PARALLEL CABAC FOR LOW POWER VIDEO CODING

Vivienne Sze
Anantha P. Chandrakasan
Massachusetts Institute of Technology

Madhukar Budagavi
Minhua Zhou
Texas Instruments

Motivation

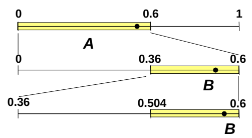
- High demand for video capture and playback on battery operated devices
- Battery capacity limited by size, weight, and cost of devices
- Need low power video coding
- Maintain performance required for real-time coding
- CABAC is a form of entropy coding used in H.264/AVC High Profile

Context Adaptive Binary Arithmetic Coding

Binary Arithmetic Coding

Example: Pr(*A*) = 0.6; Pr(*B*) = 0.4

Encoding Symbol Sequence: "***A-B-B***"



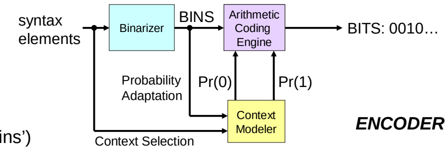
Range	Offset
1	0
0.6	0
0.24	0.36

Output Binary Bitstream: .1001

Binarizer maps syntax elements (SE) to binary symbols ('bins')

Context-Adaptive

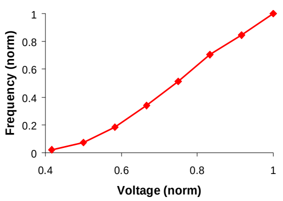
- Context (probability model)
- Adaptive estimation of probability (update context state)
- *Context can be switched and updated **every** bin*
  - *Bin-to-bin dependencies → Cycle-to-cycle dependencies*

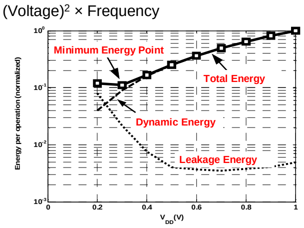


Low Power with Parallelism

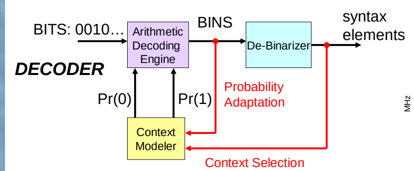
Parallelism → Reduce Cycles (Increase Throughput) → Lower Frequency → Lower Voltage → Lower Energy

Power = Capacitance × $(\text{Voltage})^2$ × Frequency

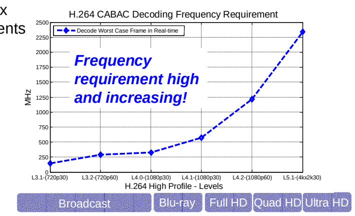




CABAC Challenges



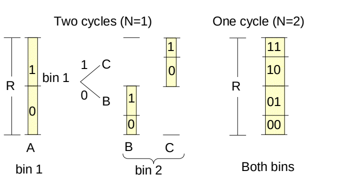
- Binary Arithmetic Coding maps bins to bits
  - Number of cycles proportional to number of bins
- Data Dependencies (difficult to parallelize)
  - Contexts and Range are updated after every bin
  - At decoder, data feedback required



Encoder: SE → Bins → Bits
Decoder: Bits → Bins → SE

Parallel CABAC

N-bins/cycle Arithmetic Coding



Quantization of Range and Multiplication

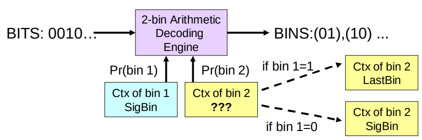
- Renormalization occurs every two bins
  - Range is quantized to 14-bits
- Quantized Multiplication
  - 2-bits by 6-bits by 6-bits (Range x Probabilities)
  - True multiplication rather than LUT

Conditional Context Selection

At decoder, context of bin 2 unknown a priori

Example: Significance map decoding

```
significant_coeff_flag (SigBin)
last_significant_coeff_flag (LastBin)
```



$$Pr_{SigBin}(\text{bin } 2) = Pr(\text{bin } 2 \mid \text{bin } 1 = 0)$$
$$Pr_{LastBin}(\text{bin } 2) = Pr(\text{bin } 2 \mid \text{bin } 1 = 1)$$

bin 1	bin 2	Probability
0	0	$Pr_{SigBin}(0) \times Pr_{SigBin}(0)$
0	1	$Pr_{SigBin}(0) \times Pr_{SigBin}(1)$
1	0	$Pr_{SigBin}(1) \times Pr_{LastBin}(0)$
1	1	$Pr_{SigBin}(1) \times Pr_{LastBin}(1)$

Skip Probability State Update

If bin 1 has context A [state σ_A], and bin 2 also has context A.

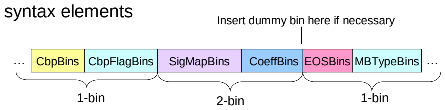
For 1-bin case
(1) Decode bin 1 with σ_A
(2) Update $\sigma_A \rightarrow \sigma'_A$
(3) Decode bin 2 with σ'_A
(4) Update $\sigma'_A \rightarrow \sigma''_A$

For 2-bin case
(1) Decode bin 1 and 2 with σ_A
(2) Update $\sigma_A \rightarrow \sigma''_A$

Update state every N bins

Dummy Bin

- For N=2, insert "dummy" bin when odd number of bins
- Encoded with skewed probability for coding efficiency
- Enables selective use of Parallel CABAC on subset of syntax elements



Results

Experiment Setup

- **Implemented Parallel CABAC (N=2) in JM 12.0**
- **Syntax elements coded with 2-bin/cycle:** `mb_type, mb_qp_delta, coded_block_pattern, Motion Vectors, Reference Index, Intra Prediction Modes, Significance Map, Coefficient Levels`
- **Based on common conditions (VCEG-AE010)**
- **Result obtained across five 720p streams**
  - **BigShips, City, Crew, Night, ShuttleStart**

Coding Efficiency

Sequence	ΔBitrate (BD delta)	ΔPSNR (BD delta)
BigShips	0.67%	-0.020
City	0.73%	-0.024
Crew	0.80%	-0.022
Night	0.80%	-0.031
ShuttleStart	0.81%	-0.029
Average	**0.76%**	**-0.025**

Cycle Reduction (Throughput Increase) for N=2

QP	Average MB	Worst MB
22	1.91×	1.98×
27	1.84×	1.97×
32	1.74×	1.98×
37	1.60×	1.97×

Estimated Power