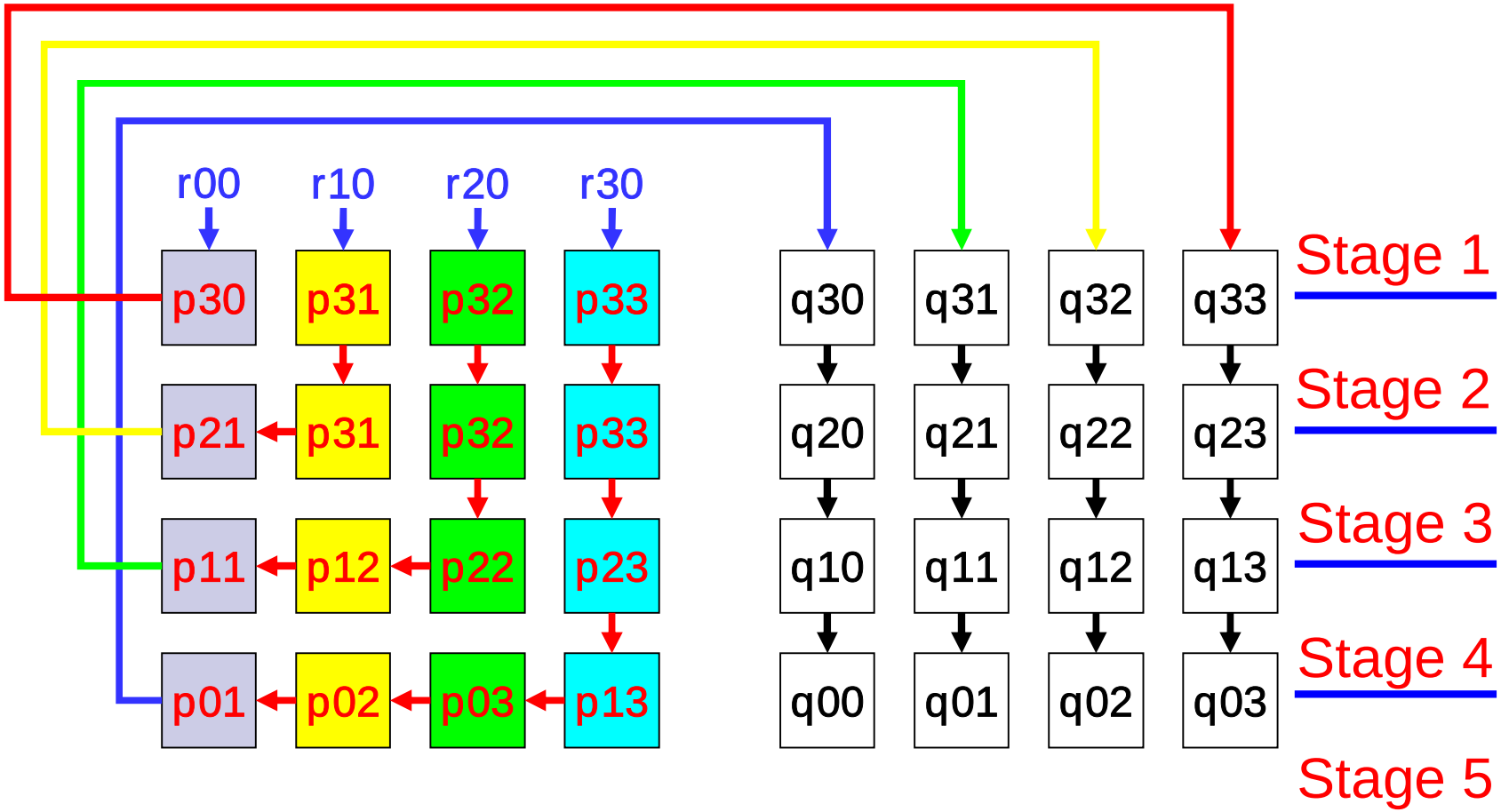
Pipeline Hazard



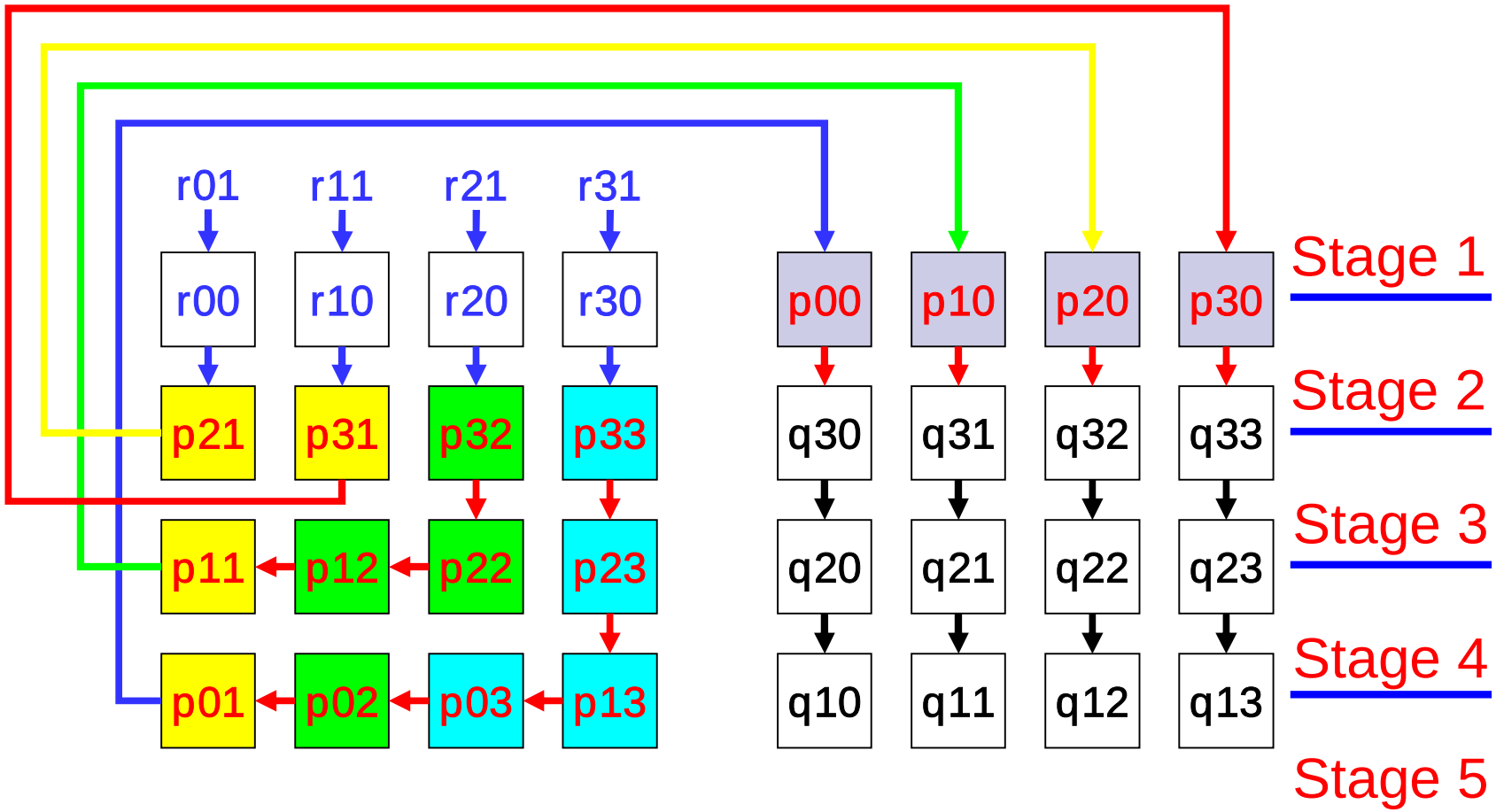
Resolving Hazard by Forwarding



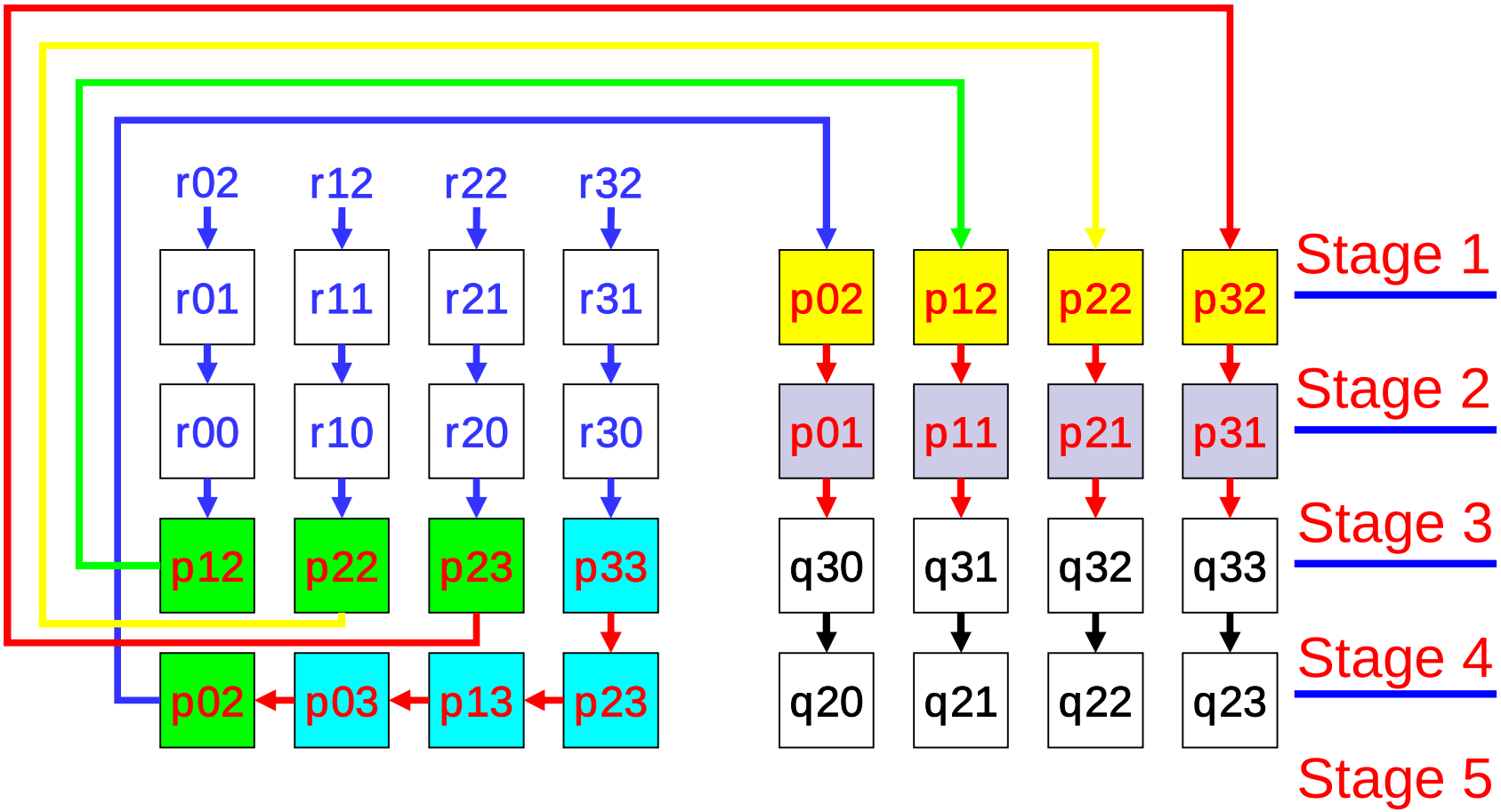
Forwarding for Filtering Step 6 (1/4)



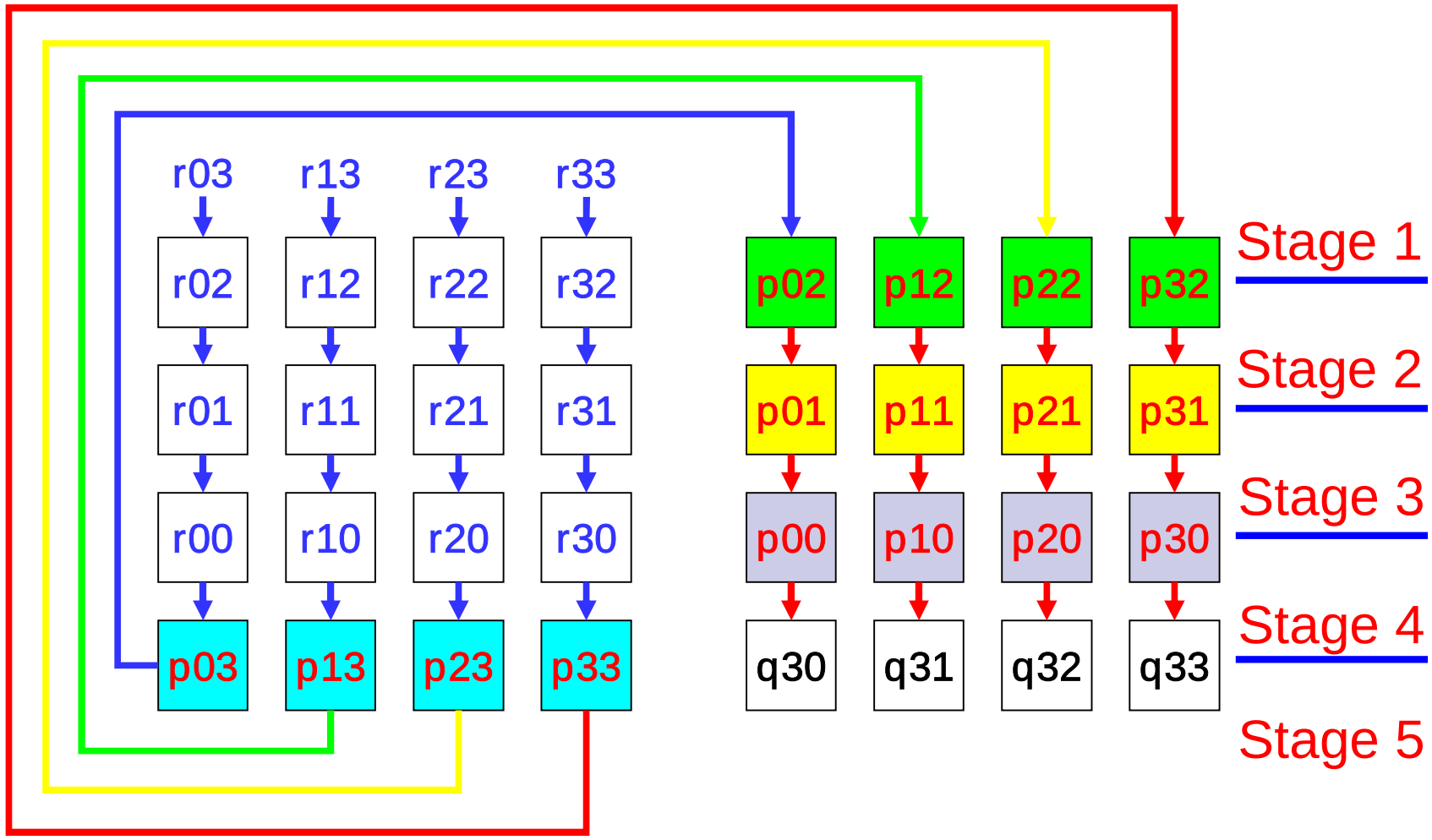
Forwarding for Filtering Step 6 (2/4)



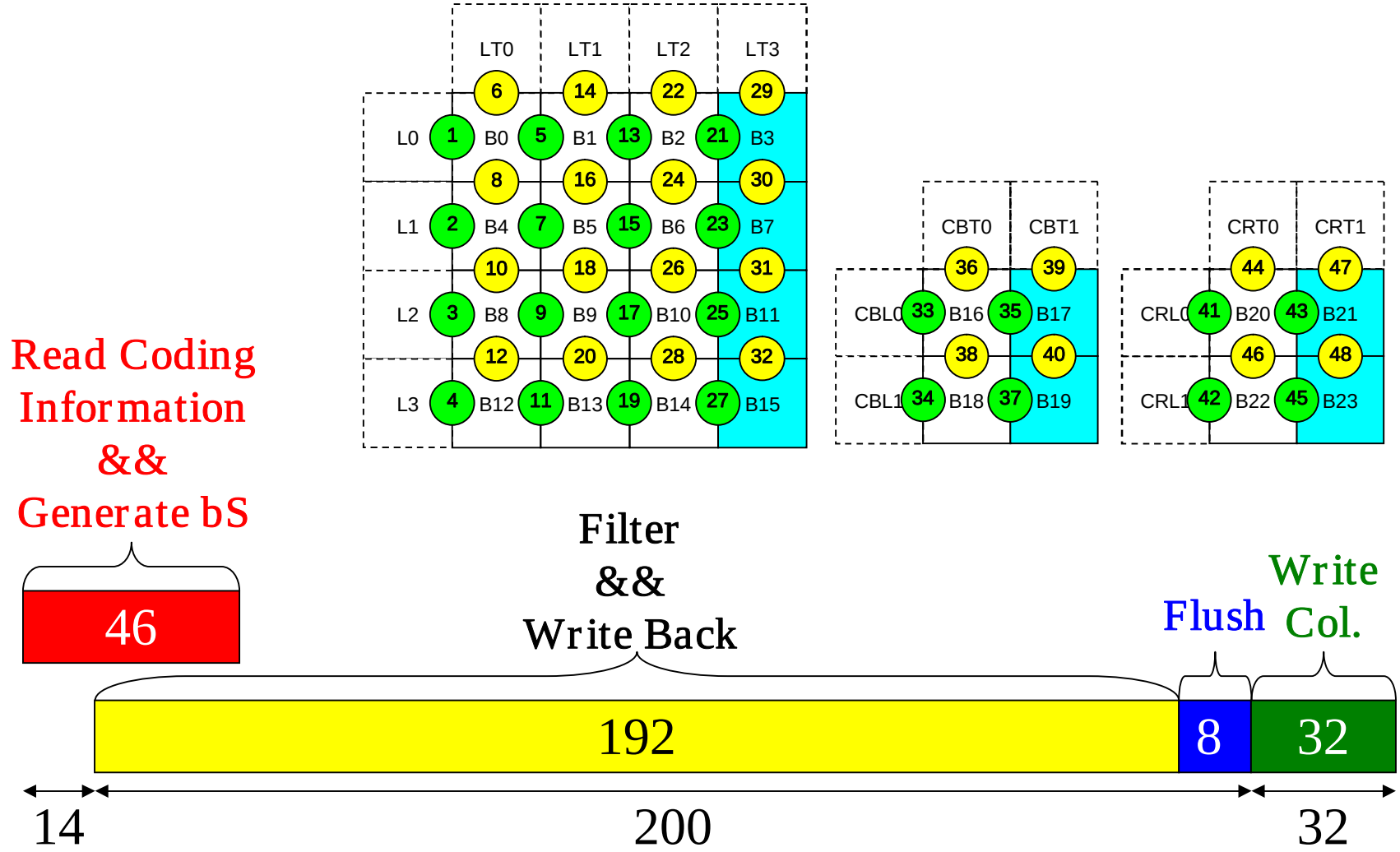
Forwarding for Filtering Step 6 (3/4)



Forwarding for Filtering Step 6 (4/4)



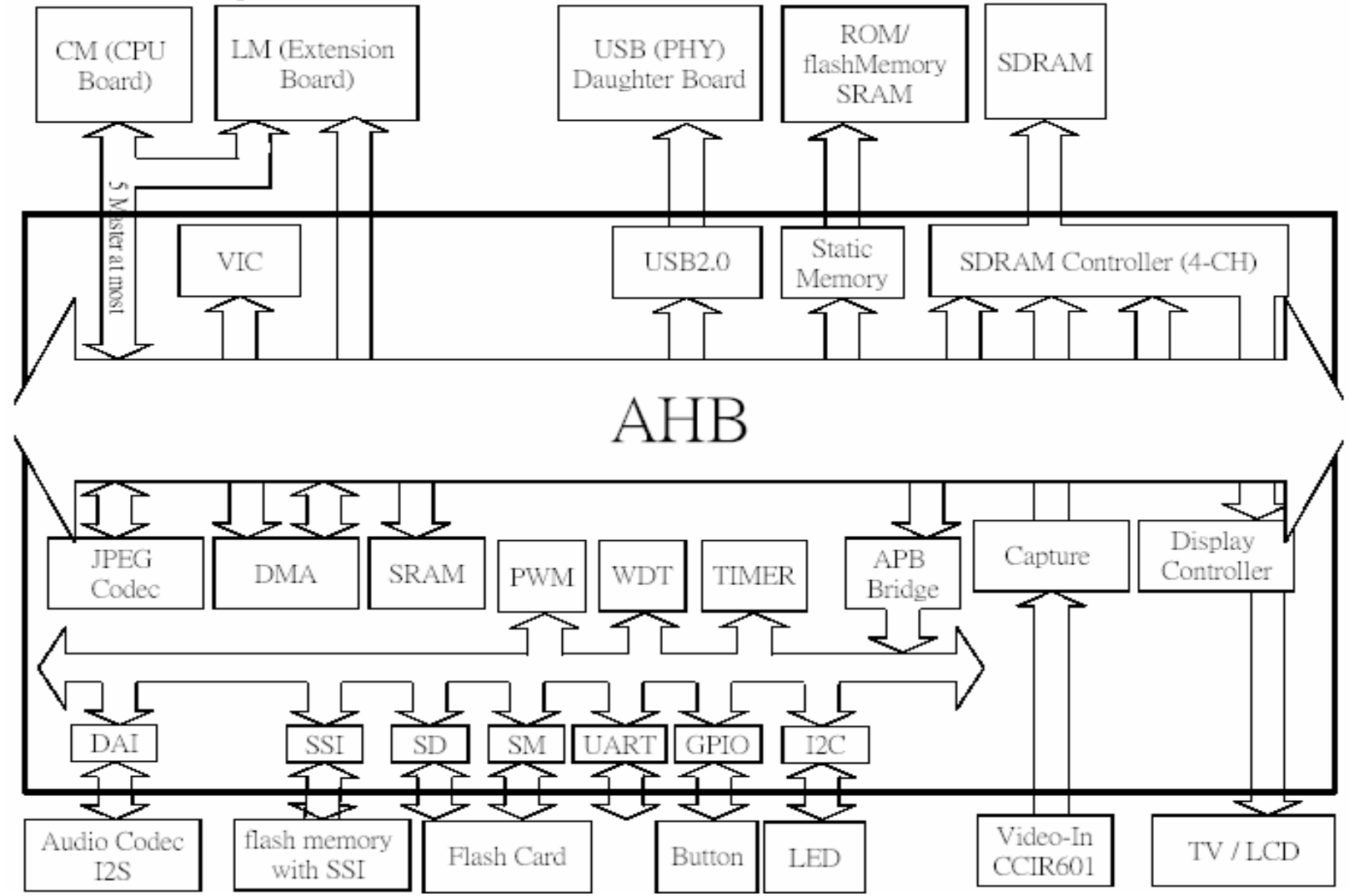
Filtering Speed (Near Optimal)



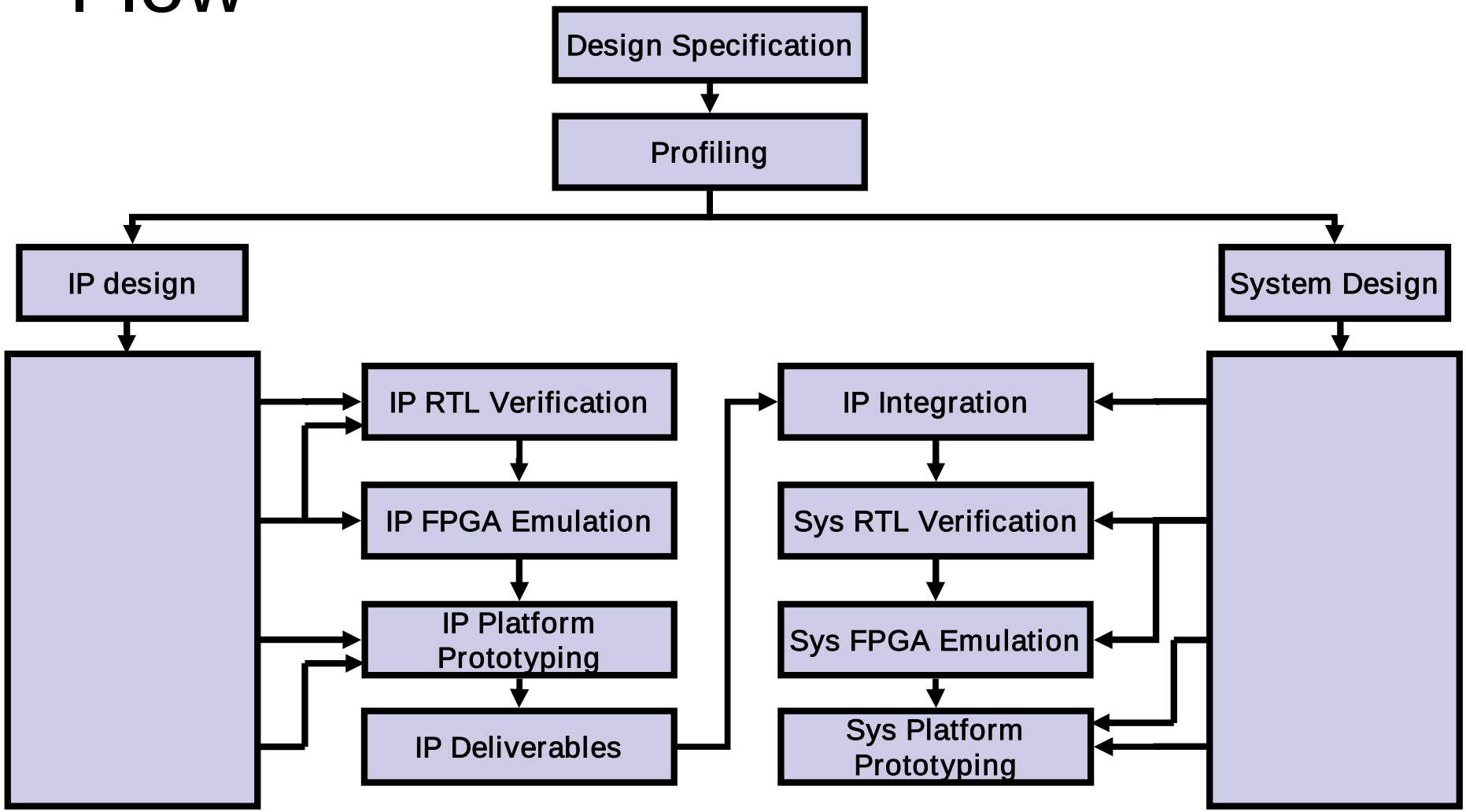
Outline

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

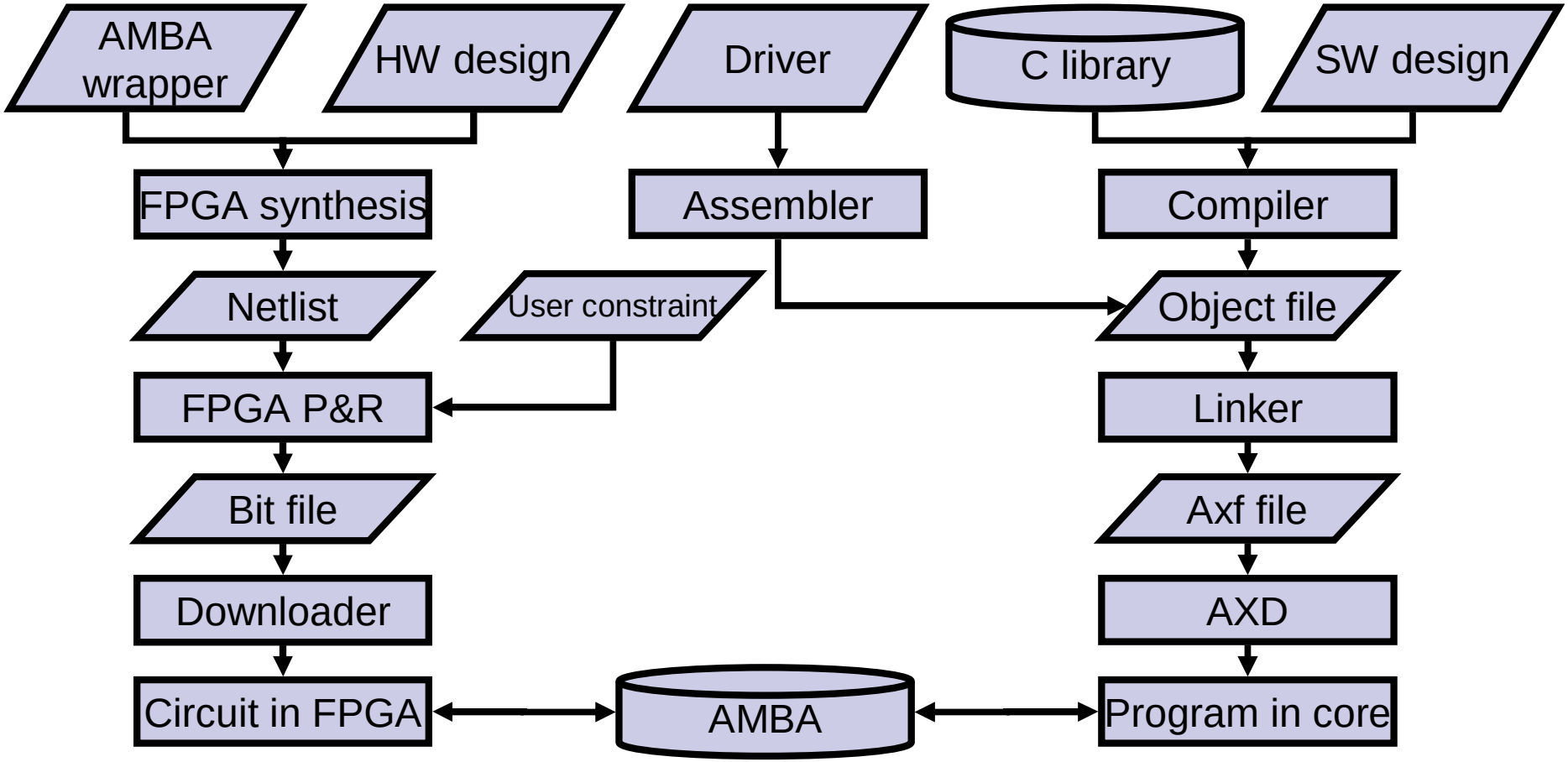
UniChip Multimedia SOC Platform



H.264/AVC Decoder IP Design Flow



SOC Platform Design Flow



IP Qualification

- Code coverage (Verification Navigator)

	Our Design	Recommended
Statement Coverage	99.8%	95%
Branch Coverage	99.1%	95%
Toggle Coverage	99.0%	95%
Path Coverage	74.0%	50%

- DFT (TetraMax)

Total Faults	Test Patterns	Fault Coverage
69,272	376	99.91%

Comparison with Platform-based Designs

	Chen's '03	Zhang's '04	Ikenaga's '05	Ling's '04	Chang's '05	Chiang's '05	Proposed
Cycles/MB	614	584	566	510	386	Max 374 Min 50 Avg. 86-244	214
Filter Cycles/MB	240	214	192	136	336	0 - 374	200
SRAM for Pixels	Dual 96x32 Dual 64x32	Dual 16x32	8 Dual 80x8	Dual 88x32 Dual 72x32 Single 32x32	Single 80x32	Single 96x32	Single 96x32 Two 32x32 Single (1.5xFrameWidth) x32
# 4x4 Registers	4	6	0	11	2	2	2
# of Edge Filter	1	1	1 Pipelined	2	1	1	1 Pipelined
Capability 1280x720p @100MHz	45.2 fps	47.5 fps	49.1 fps	54.46 fps	71.9 fps	113.8 – 322.9 fps	128.6 fps
Process (um)	.25	N/A (FPGA)	.35	N/A (FPGA)	.18	.18	.18
Gate Count	20.66K	N/A (FPGA)	9.35K	N/A (FPGA)	19.2K	21.8K	20.9K

*RAM is not included in Gate Count

Comparison with Hard Wired Designs

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

*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!!