Session II

Technical Overview

Foundations and
Fundamentals of iWarp

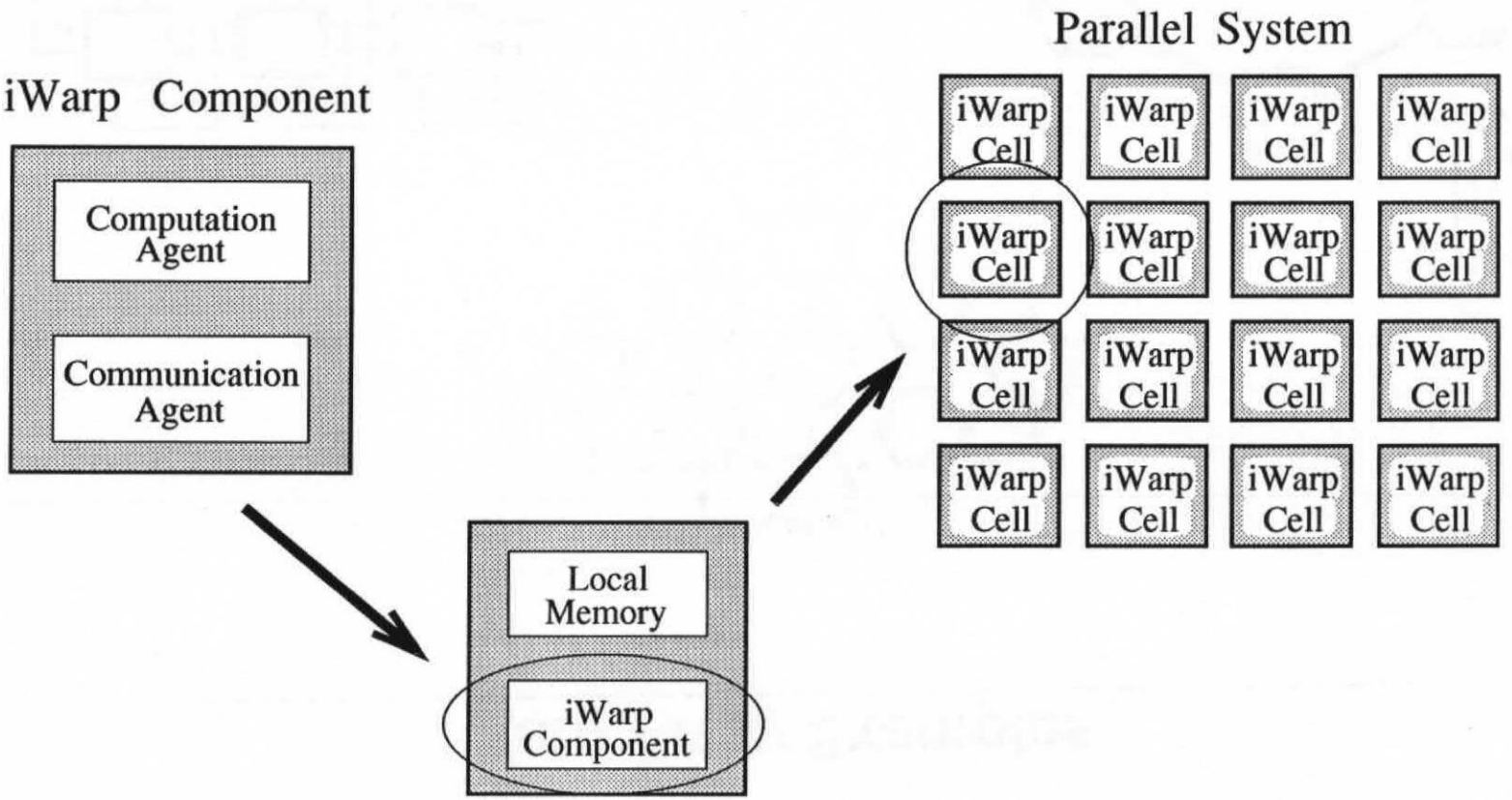
H.T. Kung
Carnegie Mellon University

Topics to Be Addressed in this Presentation

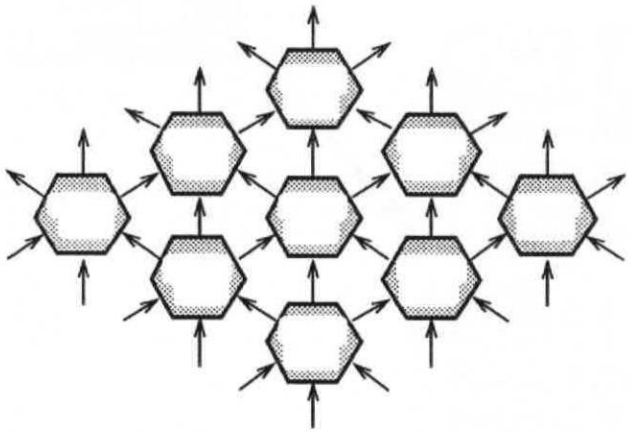
- **iWarp as a building block**
- **Application usage examples**
- **iWarp architecture and motivations**

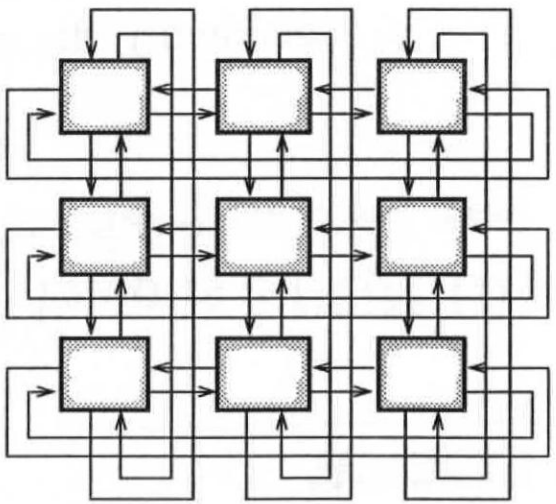
What Is Unique about iWarp ?

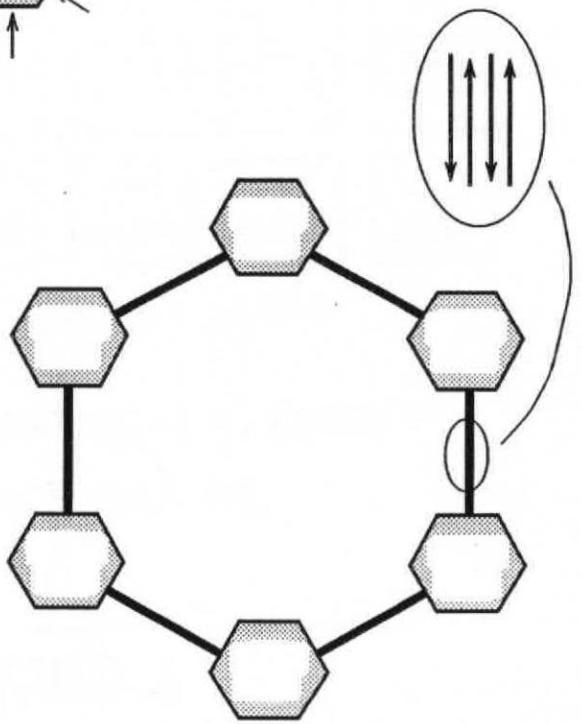
iWarp is a BUILDING BLOCK for a variety of special-purpose and general-purpose parallel systems.



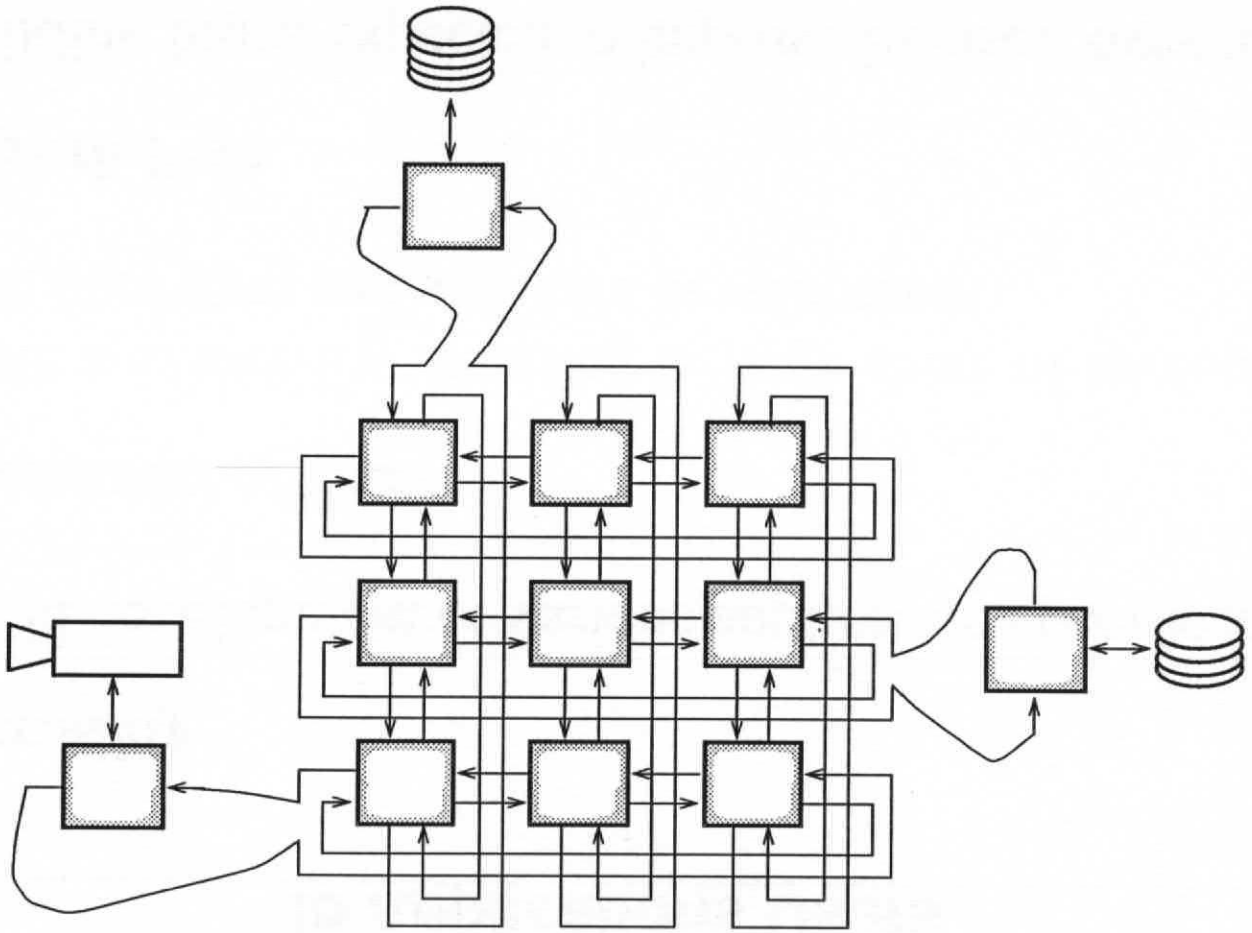
iWarp Array Examples







Direct Connections to Disks and Sensors



Benefits of the iWarp Building Block Approach to Applications Users

- ***Flexibility***

  Rapid development of various application specific systems

- ***Economies of Scale***

  Wide applicability resulting in large share of investment for both hardware and software development

- ***Growth Path***

  Building block expected to improve by itself over time

iWarp Is a Good Building Block.

1. **High Integration**

   600,000-transistor iWarp component with built-in computation and communication facility

2. **High Performance**

   Computation rate per cell: 20 MFLOPS and 20 MIPS
   Communication speed with neighboring cells:
   320 MBytes/sec bandwidth & 200 ns latency

3. **High Applicability**

   Supporting many computation and communication styles (see next slide)

iWarp Supports a Variety of Computation and Communication Models.

- **Flexible systolic communication and efficient message passing**
- **Static and dynamic scheduling of computation**
- **Reconfigurable arrays for fault tolerance**
- **. . .**

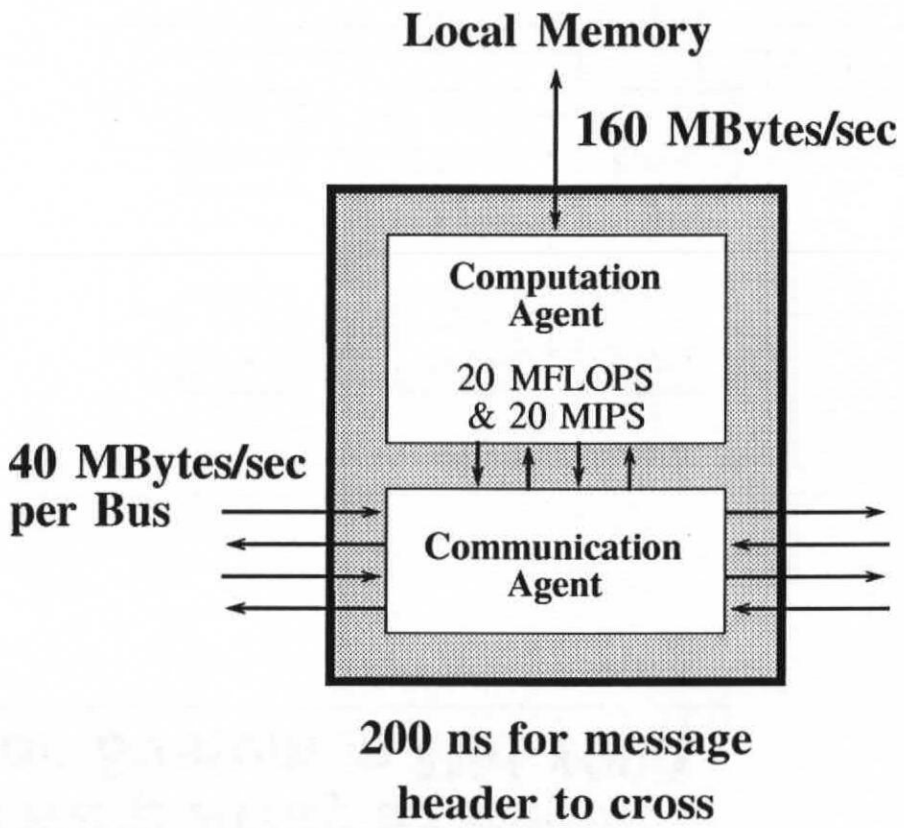
iWarp's implementation of these models will be discussed in the rest of the presentation.

The key is a powerful set of built-in intercell communication mechanisms in the iWarp.

iWarp Component

Both computation and communication hardware on the same chip:

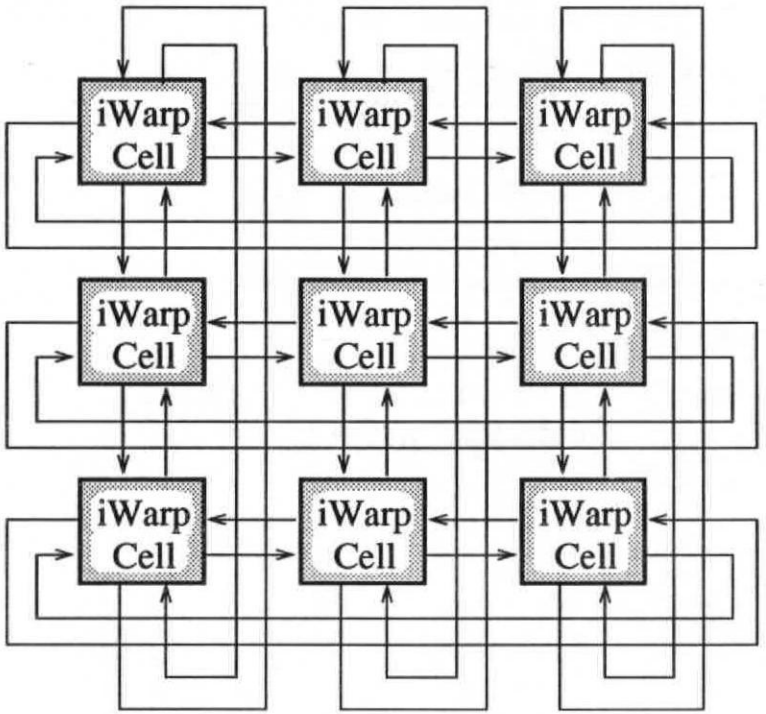
- **Separate control for easy programming**
- **Yet, close cooperation is possible.**



General-Purpose iWarp Array Example: iWarp Demonstration System in Mid-1990

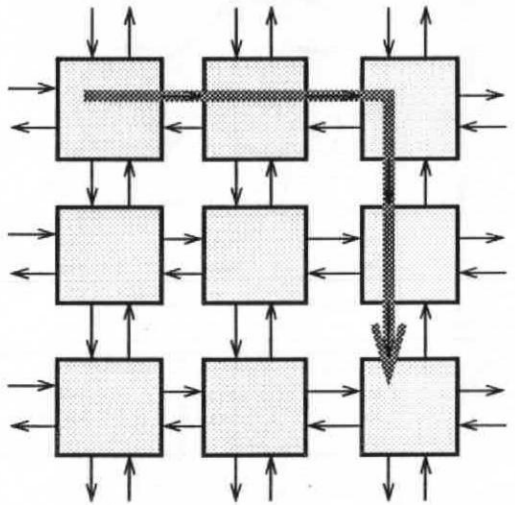
Three 1.28 GFLOPS, 8x8 torus systems at Carnegie Mellon

(System extendible to 20.48 GFLOPS, using 32x32 torus)



Why 2D Interconnection?

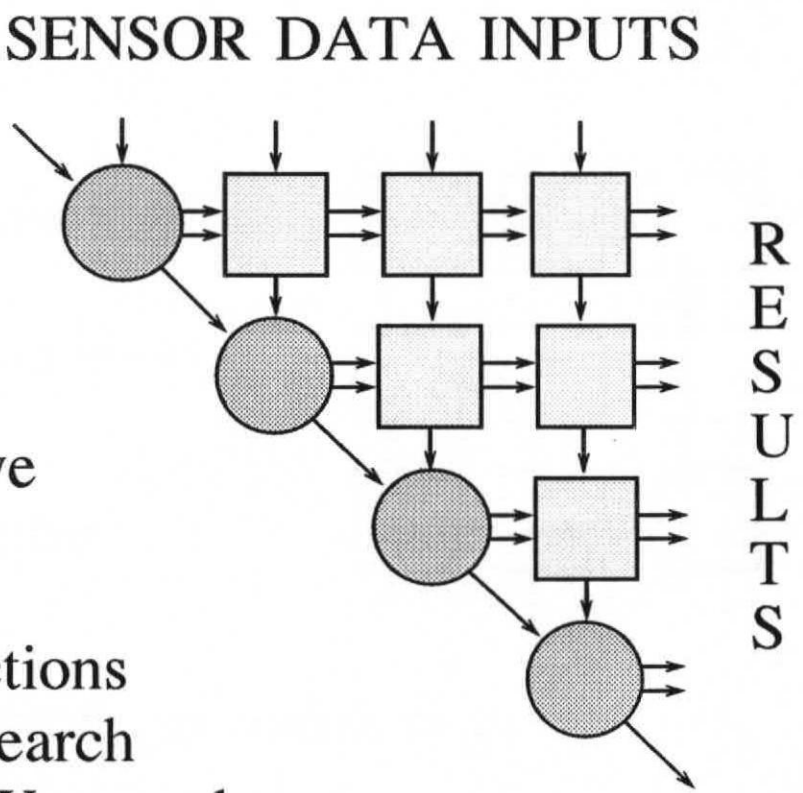
- **Better than 1D interconnection for many cells**
  - Fewer hops
  - Less intercell communication
  - More routing alternatives
  - More boundary cells for I/O ports
- **Better than higher dimension interconnections**
  - Higher bandwidth and lower latency wires
  - Higher system scalability
  - Easier to program



Special-Purpose iWarp Array Example I: Adaptive Signal Processing Arrays

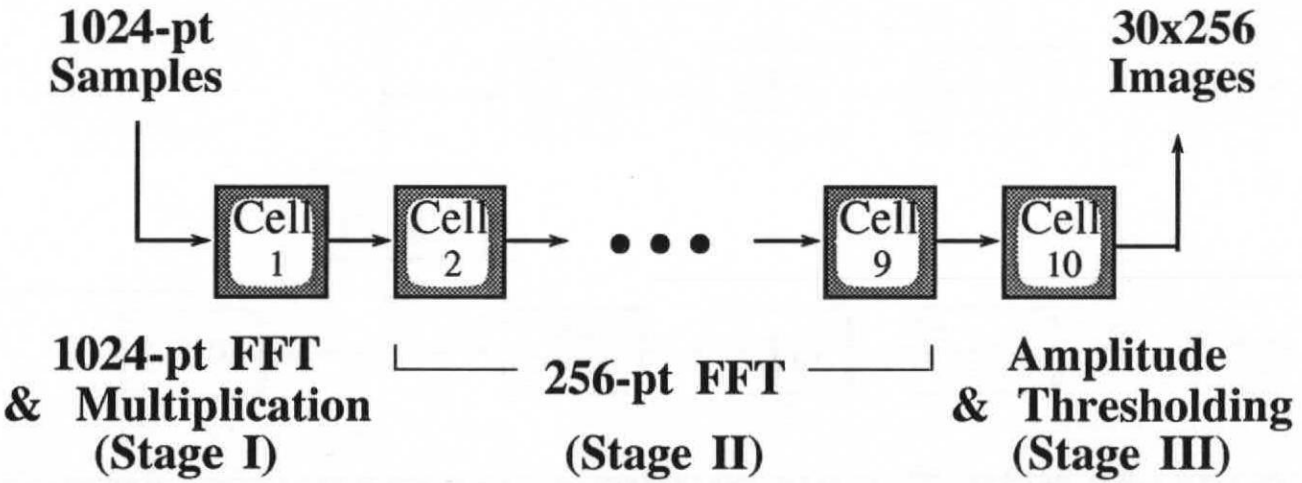
The processor array performs **on-the-fly** orthogonal triangularization on the covariance matrix formed from sensor inputs.
After the triangularization, adaptive weights can be easily computed.

Each iWarp cell performs the functions of one or more nodes. Recent research has shown that the mapping and iWarp code generation can be automated.



Special-Purpose iWarp Array Example II: Multi-Function Pipelines

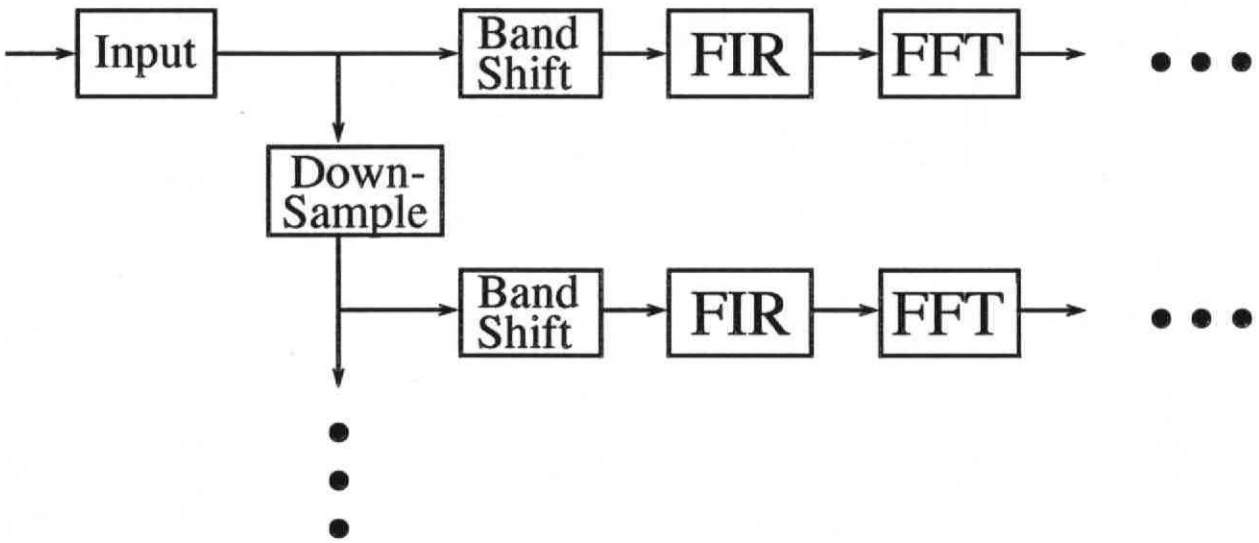
A typical radar simulation, which was implemented on a 10-cell Warp:



A dedicated set of iWarp cells implements each function stage of the pipeline.

Special-Purpose iWarp Array Example III: Heterogeneous Signal Flow Systems

In general, groups of iWarp cells can directly implement task nodes in a signal flow graph such as:

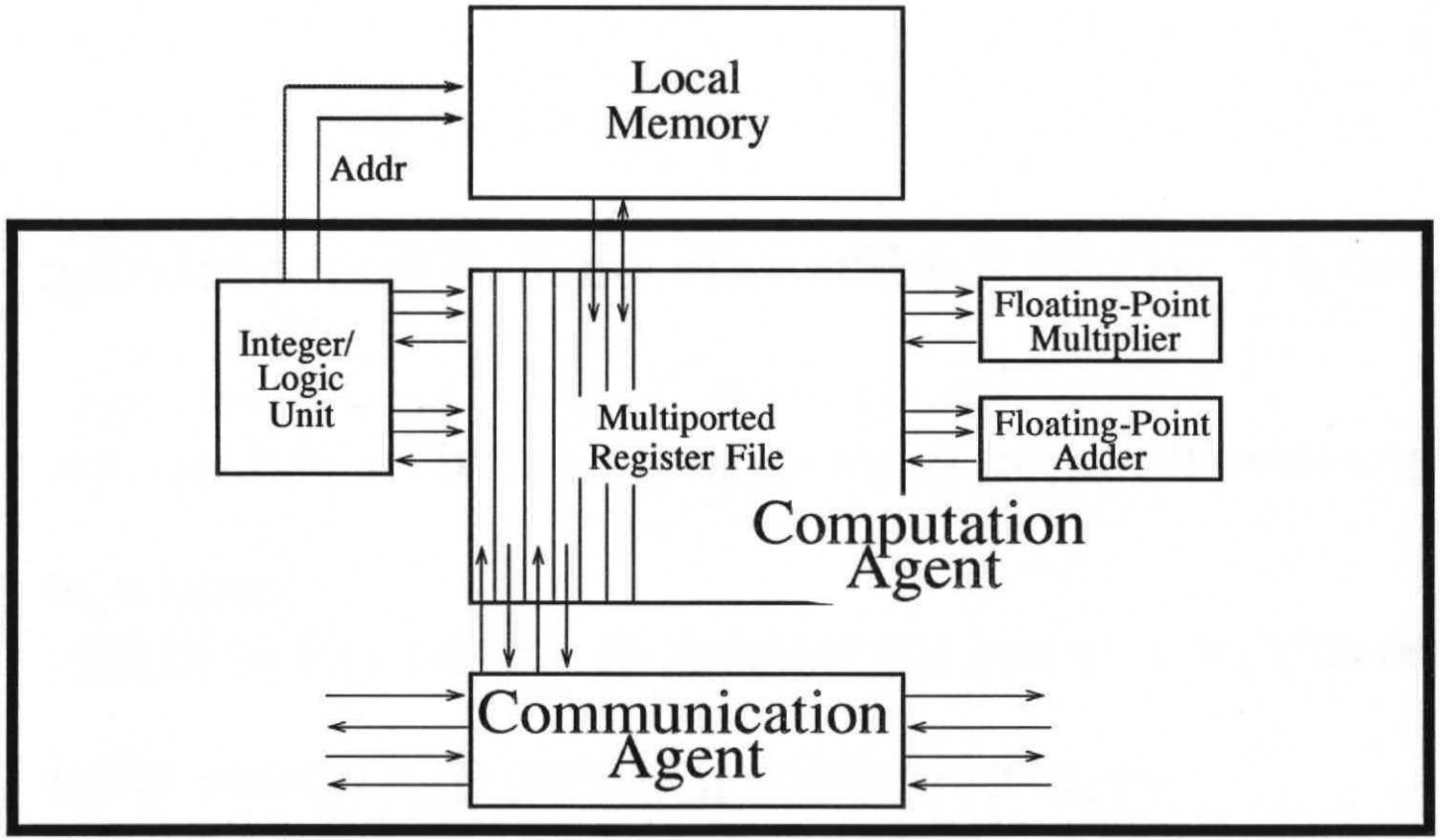


Recent research has shown the feasibility of automatic load balancing in mapping these tasks onto iWarp cells.

iWarp Support for Special-Purpose Arrays

- **High bandwidth intercell communication**
- **Multiple I/O busses to implement fan-in and fan-out of a node**
- **Hardware supported flow control between connecting cells to support heterogeneous computing**
- **Software tools to generate mapping and iWarp code**

iWarp Component Block Diagram



iWarp Component

Single-Cycle Wide-Instruction Architecture within Each iWarp Cell

Capable of performing many operations in one 100 ns cycle:

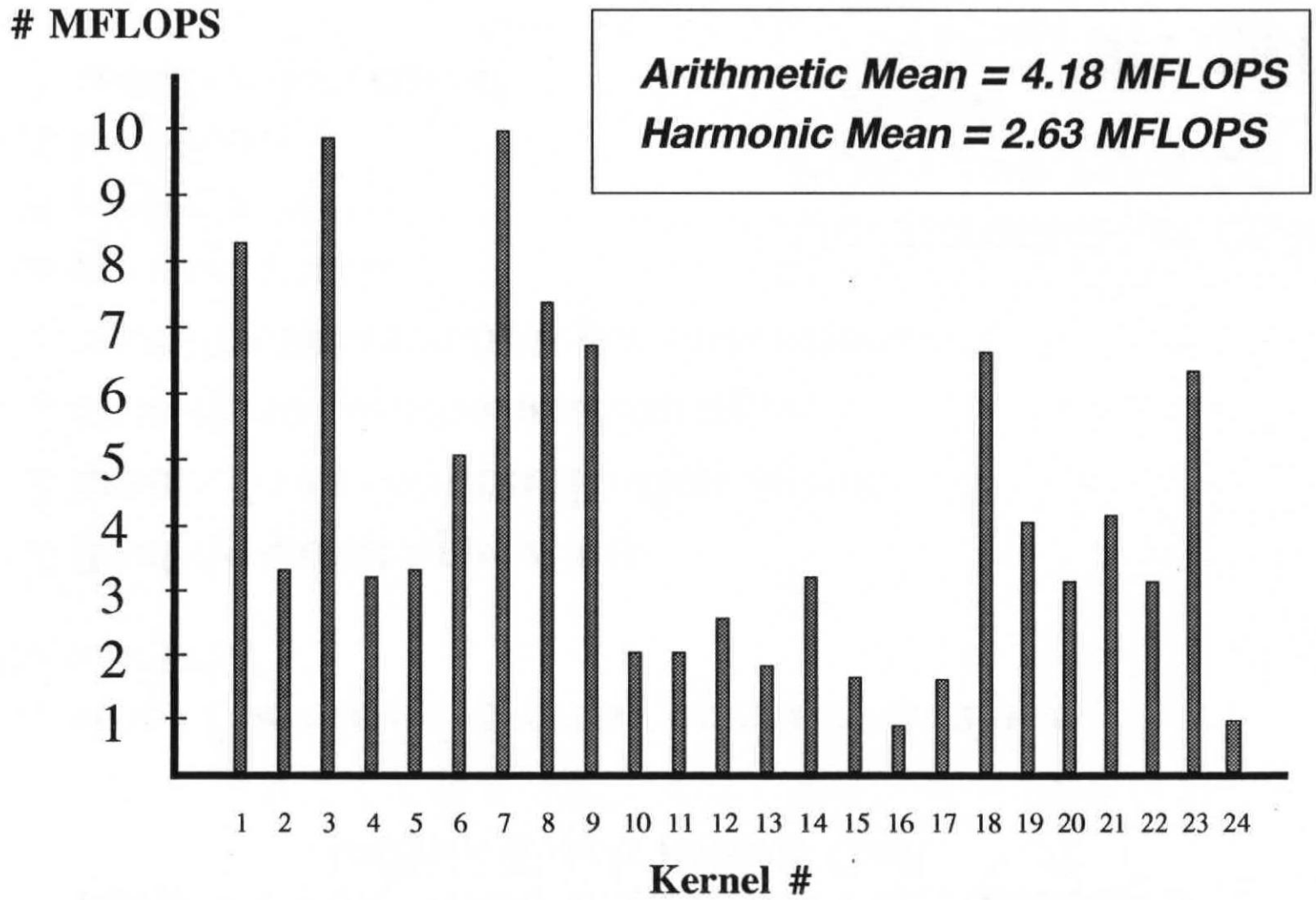
- **2 floating-point operations**
- **2 inputs from communication agent**
- **2 outputs to communication agent**
- **2 memory accesses/integer operations**
- **8 register reads**
- **5 register writes**
- **1 loop back**
- **1 counter decrement**

One iteration per cycle:

```
FOR i := 0 TO n DO
  BEGIN
  f := (A[[i]*B[3*i])+ f;
  END;
```

(How many MIPS or MFLOPS are there?)

Double Precision Performance of Livermore Loops on a Single iWarp Cell

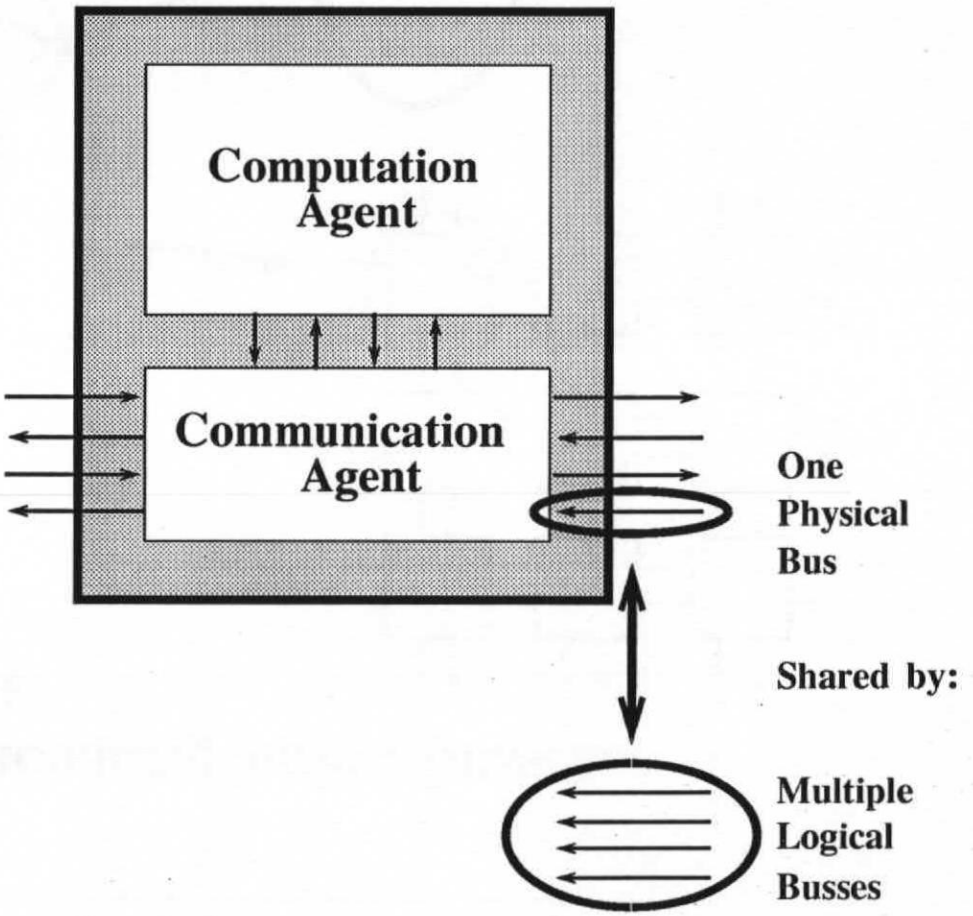


Physical and Logical Busses

Twelve 40 MBytes/sec "Physical Busses"

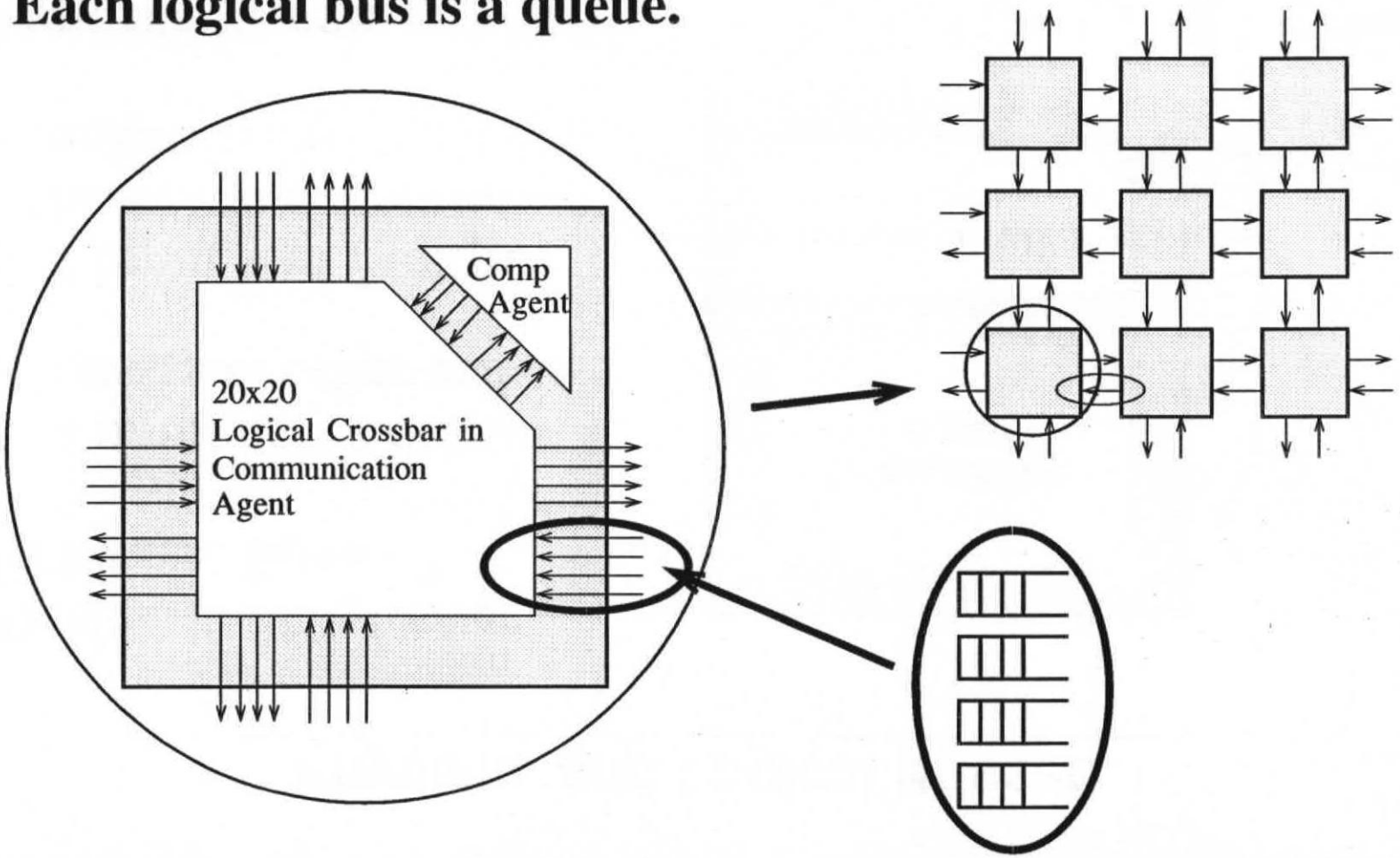
- **4 input and 4 output busses to other cells**
- **2 input and 2 output busses to computation agent**

Multiple "Logic Busses" can share a physical bus.



Logical Busses and Queues of a Cell

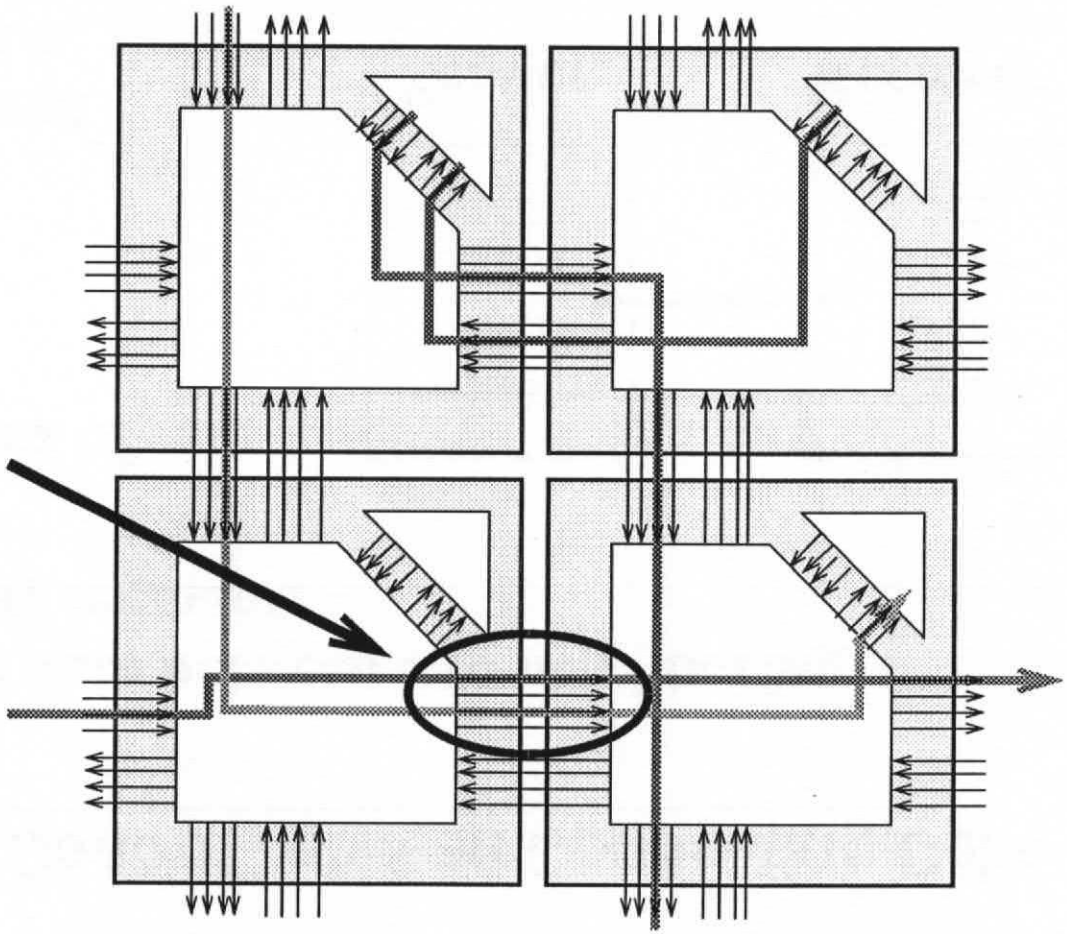
A cell can have up to 20 incoming logical busses.
Each logical bus is a queue.



Maintaining Multiple Pathways Simultaneously

A *pathway* is a sequence of connected queues over one or more cells.

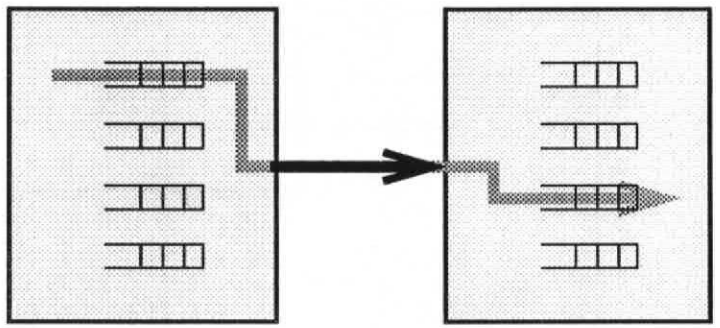
Two pathways over the same physical bus from lower left cell to lower right cell



Word-Level Flow Control between Connecting Cells

When transferring data between two neighboring cells over a pathway segment:

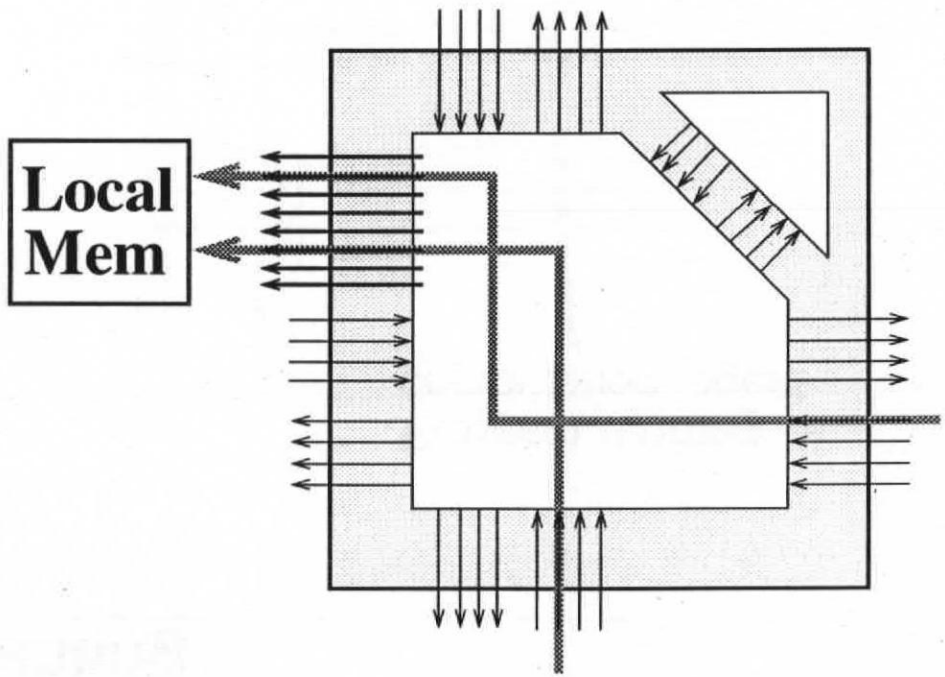
- **Sender keeps track of # empty slots in receiving queue.**
- **Receiver notifies sender when a word is dequeued from a queue.**
- **Round robin the physical bus between active pathways only.**



Spooling

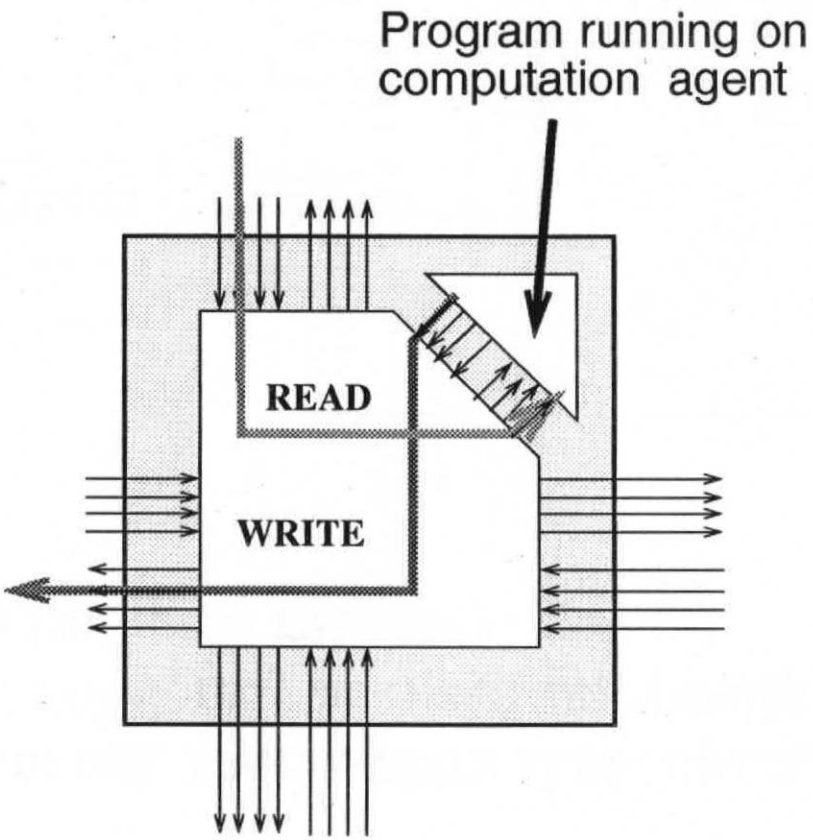
Spooling is a hardware supported mechanism that, upon the arrival of a message, can efficiently and automatically move it from the pathway to the local memory.

- **"Spooling gates" are dynamically bound to queues**
- **160 MBytes/sec total bandwidth in both directions**
- **No context switch cost**
- **Eight active spools**

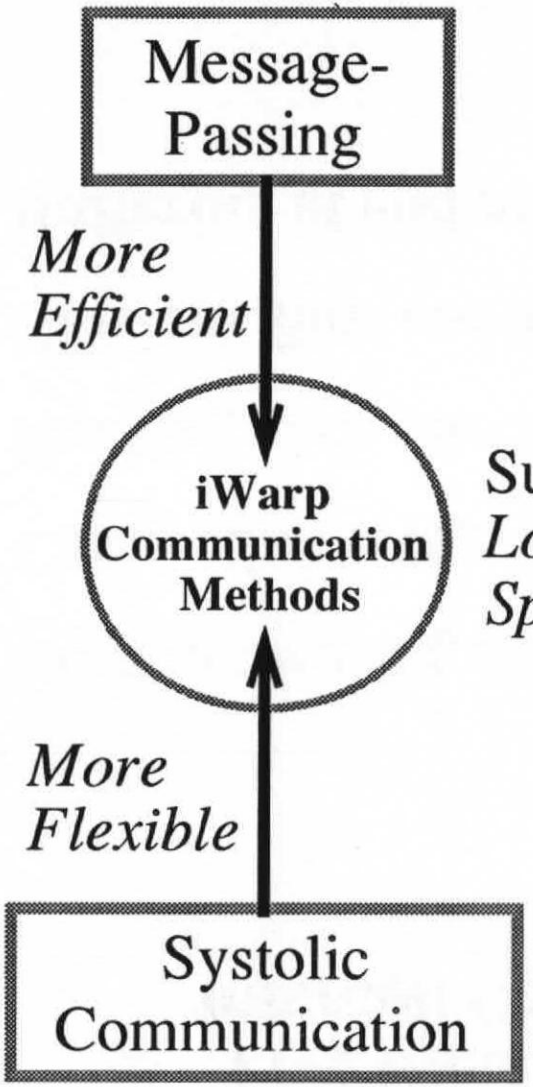


Streaming

Program can directly receive and send data from and to the communication agent, by reading and writing to the "stream gates," respectively.



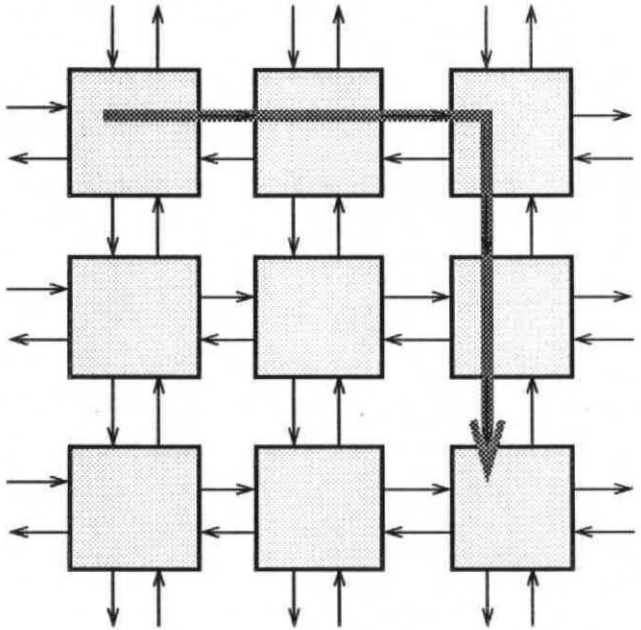
iWarp Supports Both Efficient Message-Passing and Flexible Systolic Communication



Supported by:
Logical Busses, Word-Level Flow Control, Spooling, Streaming, etc.

Five Dimensions for Classifying Intercell Communication Methods

1. **Communicating nodes**
2. **Connection setup**
3. **Routing path selection**
4. **Network flow control**
5. **Buffering at end nodes**

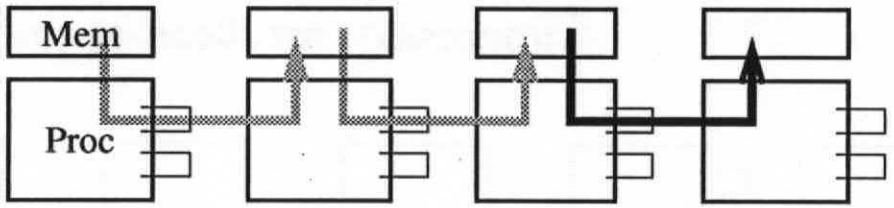


The first three dimensions are well known. The next two slides address the last two dimensions.

Network Flow Control

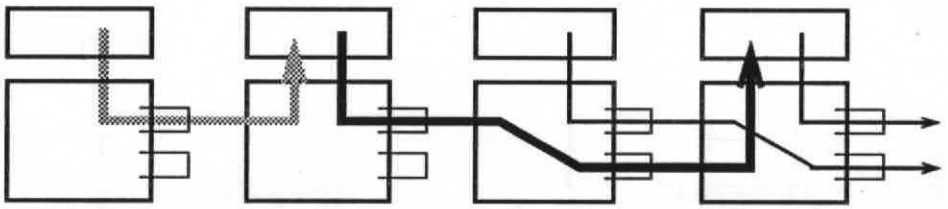
Store-and-Forward:

Message goes to Mem always.

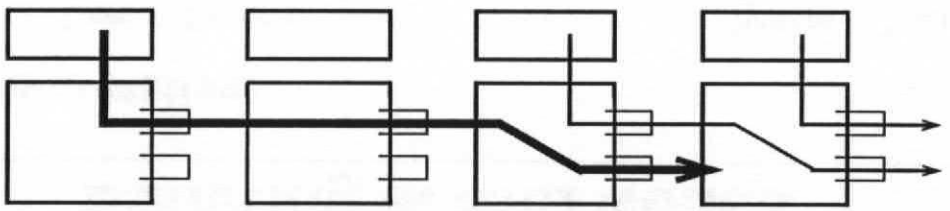


Cut-Through:

Message goes to Mem only when all output busses are congested.



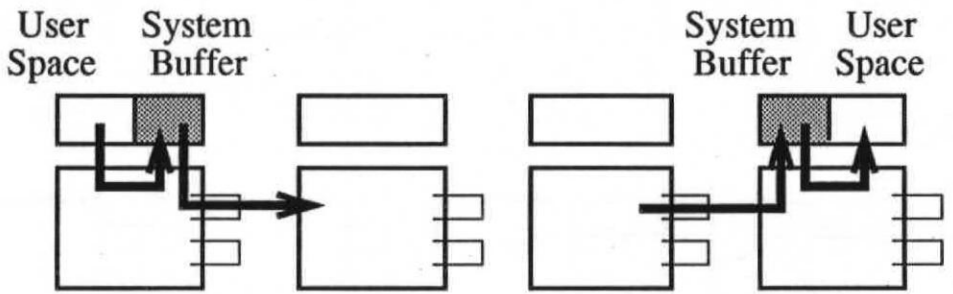
Wormhole:



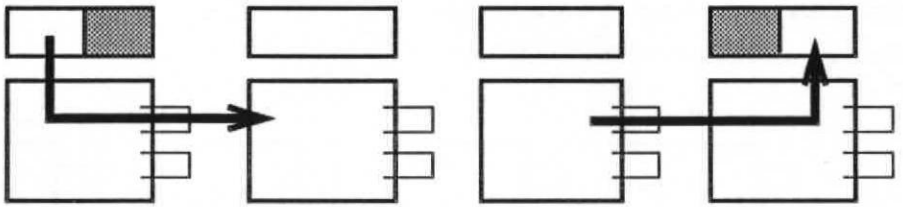
Message waits when all output busses are congested.

Buffering at End Nodes

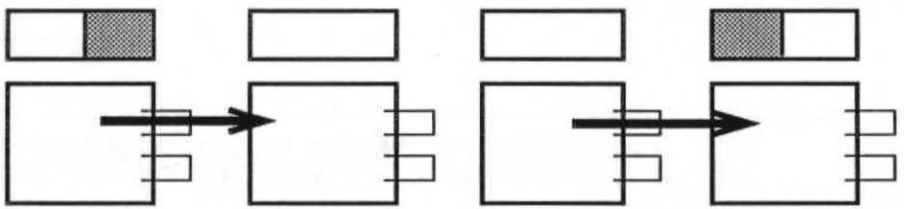
Station-to-station:



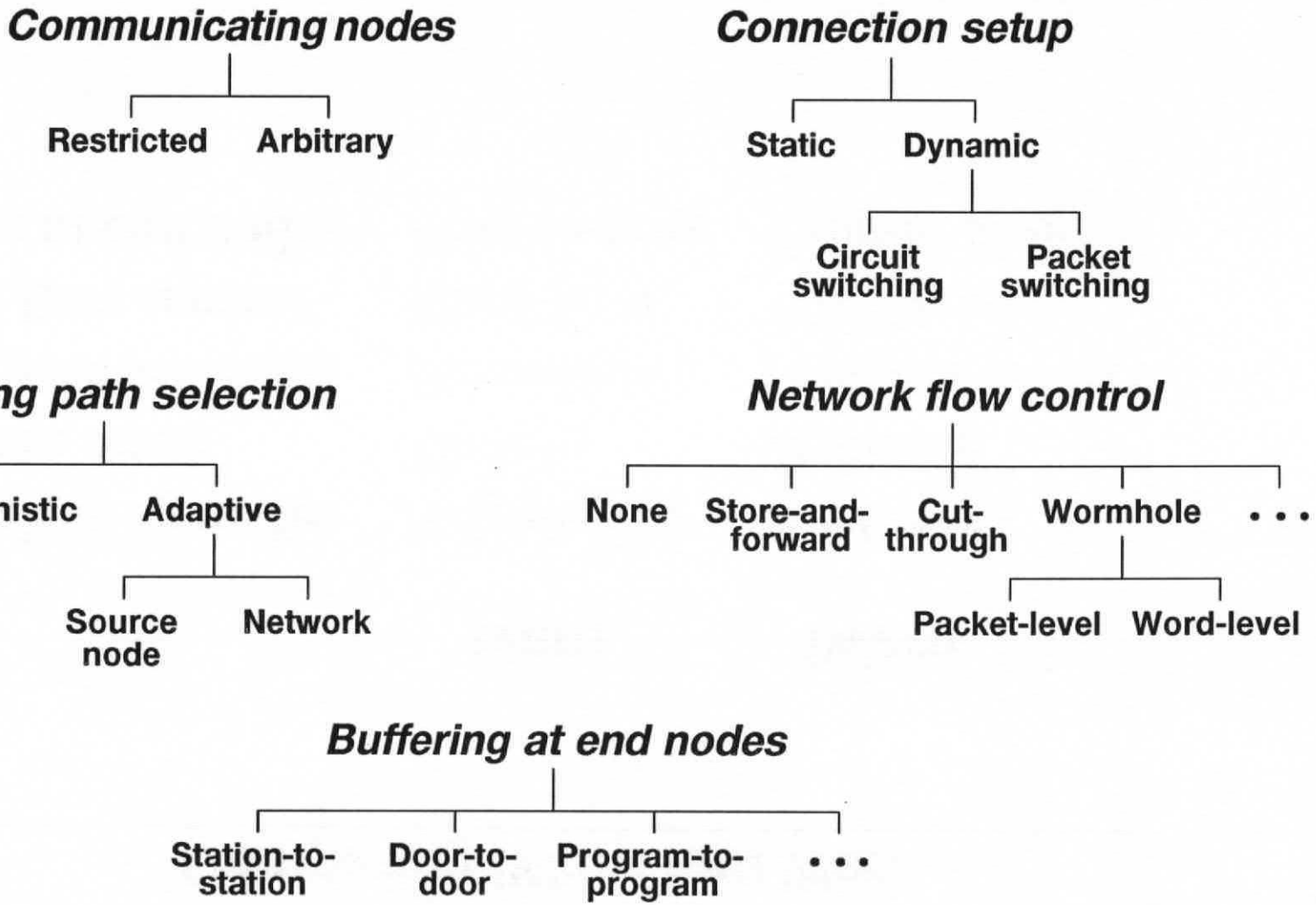
Door-to-door:



Program-to-program (systolic):



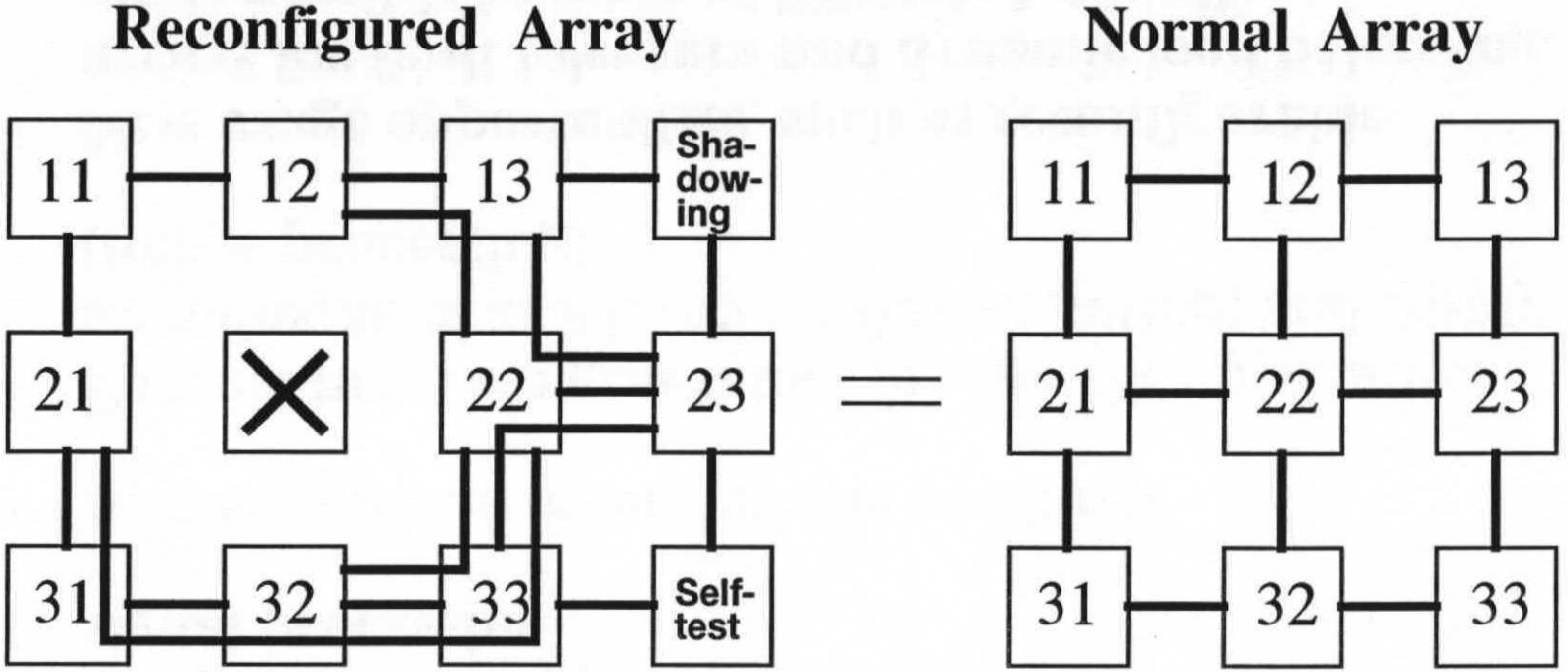
Taxonomy of Intercell Communication Methods (iWarp Supports Most of these Methods Efficiently!)



Comparing iWarp and Warp

	Warp	iWarp
Communicating nodes	*Neighboring*	***Arbitrary***
Connection setup	*Static*	***Dynamic***
Routing path selection	*Deterministic*	***Adaptive*** *(source node)*
Network flow control	*Word-level*	*Word-level*
Buffering at end nodes	*Prog-to-prog*	*Prog-to-prog*

Reconfiguration for Fault Tolerance: An Application of Logical Busses



- Using multiple logical busses between neighboring cells to implement required logical connections
- No need to change application programs for the reconfigured array

iWarp Usage Examples

- **Algorithmically specific processor arrays**
- **Large, scalable applications such as sonar or radar signal processing**
- **Applications on other parallel machines**
- **Parallel program generators for specific application domains, such as AL (scientific computing) and Apply (image processing)**
- **New usage opportunities, such as reconfigurable arrays for fault tolerance and dynamic load balancing, made possible because of iWarp's intercell communication capabilities**

Conclusions on the iWarp Approach

- **Powerful *building block* for a wide variety of high performance parallel computing systems, and to support many models of computation and communication**
- **Enabling technology for *algorithmically specific* processor arrays**
- ***System scalable* from several to thousands of cells, with *performance scalable* from 100s to 10,000s of MFLOPS**
- ***Fundamental solution* for achieving high performance computing with low costs**

Session II

Technical Overview

Meeting the Challenge in Silicon and Software

George Cox
Intel Corporation

Fundamental iWarp Design Objectives

- **1:1 Communication to Computation Ratio**
  - Fine-grain systolic
  - Near linear parallel speed-up
  - Effective use of memory and I/O
- **Independent Communication and Computation Elements**
  - Medium- and coarse-grain message passing
  - I/O can be easily and efficiently overlapped with computation
  - Heterogeneous tasks supported across the array
- **Maximize Scalar Performance**
  - Minimize pipelining
  - Greater computational efficiency from compiled code
  - Achievable peak performance for a wide range of algorithms

Meeting the Challenge

An Architectural View

- **Computational Support**
- **Communication Support**
- **Connecting to the Outside World**

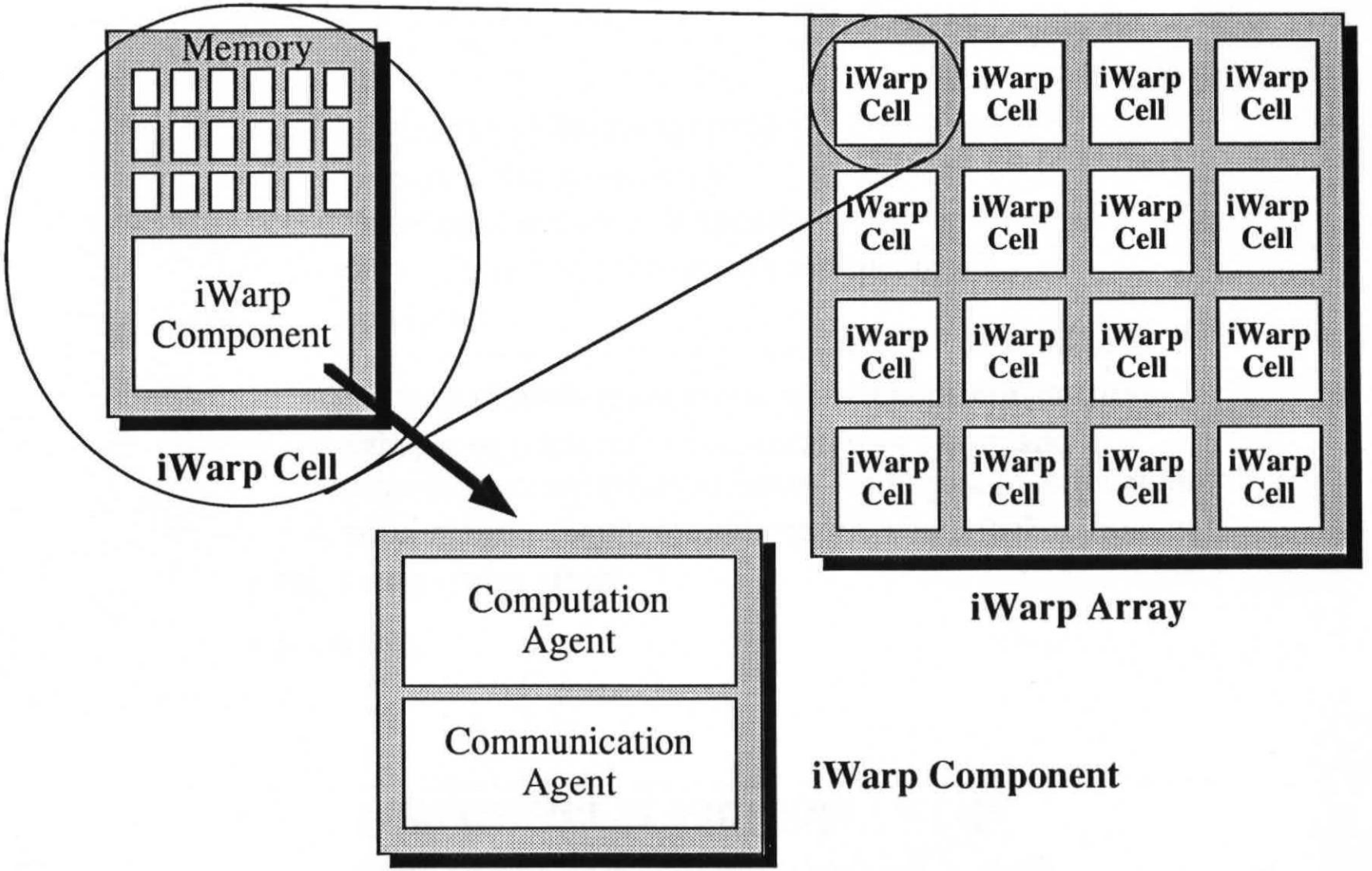
Computational Support

- **Fast Individual Cells -- 20 (40) MHz nominal clock rate**
- **Parallelism at Multiple Levels**

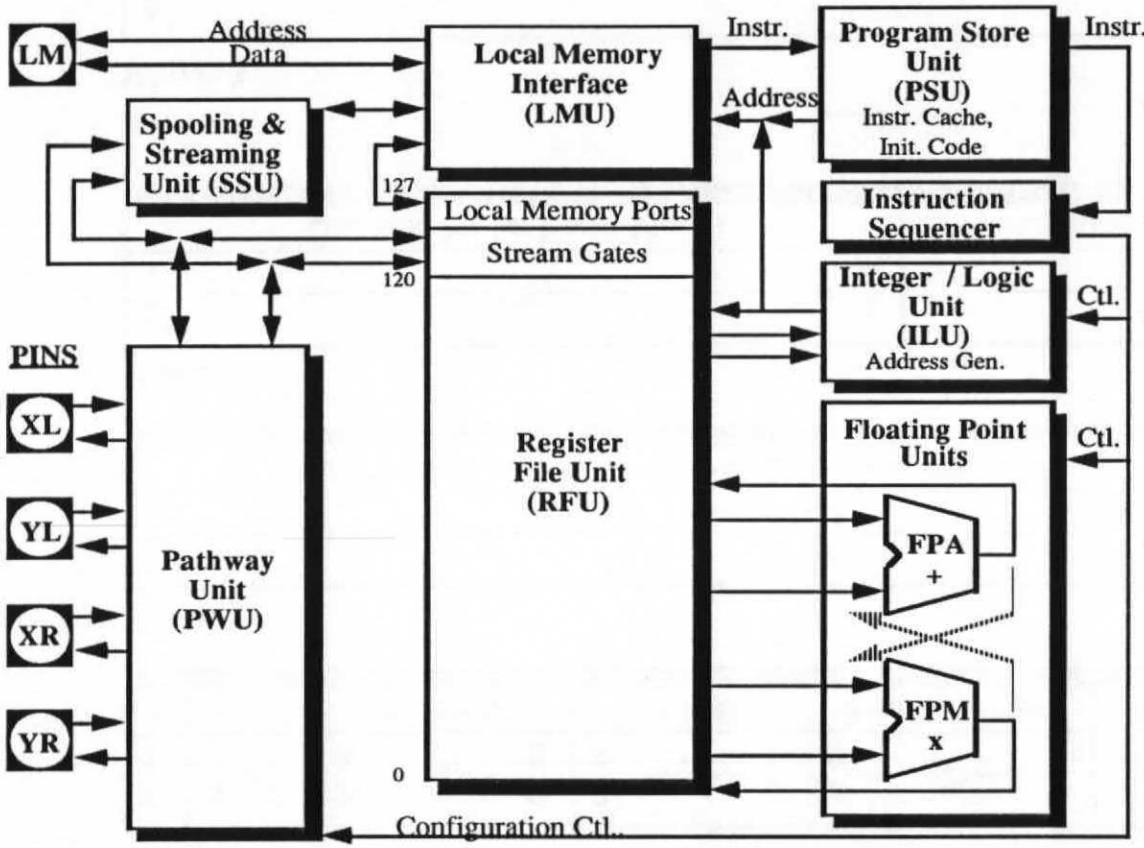
Parallelism at Multiple Levels

- **Inter-cell**
- **Intra-cell (Inter-unit)**
  - computation agent/communication agent (asynchronous)
  - intra-instruction (synchronous)—multiple functional units capable of operating concurrently—LIW-C and A
  - implicit looping (synchronous)—concurrent hardware evaluation of loop termination criteria—count and/or condition
  - inter-instruction (synchronous) scheduling
    - memory access sequencing
    - instruction sequencing
    - operand bypass/writeback

System Level Parallelism



Cell Level Parallelism



- Instruction access
- Data access
- Instruction decode
- Address calculation
- Floating-point add & multiply operations

- Access to pathway
- Access to multiported register file

Compute and Access Instruction

Word-1

31	30—29	28—(4)—25	24—(4)—21	20—(7)—14	13—(7)—07	06—(7)—00
J	1 1	Data Mode	FADD	B operand Reg	A operand Reg	K operand Reg

Word-2

31—(9)—23	22—21	20—(7)—14	13—(7)—07	06—(7)—00
Memory Control	FMUL	M operand Reg	N operand Reg	R operand Reg

Word-3

31—(16)—16	15—(16)—00
Operand for 1st Read Access	Operand for 2nd Read / Write Access

—OR—

Word-3

31—(32)—00
Full ILU (Integer Logical Unit) Instruction

Communication Support

- **Efficient Software Interface at Source and Destination**
- **No Software Involvement at Intervening Cells between Source and Destination**

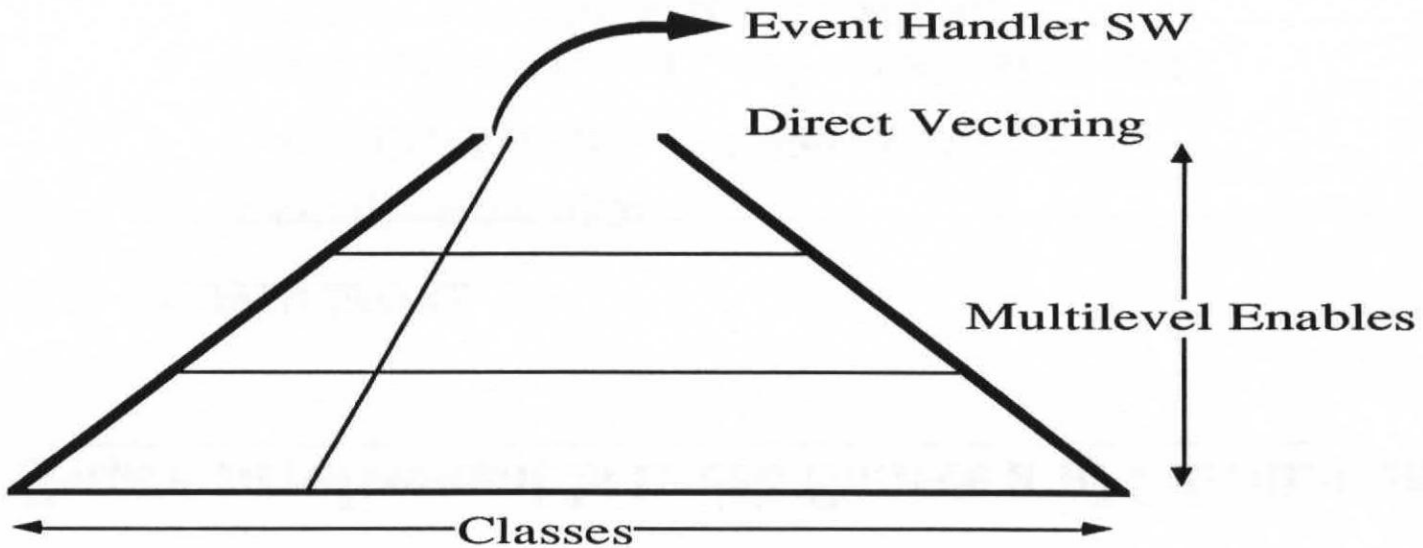
Efficient Software Interface At Source and Destination

Integrated Communication

- **Program Driven Streaming (aka "systolic" as a side effect of register assignment)**
- **Spooling (aka "DMA")**
- **Directly Visible Communication State**
  - explicit testing (via condition codes)
  - implicit testing (via events)
- **Late Resource Binding—gates, buffers (flexibility, efficient resource usage)**

Event Support

- **Uniform Model**
  - synchronous: faults/traps
  - asynchronous: interrupts
- **Event Classes**
- **Hierarchical Event Capture/Reporting (enables at each level)**
- **Direct Vectoring (low overhead getting to handler)**



No Software Involvement between Source and Destination

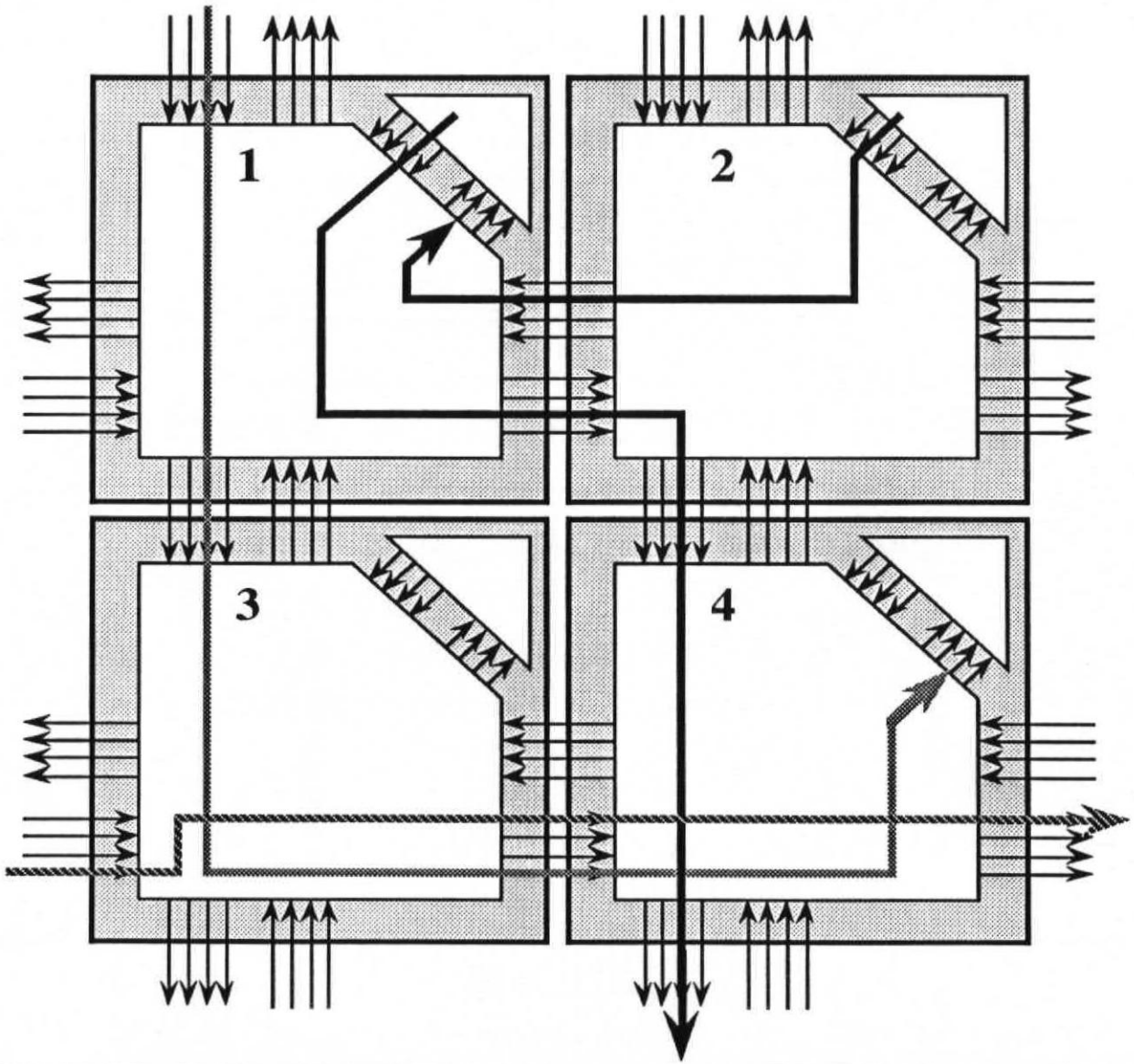
Express Service

- **Routing—streetsigns**
- **Congestion Avoidance—logical pathways**
- **Sophisticated Physical Level—longitudinal and lateral control information on pathways**

Routing

- **Streetsign routing for 2D torus**
  - "Go to Forbes, Turn Left"
    "Go to Broadway, Stop", etc.
  - One pair of "street name" and "action" in the message header for each routing step.
- **Timing**
  - 200 nsec. to pass through a cell (250 nsec. if corner turning).
  - Setting up a new pathway incurs no delay.

Logical Pathways



Connecting to the Outside World

Hosts and/or Native I/O

- **Examples:**
  - mass storage
  - communication
  - displays/video
  - sensors
- **As easy as with normal microprocessors—only higher bandwidth and lower latency potential**
- **Approaches:**
  - Memory mapped bus interface, or
  - Shared, dual ported memory

iWarp in a Nutshell

- **Modular Architecture**
  - Supports 1-d and 2-d expandable systems
- **1-GFLOP per cu. ft. computation density**
  - Commercial, air cooled package
- **Scalable performance from 20 to 20,000 MFLOPS**
- **Scalable I/O capacity with computation power**
  - Meets essential needs of Signal and Image Processing
- **Programmable systolic systems**
  - Break the "special purpose" limitations of early systolic systems
  - Support a range of computation-communication models:
    - fine-grain systolic based
    - coarse-grain message based