A Highly Parallel and Scalable CABAC Decoder for Next Generation Video Coding

Vivienne Sze, Anantha P. Chandrakasan

Massachusetts Institute of Technology

ISSCC 2011

© 2011 IEEE

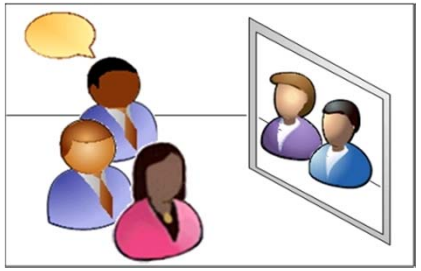
© 2011 IEEE

Video Application Space

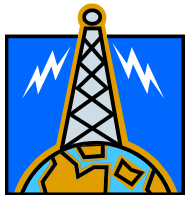
High Performance
(resolution, frame rate)

scalability

Low Power
(battery life)



Video Conferencing



Broadcast

Set-top box

Laptop, Netbook, Tablet, Smartphone

Next-Generation Video Coding

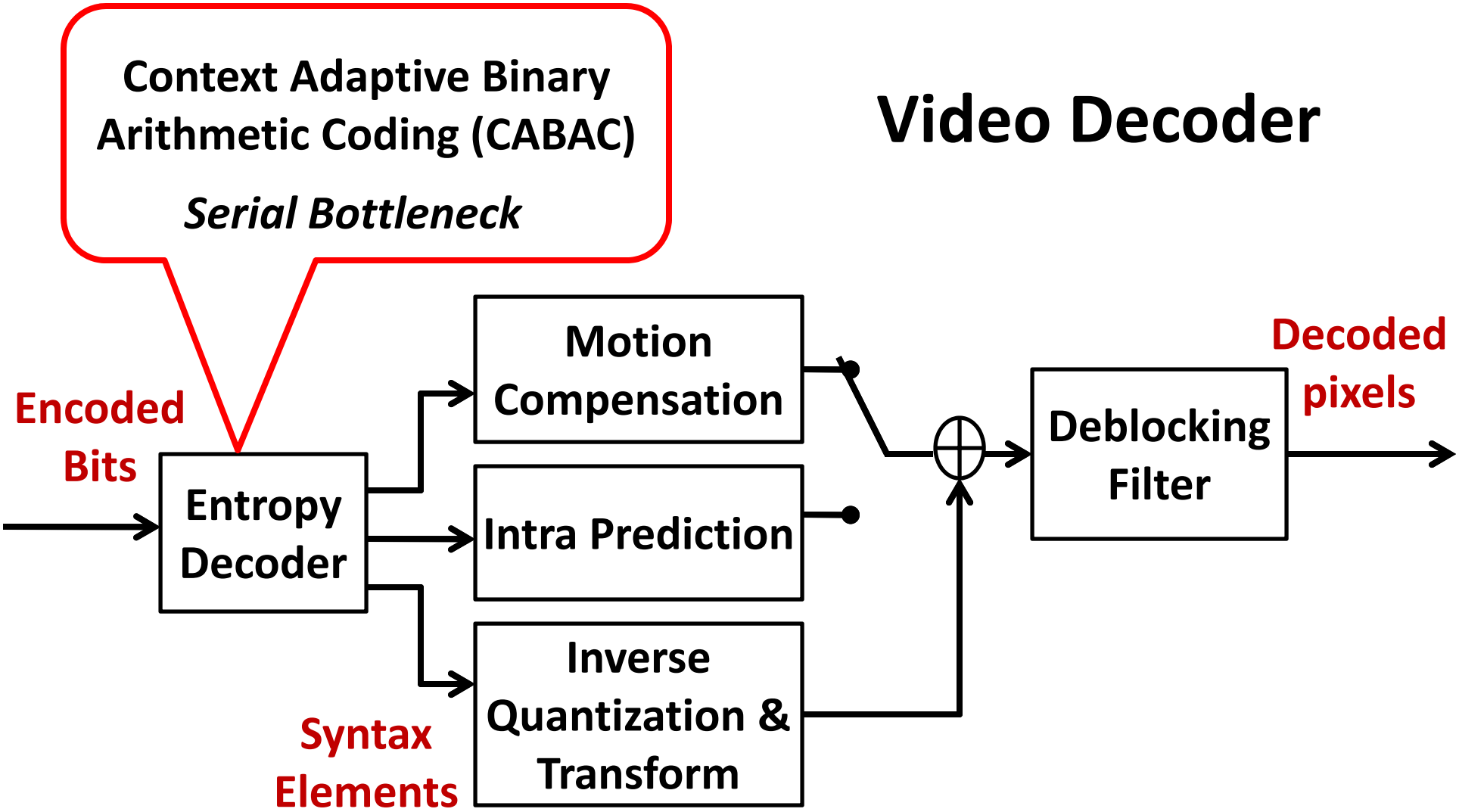
- 4kx2k and beyond: >10x processing speed of 1080HD
- New Standard: High Efficiency Video Coding (HEVC)
- 50% better coding efficiency than H.264/AVC

Our Approach: Leverage Parallelism & Joint Algorithm/Architecture Optimizations

© 2011 IEEE

© 2011 IEEE

CABAC Limits Parallelism

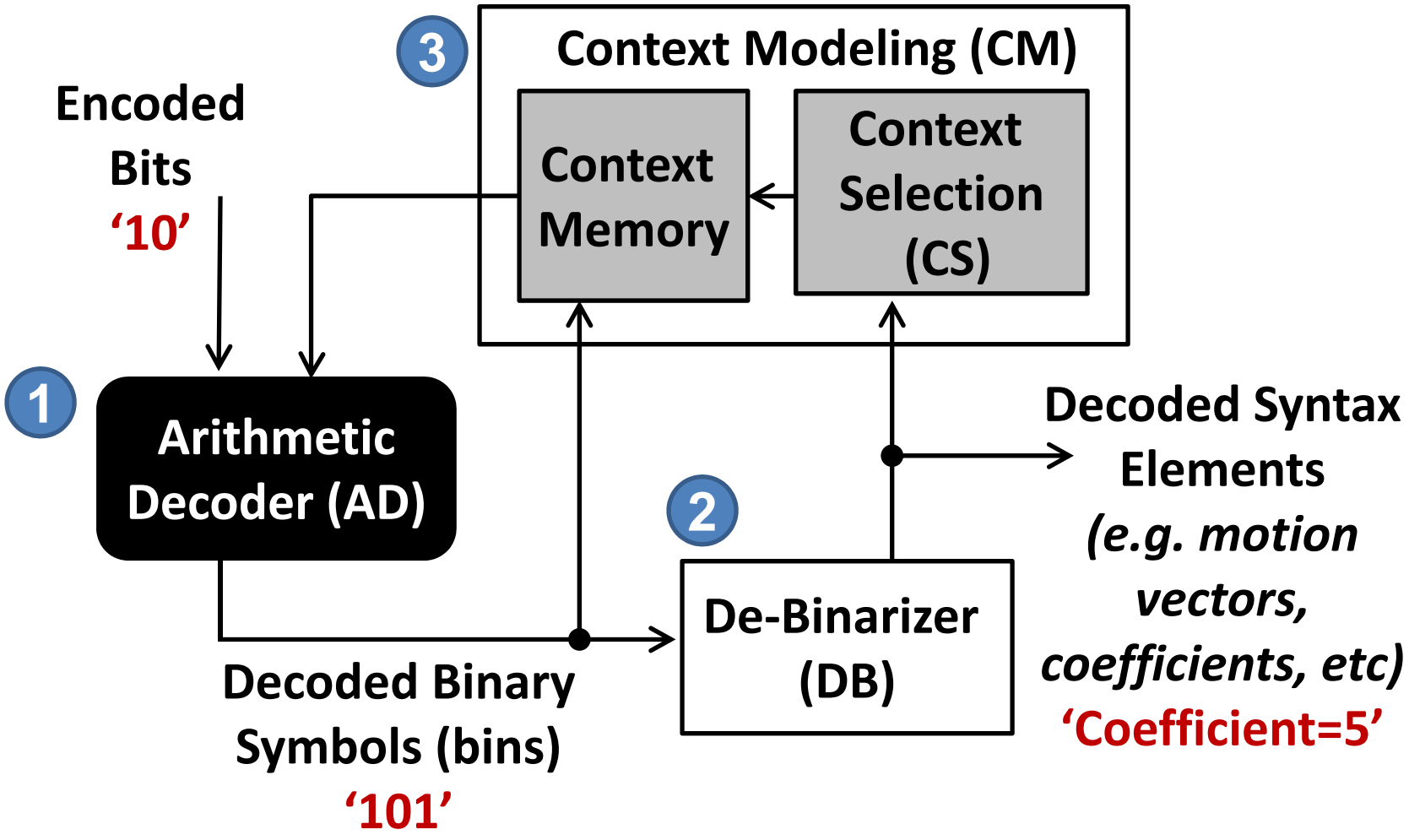


© 2011 IEEE

© 2011 IEEE

Key Blocks in CABAC Decoder

Function: Encoded Bits → Decoded Syntax Elements



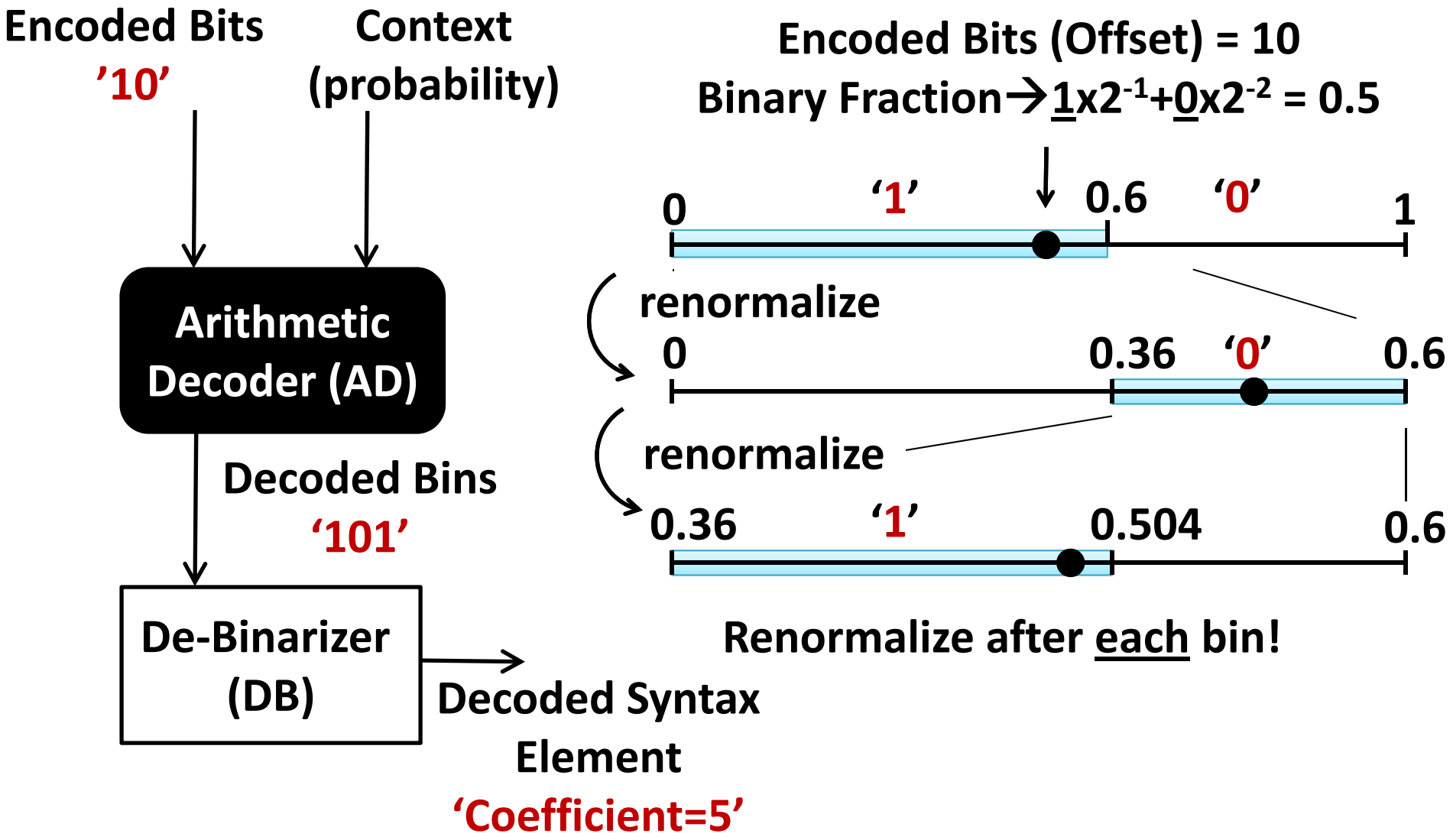
[Marpe, IEEE Trans. CSVT, July 2003]

© 2011 IEEE

© 2011 IEEE

Arithmetic Decoder (AD)

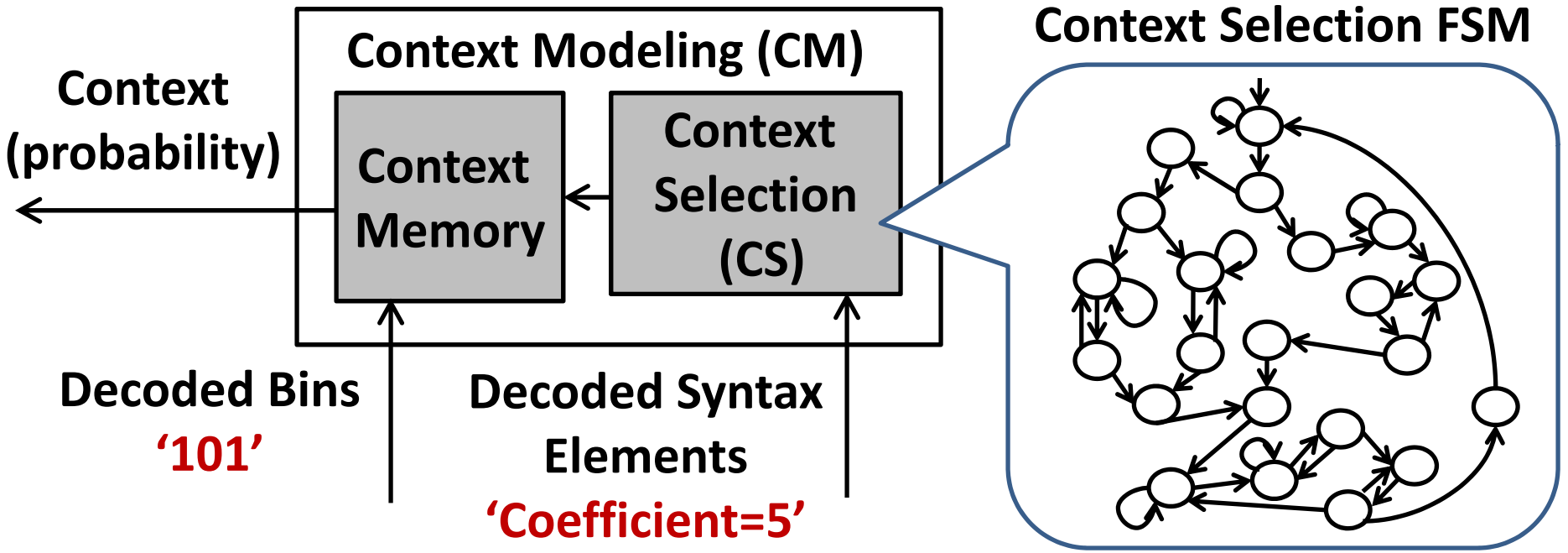
Function: Encoded Bits → Decoded Binary Symbols (Bins)



© 2011 IEEE

© 2011 IEEE

Context Modeling (CM)



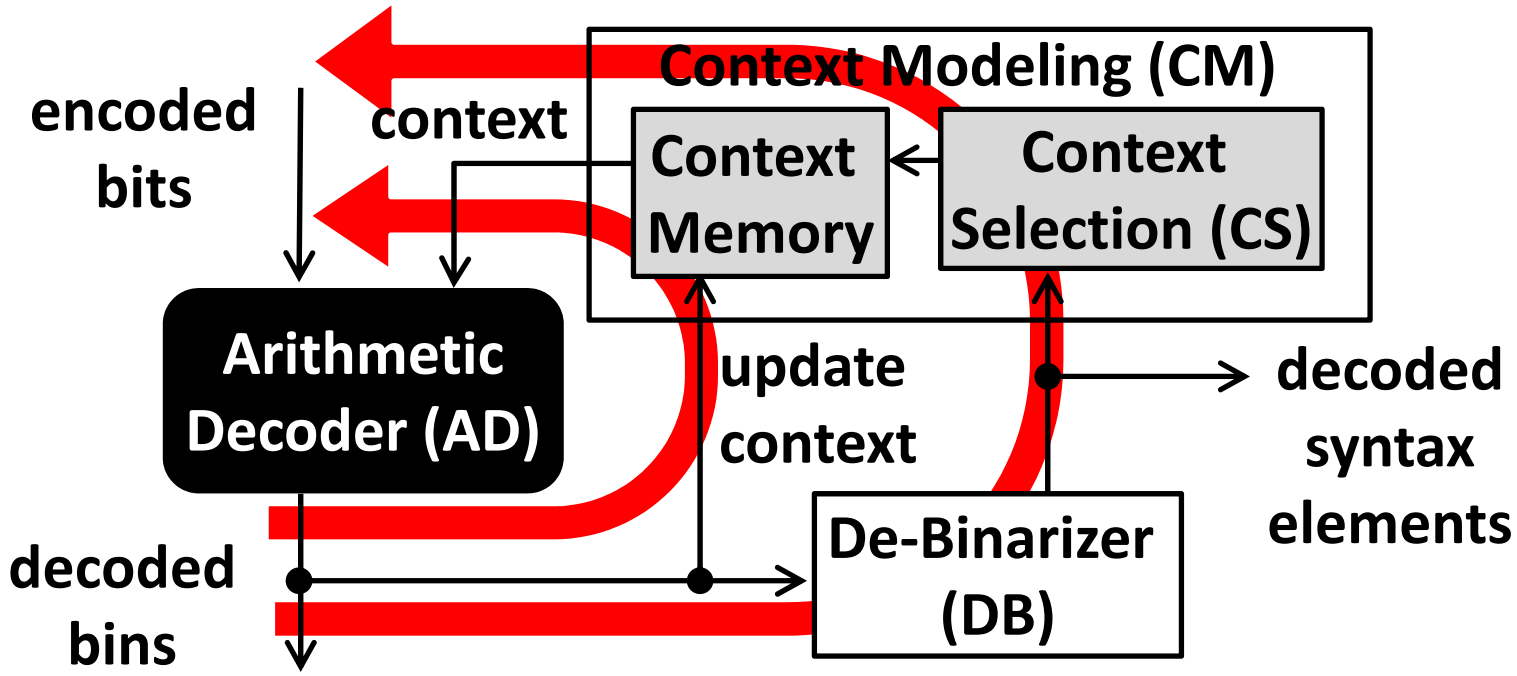
Function: Provide probability estimate to AD

- Accuracy of probability affects coding efficiency of AD
- Select one of several hundred contexts (probability models)
  - syntax element, binIdx, luma/chroma, neighbors, etc.
- Update probability after **each** decoded bin

© 2011 IEEE

© 2011 IEEE

Bottleneck Due To Feedback Loops



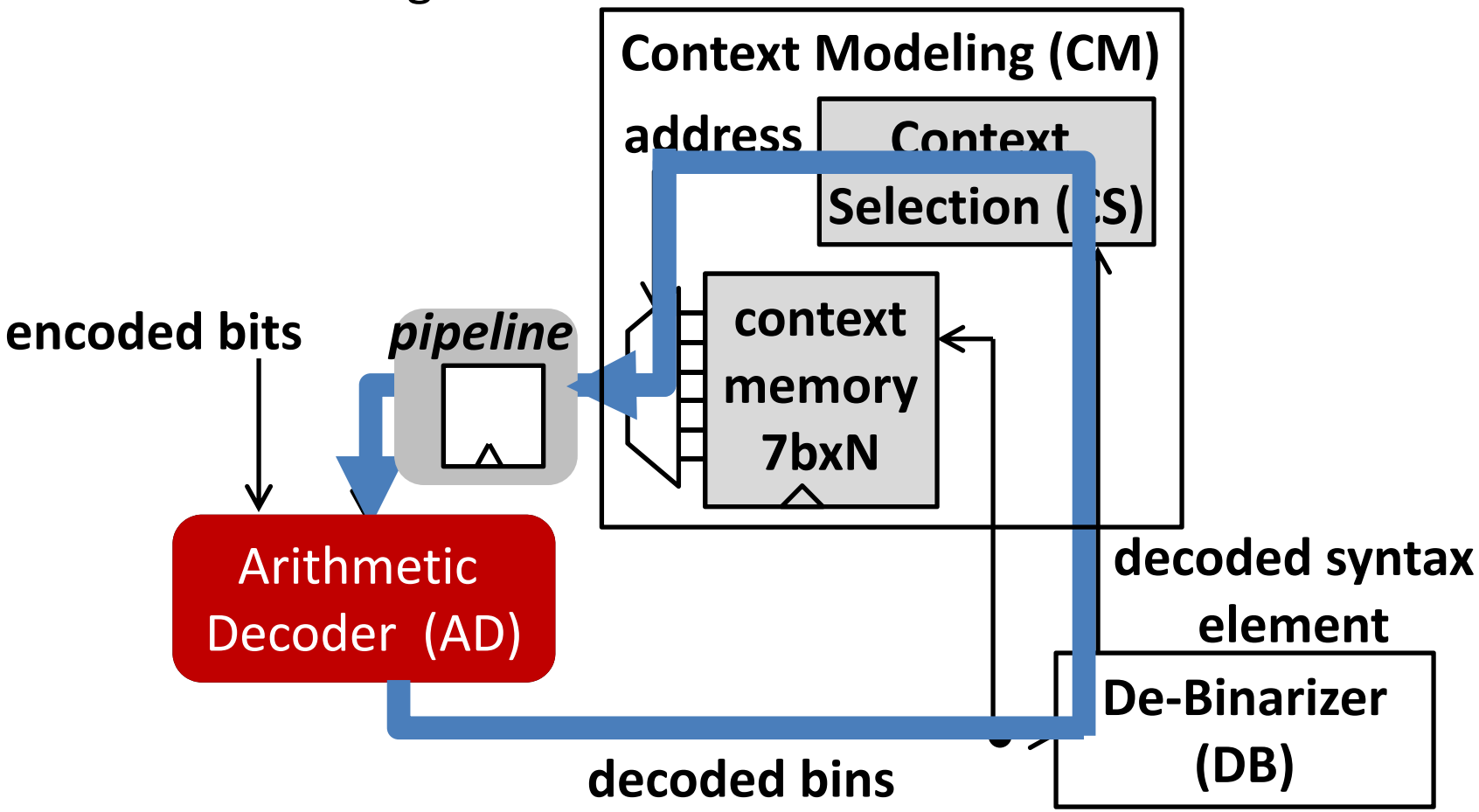
- Data dependencies result in **feedback**
  - Range and contexts are updated after **every bin**
- To increase performance(bins/sec)& throughput(bins/cycle)
  1. Speed up loop (pipeline CABAC & reduce delay of AD)
  2. Replicate loop (run multiple loops in parallel)

© 2011 IEEE

© 2011 IEEE

Speed Up Loop

- Pipeline to reduce critical path
  - Insert register between context modeler and AD

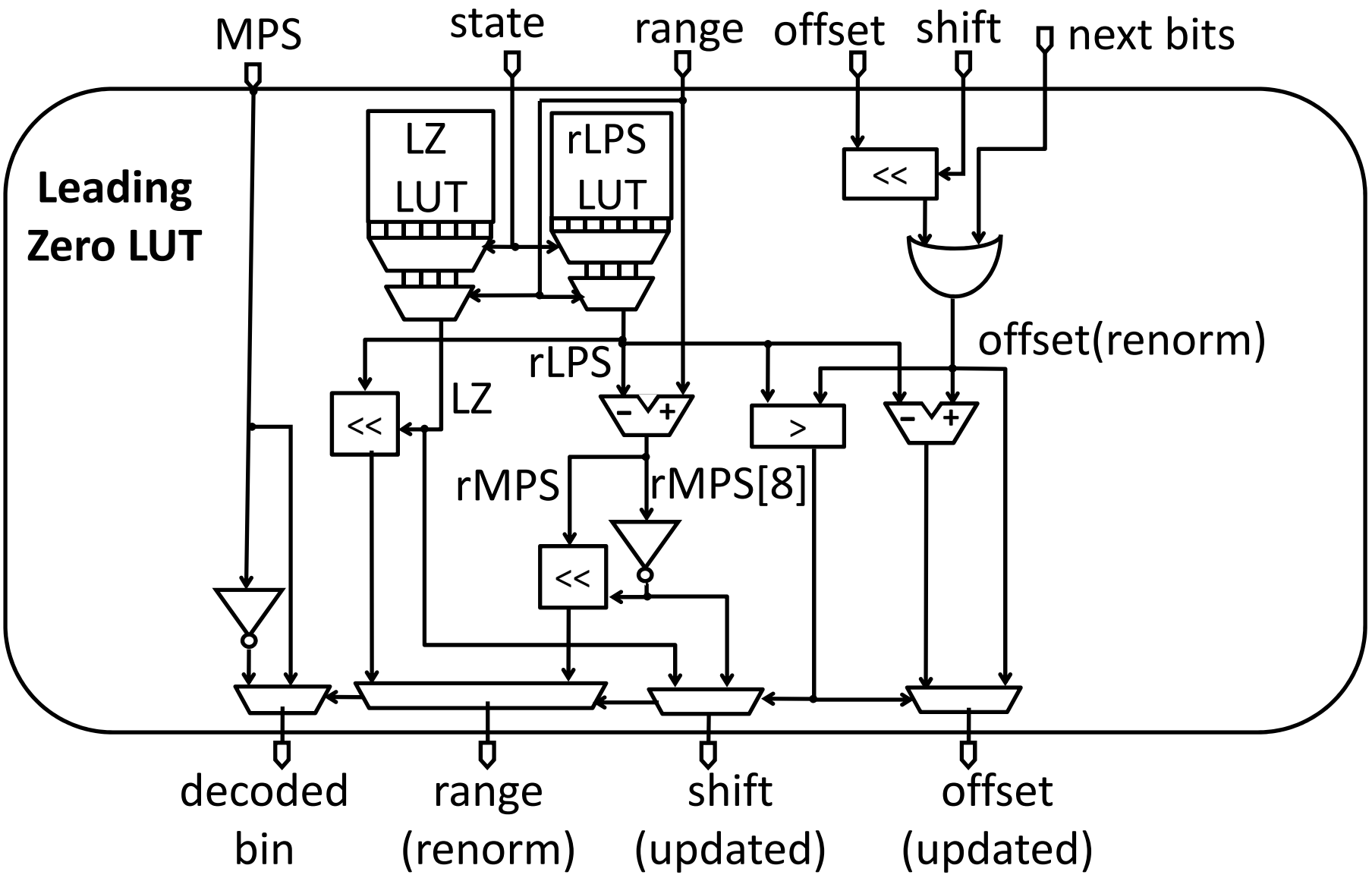


Critical path delay reduced by **40%**

© 2011 IEEE

© 2011 IEEE

Architecture Optimizations in AD

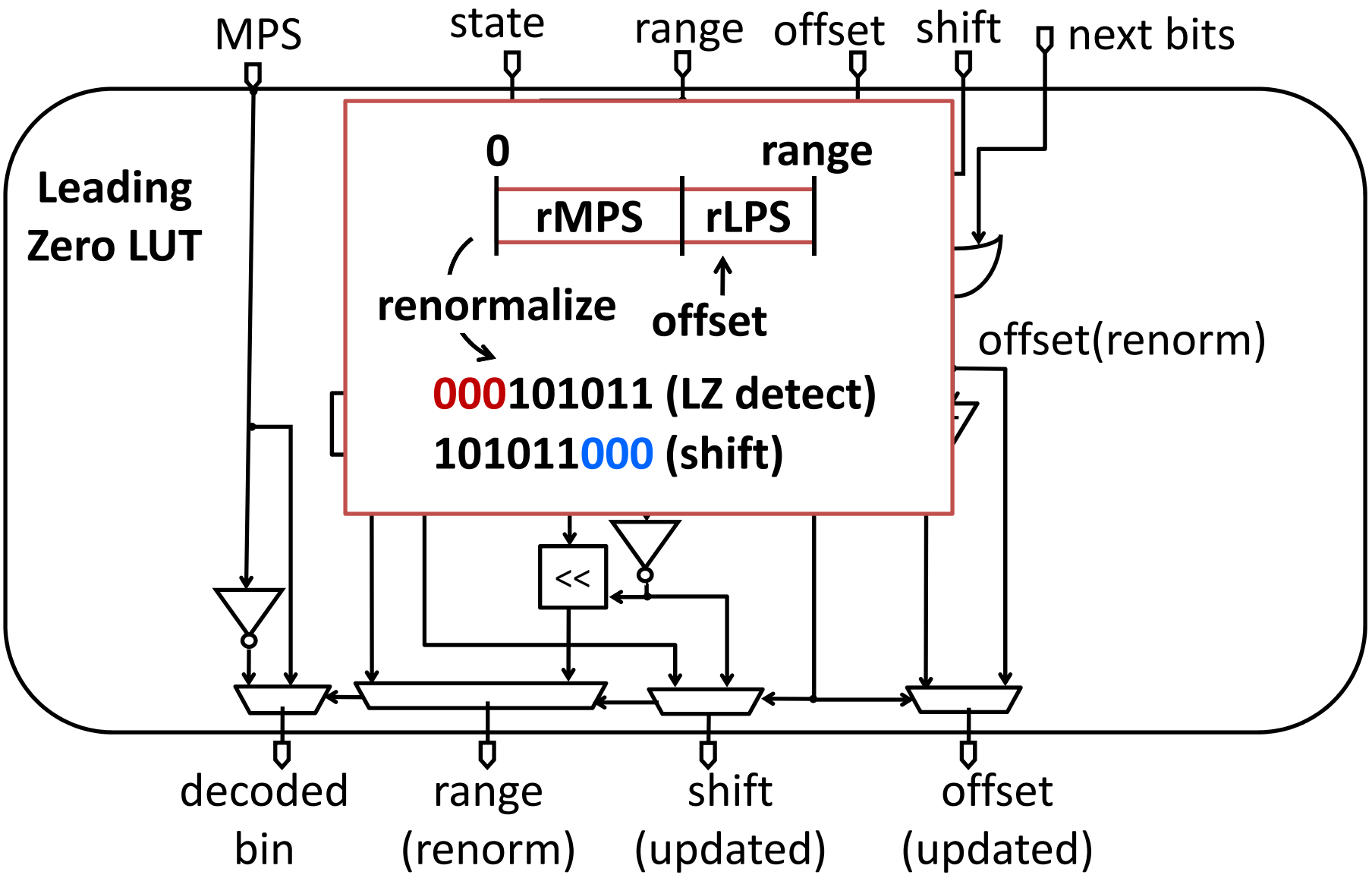


Critical path delay of AD reduced by **22%**

© 2011 IEEE

© 2011 IEEE

Architecture Optimizations in AD

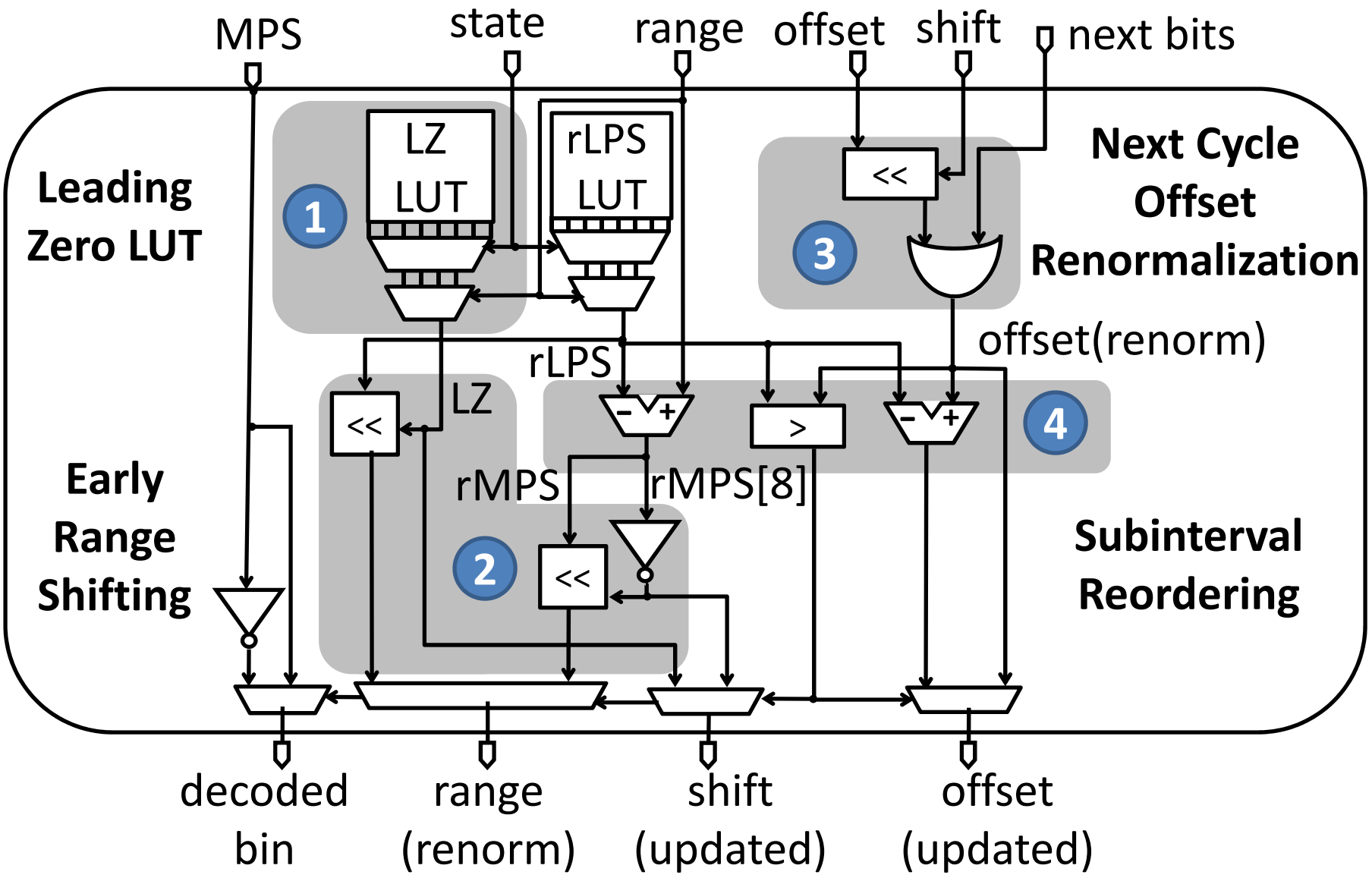


Critical path delay of AD reduced by **22%**

© 2011 IEEE

© 2011 IEEE

Architecture Optimizations in AD

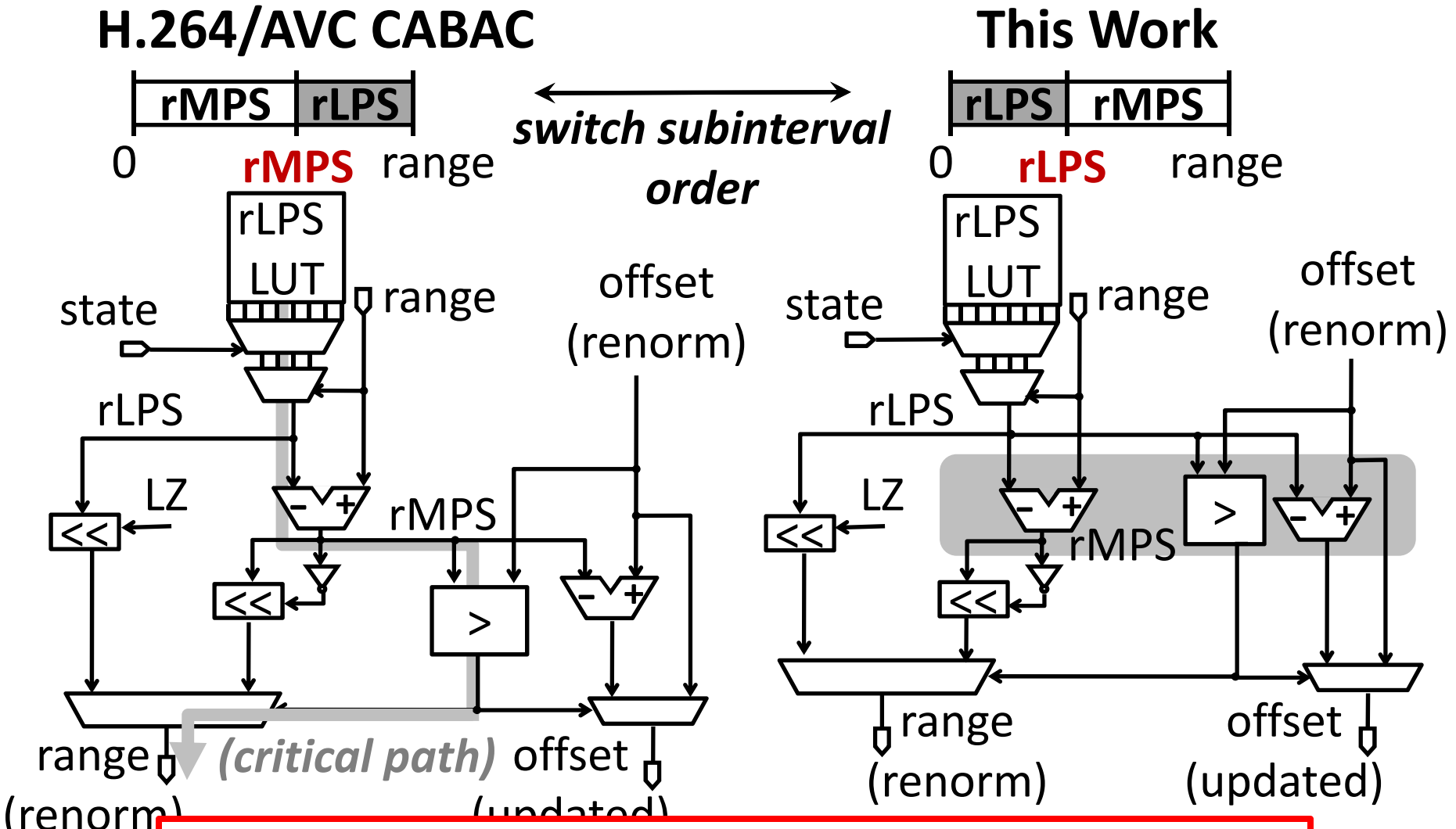


Critical path delay of AD reduced by **22%**

© 2011 IEEE

© 2011 IEEE

Subinterval Reordering



Accounts for **half** of the **critical path reduction**

No impact on coding efficiency

© 2011 IEEE

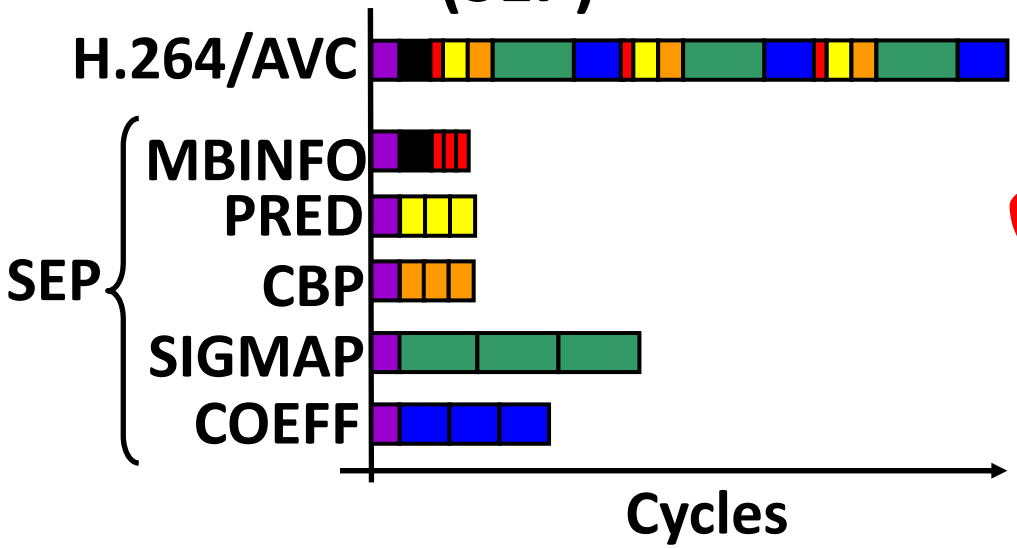
© 2011 IEEE

Massively Parallel CABAC

Our proposed approach for HEVC

Efficient trade-off between throughput and coding efficiency

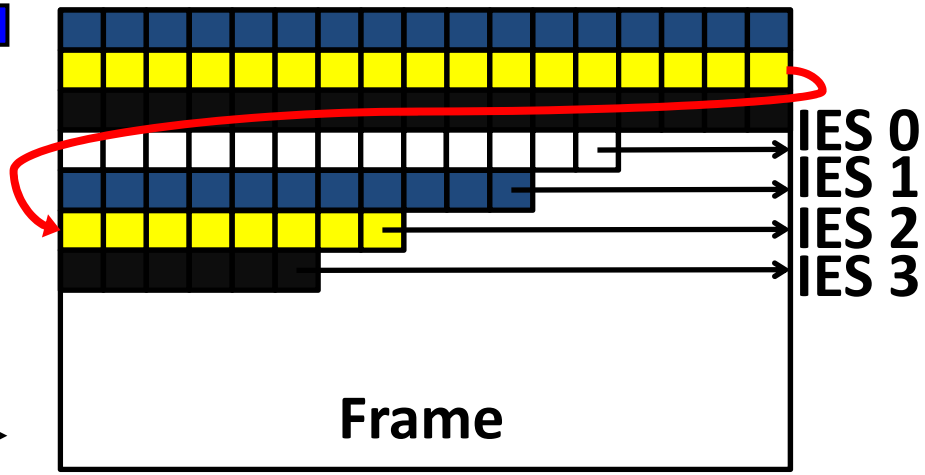
Syntax Element Partitions (SEP)



Proposed SEP can be processed in parallel

[Sze, IEEE ICIP, Nov. 2009]

Interleaved Entropy Slices (IES)



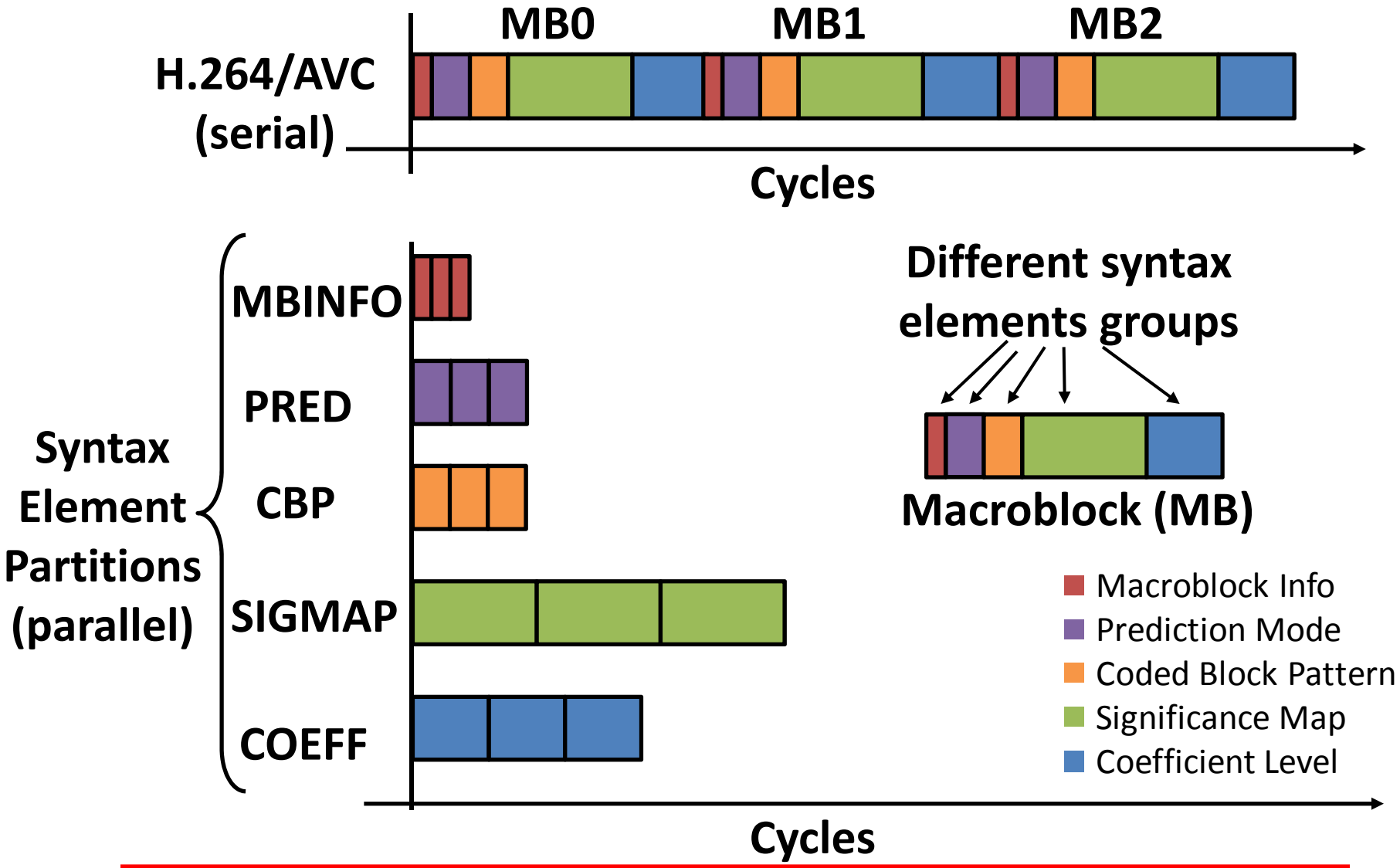
Proposed IES can be processed in parallel

[Finchelstein, IEEE Trans. CSVT, Nov. 2009]

© 2011 IEEE

© 2011 IEEE

Syntax Element Partitions (SEP)

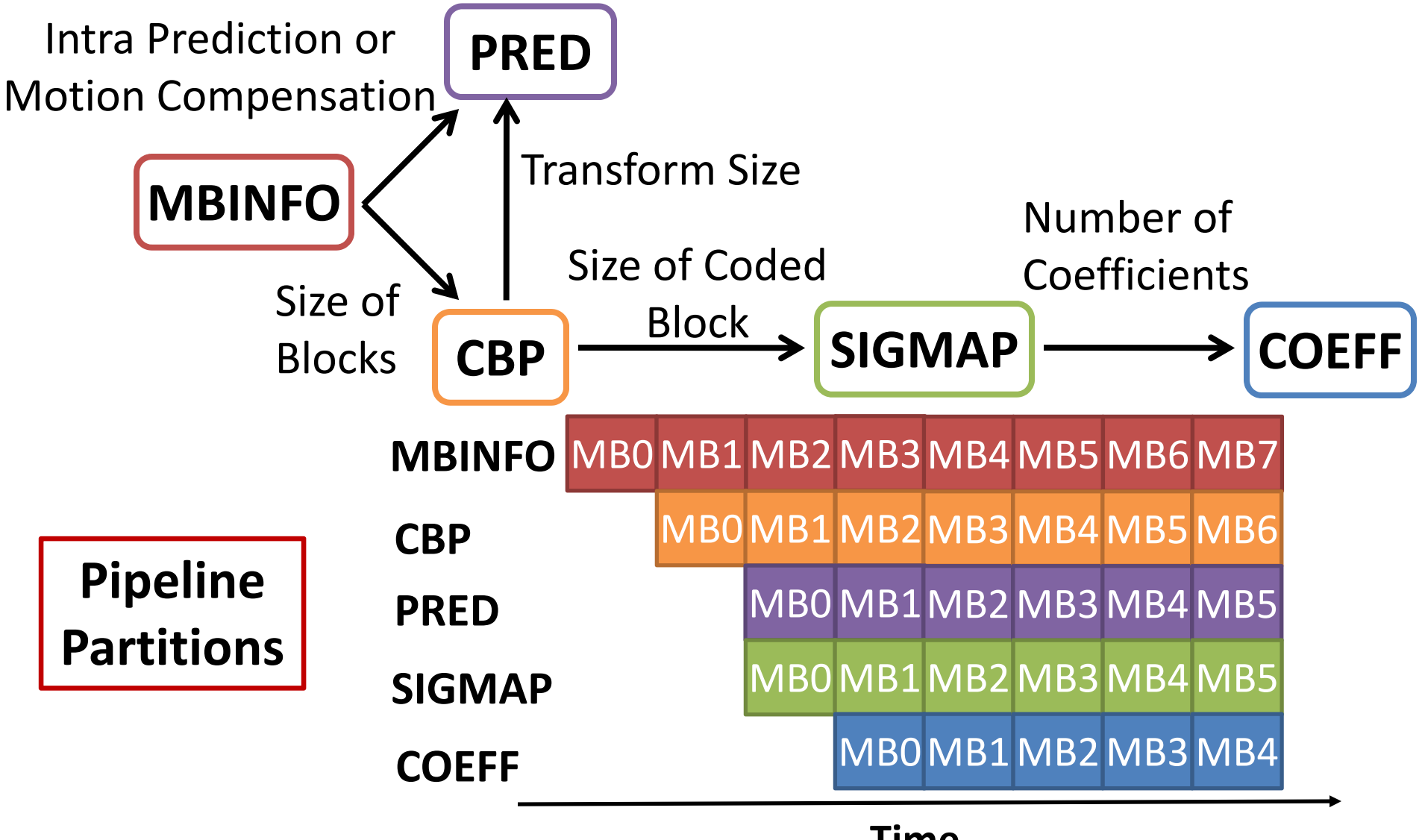


Syntax elements processed in **parallel**

© 2011 IEEE

© 2011 IEEE

Dependencies Between Partitions



Pipeline Partitions

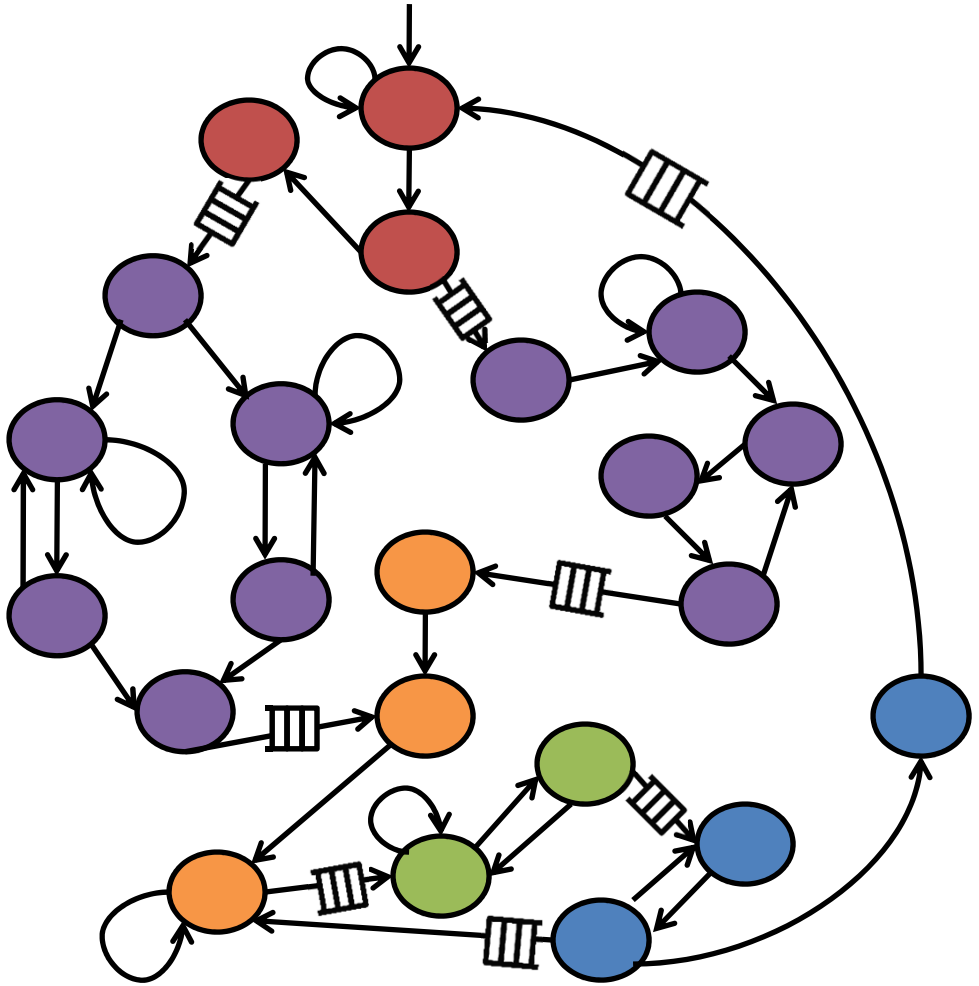
Use **FIFOs** to **synchronize partitions**

© 2011 IEEE

© 2011 IEEE

Distribute FSM for SEP

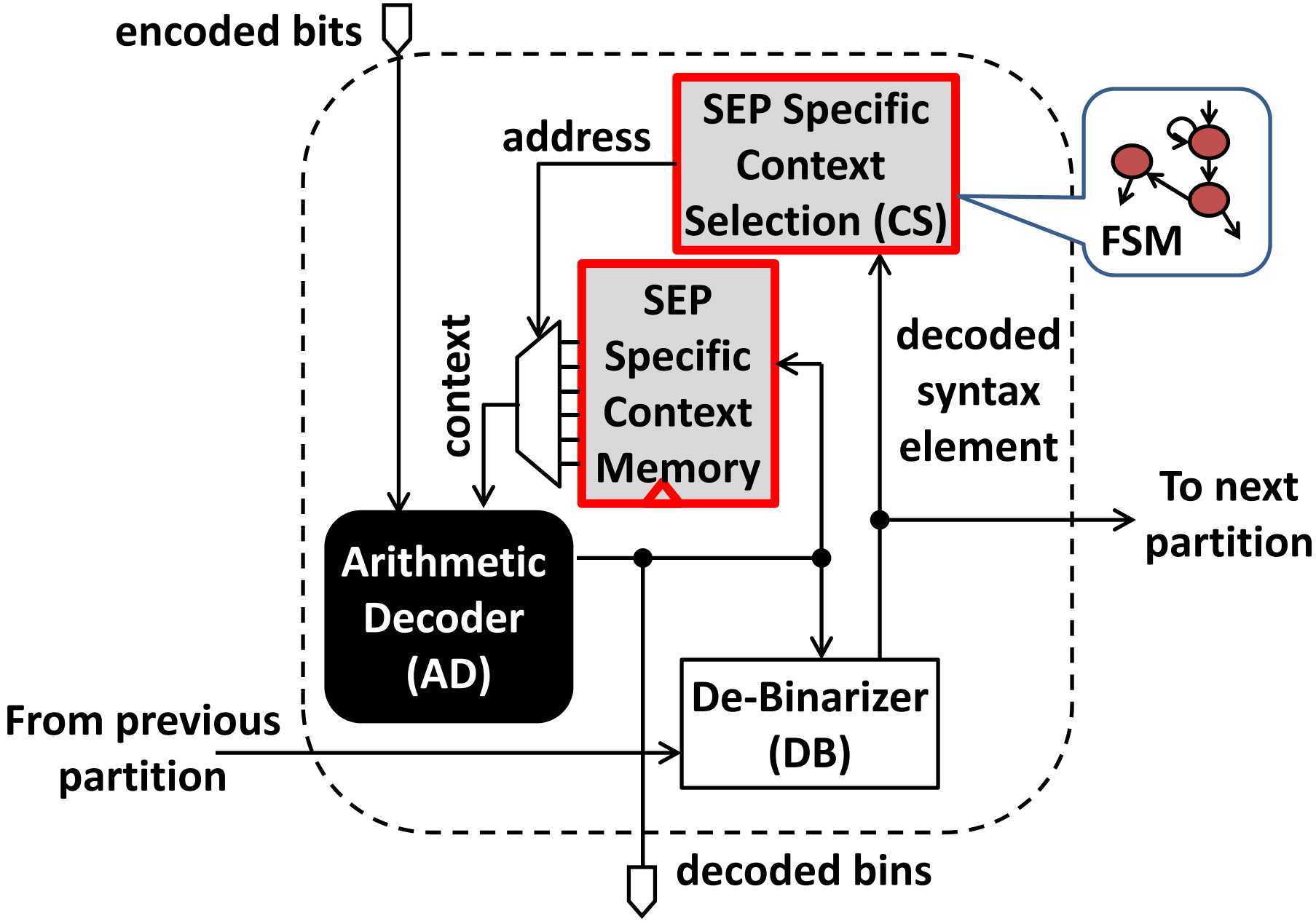
- Divide FSM and context memory into smaller ones based on SEP
- **Partition engines** operate in parallel on different SEP
- **<u>Low area cost</u>** since only replicate AD



© 2011 IEEE

© 2011 IEEE

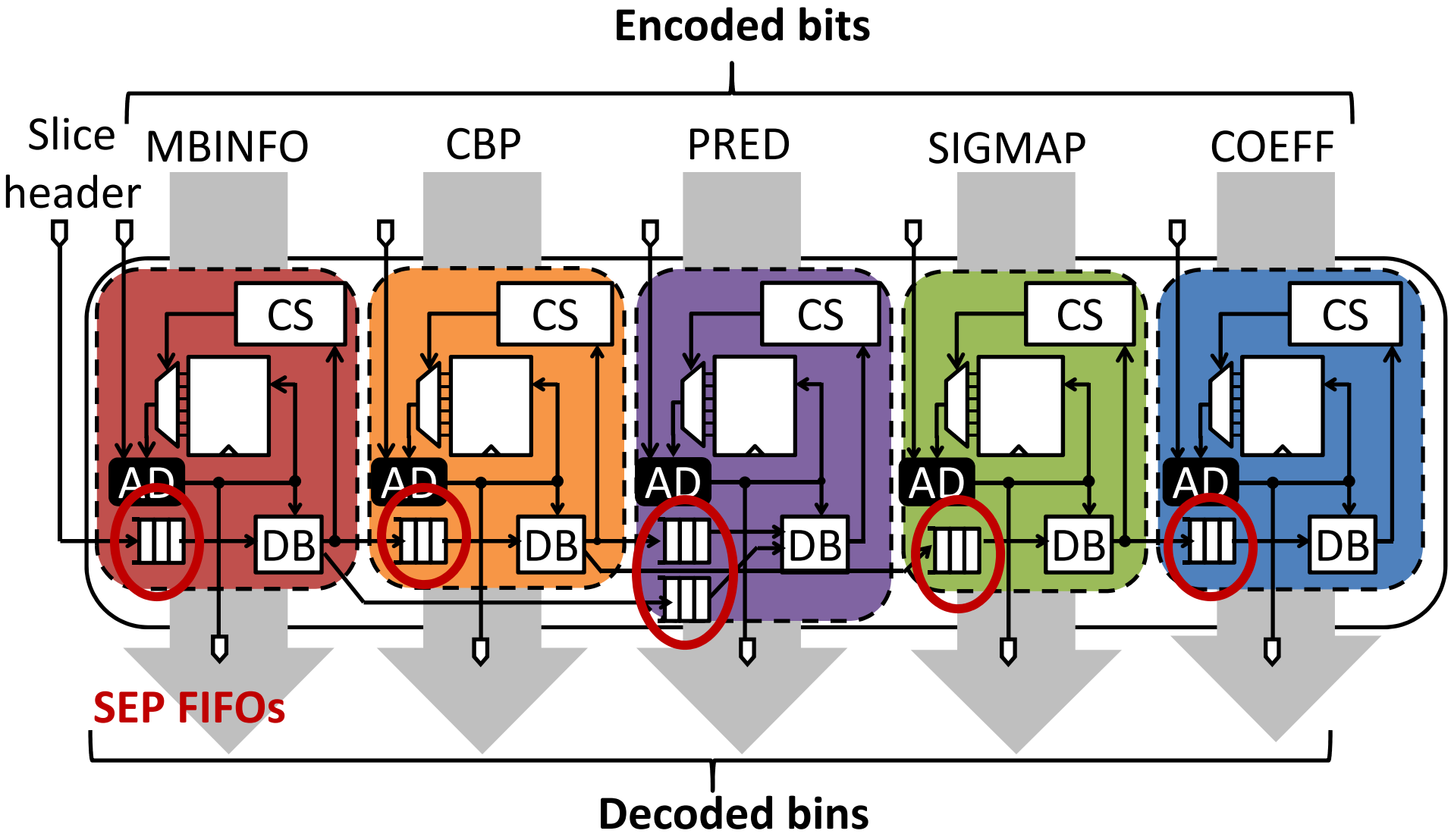
Partition Engine (PE)



© 2011 IEEE

© 2011 IEEE

Slice Engine Architecture

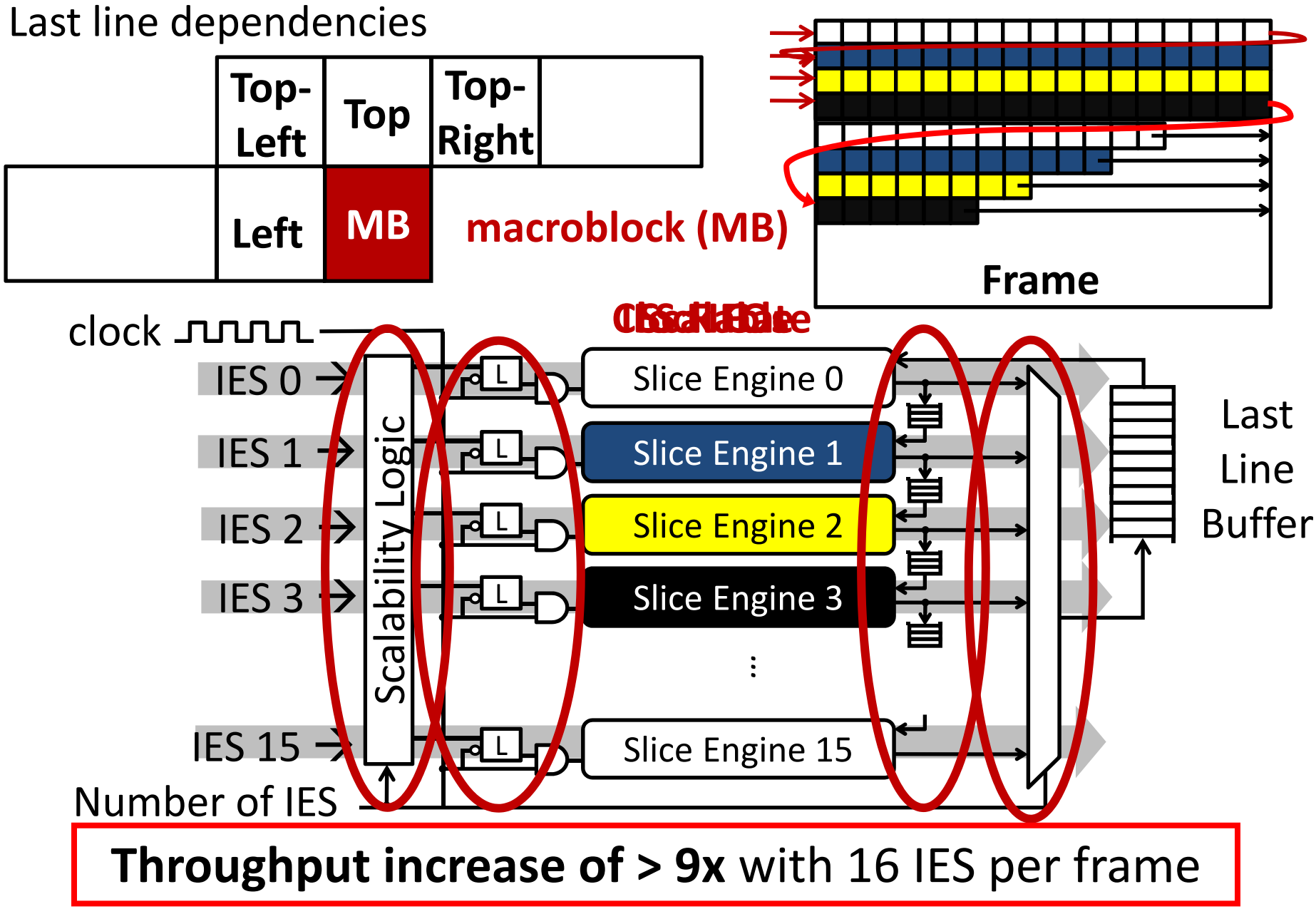


Average **throughput** increase of **2.4x**

© 2011 IEEE

© 2011 IEEE

Interleaved Entropy Slices (IES)



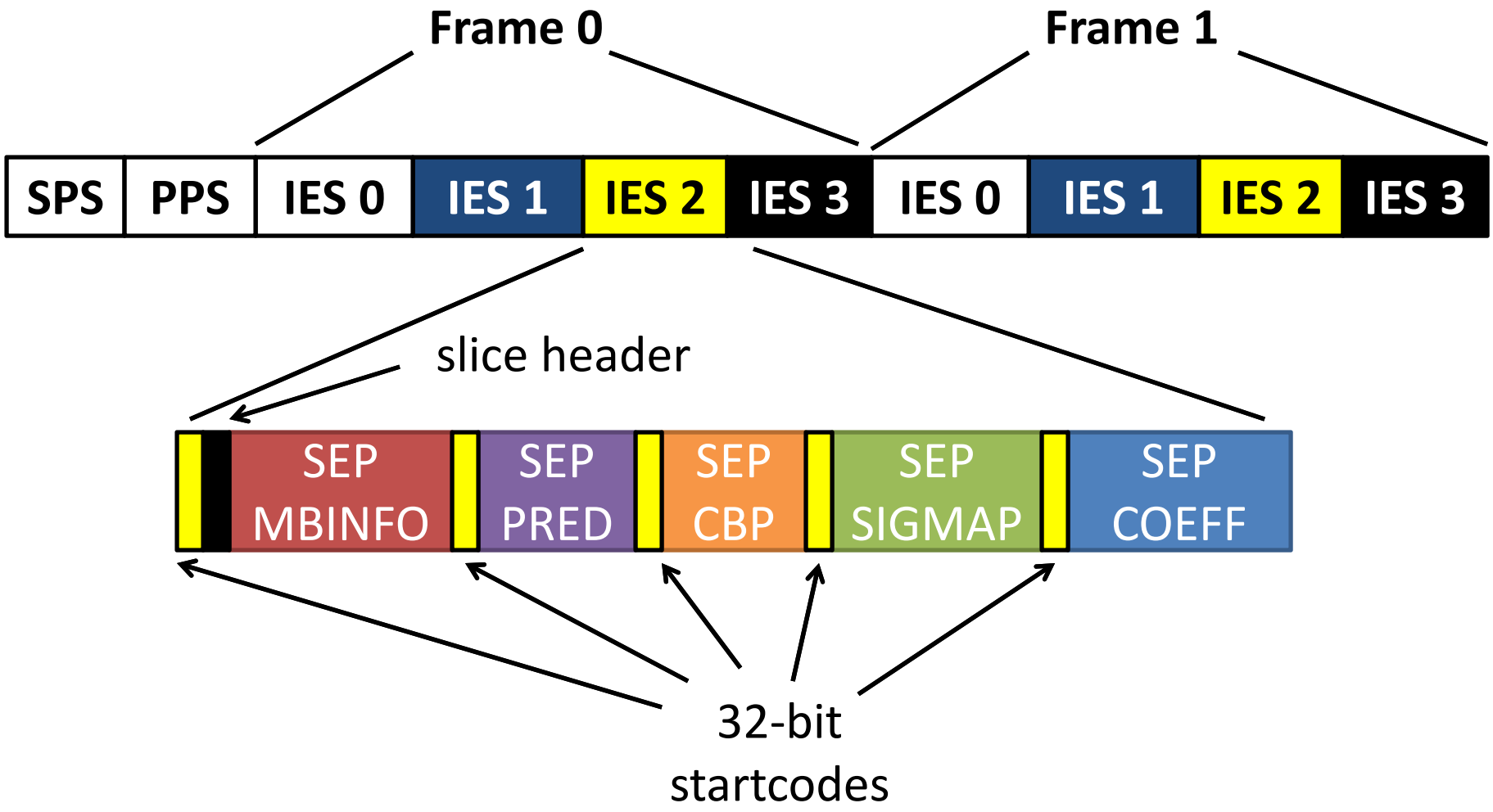
Throughput increase of > 9x with 16 IES per frame

© 2011 IEEE

© 2011 IEEE

Massively Parallel CABAC Data Structure

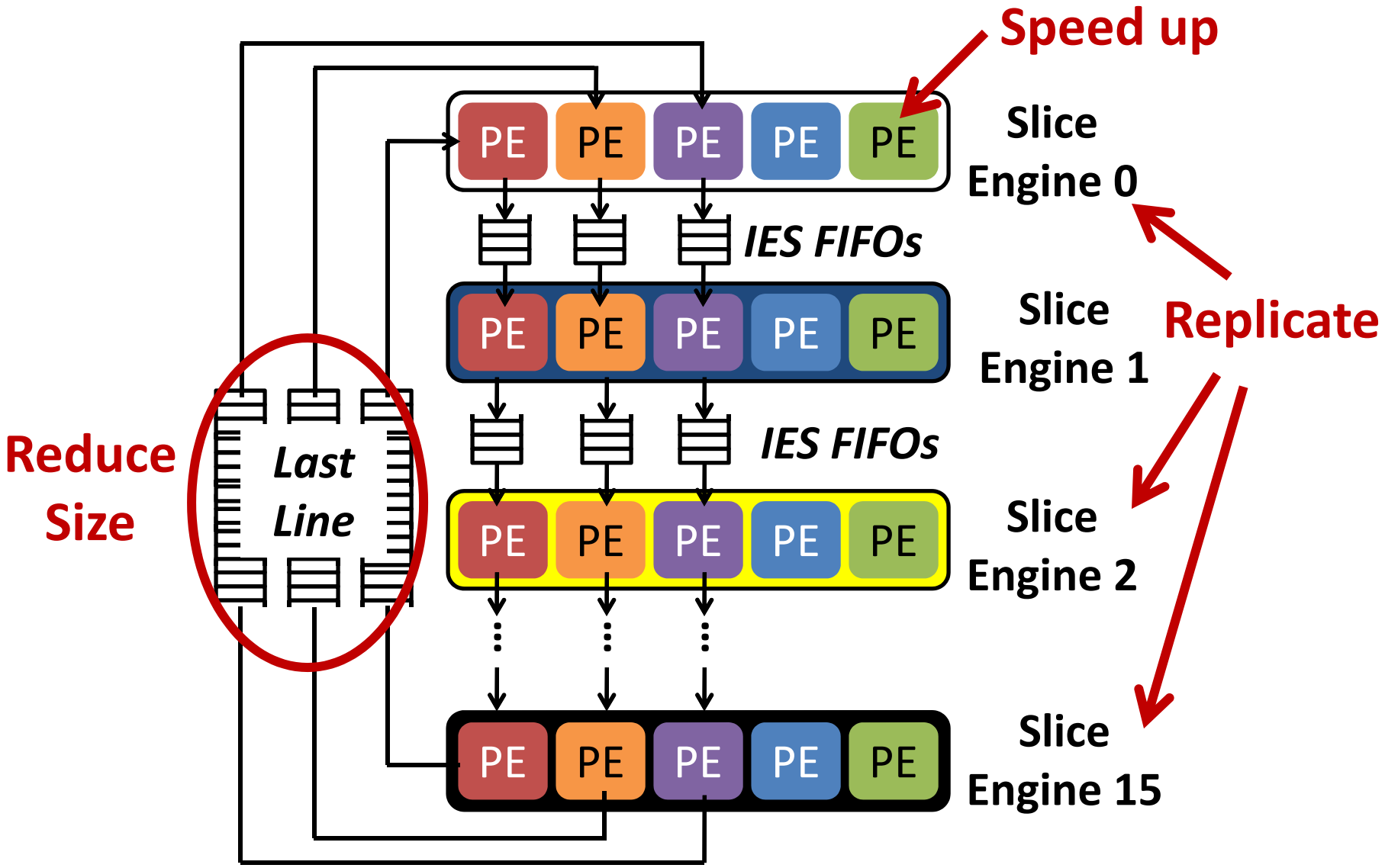
Example with four of IES per frame



© 2011 IEEE

© 2011 IEEE

Optimize MP-CABAC Architecture

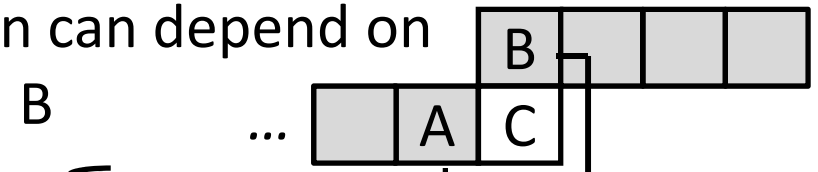


© 2011 IEEE

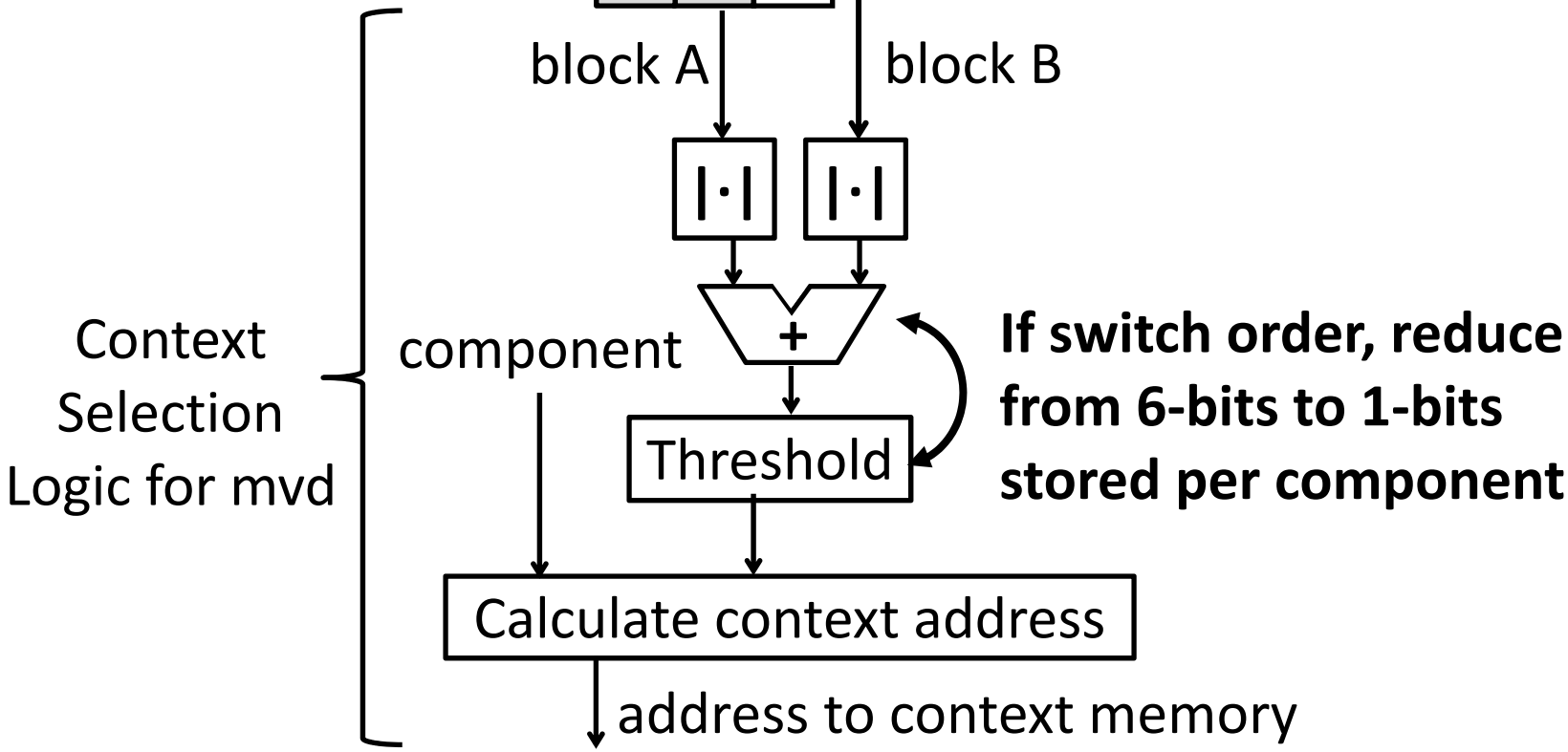
© 2011 IEEE

Reduce Last Line Buffer Size

Context selection can depend on neighbors A and B



... store last line info (grey) in buffer



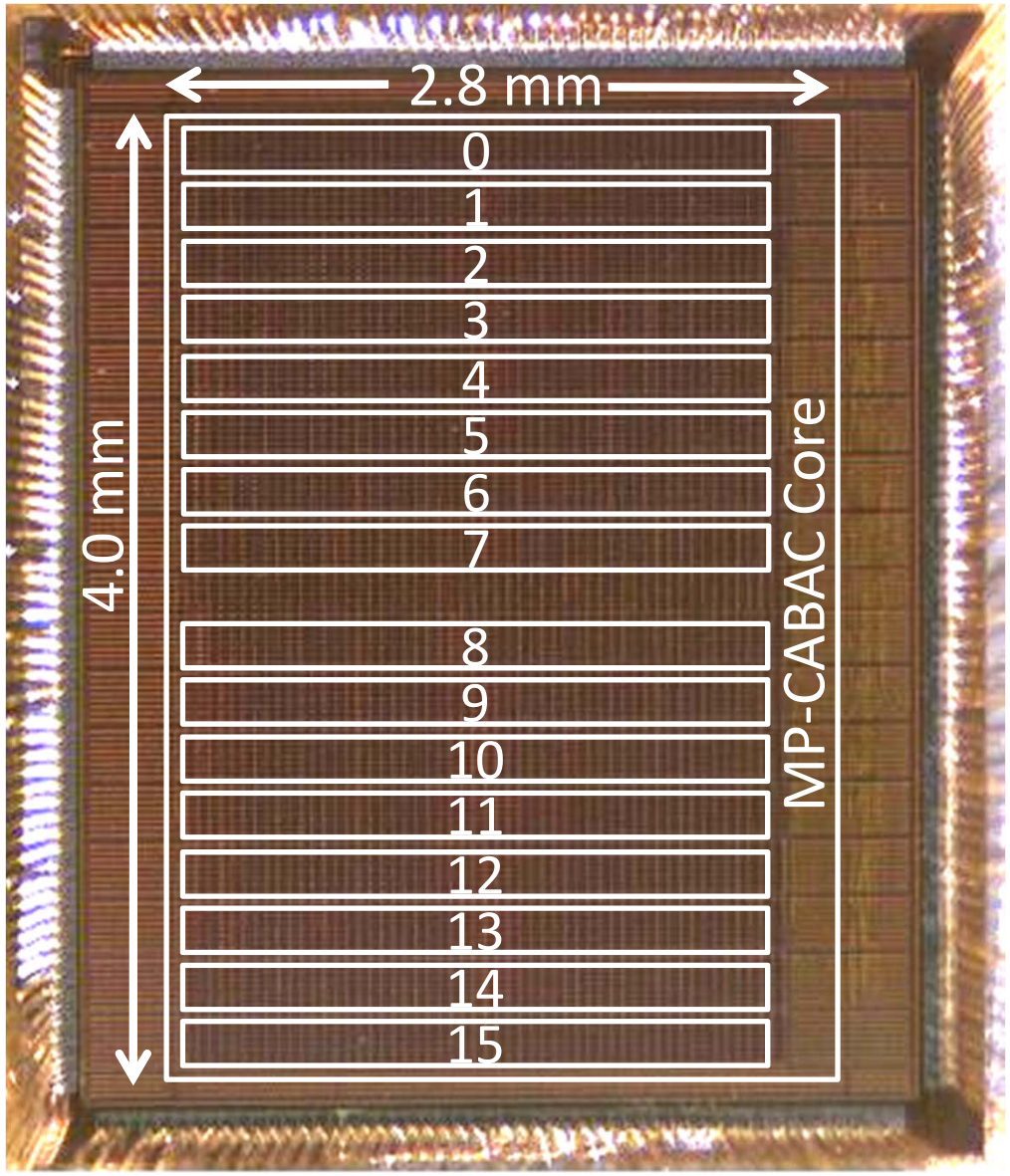
Reduce overall CABAC **Last Line Buffer Size by 50%**
Negligible impact on coding efficiency (0.02%)

© 2011 IEEE

© 2011 IEEE

Massively Parallel CABAC Test-Chip

Standard	Massively Parallel CABAC proposed for HEVC
Process	65-nm CMOS
Transistor Count	9.74 M (includes all memory)
Max Resolution	QFHD/4kx2k (4096x2160)



© 2011 IEEE

© 2011 IEEE

Other CABAC Implementations

Publication	This Work	ISSCC' 11 paper 7.1	TCE '09	TVLSI '09	TCSVT '09
Standard	proposed for HEVC	H.264/ AVC	H.264/ AVC	H.264/ AVC	H.264/ AVC
Peak bins/cycle	80	2	2	16	2*
Avg. bins/cycle	**24.11**	**1.95**	**1.32**	**2.27**	**0.86**
Process (nm)	65	40	130	180	180
Voltage	1.0	1.0	1.2	1.8	1.8
Results	Measured	Measured	Synthesis	Synthesis	Synthesis
Max Freq. (MHz)	125.5	240	238	45	105
Perf. (Mbins/s)	**3026**	**468**	**314**	**102**	**90**

* 4 bins/cycle for bypass

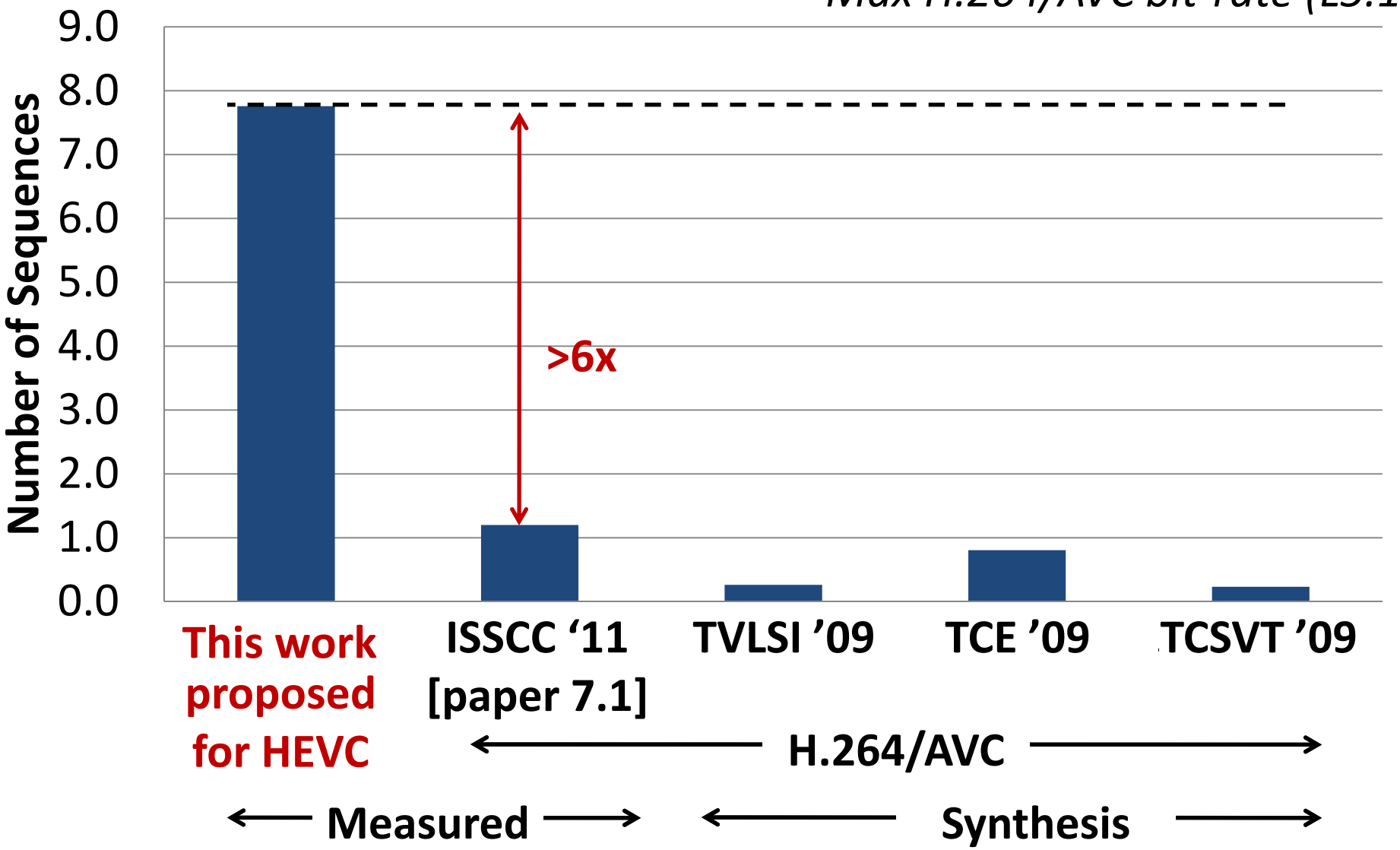
© 2011 IEEE

© 2011 IEEE

High Performance Comparison

Max Number Sequences with 300 Mb/s* Decoded by CABAC

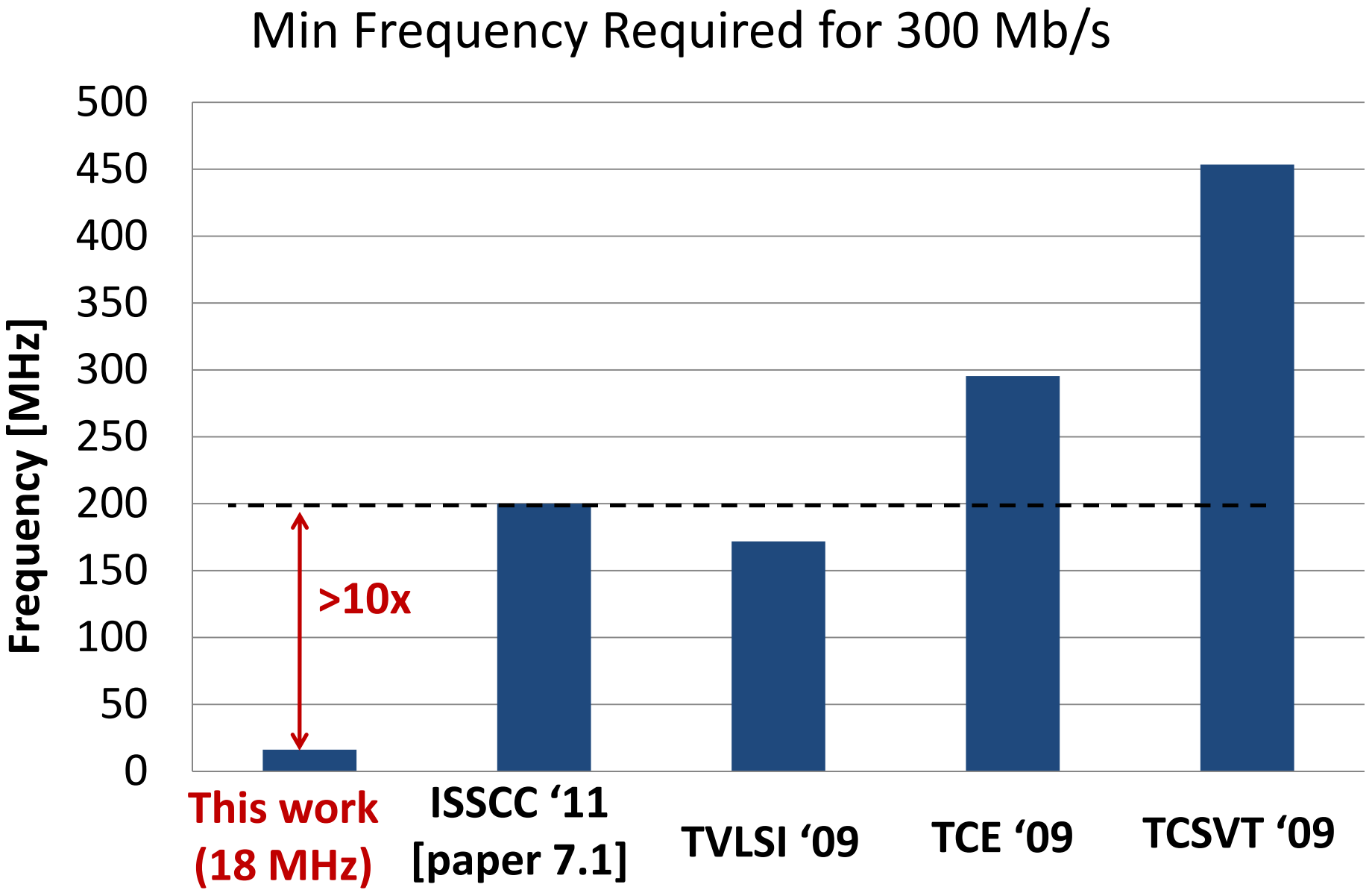
**Max H.264/AVC bit-rate (L5.1)*



© 2011 IEEE

© 2011 IEEE

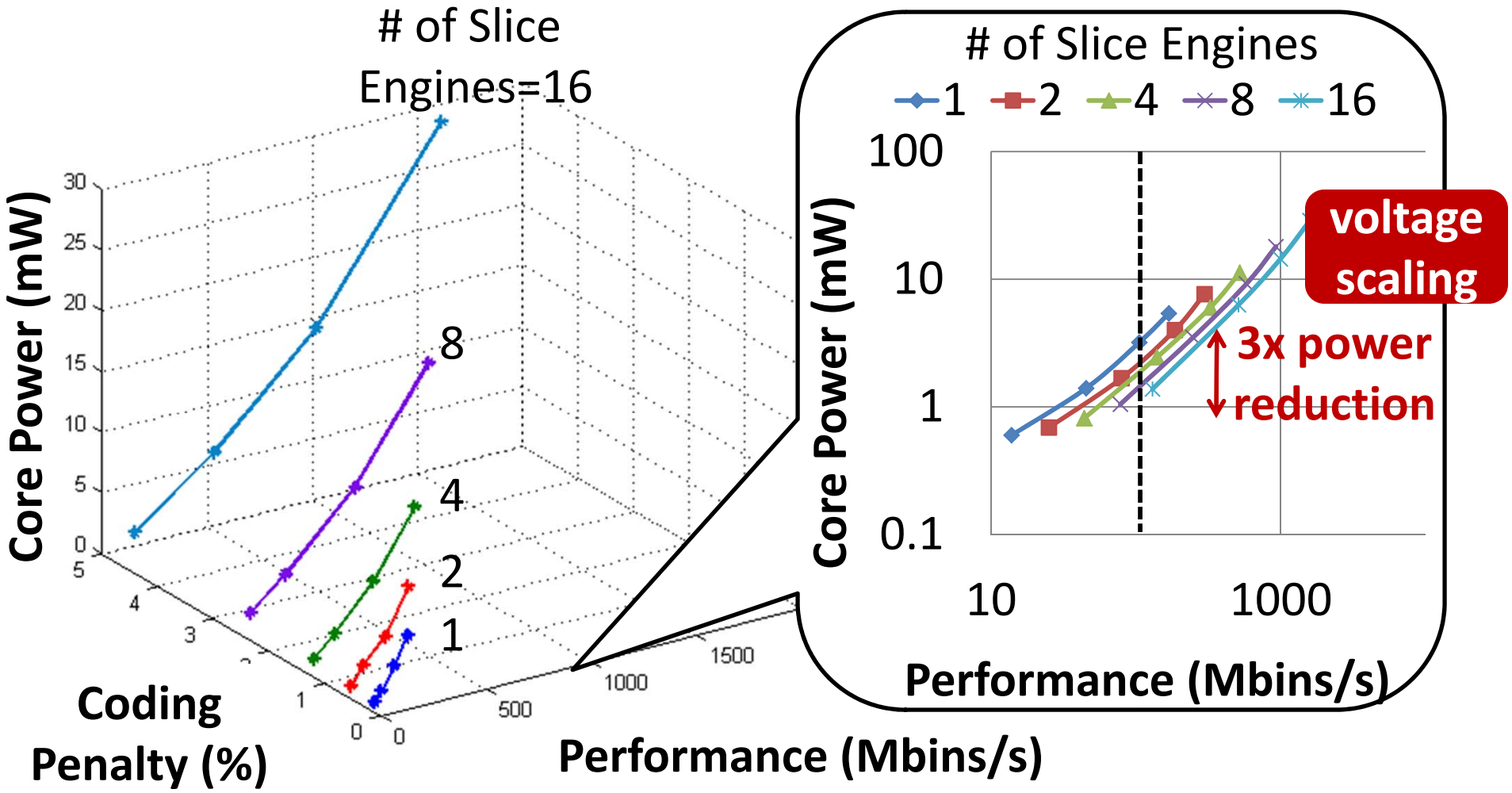
Low Power Comparison



© 2011 IEEE

© 2011 IEEE

Trade-offs



Scalability allows for efficient **trade-off** in three dimensions

4x lower coding penalty than H.264/AVC slices

© 2011 IEEE

© 2011 IEEE

Massively Parallel CABAC Summary

Standard	Massively Parallel CABAC proposed for HEVC
Process	65-nm CMOS
Transistor Count	9.74 M (includes all memory)
Max Resolution	QFHD/4kx2k (4096x2160)
Voltage	0.6 to 1.0 V
Operating Frequency	5.5 to 125.5 MHz
Bin-rate	14 to 3026 Mbins/s
Core Power	0.6 to 77 mW
Energy Metric	10.5 pJ/bin @ 0.6 V (Min Energy Point)
	12.3 pJ/bin @ 0.7 V (Max H.264/AVC bit-rate)

© 2011 IEEE

© 2011 IEEE

Conclusions

- **New CABAC algorithm** proposed for next generation standard **HEVC** provides an efficient **trade-off** between throughput and coding efficiency
  - Over **4x lower coding penalty** than H.264/AVC slices
- Algorithm mapped to a **highly parallel** architecture
  - Over **10x higher throughput** than state-of-the-art H.264/AVC
- **Joint Algorithm/Architecture Optimizations** can reduce delay and memory requirements with little or no cost in coding efficiency

Acknowledgements: Texas Instruments for funding, fabrication and algorithm support

© 2011 IEEE

© 2011 IEEE