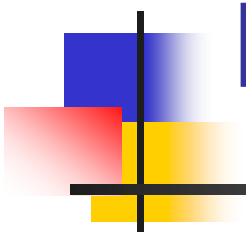
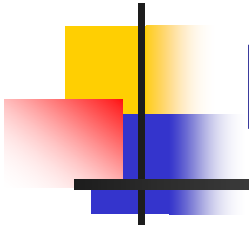


# Hybrid Computer Architecture

Brian Van Essen

Benjamin Ylvisaker

Carl Ebeling

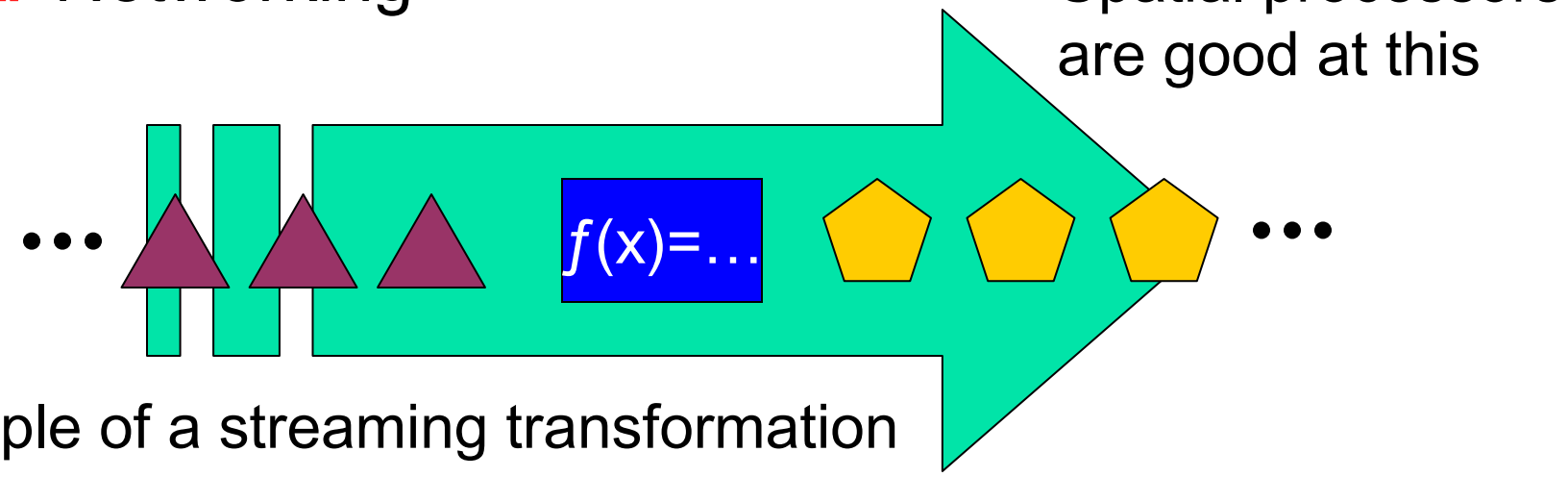


# Moore's Law: Is it Over?

- von Neumann processors no longer scale
  - Overhead of speculative execution is too high
  - Complexity of superscalar OOO core is $n^2$
  - Optimum power / performance pipeline depth is ~7 stages
- Spatial processors benefit from added transistors
  - Reconfigurability allows virtualization
    - Enables programming abstraction

# Keeping up with streams is hard

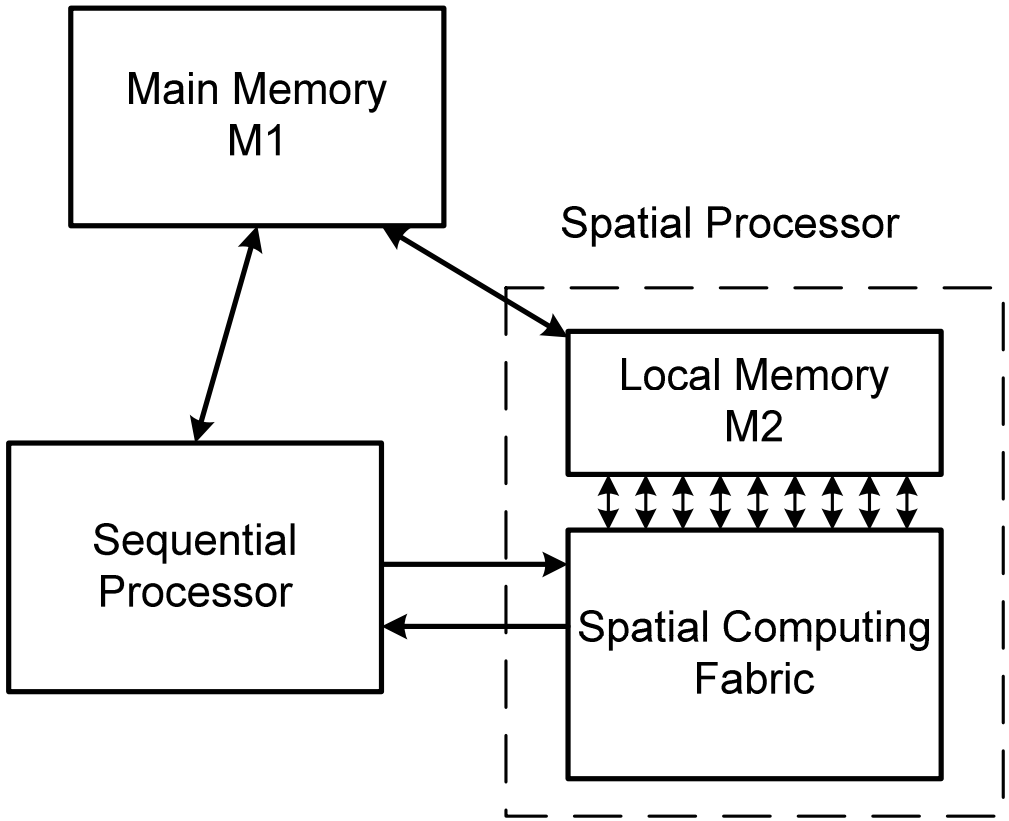
- Multimedia workloads
  - Audio & Video
- Communication workloads
  - Networking

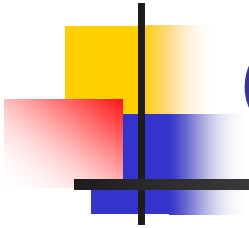



Example of a streaming transformation

# Hybrid Architecture Research

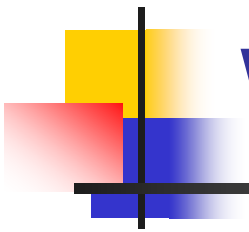
- Blend sequential and spatial computing
  - One program executes both types of computation





# Overview

- What is spatial computing
  - Why is it interesting
- Hybrid Architectures
  - What is hard about hybrid architectures
- Future Research



# What is spatial computing?

- Spatial processors:
  - Parallel array of compute elements (fabric)
  - Assign operations to different physical resources
  - Stream operands through the fabric
  - Execute many operations in parallel

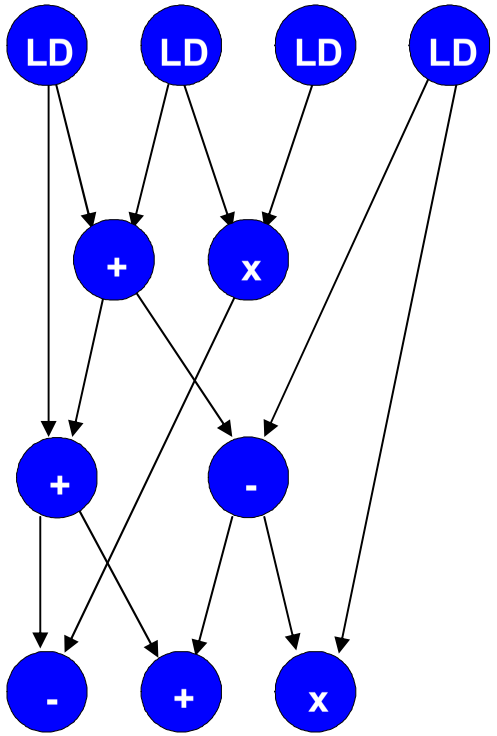
- Sequential processors:
  - Step through a sequence of instructions

# Encoding a program

## Instruction Stream

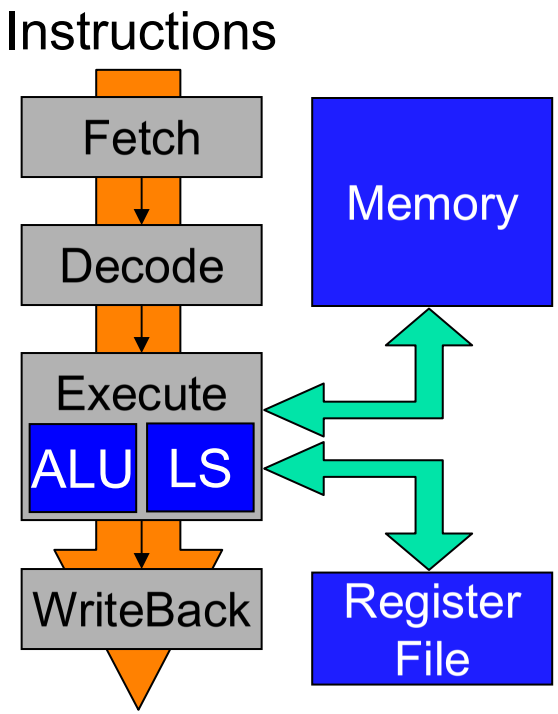
```
Load  r1, A
Load  r2, B
Load  r3, C
Load  r4, D
Add   r5, r1, r2
Mul   r6, r2, r3
Add   r7, r1, r5
Sub   r8, r5, r4
Sub   r9, r7, r6
Add  r10, r7, r8
Mul  r11, r8, r4
```

## Dataflow Graph

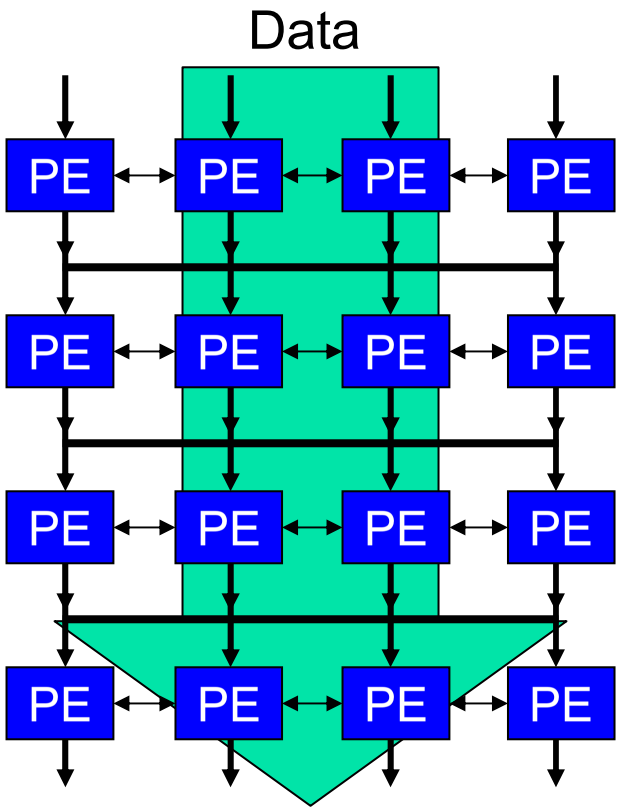


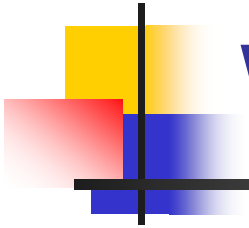
# Processors: Under the hood

Traditional Computer
(Load / Store Arch)




Spatial Computer
(e.g. FPGA, PipeRench)





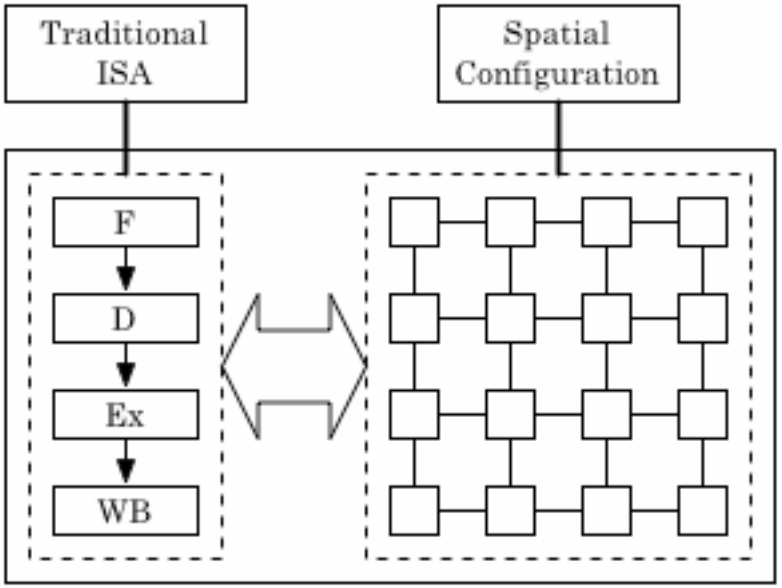
# Why spatial processors?

- Extremely efficient for certain applications
  - Regular computation
  - Regular communication
    - e.g. Streaming Data
- Excellent performance / power ratio

- Limitations:
  - Difficult to execute control flow
  - Hard to program

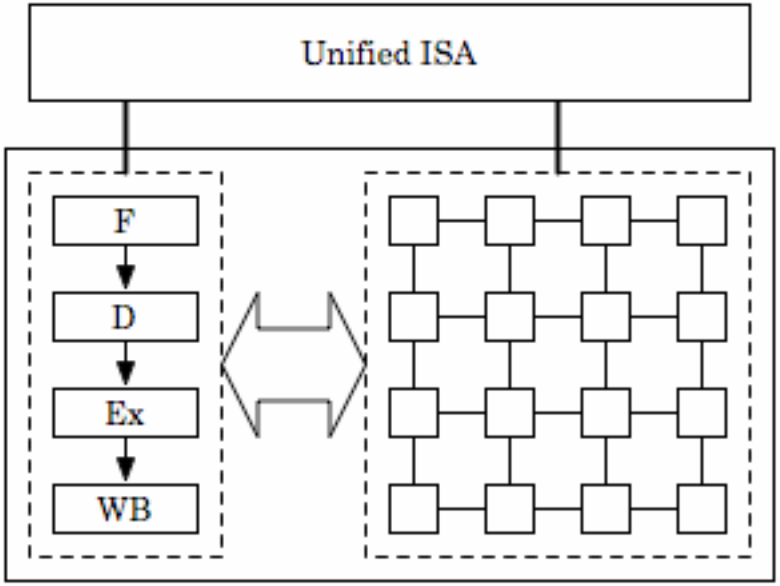
# Basic Hybrid Architectures

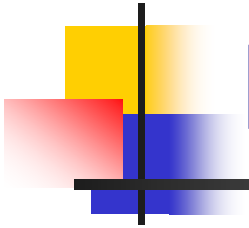
- Two processors on a single chip
  - Integrates control plane and data plane processors
  - Provide high speed interconnect
  - Share memory
- Execute independent programs
  - Manage synchronization



# Unified Hybrid Architecture

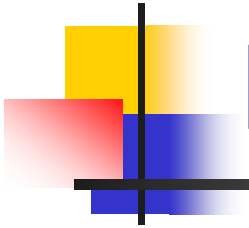
- Single programming model
  - Collapses control plane and data plane processors into single abstraction
  - Implicit synchronization
  - Simplified programming abstraction
  - Program “Automagicly” executes on appropriate processor
    - Runtime system manages fabric configuration





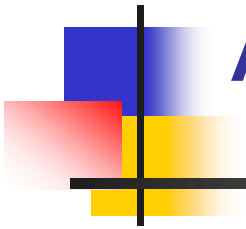
# Research Challenges

- Creating a new Instruction Set Architecture (ISA)
  - Provides canonical sequential interpretation
  - Exposes good spatial configuration
  - Efficient synchronization of runtime control
- Virtualization of spatial processors is hard
  - Necessary to provide abstract programmers model
  - Use dynamic reconfiguration
- Programming Language
  - Explicit stream operations
  - Disambiguate memory references



# Research Synopsis

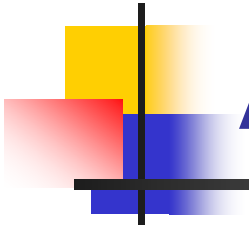
- Define new processor architecture and ISA
  - New level of ease of use
    - Unified programming model
  - Blend sequential and spatial computing
    - Excels at streaming data applications
    - One program executes both types of computation
  - Implicit communication
- Efficient virtualization of spatial processors
- System-level programming language



# Appendix

Type Architectures

Programming Languages



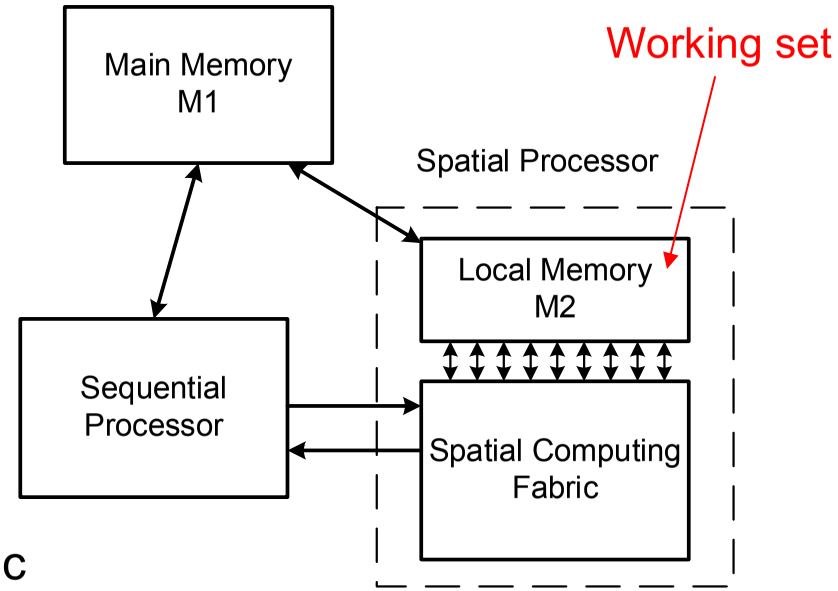
# Abstract processor models

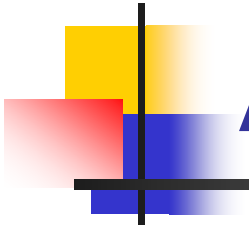
- von Neumann Type Architecture - RAM Model
  - A processor interpreting 3-address instructions
  - PC describing the next instruction of program in memory
  - Flat, randomly accessed memory requires 1 time unit
  - Memory is composed of fixed sized addressable units
  - One instruction executes at a time, and is completed before the next instruction executes
- Modern RISC & CISC processors emulate this model

C directly implements this model

# Hybrid Type Architecture

- von Neumann sequential processor
- Spatial Fabric
  - P operations per cycle
  - Statically scheduled
- Main Memory
  - ~ 1 access per cycle
- Local Memory (Workspace)
  - ~ P accesses per cycle
  - enough to maintain P ops
- Alternating Execution
  - Sequential program executes
  - Control transferred to spatial fabric
    - Shared state transferred
    - Atomic execution of spatial section
    - Shared state transferred back





# A new Programming Language

- "System level"
  - Full control of underlying ISA
  - Explicit resource management
- Key Issues
  - Expressing parallel portions of computation
    - Easily mapped to spatial processor
  - "Relaxed" memory access ordering
    - e.g. streams
  - Disambiguate memory references
    - mitigate aliasing
  - Reflect constraints of type architecture
    - e.g. low main memory bandwidth