Brook
A Streaming Programming Environment
Ian Buck, Mattan Erez, Ben Chelf, Pat Hanrahan

Stream Operators
- Load
- Store
- Group
- Stencil
- Flatten
- Repeat
- Stride
- SelfProduct
- MakeRef
- Gather
- GatherOp
- Scatter
- ScatterOp

Current Status
- Over 15,000 lines of Brook code
- 8 application designers
- Compiler support on Win32, Linux, MacOS X

Reference Streams
- values: [...]
- references: [...]
- streamScatterOp (values, references, FLOAT_ADD)

Why Stream Programming
- High arithmetic intensity
- Hide memory latency
- Efficient use of VLSI resources
- Exploit parallelism

Multidimensional Streaming
- streamShape (a, 2, 3, 3)
- streamStencil (b, a, 2, HALO, 0, 1, 0, 1)

Data & Task Parallelism
- Parallel execution hidden from programmer
- Task level parallelism
- Multiple kernels executing at once
- Data level parallelism
- Multiple stream elements processed simultaneously

Design Goals
- Encourage programmer write parallel code with high arithmetic intensity
- Place restrictions on kernels and streams
- General purpose streaming language
- No domain specific operators
- Easy to program and understand code
- Minor extensions to C

Multinode issues
- Memory synchronization
  - primitves
- Blocking primitives

Why Stream Programming
- High arithmetic intensity
- Hide memory latency
- Efficient use of VLSI resources
- Exploit parallelism

Kernel
- Functions which operate only on streams
  - Arguments are read-only, write-only, or reduce (associative operations only)
  - Called on every element of the input streams
  - Limited communication between elements
  - No “static” variables
  - No global memory access

Operation: streamScatterOp (values, references, FLOAT_ADD)

Design Goals
- Encourage programmer write parallel code with high arithmetic intensity
- Place restrictions on kernels and streams
- General purpose streaming language
- No domain specific operators
- Easy to program and understand code
- Minor extensions to C

General Streaming Language
- Parallel execution hidden from programmer
- Task level parallelism
- Multiple kernels executing at once
- Data level parallelism
- Multiple stream elements processed simultaneously

Research Directions
- Multinode issues
- Memory synchronization
  - primitves
- Blocking primitives

Current Status
- Over 15,000 lines of Brook code
- 8 application designers
- Compiler support on Win32, Linux, MacOS X

Streams
- Contiguous 1D view of records in memory
- Operated on in parallel
  - Declarations:
    - stream float floats;
  - Accessing stream values not permitted outside of kernels

Kernels
- Functions which operate only on streams
  - Arguments are read-only, write-only, or reduce (associative operations only)
  - Called on every element of the input streams
  - Limited communication between elements
  - No “static” variables
  - No global memory access

Reference Streams
- values: [...]
- references: [...]
- streamScatterOp (values, references, FLOAT_ADD)

Multidimensional Streaming
- streamShape (a, 2, 3, 3)
- streamStencil (b, a, 2, HALO, 0, 1, 0, 1)

Data & Task Parallelism
- Parallel execution hidden from programmer
- Task level parallelism
- Multiple kernels executing at once
- Data level parallelism
- Multiple stream elements processed simultaneously

Research Directions
- Multinode issues
- Memory synchronization
  - primitves
- Blocking primitives

Current Status
- Over 15,000 lines of Brook code
- 8 application designers
- Compiler support on Win32, Linux, MacOS X

Streams
- Contiguous 1D view of records in memory
- Operated on in parallel
  - Declarations:
    - stream float floats;
  - Accessing stream values not permitted outside of kernels

Kernels
- Functions which operate only on streams
  - Arguments are read-only, write-only, or reduce (associative operations only)
  - Called on every element of the input streams
  - Limited communication between elements
  - No “static” variables
  - No global memory access

Reference Streams
- values: [...]
- references: [...]
- streamScatterOp (values, references, FLOAT_ADD)

Operations supported: addition, substraction multiplication, reciprocal, maximum, minimum, logical or, logical and.