Distributed Microarchitectural Protocols in the TRIPS Prototype Processor

Sankaralingam et al.

Presented by Cynthia Sturton

CS 258

3/3/08

Tera-op, Reliable, Intelligently adaptive Processing System (TRIPS)

- Trillions of operations on a single chip by 2012!
- Distributed Microarchitecture
  - Heterogenous Tiles - Uniprocessor
  - Distributed Control
  - Dynamic Execution
- ASIC Prototype Chip
  - 170M transistors, 130nm
  - 2 16-wide issue processor cores
  - 1MB distributed Non Uniform Cache Access (NUCA)

Why Tiled and Distributed?

- Issue width of superscalar cores constrained
  - On-chip wire delay
  - Power constraints
  - Growing complexity
- Use tiles to simplify design
  - Larger processors
  - Multi-cycle communication delay across the processor
- Use a distributed control system

TRIPS Processor Core

- Explicit Data Graph Execution (EDGE) ISA
  - Compiler-generated TRIPS blocks
- 5 types of tiles
- 7 micronets
  - 1 each data and instruction
  - 5 control
- Few global signals
  - Clock
  - Reset tree
  - Interrupt

EDGE Instruction Set Architecture

- TRIPS block
  - Compiler-generated dataflow graph
- Direct intra-block communication
  - Instructions can send results directly to dependent consumers
- Block-atomic execution
  - 128 instructions per TRIPS block
  - Fetch, execute, and commit

TRIPS Block

- Blocks of instructions built by compiler
  - One 128-byte header chunk
  - One to four 128-byte body chunks
  - All possible paths emit the same number of outputs (stores, register writes, one branch)
- Header chunk
  - Maximum 32 register reads, 32 register writes
- Body chunk
  - 32 instructions
  - Maximum 32 loads and stores per *block*

Processor Core Tiles

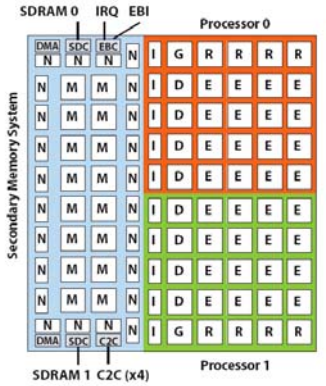
- Global Control Tile (1)
- Execution Tile (16)
- Register Tile (4)
  - 128 registers per tile
  - 2 read ports, 1 write port
- Data Tile (4)
  - Each has one 2-way 8KB L1 D-cache
- Instruction Tile (5)
  - Each has one 2-way 16KB bank of the L1 I-cache
- Secondary Memory System
  - 1MB, Non Uniform Cache Access (NUCA), 16 tiles, Miss Status Holding Register (MSHR)
  - Configurable as L2 cache or scratch-pad memory using On Chip Network (OCN) commands
  - Private port between memory and each IT/DT pair

Processor Core Micronetworks

- Operand Network
  - Connects all but the Instruction Tiles
- Global Dispatch Network
  - Instruction dispatch
- Global Control Network
  - Committing and flushing blocks
- Global Status Network
  - Information about block completion
- Global Refill Network
  - I-cache miss refills
- Data Status Network
  - Store completion information
- External Store Network
  - Store completion to L2 cache or memory information

TRIPS Block Diagram

- Composable at design time
- 16-wide out-of-order issue
- 64KB L1 I-cache
- 32KB L1 D-cache
- 4 SMT Threads
- 8 TRIPS blocks in flight



Distributed Protocols – Block Fetch

- GT sends instruction indices to ITs via Global Dispatch Network (GDN)
- Each IT takes 8 cycles to send 32 instructions to its row of ETs and RTs (via GDN)
  - 128 instructions total for the block
- Instructions enter read/write queues at RTs and reservation stations at ETs
- 16 instructions per cycle in steady state, 1 instruction per ET per cycle.

Block Fetch – I-cache miss

- GT maintains tags and status bits for cache lines
- On I-cache miss, GT transmits refill block's address to every IT (via Global Refill Network)
- Each IT independently processes refill of its 2 64-byte cache chunks
- ITs signal refill completion to GT (via GSN)
- Once all refill signals complete, GT may issue dispatch for that block.

Distributed Protocols - Execution

- RT reads registers as given in read instruction
- RT forwards result to consumer ETs via OPN
- ET selects and executes enabled instructions
- ET forwards results (via OPN) to other ETs or to DTs

```
R[0]        read R4    N[1,L] N[2,L]
N[0]        movi #0    N[1,R]
N[1]        teq        N[2,P] N[3,P]
N[2]  p_f   muli #4    N[32,L]
N[3]  p_t   null       N[34,L] N[34,R]
N[32]       lw #8      N[33,L]   LSID=0
N[33]       mov        N[34,L] N[34,R]
N[34]       sw #0                LSID=1
N[35]       callo      $func1
```

Distributed Protocols – Block/Pipeline Flush

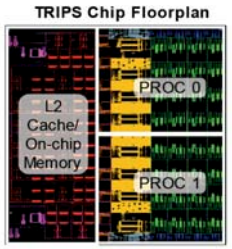
- GT initiates flush wave on GCN on branch misprediction
- All ETs, DTs, and RTs are told which block(s) to flush
- Wave propagates at one hop per cycle
- GT may issue new dispatch command immediately – new command will never overtake flush command.

Distributed Protocols – Block Commit

- Block completion – block produced all outputs
  - 1 branch, <= 32 register writes, <= 32 stores
  - DTs use DSN to maintain completed store info
  - DT and RTs notify GT via GSN
- Block commit
  - GT broadcasts on GCN to RTs and DTs to commit
- Commit acknowledgement
  - DTs and RTs notify GT via GSN

Prototype Evaluation - Area

- Area Expense
  - Operand Network (OPN): 12%
  - On Chip Network (OCN): 14%
  - Load Store Queues (LSQ) in DTs: 13%
  - Control protocol area overhead is light



IBM CU-11 process (130nm)
18x18 mm chip area
533MHz target clock rate
2 16-wide EDGE processors
16 GFlops/ops peak
1MB on-chip memory

Prototype Evaluation - Latency

- Cycle-level simulator (tsim-proc)
- Benchmark suite:
  - Microbenchmarks (dct8x8, sha, matrix, vadd), Signal processing library kernels, Subset of EEMBC suite, SPEC benchmarks
- Components of critical path latency
  - Operand routing largest contributor:
    - Hop latencies: 34%
    - Contention accounting: 25%
    - Operand replication and fan out: up to 12%
- Control latencies overlap with useful execution
- Data networks need optimization

Prototype Evaluation - Comparison

- Compared to 267 MHz Alpha 21264 processor
  - Speedups range from 0.6 to over 8
  - Serial benchmarks see performance degrade