Instruction Fetch Characteristics of Media Processing

Jason Fritts
Assistant Professor
Department of Computer Science
Co-Author: Wayne Wolf

Overview

- Architectures for Media Processing
- Workload-Dependent Characteristics
  - branch characteristics
  - instruction memory characteristics
  - loop characteristics
- Architecture-Dependent Characteristics
  - fetch architectures
  - dynamic branch prediction
  - pre-execution pipeline length
- Conclusions
Programmable Architectures for Media Processing

- General-purpose processors (GPPs) w/ multimedia extensions
  - good programmability at little added cost
  - some speedup with subword parallelism
  - optimized for general-purpose processing

- High-performance DSPs (aka Media Processors)
  - good performance
    - specialized hardware
    - subword parallelism
    - ILP
  - good programmability (w/ special programming libraries)
  - moderate frequency

- GPPs and DSPs continually evolving towards media processing
  - computing workloads becoming increasingly dominated by media applications

Media Processing vs. General-Purpose Processing

- General-purpose processing:
  - streaming instructions
  - static data

- Media processing:
  - streaming data
  - static instructions
Workload-Dependent Characteristics

Evaluation Environment

- **MediaBench benchmarks**
  - Developed at UCLA
  - Augmented w/ MPEG-4 and H.263
  - Variety of media applications, including video, audio, graphics, image, security, and speech

- **IMPACT compilation & simulation environment**
  - Aggressive ILP research compiler
  - Cycle-accurate simulation
  - Large, generic RISC instruction set
  - Three levels of optimizations
    - Classical
    - Superscalar
    - Hyperblock
      - classical optimizations only
      - adds loop unrolling and superblock formation (speculation)
      - adds hyperblock optimization (predication)
Static Branch Prediction

- 5.5 dynamic operations per basic block
  - similar to general-purpose applications
- 89.5% prediction rate on training input
- 85.9% prediction rate on evaluation input

Instruction Working Set Size

- Cache Regression
  - cache sizes: 1K to 4MB
  - assumed line size of 64 bytes
  - 8 KB working set size
**Instruction Spatial Locality**

- **Cache Regression**
  - line sizes: 8 to 1024 bytes
  - assumed cache size of 64 KB
  - 84.8% spatial locality (up to 256 bytes)

\[
\text{spatial locality} = \frac{(A - B)}{A}
\]

**Looping Characteristics**

- **Highly Loop Centric**
  - nearly 95% of execution time spent within two innermost loop levels
Iterations per Loop

- Large Number of Iterations
  - average of 10 iterations per loop
  - significant processing regularity

Architecture-Dependent Characteristics
**Base Architecture Model**

- **Architecture model**
  - 8-issue media processor
  - operation latencies targeting 800 MHz to 1.2 GHz
  - 64 integer and floating-point registers

- **L1 Cache**
  - 16 KB direct-mapped L1 instruction cache w/ 256 byte lines
  - 32 KB direct-mapped L1 data cache w/ 64 byte lines

- **On-Chip L2 Cache**
  - 256 KB 4-way set associate w/ 64 byte lines

- **External Memory**
  - 6:1 Processor to bus frequency ratio

**Aggressive vs. Conservative Fetch**

- **Aggressive Fetch**
  - decoupled fetch-execute pipeline
  - 12-entry instruction buffer

- **Conservative Fetch**
  - lock-step pipeline
**History-Based Dynamic Branch Prediction**

- PAs(k,p)
  - k - # of history bits
  - p - # of history tables

- Good performance with small predictors
  - 128-256 entries
  - 4-8 history bits
  - 1 history table

**Performance vs. Size for Dynamic Branch Predictors**

- Comparison of four 1 Kb dynamic branch predictors
  - k - # of history bits
  - p - # of history tables
  
  \[ \text{size}_{\text{PAs}(k, p)} = bk + 2^{k+1} p \]

- Minor variation between 1 Kb predictors
  - large predictors only provide 2-3% better performance
Conclusions

- Instruction fetch characteristics of Media Processing are **Idealistic**

- **Workload-Dependent Characteristics**
  - 5.5 operations per basic block
  - 85-90% static branch prediction
  - 8 KB working set size
  - 84.8% spatial locality (up to 256 bytes)
  - 95% of execution in 2 innermost loops
  - 10+ iterations per loop

- **Architecture-Dependent Characteristics**
  - Aggressive fetch provides little benefit
  - Small dynamic branch predictors sufficient
  - 1K predictors reduce miss rate by 2x
  - 2% performance reduction for each extra pre-execution pipeline stage