ECE 498AL

Programming Massively Parallel Processors

Lecture 13:
Structuring Parallel Algorithms

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009

Key Parallel Programming Steps

1) To find the concurrency in the problem
2) **To structure the algorithm to translate concurrency into performance**
3) To implement the algorithm in a suitable programming environment
4) To execute and tune the performance of the code on a parallel system

Unfortunately, these have not been separated into levels of abstractions that can be dealt with independently.

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009

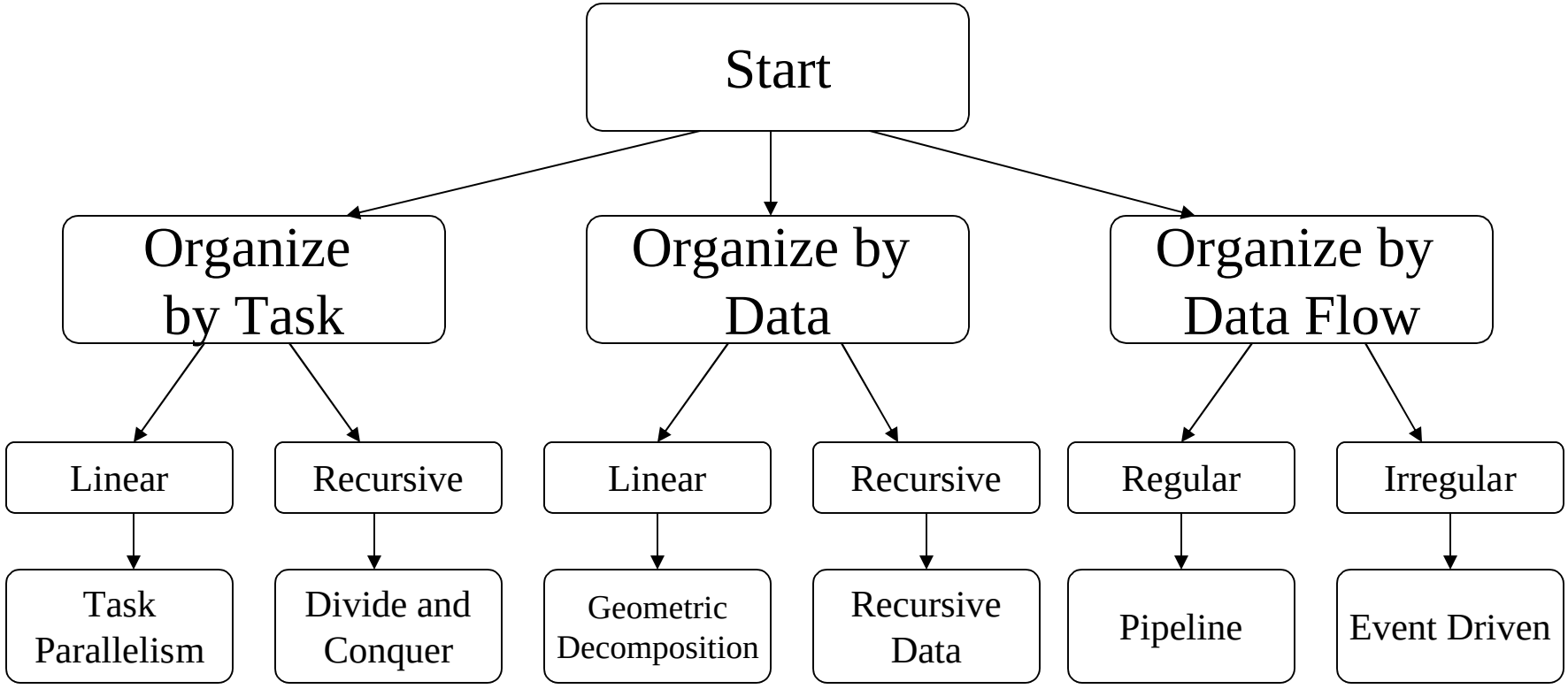
Algorithm

- A step by step procedure that is guaranteed to terminate, such that each step is precisely stated and can be carried out by a computer
  - Definiteness – the notion that each step is precisely stated
  - Effective computability – each step can be carried out by a computer
  - Finiteness – the procedure terminates

- Multiple algorithms can be used to solve the same problem
  - Some require fewer steps
  - Some exhibit more parallelism
  - Some have larger memory footprint than others

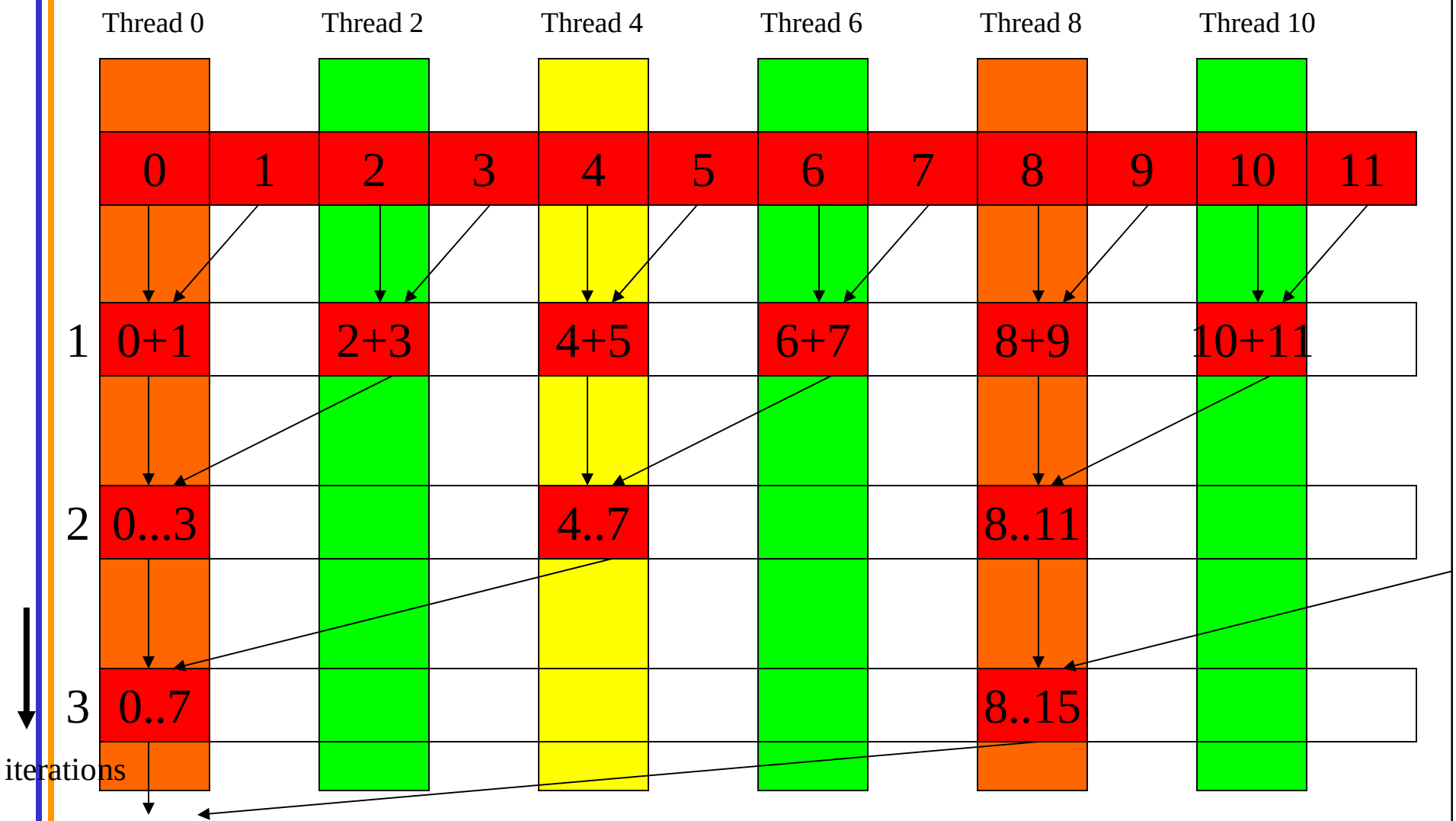
© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009

Choosing Algorithm Structure



© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009

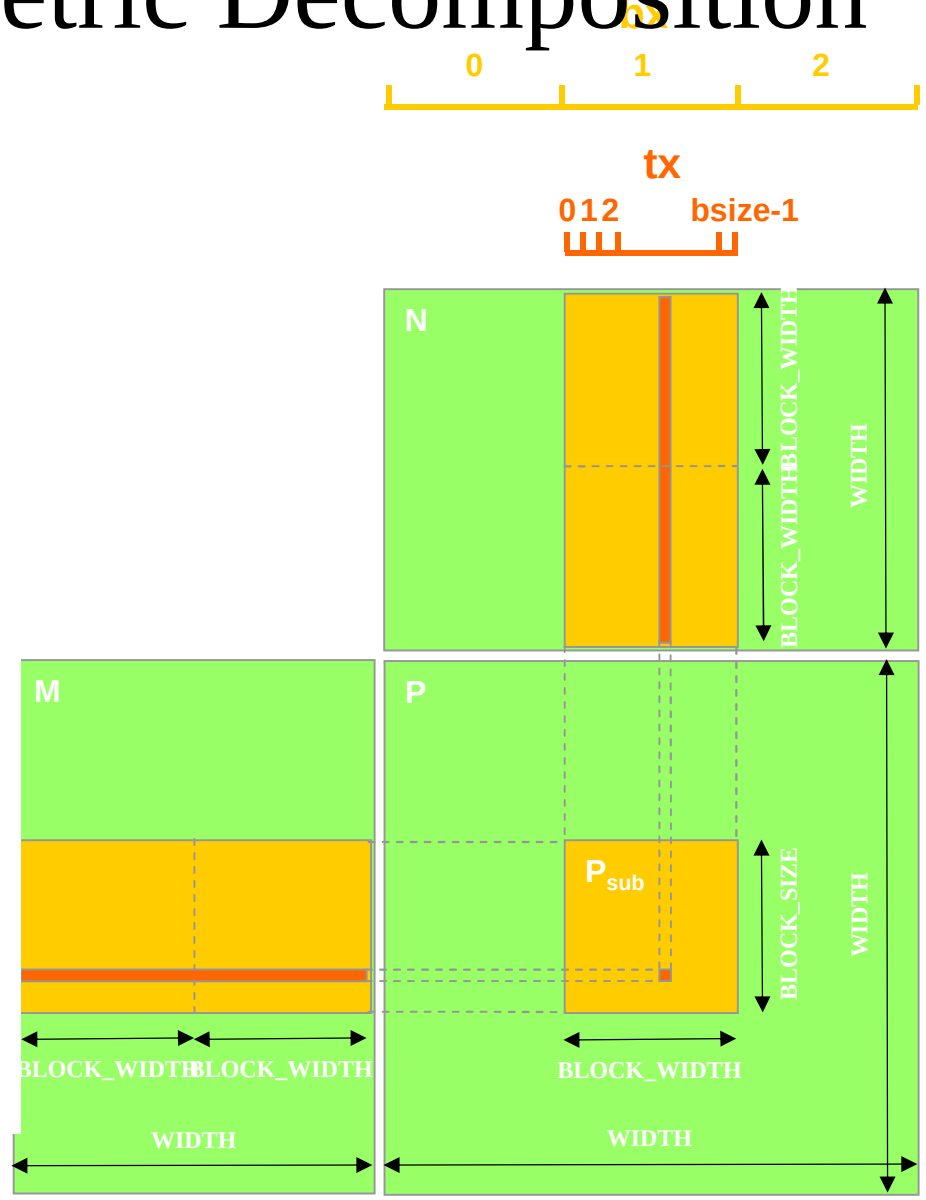
Mapping a Divide and Conquer Algorithm



© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009
ECE 498AL, University of Illinois, Urbana-Champaign

Tiled (Stenciled) Algorithms are Important for Geometric Decomposition

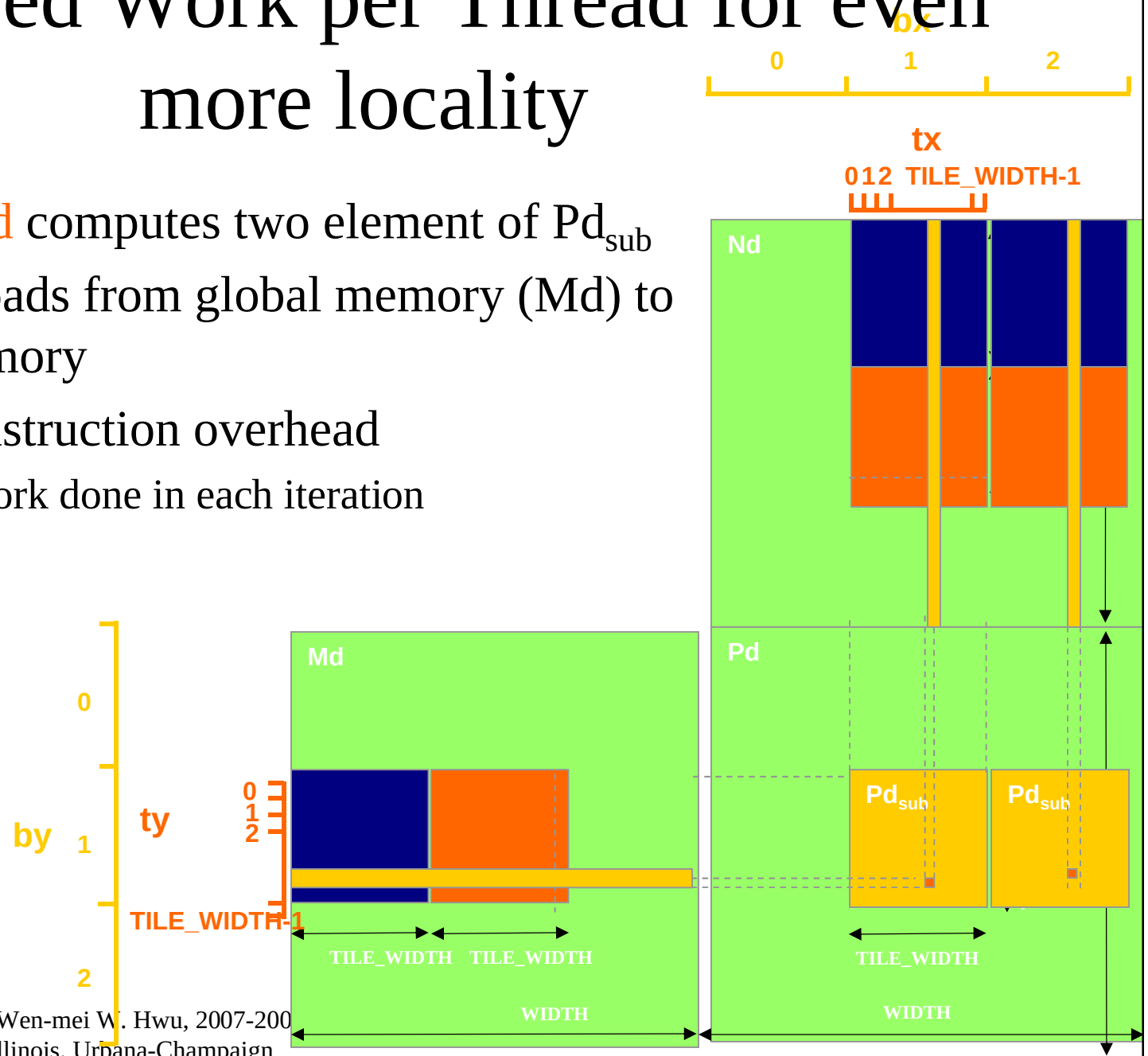
- A framework for memory data sharing and reuse by increasing data access locality.
  - Tiled access patterns allow small cache/scartchpad memories to hold on to data for re-use.
  - For matrix multiplication, a 16X16 thread block perform 2*256 = 512 float loads from device memory for 256 * (2*16) = 8,192 mul/add operations.
- A convenient framework for organizing threads (tasks)



© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009
ECE 498AL, University of Illinois, Urbana-Champaign

Increased Work per Thread for even more locality

- Each thread computes two element of Pd_{sub}
- Reduced loads from global memory (Md) to shared memory
- Reduced instruction overhead
  - More work done in each iteration



© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-200
ECE 498AL, University of Illinois, Urbana-Champaign

Double Buffering - a frequently used algorithm pattern

- One could double buffer the computation, getting better instruction mix within each thread
  - This is classic software pipelining in ILP compilers

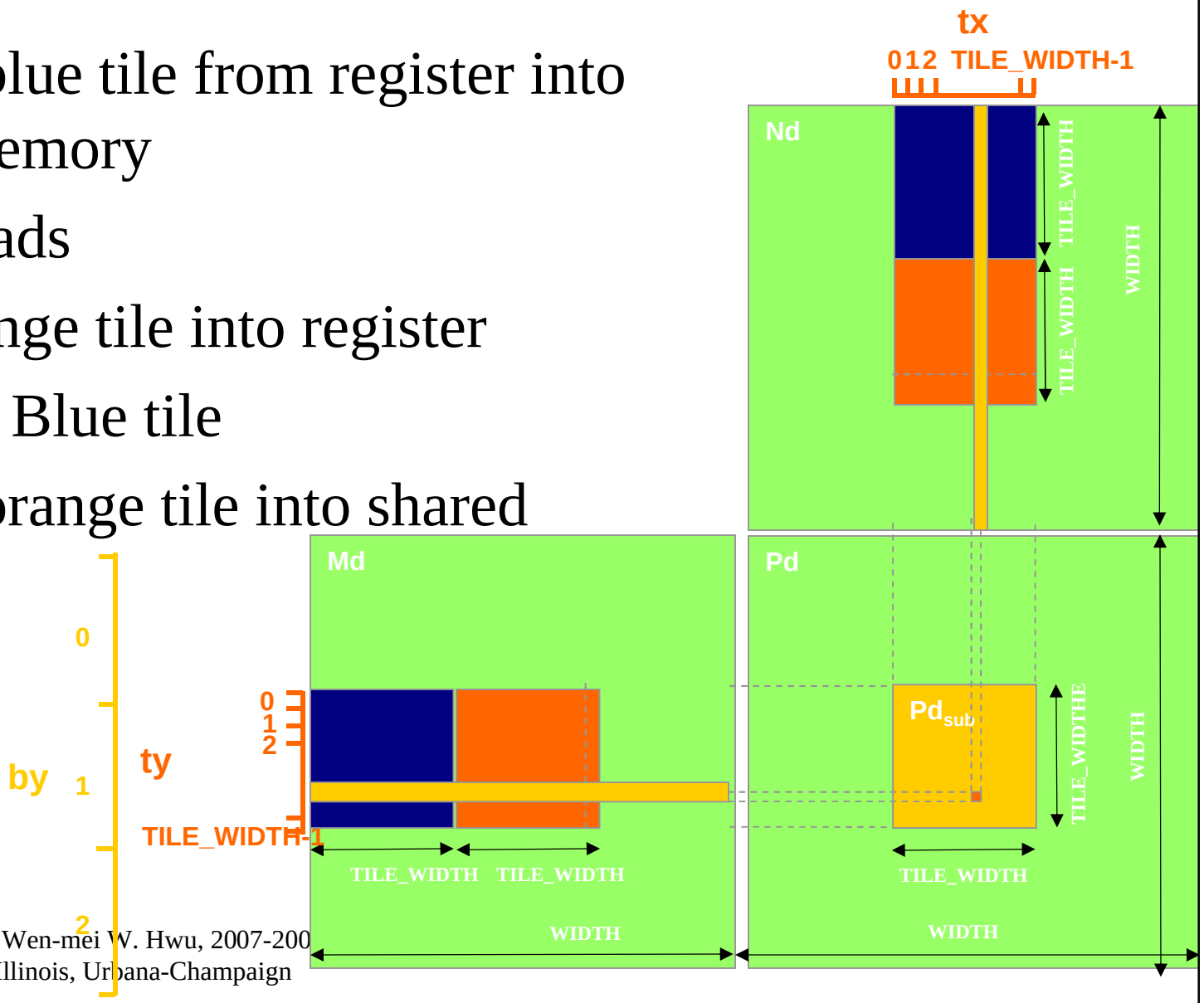
```
Loop {

  Load current tile to shared memory

  syncthreads()

  Compute current tile

  syncthreads()
}
```

```
Load next tile from global memory

Loop {
  Deposit current tile to shared memory

  syncthreads()

  Load next tile from global memory

  Compute current tile

  syncthreads()
}
```

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009
ECE 498AL, University of Illinois, Urbana-Champaign

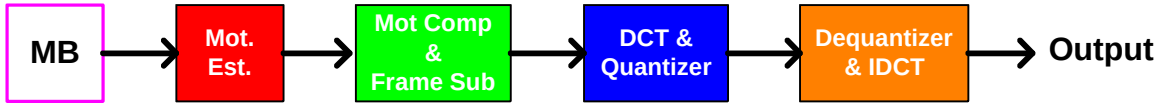
Double Buffering

- Deposit blue tile from register into shared memory
- Syncthreads
- Load orange tile into register
- Compute Blue tile
- Deposit orange tile into shared memory
- ….

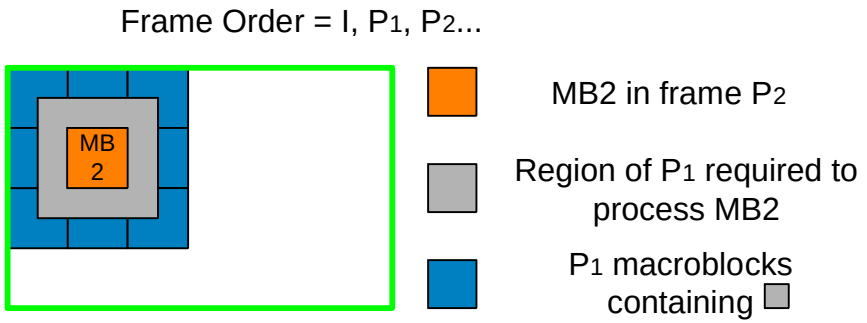


© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-200
ECE 498AL, University of Illinois, Urbana-Champaign

One can trade more work for increased parallelism

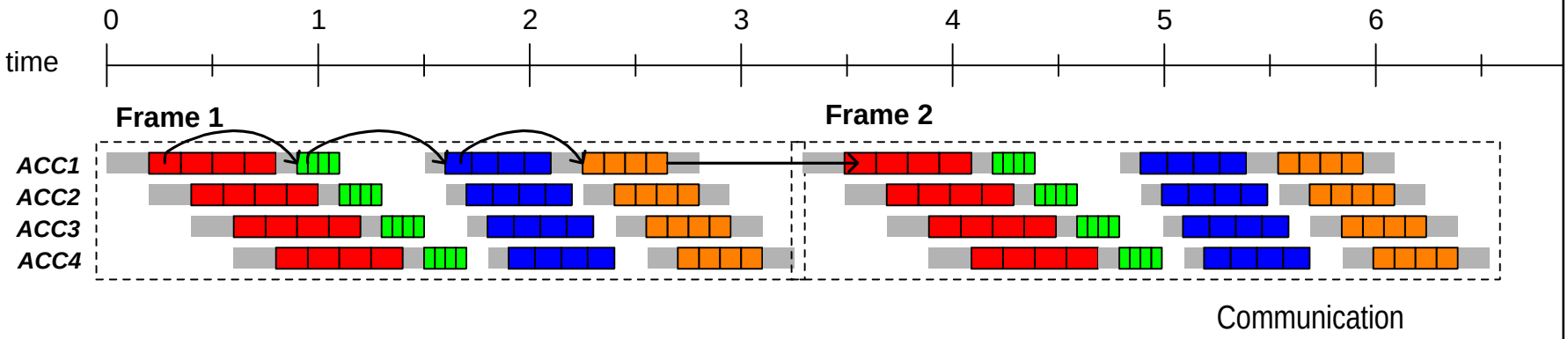


- Diamond search algorithm for motion estimation work efficient but sequential
  - Popular in traditional CPU implementations
- Exhaustive Search totally parallel but work inefficient
  - Popular in HW and parallel implementations



© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009

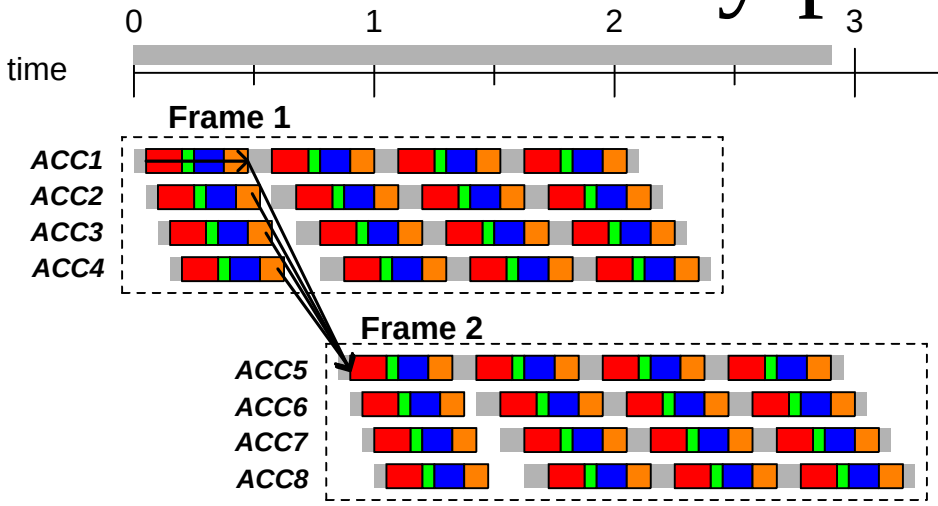
An MPEG Algorithm based on Data Parallelism



- Loops distributed – `DOALL` style
  - Replicates instructions and tables across accelerators
- If instructions and data are too large for local memory…
  - Large memory transfers required to preserve data
  - Saturation of and contention for communication resources can leave computation resources idle

Memory bandwidth constrains performance

Loop fusion & memory privatization

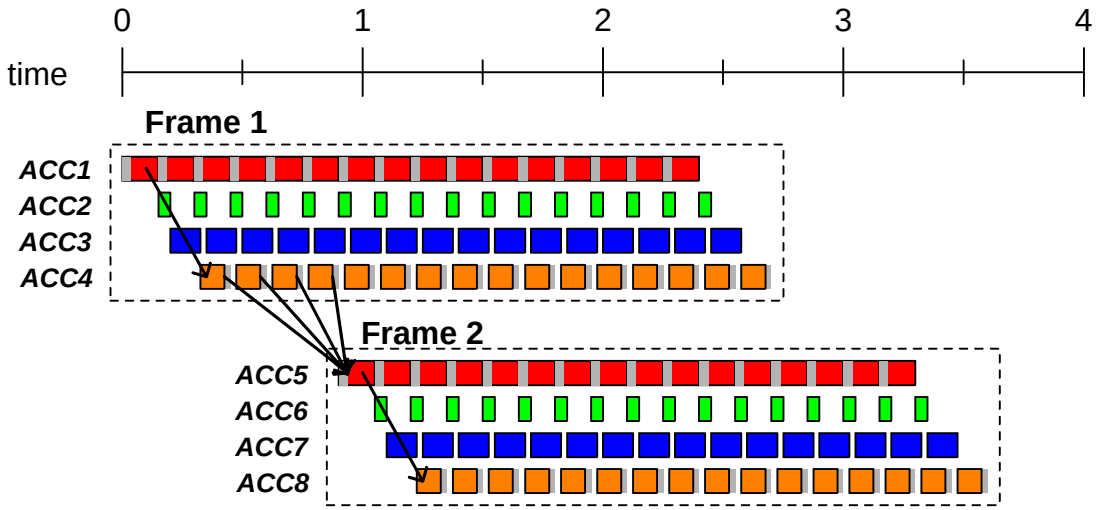


- Stage loops fused into single DOALL macroblock loop
  - Memory privatization reduces main memory access
- Replicates instructions and tables across processors
  - Local memory constraints may prevent this technique

Novel dimensions of parallelism reduce communication

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009

Pipeline or “Spatial Computing” Model

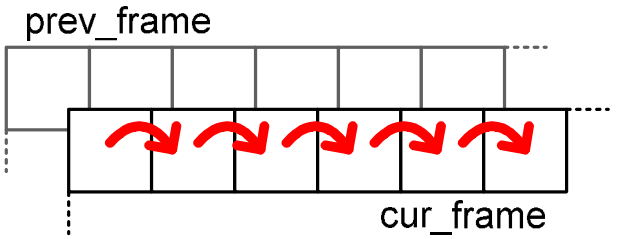


- Each PE performs as one pipeline stage in macroblock processing
- Imbalanced stages result in idle resources
- Takes advantage of direct, accelerator to accelerator communication
- Not very effective in CUDA but can be effective for Cell

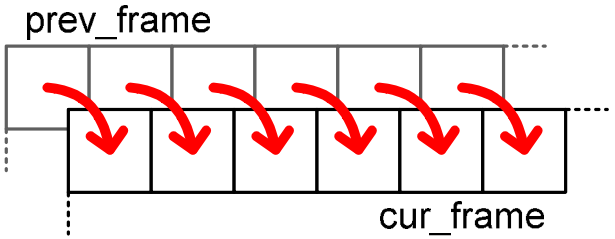
Efficient point-to-point communication can enable new models

Small decisions in algorithms can have major effect on parallelism.
(H.263 motion estimation example)

- Different algorithms may expose different levels of parallelism while achieving desired result
- In motion estimation, can use previous vectors (either from space or time) as guess vectors



(a) Guess vectors are obtained from the previous macroblock.



(b) Guess vectors are obtained from the corresponding macroblock in the previous frame.

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2009
ECE 498AL, University of Illinois, Urbana-Champaign