# *Multi-Level Cache Hierarchy Evaluation for Programmable Media Processors*

***Jason Fritts***

***Assistant Professor***

***Department of Computer Science***

***Co-Author: Prof. Wayne Wolf***

Washington University in St. Louis

## *Overview*

- **Why Programmable Media Processors?**
- **Evaluation Environment**
- **Cache Memory Hierarchy Evaluation**
  - preliminary investigation of memory hierarchy for media processing
- **Conclusions**
- **Future Research**

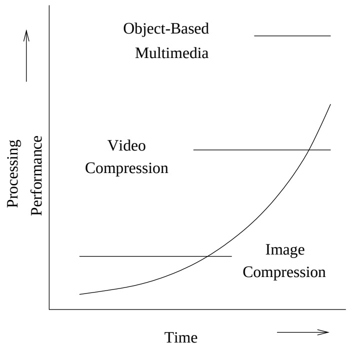
# *Multimedia Applications*

- **Wide range of applications**
  - *Communication*
    - video conferencing
    - World Wide Web
    - digital/video libraries
    - videophones
  - *Entertainment*
    - video/computer games
    - movies
    - animation
  - *Computer Vision*
    - image understanding
    - surveillance
    - tracking
  - *Education*
    - interactive learning
    - virtual classrooms
  - *Art and Architecture*

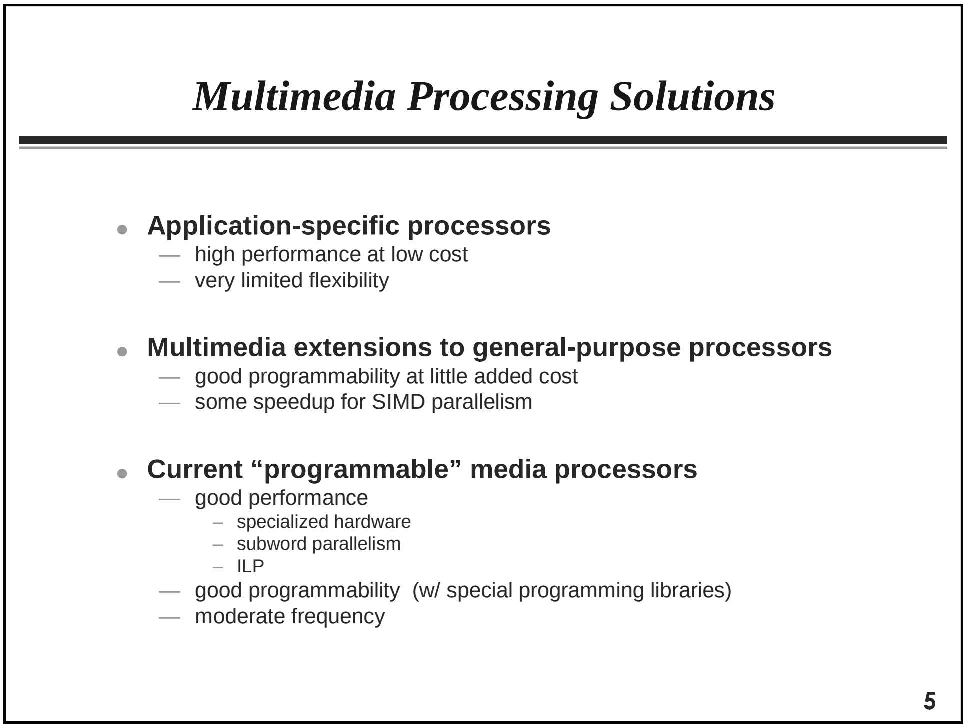
***Multimedia is primarily a communication media***

# *Future of Multimedia*

**Multimedia industry evolves with processor performance.**



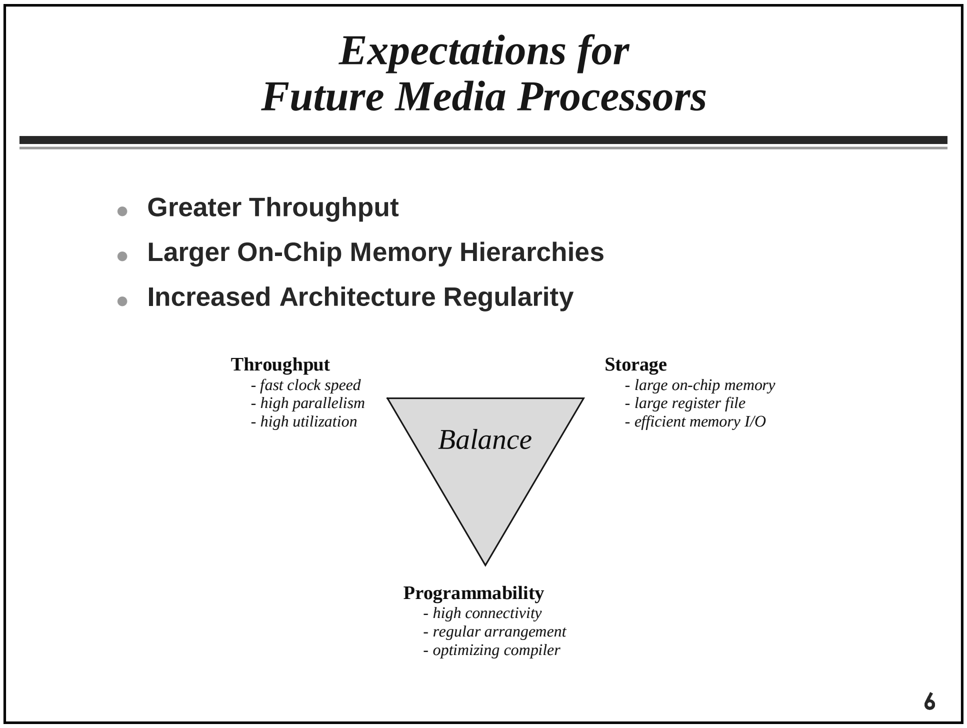
**Multimedia is moving towards advanced representations**



## Multimedia Processing Solutions

- **Application-specific processors**
  - high performance at low cost
  - very limited flexibility
- **Multimedia extensions to general-purpose processors**
  - good programmability at little added cost
  - some speedup for SIMD parallelism
- **Current “programmable” media processors**
  - good performance
    - specialized hardware
    - subword parallelism
    - ILP
  - good programmability (w/ special programming libraries)
  - moderate frequency

5



## Expectations for Future Media Processors

- **Greater Throughput**
- **Larger On-Chip Memory Hierarchies**
- **Increased Architecture Regularity**

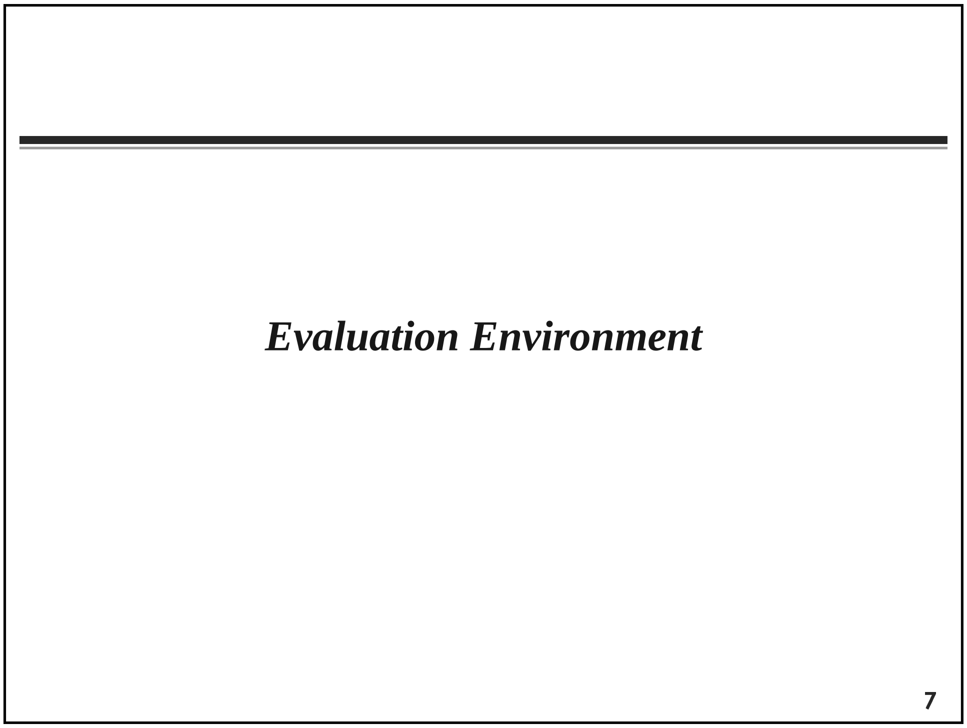
**Throughput**
*- fast clock speed*
*- high parallelism*
*- high utilization*

**Storage**
*- large on-chip memory*
*- large register file*
*- efficient memory I/O*

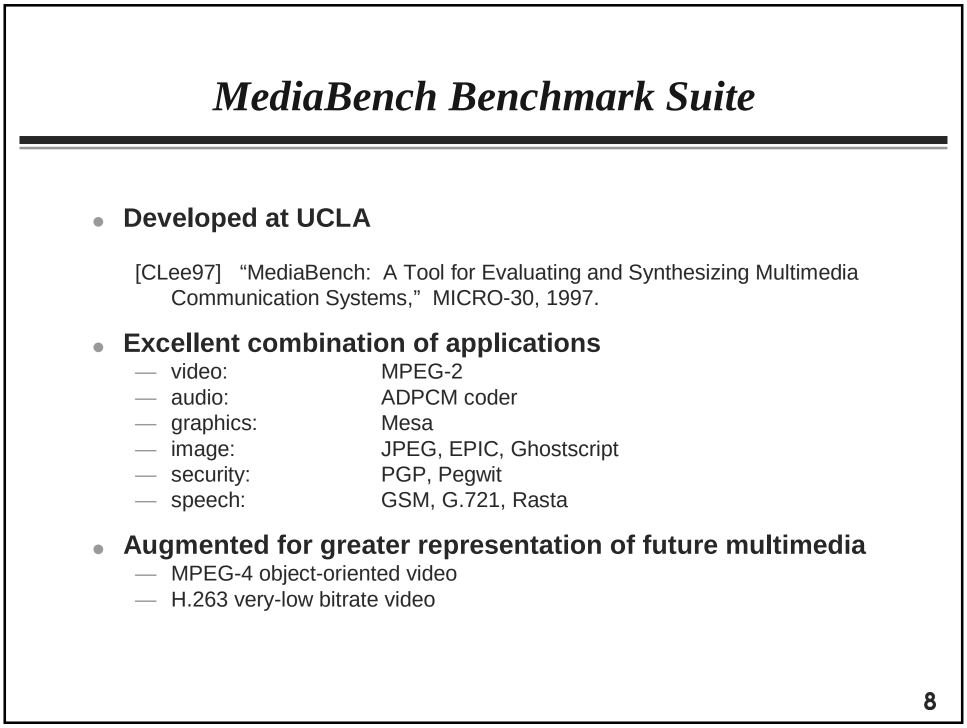
*Balance*

**Programmability**
*- high connectivity*
*- regular arrangement*
*- optimizing compiler*

6



# *Evaluation Environment*



# *MediaBench Benchmark Suite*

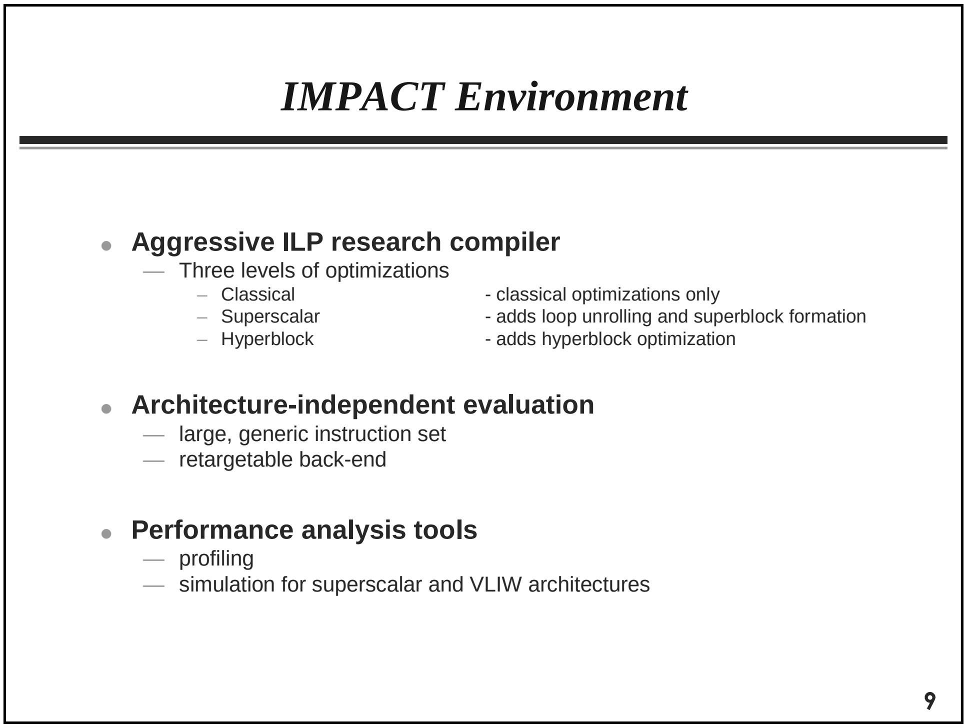
- **Developed at UCLA**

  [CLee97] "MediaBench: A Tool for Evaluating and Synthesizing Multimedia Communication Systems," MICRO-30, 1997.

- **Excellent combination of applications**

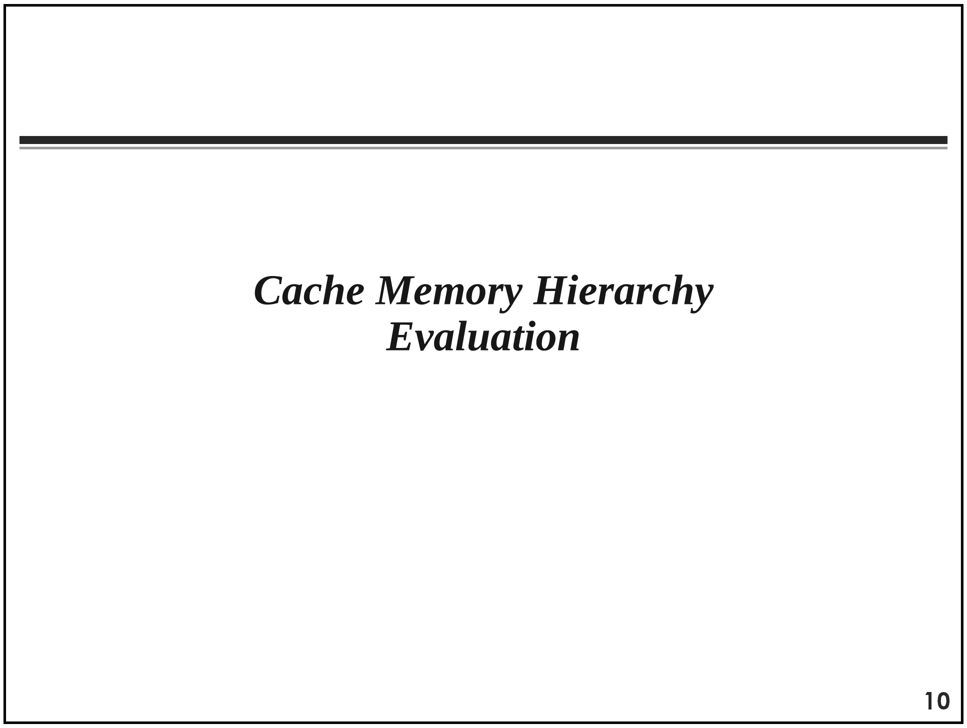
| | |
|---|---|
| — video: | MPEG-2 |
| — audio: | ADPCM coder |
| — graphics: | Mesa |
| — image: | JPEG, EPIC, Ghostscript |
| — security: | PGP, Pegwit |
| — speech: | GSM, G.721, Rasta |

- **Augmented for greater representation of future multimedia**
  - MPEG-4 object-oriented video
  - H.263 very-low bitrate video



# IMPACT Environment

- **Aggressive ILP research compiler**
  - Three levels of optimizations
    - Classical - classical optimizations only
    - Superscalar - adds loop unrolling and superblock formation
    - Hyperblock - adds hyperblock optimization
- **Architecture-independent evaluation**
  - large, generic instruction set
  - retargetable back-end
- **Performance analysis tools**
  - profiling
  - simulation for superscalar and VLIW architectures



# Cache Memory Hierarchy Evaluation

# *Architecture Evaluation*

- **Variety of Memory Hierarchy Options**
  - — Cache vs. Memory
  - — Automatic Prefetching vs. Software Prefetching
  - — Streaming Memory vs. DMA Prefetching
  - — Organization of hierarchy?
- **Related Work**

[CLee97] "MediaBench: A Tool for Evaluating and Synthesizing Multimedia Communications Systems," MICRO-30, 1997.

[ZWu97] "Study of Cache Systems in Video Signal Processors," SiPS-98, 1998.

[DZucker97] "Architecture and Arithmetic for Multimedia Enhanced Processors," Ph.D. Thesis, Dept. of Electrical Engineering, Stanford Univ., 1997.

[DZucker95] "A comparison of hardware prefetching techniques for multimedia benchmarks," Technical Report CSL-TR-95-683, Stanford University, 1995.

[YChen98] "Multimedia Signal Processors: An Architectural Platform with Algorithmic Compilation," Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, vol. 20, 1998.

[FCatthoor98] "Custom Memory Management Methodology: Exploration of Memory Organisation for Embedded Multimedia System Design," Kluwer Academic Publishers, 1998.

# *Base Architecture Model*

- **Architecture model**
  - — 8-issue VLIW media processor
  - — operation latencies targeting 500 MHz to 1 GHz processor frequency
  - — 64 integer and floating-point registers
  - — pipeline: 1 fetch, 2 decode, 1 write back, variable execute stages
- **L1 Cache**
  - — 16 KB direct-mapped L1 instruction cache w/ 256 byte lines
  - — 32 KB direct-mapped L1 data cache w/ 64 byte lines
    - – non-blocking w/ 8-entry miss buffer
    - – no-write allocate w/ 8-entry write buffer
  - — currently no streaming memory support
- **On-Chip L2 Cache**
  - — 256 KB 4-way set associate w/ 64 byte lines
    - – non-blocking w/ 8-entry miss buffer
    - – write allocate w/ 8-entry write buffer
- **External Memory**
  - — 4:1 Processor to external bus frequency ratio

| *Cache* | *Miss Latency* |
|---|---|
| L1 I-Cache | 20 |
| L1 D-Cache | 15 |
| L2 Cache | 50 |

# L1 Cache

- **Results from earlier workload evaluation:**

| | |
|---|---|
| — i-cache working set size: | < 8 KB |
| — i-cache spatial locality: | 84.8% locality within 256 bytes |
| — d-cache working set size: | < 32 KB |
| — d-cache spatial locality: | 60.8% locality within 128 bytes |

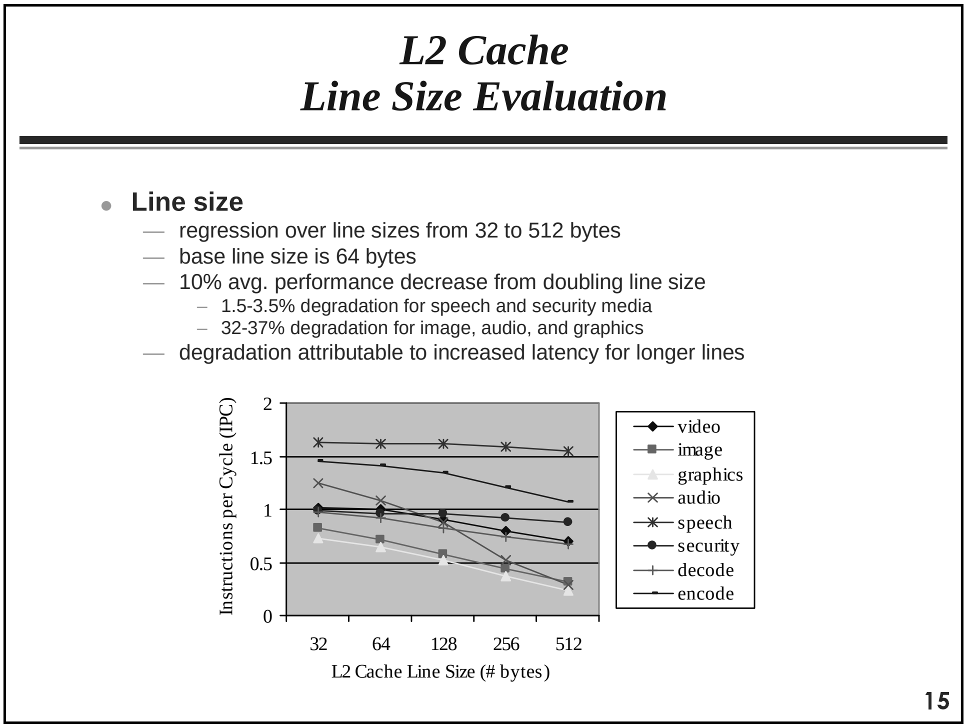
[JFritts99] “Understanding multimedia application characteristics for designing programmable media processors,” SPIE Photonics West, Media Processors ’99, 1999.

- **No streaming memory support**
  - to be evaluated in future work

# L2 Cache Evaluation

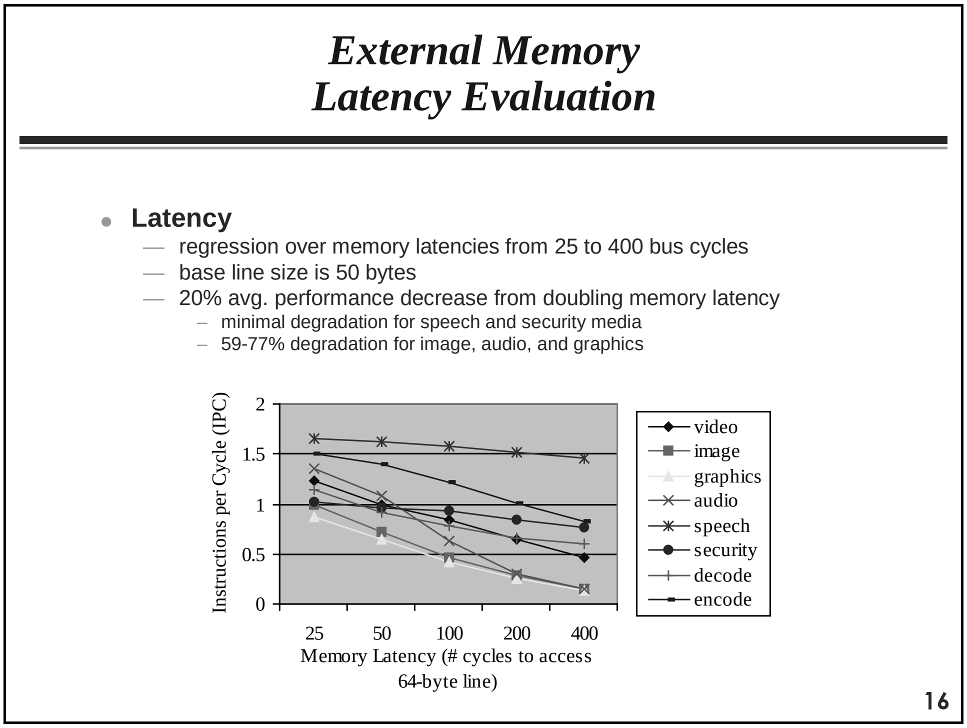
- **Cache size**
  - regression over cache sizes from 128 KB to 1 MB
  - base cache size is 256 KB
  - 0.5% avg. performance increase from doubling cache size
    - ~7% difference for *unepic* and *mpeg4dec*

- **Access latency**
  - regression over access latencies of 8, 15, 30, 60 cycles
  - base access latency is 15 cycles
  - 5.6% avg. performance decrease from doubling access latency
    - ~35% difference for pegwitdec and pegwitenc (large working set size)
    - ~16% difference for mpeg2dec
  - attributable to increasing memory access latency



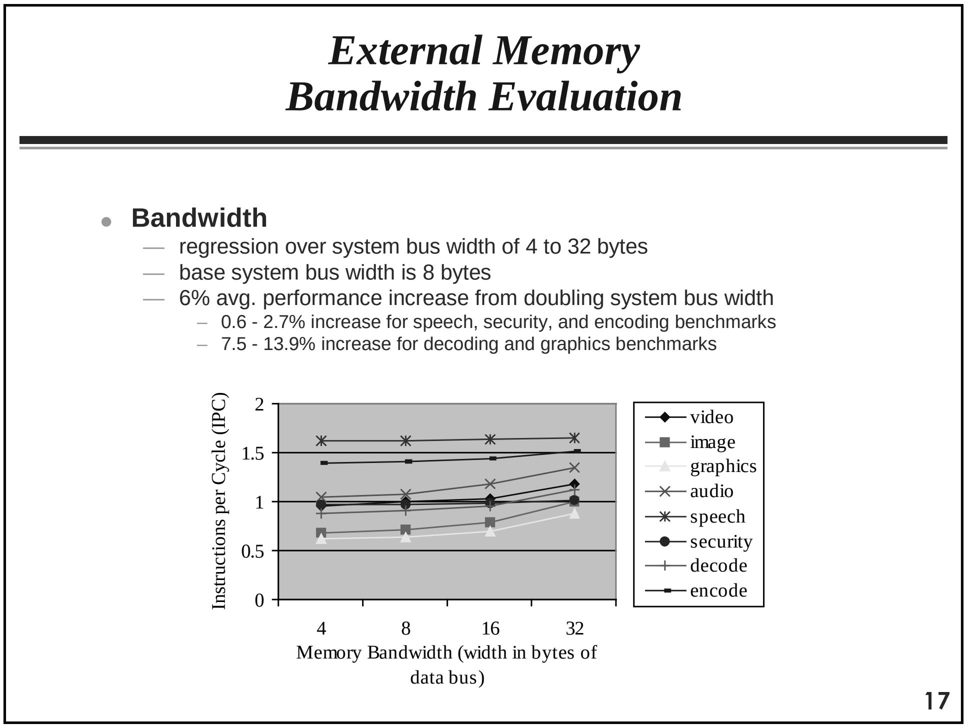
# L2 Cache Line Size Evaluation

- **Line size**
  - regression over line sizes from 32 to 512 bytes
  - base line size is 64 bytes
  - 10% avg. performance decrease from doubling line size
    - 1.5-3.5% degradation for speech and security media
    - 32-37% degradation for image, audio, and graphics
  - degradation attributable to increased latency for longer lines



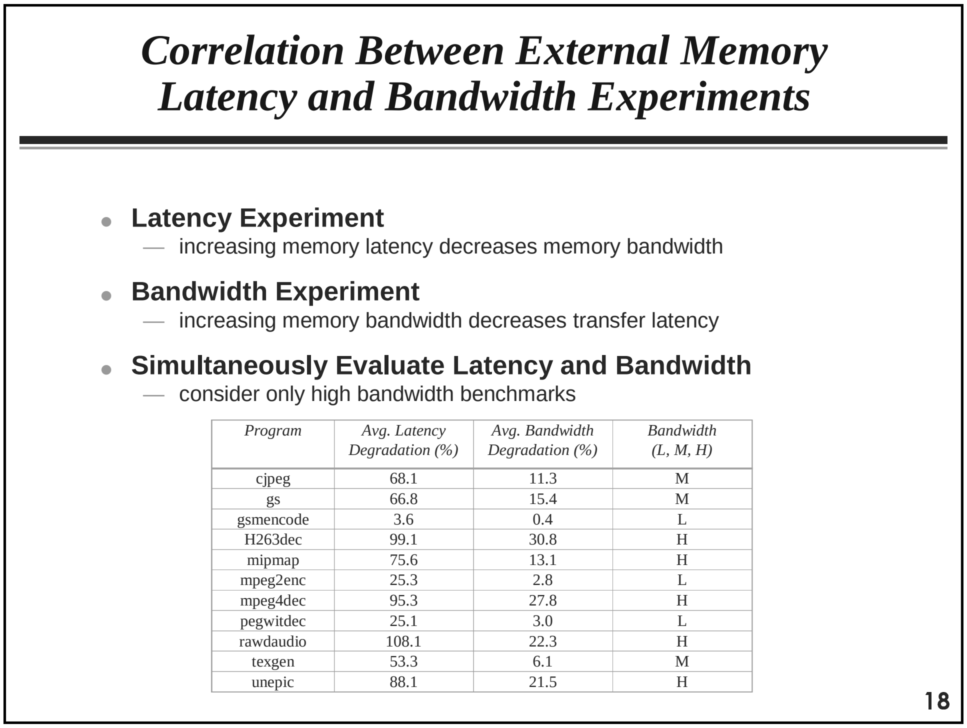
# External Memory Latency Evaluation

- **Latency**
  - regression over memory latencies from 25 to 400 bus cycles
  - base line size is 50 bytes
  - 20% avg. performance decrease from doubling memory latency
    - minimal degradation for speech and security media
    - 59-77% degradation for image, audio, and graphics



# External Memory Bandwidth Evaluation

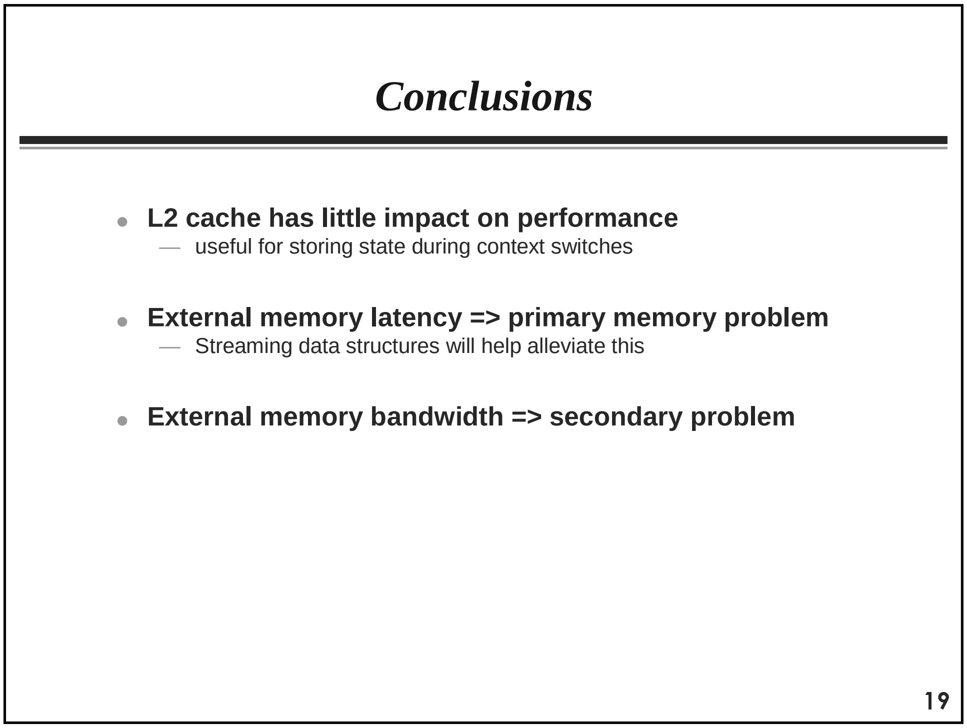
- **Bandwidth**
  - regression over system bus width of 4 to 32 bytes
  - base system bus width is 8 bytes
  - 6% avg. performance increase from doubling system bus width
    - 0.6 - 2.7% increase for speech, security, and encoding benchmarks
    - 7.5 - 13.9% increase for decoding and graphics benchmarks



# Correlation Between External Memory Latency and Bandwidth Experiments

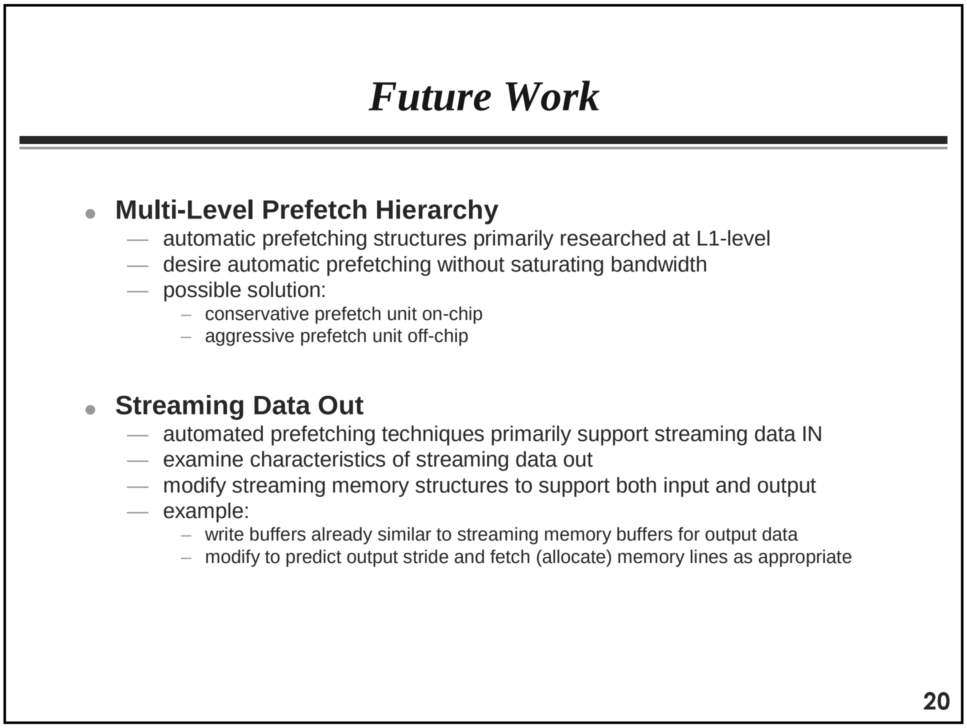
- **Latency Experiment**
  - increasing memory latency decreases memory bandwidth
- **Bandwidth Experiment**
  - increasing memory bandwidth decreases transfer latency
- **Simultaneously Evaluate Latency and Bandwidth**
  - consider only high bandwidth benchmarks

| *Program* | *Avg. Latency Degradation (%)* | *Avg. Bandwidth Degradation (%)* | *Bandwidth (L, M, H)* |
|---|---|---|---|
| cjpeg | 68.1 | 11.3 | M |
| gs | 66.8 | 15.4 | M |
| gsmencode | 3.6 | 0.4 | L |
| H263dec | 99.1 | 30.8 | H |
| mipmap | 75.6 | 13.1 | H |
| mpeg2enc | 25.3 | 2.8 | L |
| mpeg4dec | 95.3 | 27.8 | H |
| pegwitdec | 25.1 | 3.0 | L |
| rawdaudio | 108.1 | 22.3 | H |
| texgen | 53.3 | 6.1 | M |
| unepic | 88.1 | 21.5 | H |



## Conclusions

- **L2 cache has little impact on performance**
  - useful for storing state during context switches
- **External memory latency => primary memory problem**
  - Streaming data structures will help alleviate this
- **External memory bandwidth => secondary problem**



## Future Work

- **Multi-Level Prefetch Hierarchy**
  - automatic prefetching structures primarily researched at L1-level
  - desire automatic prefetching without saturating bandwidth
  - possible solution:
    - conservative prefetch unit on-chip
    - aggressive prefetch unit off-chip
- **Streaming Data Out**
  - automated prefetching techniques primarily support streaming data IN
  - examine characteristics of streaming data out
  - modify streaming memory structures to support both input and output
  - example:
    - write buffers already similar to streaming memory buffers for output data
    - modify to predict output stride and fetch (allocate) memory lines as appropriate