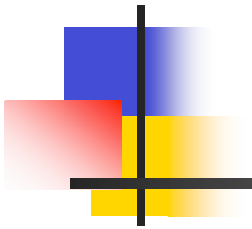
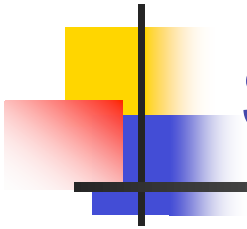


Programming Hybrid Sequential / Micro-parallel Computers

Presented by Brian Van Essen

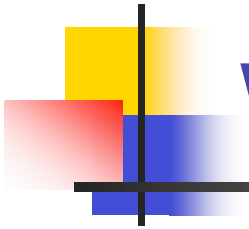
Benjamin Ylvisaker & Brian Van Essen

Faculty: Carl Ebeling



Spatial Computing: On its way?

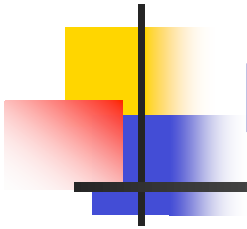
- Excellent performance
- Great power efficiency
- Good for exploiting micro-parallelism
- Hard to program
  - Typically requires hardware designers



What is “micro-parallelism”?

"micro-parallelism" describes both computations and architectures

- Features of micro-parallel computations are :
  - parallel, closely communicating, operations
  - predictable communication
  - significant repetition
- Features of micro-parallel architectures are:
  - simple, concurrent, compute units
  - a scalable interconnection fabric
  - simple and efficient control

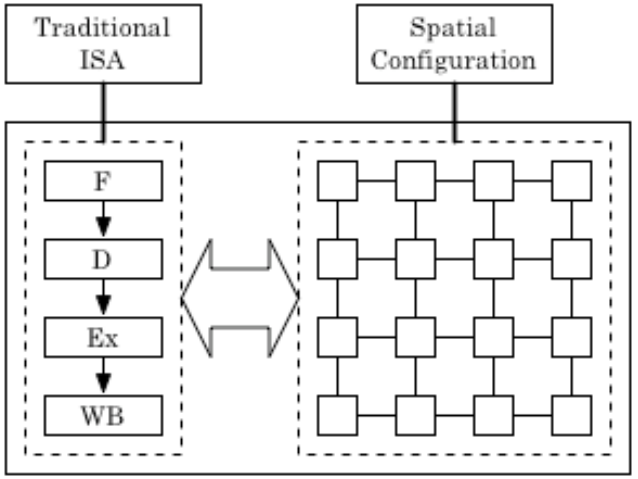


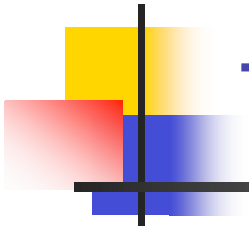
Research Goals

- Define an effective abstract model for hybrid micro-parallel architectures
- Simplify the process of programming micro-parallel engines
- Enable high-performance processor architectures for micro-parallel applications

Hybrid Micro-Parallel Computers

- Sequential processor
  - + Good for control flow
  - + Random Data access
  - + Ease of programming
- Micro-parallel engines
  - + Recurring operations (streaming data)
  - + Excellent performance
  - + Good power efficiency
- Integration is Hard
  - - Multiple programming models
  - - Communication and synchronization
  - - Sharing data
  - - Application partitioning





Type Architecture

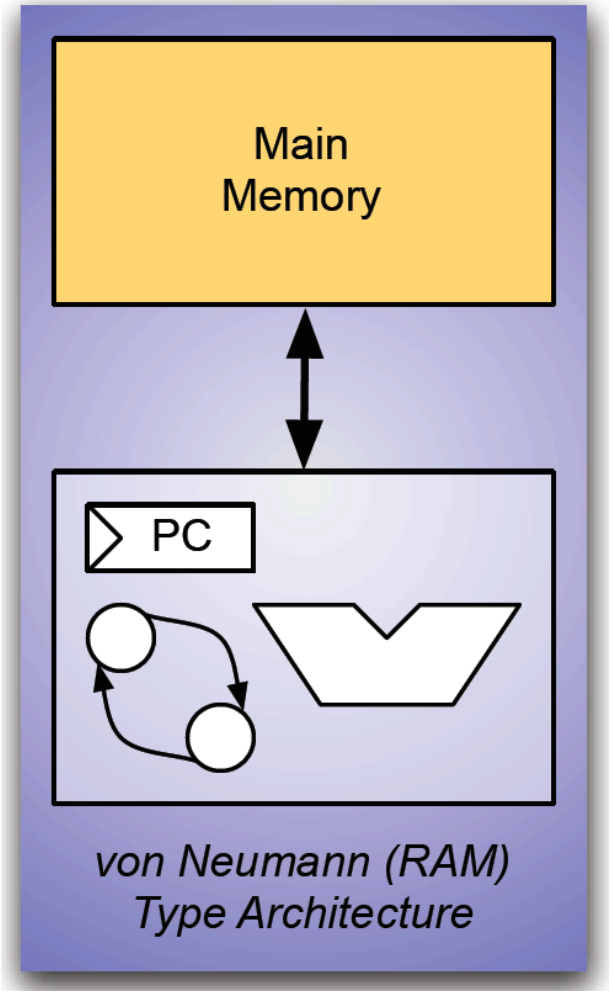
- A model that is a:

  "region of consensus … explicit about a few salient features [of a family of computers] and mute on everything else"

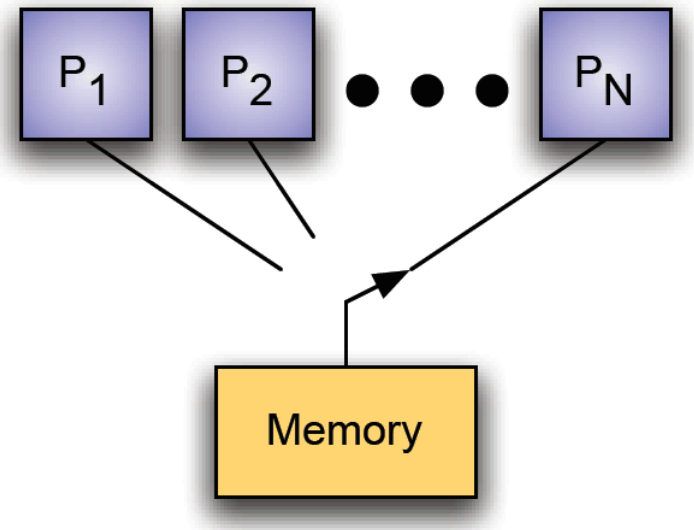
- A type architecture defines:
  - Resources available
  - Behavior (execution model)
  - Performance characteristics
- Programmer can rely on model
- Computer architect is obliged to implement the model

e.g. von Neumann Type Architecture

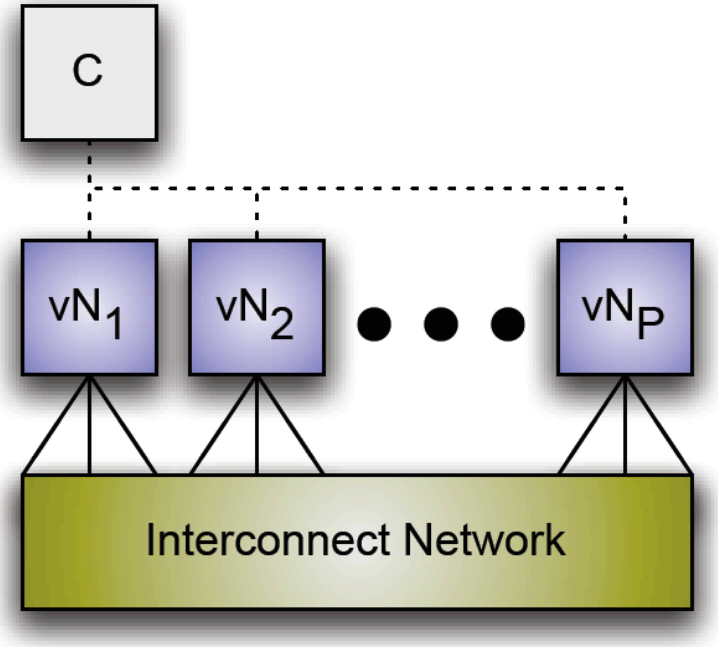
- Modern RISC & CISC processors realize this model
- C directly implements this model
- C programmers understand the cost of a given program
- Not sufficient for hybrid systems



e.g. Parallel Computers



Parallel RAM (PRAM)
Type Architecture

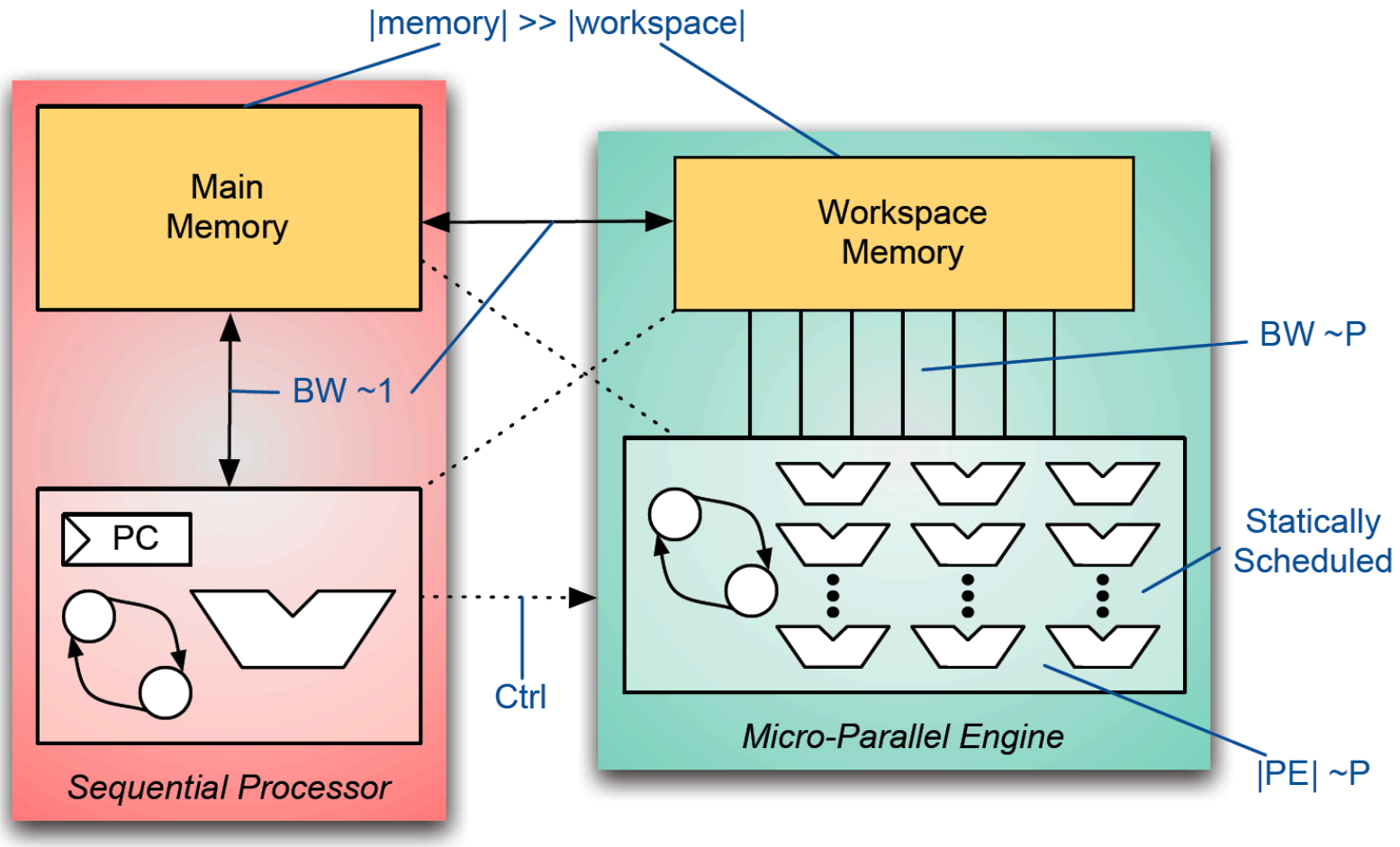


Candidate Type Architecture (CTA)

Model does not match reality

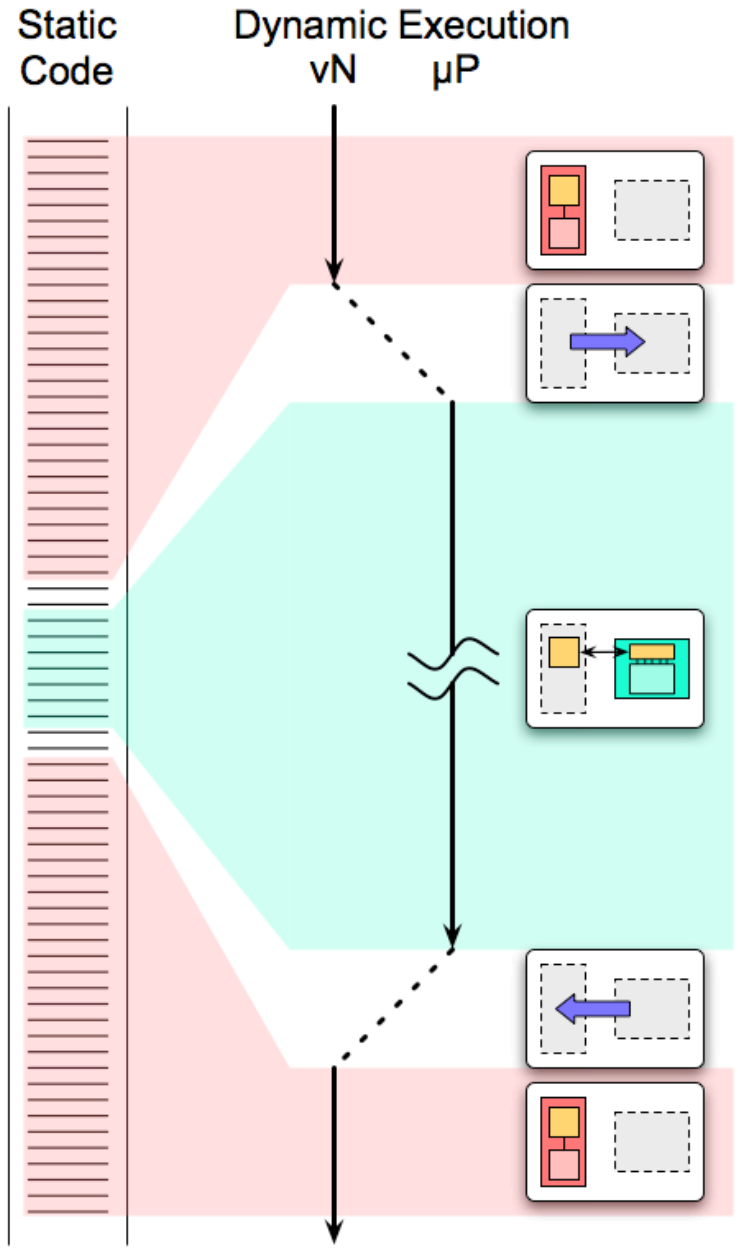
Cost(Remote Mem >> Local Mem)

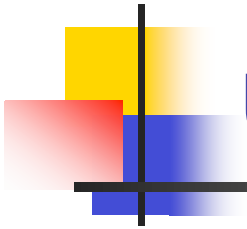
Hybrid Micro-Parallel Type Architecture



Execution Model

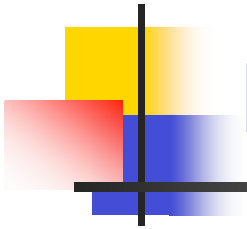
- Single thread of execution
  - Simplifies programmers' reasoning
- Two modes:
  - Sequential execution
  - Micro-parallel execution
- 90% of the runtime is in 10% of the code
  - aka. kernels





Using the HMP for performance analysis

- Is there sufficient parallelism in application?
- Does working set fit in workspace memory?
- Can the workspace be replenished quickly enough?
- Does micro-parallel execution speedup outweigh overhead of switching execution modes?



HMP Type Architecture Summary

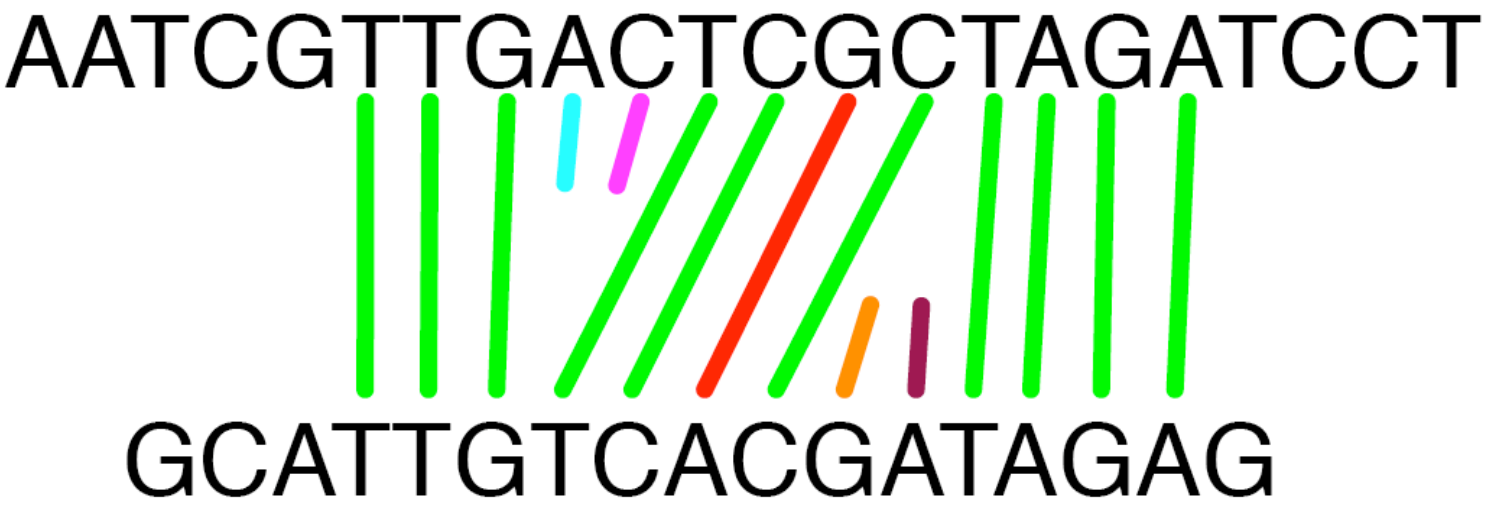
- Integrates sequential processor and micro-parallel engine into a unified model
- Highlights salient details of micro-parallel execution to programmer

HMP type architecture models:

- Resources
  - Large shared main memory
  - Smaller, distributed, workspace memory
  - ~P parallel processing elements
- Execution
  - Single thread of execution

Example: Smith/Waterman

- Smith/Waterman is an algorithm that assigns scores to strings that "almost" match.
  - It is used widely in biology for protein and DNA strings



Breaking down the algorithm

- Two ways to think about Smith/Waterman:

Recursively defined equations for Smith/Waterman string matching

$$E_{i,j} = \max\{H_{i,j-1} - u_d,\ E_{i,j-1} - v_d\}$$
$$F_{i,j} = \max\{H_{i-1,j} - u_q,\ F_{i-1,j} - v_q\}$$
$$H_{i,j} = \max\{H_{i-1,j-1} + S(q_i, d_j),\ E_{i,j},\ F_{i,j}\}$$

Recursive references to E, F and H

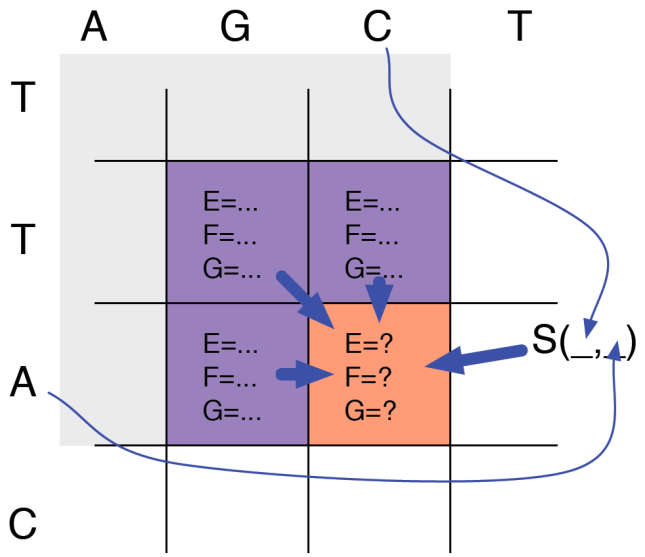
Constants

Comparison of two individual letters

Base cases:

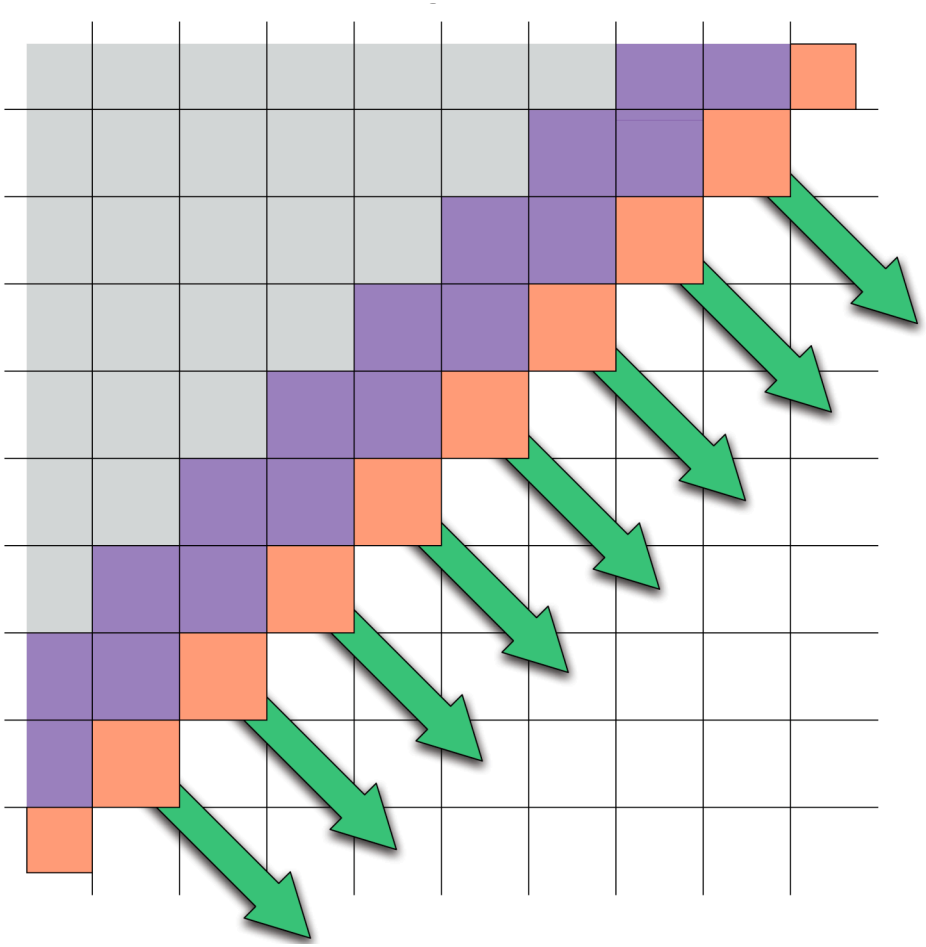
$$E_{i,0} = E_{0,j} = F_{i,0} = F_{0,j} = H_{i,0} = H_{0,j} = 0$$

Filling in a 2D table



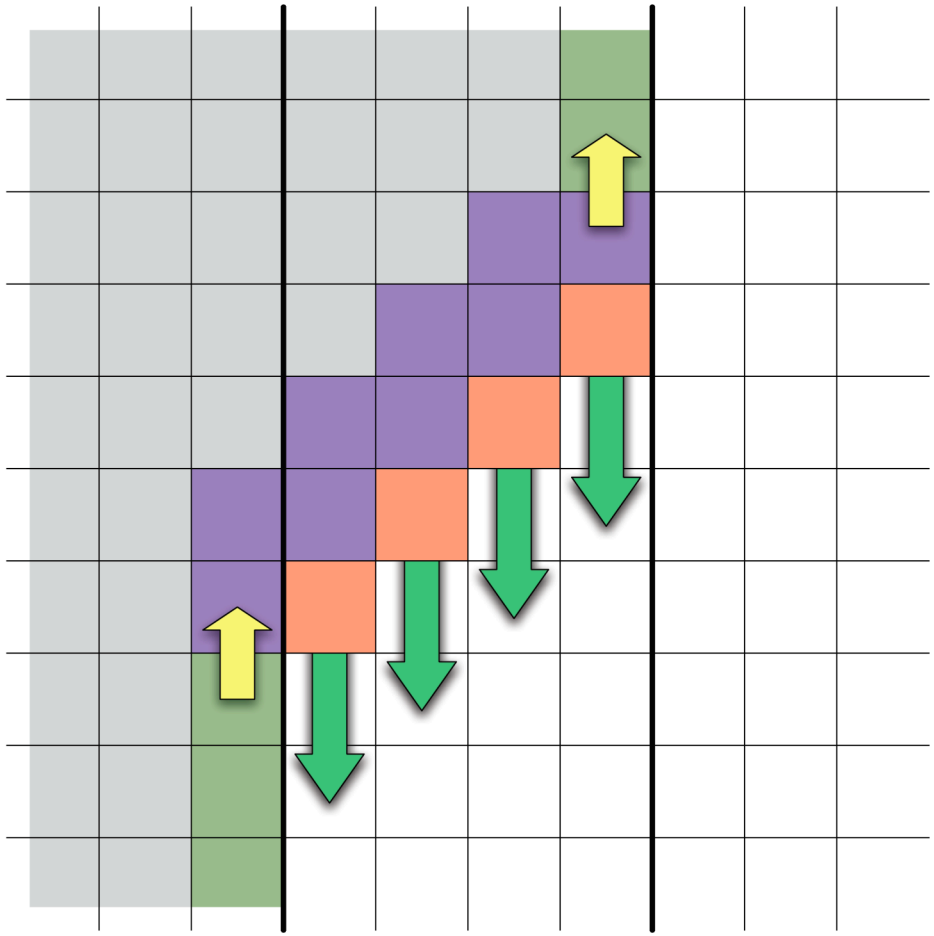
Step 1: Identify Parallelism

- Compute the diagonal and sweep across the table



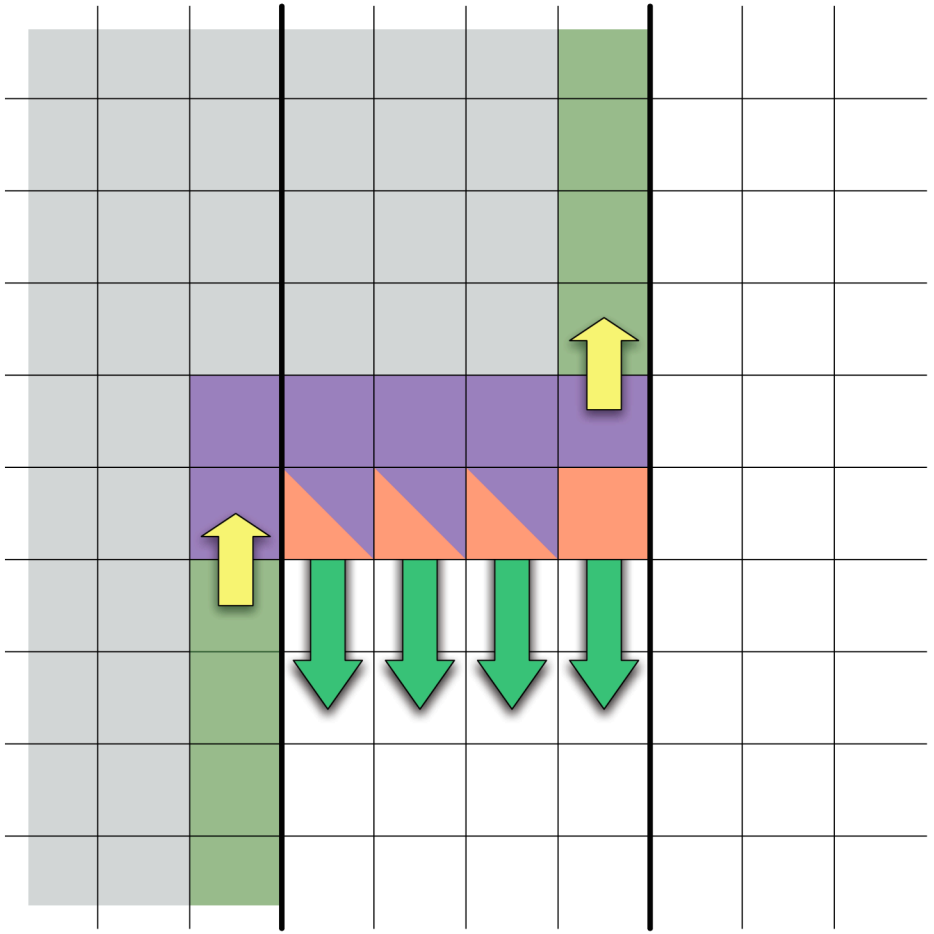
Step 2: Observe Constraints

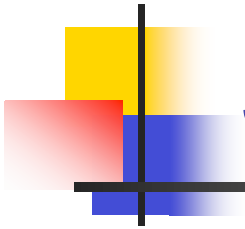
- Workspace cannot hold two entire diagonals of table
- Tile the table in manageable pieces



Step 3: Simplify

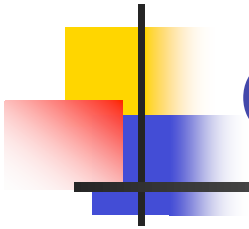
- Write down as a loop over a single row
- Compiler will pipeline automatically
  - e.g. software pipelining





Summary

- Introduced the Hybrid Micro-Parallel Type Architecture
  - Integrates sequential processor and micro-parallel engine into a unified model
  - Highlights salient details of micro-parallel execution to programmer
- Programmer can rely on model
- Computer architect is obliged to implement the model
- Develop language that reflects HMP type architecture
- Explore architecture space for HMP implementation



Questions?

- Visit us at the poster session in:

Hardware and Embedded Systems Lab, CSE 505

Brian Van Essen & Benjamin Ylvisaker
Faculty: Carl Ebeling