

---

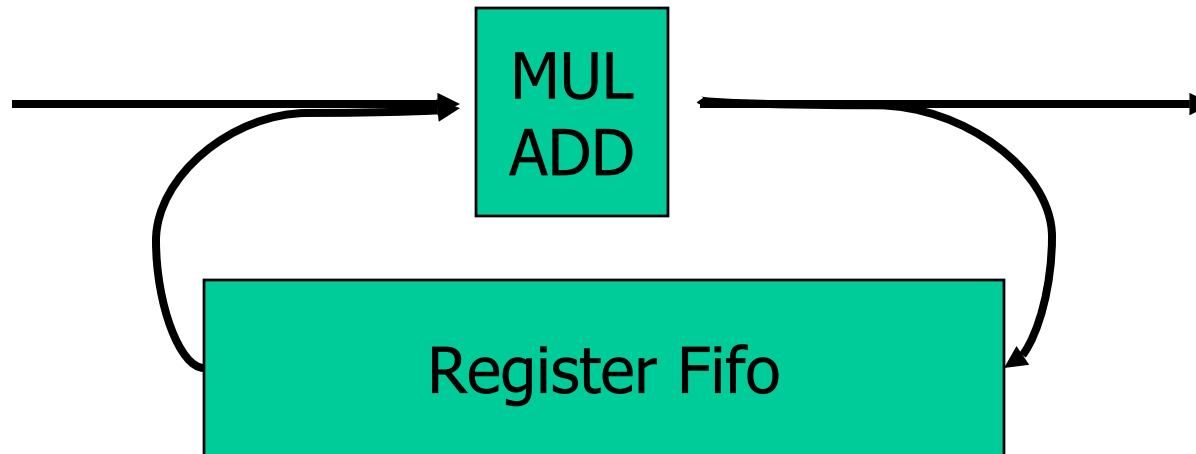
# Compilation Targets

Ian Buck, Francois Labonte  
February 04, 2003

# GPU: Architectural Differences

---

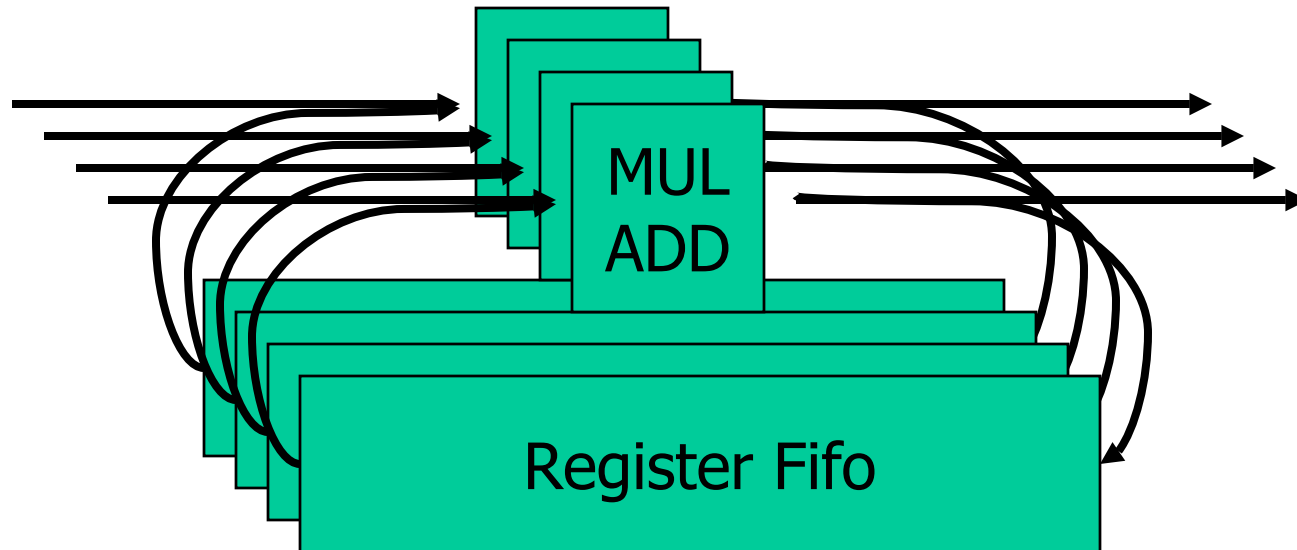
- No SRF
- Pipelined MADD units
- Multiplexed Register File



# GPU: Architectural Differences

---

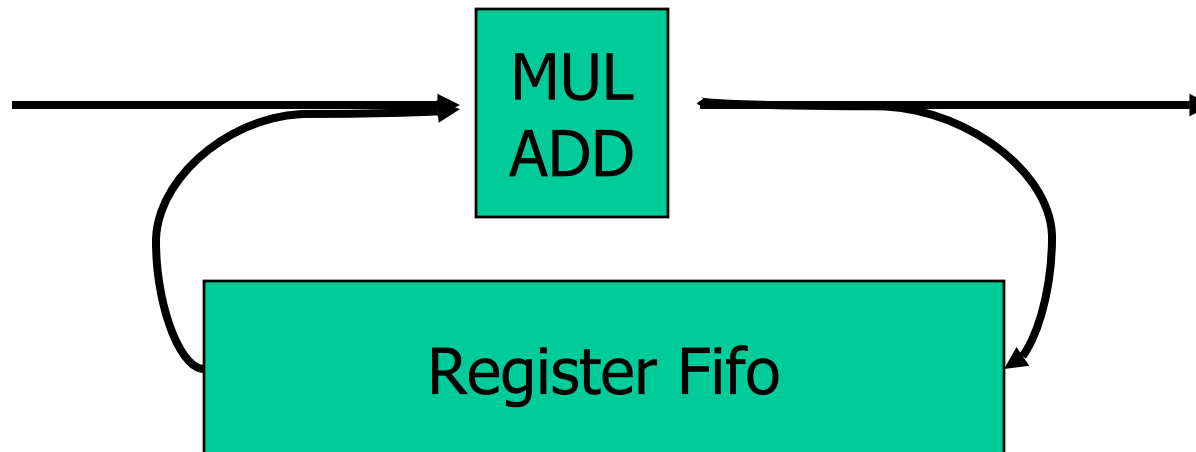
- No SRF
- Pipelined MADD units
- Multiplexed Register File



# GPU: Architectural Differences

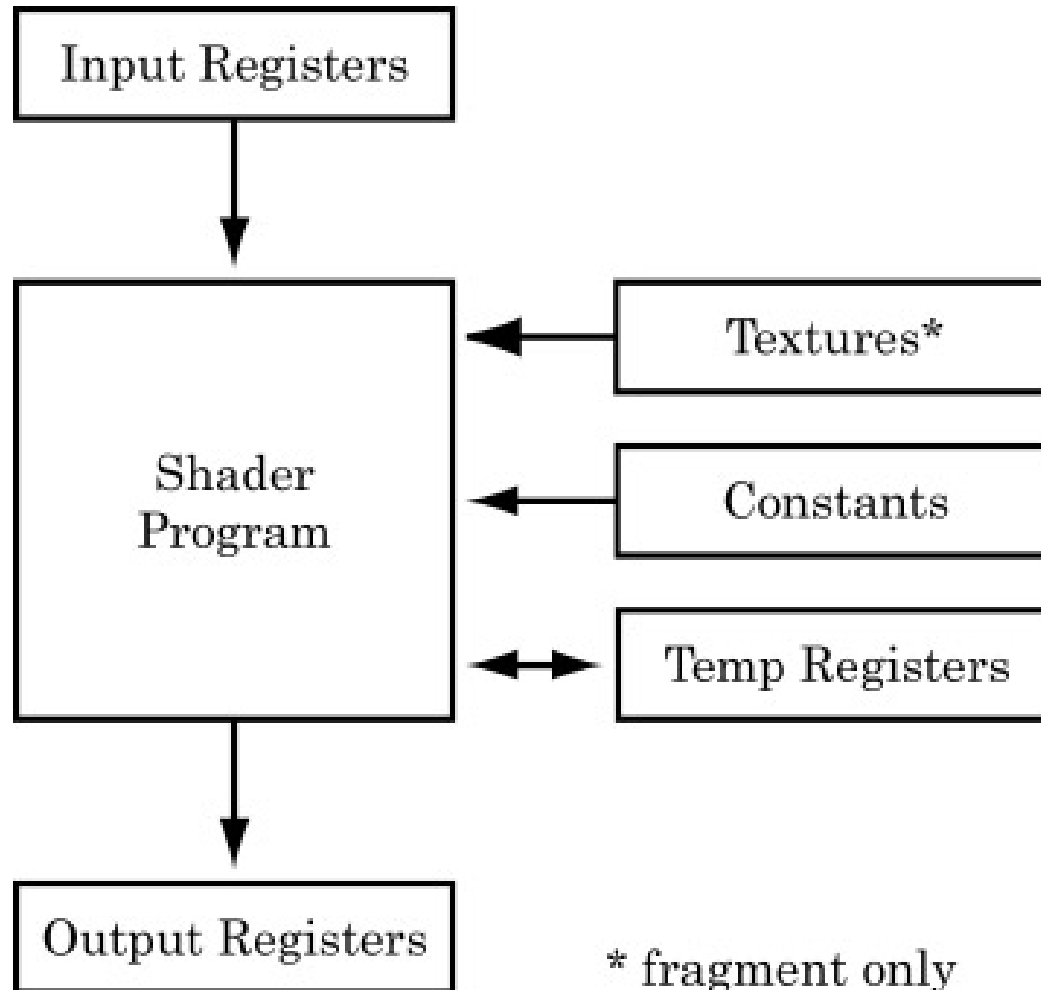
---

- No SRF
- Multiplexed Register File
- Data Parallelism
- Arithmetic Intensity
- Gather inside kernels



# GPU: Programming Model

---



\* fragment only

# GPU: Programming Model

---

- Positives
  - 4-vector fp32 SIMD instruction set
  - Gathers allowed inside kernels
  - High level compilers (Cg & HLSL)

# GPU: Programming Model

---

- Negatives
  - No exposed SRF
  - Limited Scatter capabilities
  - No branching
  - No retained state between stream elements

# GPU: Compilation Target

---

- Compile Brook kernels to Cg
- Streams = Textures
- Roll Operators into gathers
  - Stencil, Group
- Compile stream graph into large kernels

# GPU Compilation Target

---

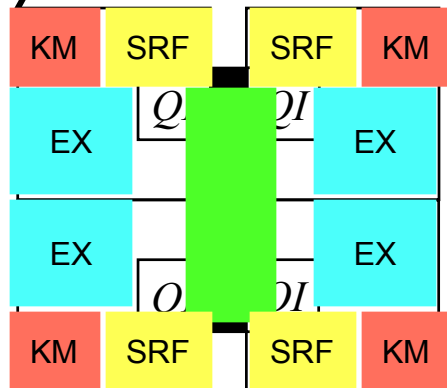
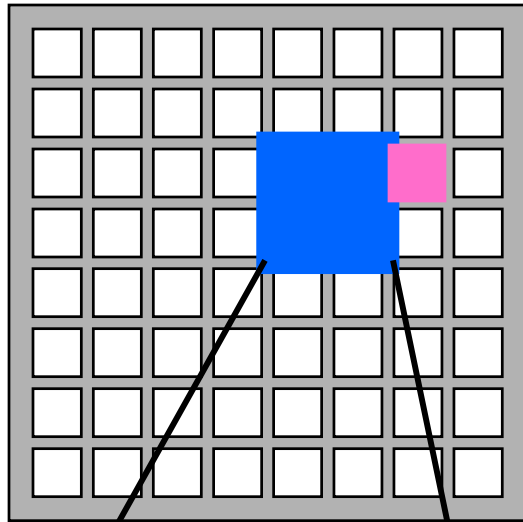
- Challenges
  - Reductions require  $\lg(N)$  passes
  - Scatter requires host assist
    - May be fixed soon
  - Limited resources
    - registers
    - inputs / outputs
    - instruction counts
  - Needs generalized RDS

# GPU Compilation Target

---

- Questions
  - How does a GPU fit into the SVM?
    - Texture memory ~ SRF?
  - Do we allow gather operations inside of kernels?
  - Multinode issues?
    - Not a shared memory machine.

# Smart Memories



- Original Smart Memories

- 4 CPUs in a quad could be configured as a 4 cluster machine working in SIMD
- Control node was one processor node
- Memory tiles could be configured as SRF banks, kernel instruction memory stream buffers.

# Smart Memory Implementation Status

---

- Instead of creating the whole processor core, Smart Memories is looking at using a processor core from Tensilica
- Tensilica provides extensible (add instructions) synthesizable processor cores.
- The status of streaming is uncertain because
- Until this is resolved, it is not worthwhile discussing

# X86 Workstation cluster - Diff

---

- No SRF per se
  - Could try to exploit cache as SRF (similar to Sandia's Sierra)
- Indexing in kernels is possible
  - Though degrades performance if outside the cache
- Conditionals: branches are possible, predication not (single cluster)
- SIMD instructions – SSE/MMX provide extra ILP
- Simultaneous Multithreading – Chance to overlap memory and kernel execution.

# Multinode issues

---

- Not shared memory environment
  - Do we need software address translation?
  - Would be simpler to implement on SGI Origin or Flash
- ScatterOps across multiple nodes need to go through the CPU of the concerned memory location

# Compilation Paths

---

- Brook -> Mattan/Jayanth compiler -> SVM -> pthreads
  
- Brook on multiple threads - Christos