Streaming Virtual Machine to Stanford Streaming Architecture (part 1)

Abhishek Das
Mattan Erez
3/9/2004
Merrimac ISA

- **Scalar unit**
  - Standard scalar ISA
  - Interface instructions to stream unit
    - MIPS COP2 instructions

- **Stream unit**
  - Kernels
    - Interface through kernelC
  - Streams
    - Memory instructions
    - Kernel control
    - Scalar interface
Streams

- Contiguous elements in the SRF
  - Aligned to 64 word SRF lines
  - No real allocation – just access

- Arbitrary number of live streams
  - Scalar code / compiler maintained

- Limited number of active streams
  - Maximum 1 kernel and 2 memory instructions
    - $10 + 3 + 3 = 16$ active streams
  - Accessed through Stream Descriptor Registers
    - Hardware registers describing location and length
    - Allocated by the software system
Memory

- Contiguous blocks in main memory
  - Allocated using scalar code (malloc)
- Scalar unit access
  - Arbitrary
- Stream unit access
  - Through stream memory instructions
  - Memory described by Memory Access Registers
    - Hardware registers describing location, stride and record size
    - Allocated by the software system
Kernels and scalar interface

• Kernel code
  – Transferred from memory to SRF then to µstore
  – µstore locations “allocated” by compiler
  – PC set by scalar code

• Scalar interface
  – Scalar registers in the stream unit
    • Allocate, read, and write
  – Control registers
    • PC, status, ...
  – Barrier instructions
  – Cache control
Stream Instructions

• Transfer data between memory and SRF
  – Strided
  – Indexed
  – withOp

• Kernels
  – Load μcode
  – Run kernel

• Miscellaneous
  – Set SDR, MAR, scalar register, control register
  – Read scalar and status registers
  – Barrier
  – Cache control
Stream Instruction Execution

• Dynamically executed by the stream-controller
  – Out-of-order instruction queue
• Software places instructions in specific queue slots
• Software explicitly communicates dependencies
  – Based on slot numbers
  – Start or completion dependencies
• Alternative instruction queues under evaluation
Low Level Compilation

• Allocation
  – SDRs
  – MARs
  – scalar register in stream unit
  – instruction-queue slots
  – Code in μstore
  – Memory/SRF allocated by high-level compiler

• Scalar/stream interface
  – Translate SVM instructions to Merrimac
  – Insert synchronization and communication instructions
SVM Stream Instructions

- Memory initialization
- Memory transfer
- Kernel control
• Handled by library calls
  – streamInitRAM
  – streamInitWithDataRAM
  – blockInit
• Not handled
  – streamInitFifo
Memory transfer instructions

• Simple translation involving SDR and MAR allocation
  – Movelnit
  – StridedScatter/StridedGather
  – IndexedScatter/IndexedGather

• Not handled
  – Pop/Push/Peek/…
  – blockRead/Write
Kernel control

- Used only to communicate information to LLC
  - kernelInit – allocate µcode store
- Fairly straightforward compilation:
  - kernelAddDependence
  - kernelReady
  - kernelRun
  - kernelWait
  - kernelGetStatus
- Not handled
  - kernelRunMultiplexed
  - kernelPause
  - kernelWaitMultiple