Welcome

To Advance through Presentation
Use Page Up and Page Down Keys

Data Prefetching

Kalpesh Gala and Jim Robertson

MOTOROLA

Outline

Motivation/Overview
Overview of Prefetch Instructions
Instruction Format and Arguments
Memory Access Example
Stopping Prefetching
Other Considerations

Digital DNA from Motorola

Motivation

- By prefetching the data BEFORE it is needed:
  - The data will be in the local (L1) cache
  - The page table entry will be in the TLB when the memory access
  - load/store instructions accessing memory execute quickly (no bus access)

Digital DNA from Motorola

Overview

- G4 supports software-directed prefetch
- Uses idle bus cycles to load data into cache before it is needed
- When the load/store instructions are actually executed, data is in cache
- “Data Stream Touch” instructions control software-directed prefetch

Digital DNA from Motorola

Prefetch Instructions

- Four instructions initiate software prefetching
  - Dst—Data Stream Touch
  - Dstt—Data Stream Touch Transient (used for last access)
  - Dstst—Data Stream Touch-for-Store (should not be used)
  - Dststt—Data Stream Touch-for-Store Transient (should not be used)

Digital DNA from Motorola

Prefetch Instructions (Cont.)

- Transient instructions indicate the data does not have a long lifetime
  - Transient data will not be castout to the L2 for future use
  - Modified data is written directly to the memory, unmodified data is discarded
- Touch-for-Store instructions mark data as exclusive
  - Inefficient because they use different internal resources

Digital DNA from Motorola

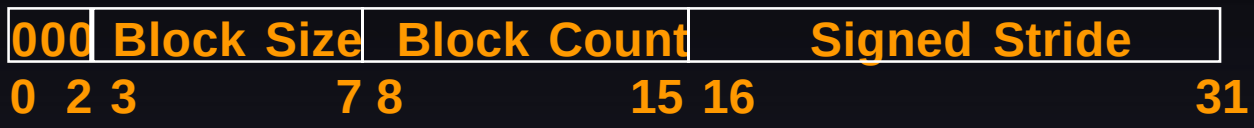
dstX Instruction Format

- Usage: **dstX rA, rB, STRM**

  dstX is one of **dst, dstt, dstst**, or **dststt**

  rA is the Address of the first block to prefetch

  rB encodes the Block Size, Block Count and Stride



STRM is the stream to use; 0 - 3

Digital DNA from Motorola

dstX Arguments

- Stream ID—which stream engine to use
  - There are four stream engines, 0-3
- Address—Initial address of the sequence
- Block Size—The number of quad words (16 bytes) in each block
  - Block Size is between 1 and 32
  - Should be at least 2 to fill a single 32-byte G4 cache line

Digital DNA from Motorola

dstX Arguments (Cont.)

- Count—Number of blocks in the sequence
  - Count is between 1 and 256
- Stride—Number of bytes between blocks
  - Valid stride values:
  - -32768 < stride < 0 or 0 < stride < 32768
  - To avoid redundant loads, stride must be used correctly

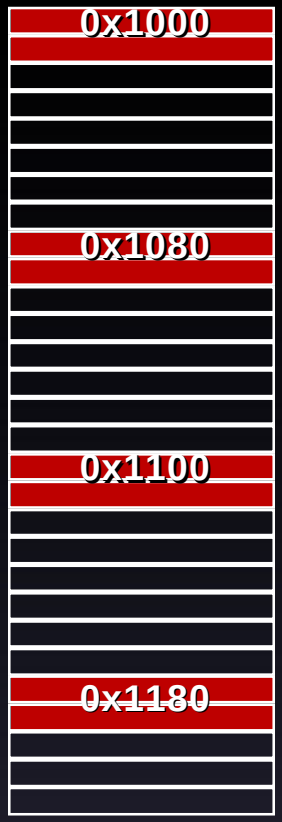
Digital DNA from Motorola

dst Memory Access

Address = 0x1000
(First vector is at 0x1000)

Stride = 128
(16 bytes/vector * 8 vectors)

Each ▭ represents 16 bytes of memory



Block Size = 2 vectors
(2 vectors = 32 bytes)

rA = 0x00001000
rB = ((2<<24)
| (4<<16) | 128)

dst rA,rB,0

Count = 4
(4 total blocks loaded)

Digital DNA from Motorola

dst Termination

- Two different instructions to allow the user to stop dst streams
  - Dss (Data Stream Stop)—stops a single stream
  - Dssall (Data Stream Stop All)—stops ALL active streams

Digital DNA from Motorola

dst Termination (Cont.)

- Prefetching may also terminate for any of the following reasons:
  - Successfully reached end of stream
  - Another dst instruction to the same stream is executed
  - Currnet line-fetch causes a table walk which results in a page table miss
  - Current line-fetch is translated as cache inhibited

Digital DNA from Motorola

dst Termination (Cont.)

- There is no way to identify if a stream has stopped fetching
  - Should re-issue dst instructions periodically “just in case”

Digital DNA from Motorola

Other Considerations

- Prefetching is context aware
  - Prefetching is paused if the processor switches from user to supervisor mode
  - Prefetching resumes when switching back to user mode
  - This prevents prefetching from happening during exceptions

Digital DNA from Motorola

Other Considerations (Cont.)

- No arbitrary address boundaries which stop the progress of a stream
- dstX instructions handle address alignment issues automatically
- All four prefetch engines can be active at the same time

Digital DNA from Motorola

Conclusion

- Software prefetching can be a useful tool for increasing performance
- dstX instructions are directly supported by the AltiVec programming model
- Use transient versions of dstX for the last access of the data
- Avoid using the touch-for-store variants of dstX

Digital DNA from Motorola

Sim_G4 in Depth & Details

Kalpesh Gala and Jim Robertson

MOTOROLA

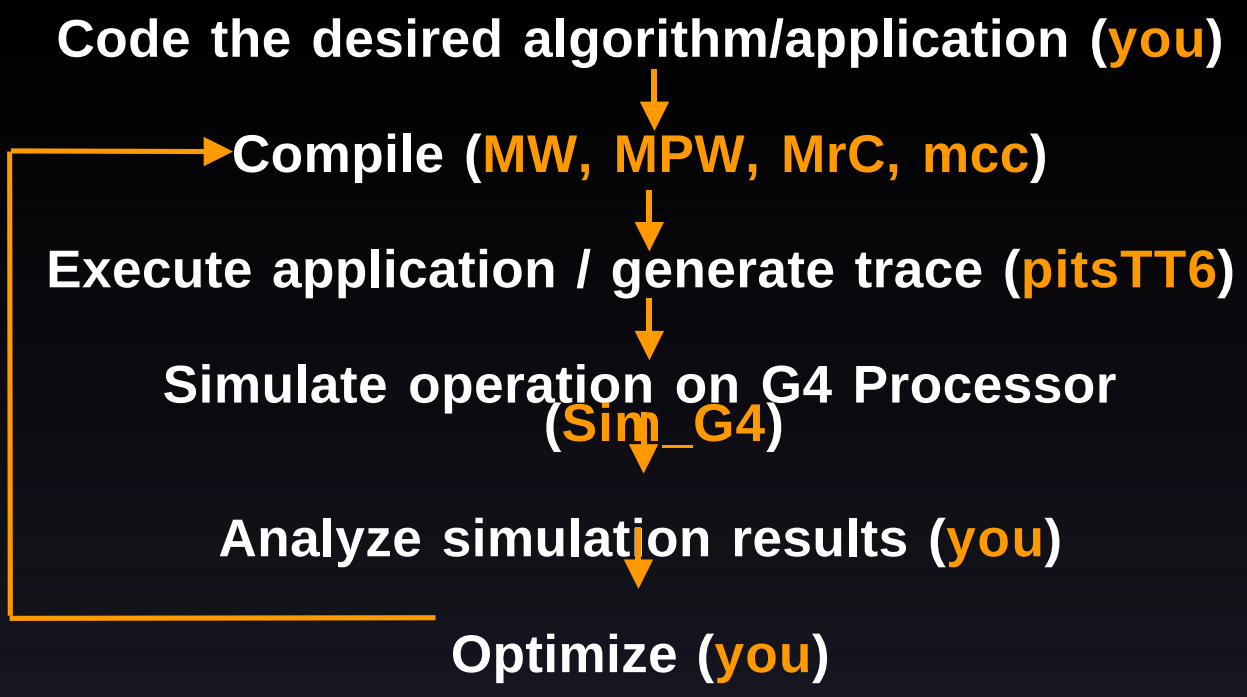
Presentation Summary

- Introduction
- Methodology & Tool Flow
- Configuring Sim_G4
- 64-bit Multiply Example
- Conclusion

Introduction

- Sim_G4 is a trace driven, cycle accurate timing simulator
- Sim_G4 was developed by Motorola's PowerPC G4 Design Team for architectural decisions
- Limitations
    - No notion of data dependencies
    - Most applications are 95-100% accurate

Methodology & Tool Flow



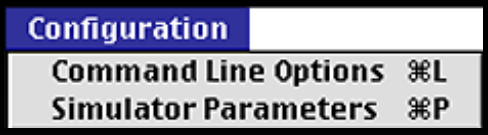
High Performance Code!!! = $$$

Digital DNA from Motorola

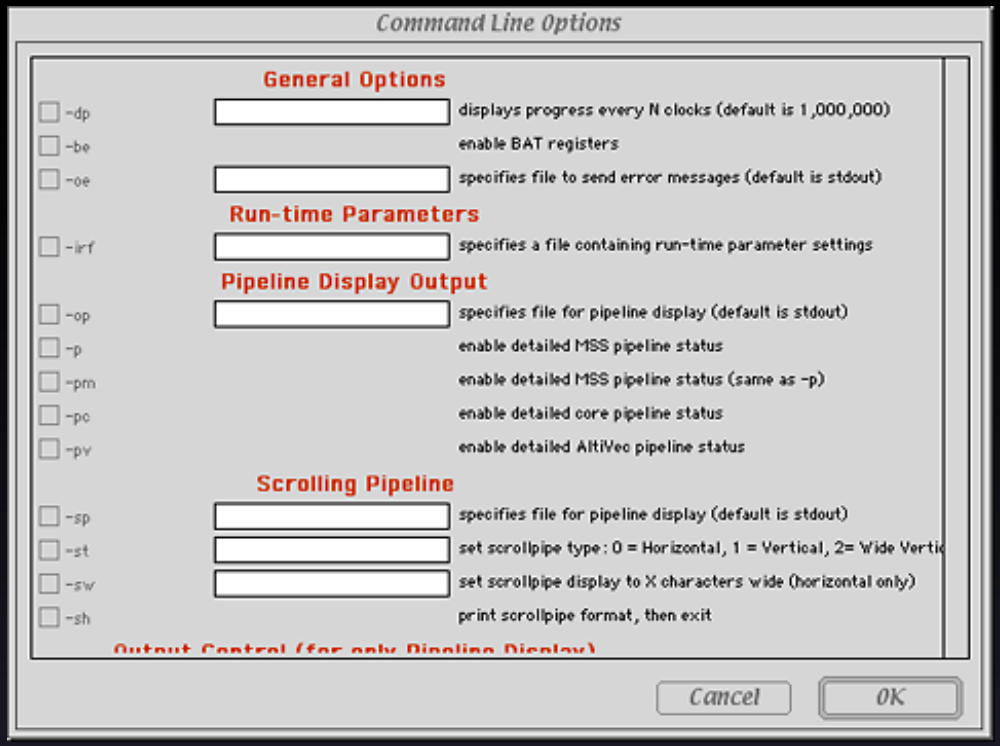
Sim_G4 Configuration

- Command Line Options
- Simulation Parameters
- Current Sim_G4 Defaults
  - Processor is a 300 MHz G4
  - System bus running at 75 MHz
  - System is a PowerMac G3

Output Configuration

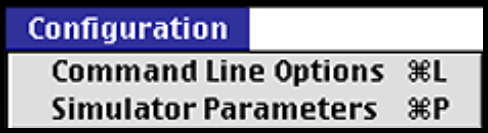


A variety of control parameters can be set through the Command Line Options popup window

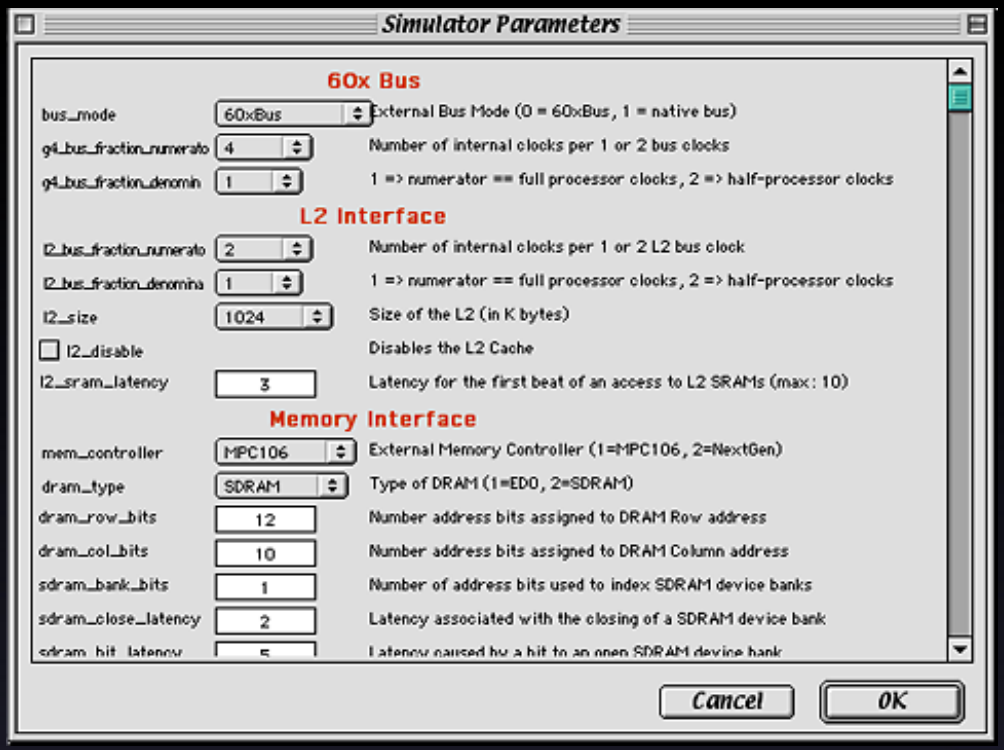


Digital DNA from Motorola

System Configuration



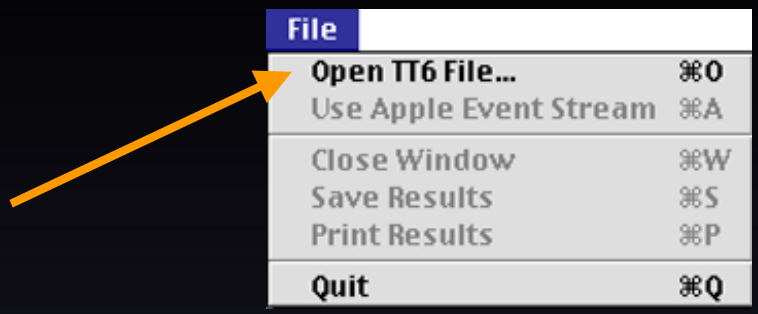
A variety of system configuration variables can be set through the Simulation Parameters popup window



Digital DNA from Motorola

Processing the Trace

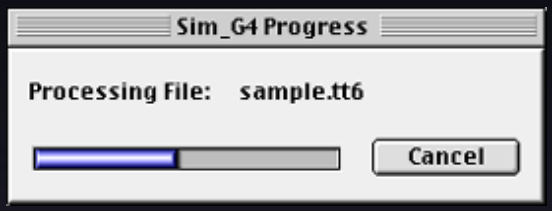
- Assumption: a TT6 trace file has been generated as input to Sim_G4



- After invoking the application and selecting the desired configuration, an input TT6 file must be designated

Sim_G4 the Last Step…

After configuring and supplying the desired trace Sim_G4 provides profiling information



sample.tt6.out

```
Clocks: 1510  Retired: 77  Folded: 5  IPC=  0.0543

Instruction Flow Stats:
  Fetched: 82  Dispatched: 79  Retired: 77  Branches Folded: 5


Dispatch Stalls (in evaluation order):
  Drain:          0.00% (0)
  IB_empty:      90.53% (1367)
  CB_full:        0.00% (0)
  GPR_rename:     0.00% (0)
  FPR_rename:     0.00% (0)
  VR_rename:      0.00% (0)
  Unit_busy:      3.84% (58)
    FXU1:                    0.07% (1)
    FXU2:                    0.00% (0)
    FPU:                     0.00% (0)
    VAUS:                    0.00% (0)
    VAUC:                    0.00% (0)
    VAUF:                    0.00% (0)
    VPU:                     3.77% (57)
    SYS:                     0.00% (0)
    LSU:                     0.00% (0)
  Disp_serial:    0.00% (0)
  Tail_serial:    0.00% (0)

Execution Unit Stats/Stalls:
  FXU1: idle:  98.81%  dispatch:        17  depend_stall:         0  ser_stall:         0
  FXU2: idle:  99.87%  dispatch:         1  depend_stall:         0  ser_stall:         0
  FPU:  idle:  99.93%  dispatch:         0  depend_stall:         0  ser_stall:         0
cbl_stall:         0
  VAU:  double dispatch attempts:         1
  VAUS: idle:  99.87%  dispatch:         1  depend_stall:         0  ser_stall:         0
  VAUC: idle:  99.93%  dispatch:         0  depend_stall:         0  ser_stall:         0
```

Digital DNA from Motorola

Demo

Conclusion

- Sim_G4 models the G4 Architecture NOT just the AltiVec engine
- Sim_G4 can be useful in fine-tuning pieces of code
- Analysis of memory intensive applications may not reflect system
- Sim_G4 not always 100% accurate

Digital DNA from Motorola

Call to Action

- Download the SDK:
  developer.apple.com/hardware/altivec
- www.mot.com/AltiVec
- Identify data parallelism in your programs
- Vectorize computation intensive code
- Use Sim G4 to tune performance
- Sign up for the next AltiVec kitchen

Digital DNA from Motorola

Other AltiVec Sessions

AltiVec Workshops Hands-on introduction to AltiVec (pre-registration only)	Room L **Fri.**
AltiVec Feedback Session Open Q&A session	Hall J2 **Fri., 10:15am**

Digital DNA from Motorola

Q&A

Think different.™

Digital DNA from Motorola

Welcome

To Advance through Presentation
Use Page Up and Page Down Keys