Background & Motivation

End-user can readily make use of high quality and high capacity multimedia data.

For Fast Processing

- **Repeated operation**: arithmetic-logic operations carried out for all data.
  - **parallel processing**

- **Table-lookup-coding**: coding operation using a code word table
  - Reading from SRAM and comparing
  - Coding by CAM or hard-wired logic
  - Efficient parallel processing is difficult

Architecture Concept

- Combining parallel and sequential processing blocks
  - Decreases the frequency of bus conflicts

- **Saving processing time**
- **Small area and low power consumption**

Media processor

Parallel processing block
- RGB -> YCbCr
- Level shift
- DCT
- Quantization etc

FMCAM
- Huffman coding

Sequential processing block

Internal bus

External data
- (Picture etc)

External memory controller
- (SDRAM etc)

External bus controller
- (PCI etc)

Data / Instruction cache
- (SRAM etc)

Media processor

GOPS: Gig Operations Per Second

Adapted Flexible Multi-ported CAM (FMCAM)

- Bit-Parallel and Block-Parallel (BPBP) search
- Categorization of the stored reference words
- P input/output

- Asynchronous processing at each port
- Multiple/Single search mode
- Counting value setting mode

Adapted FMCAM can stop the process immediately after the first matching symbol.

Implementation results for JPEG Application

Adapted FMCAM processing clock cycles are 43% less than for original FMCAM and 93% less than for DSP.

Comparison of Processing Efficiency

Performance per area size of adapted FMCAM is 1.7 times higher than for original FMCAM and 3.8 times higher than for DSPs (for 16 ports).

Conclusion

- Multi-ported CAM for parallel coding is proposed as a novel architecture for table-lookup-coding.
- In the JPEG application, the clock cycle number of the adapted FMCAM is up to 93% smaller than for a conventional DSP.
- The efficiency of the adapted FMCAM in MOPS/mm² is up to 3.8 times higher than that of conventional parallel operated DSPs.