

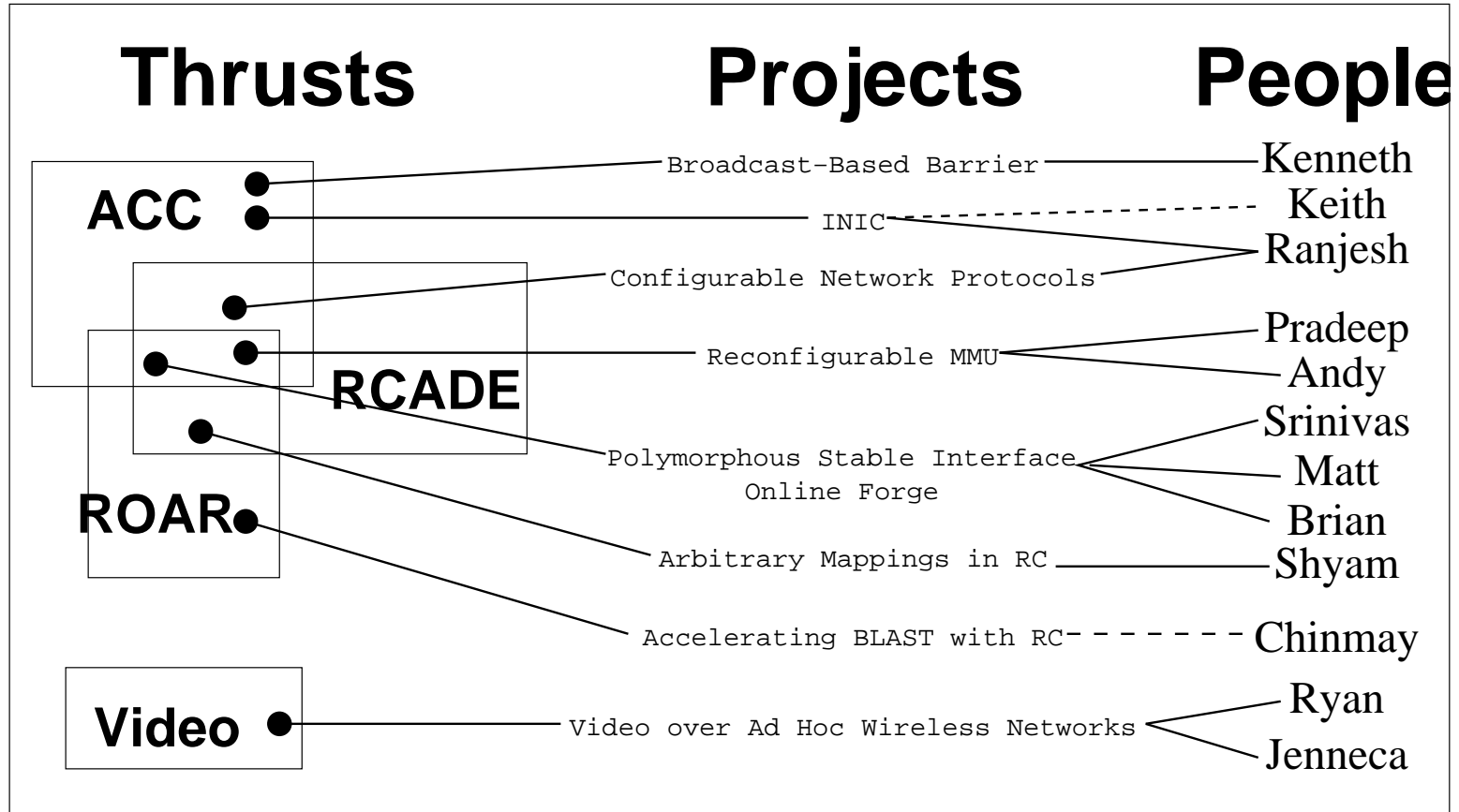
# Reconfigurable Computing — Spring 2002

- ⇒ Research Issues
  - Three Major Thrusts (one ancillary)
  - Eight Projects
  - Eleven Supported Students
- ⇒ Scheduling
- ⇒ Lab Status
- ⇒ Goals

## Thrusts

- ① Adaptable Computing Cluster (ACC)
- ② Reconfigurable Computing Application Development Environment (RCADE)
- ③ Reconfigurable Online Architectures Research (ROAR)
- ④ Video over *Ad Hoc* Wireless Networks (Video)

# Big Picture



## Supported Students

- ⇒ Ranjesh (MS Thesis)
- ⇒ Pradeep (MS Thesis)
- ⇒ Shyam (MS Thesis)
- ⇒ Chinmay (MS Thesis)
- ⇒ Srinivas (MS Thesis)
- ⇒ Ryan (MS Thesis)
- ⇒ Andy (Honors Thesis)
- ⇒ Brian (Honors Thesis)
- ⇒ Kenneth (Honors Thesis)
- ⇒ Jenneca (SURE)

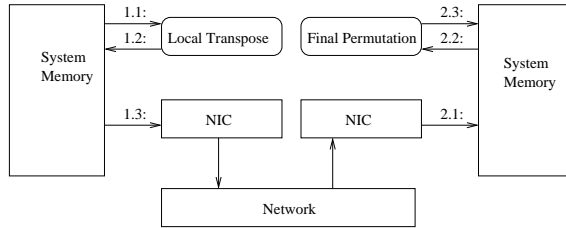
## New Students

- ⇒ Sudhanshu (MS Thesis)
- ⇒ N.T.?
- ⇒ Krishna?
- ⇒ Amol?

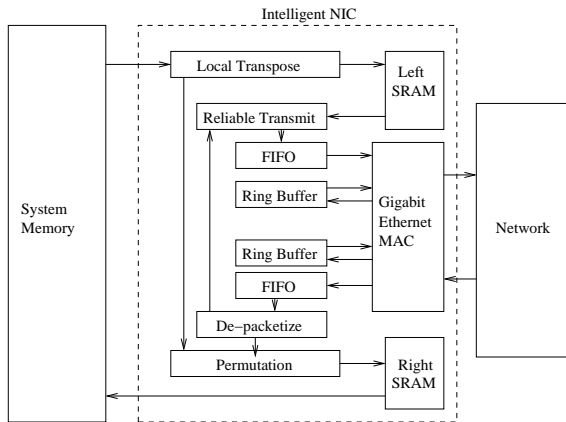
## Adaptable Computing Cluster

- ⇒ place RC in network interface, forming an INIC, to make a cost-effective, performance enhancement to typical Beowulf cluster
- ⇒ Keith has shown the feature to have a performance gain and cost-effective
- ⇒ next big question: how to make the feature accessible?
  - Keith, Kenneth: provide library of features (such as barrier synchronization)
  - Ranjesh: make configurable network protocols

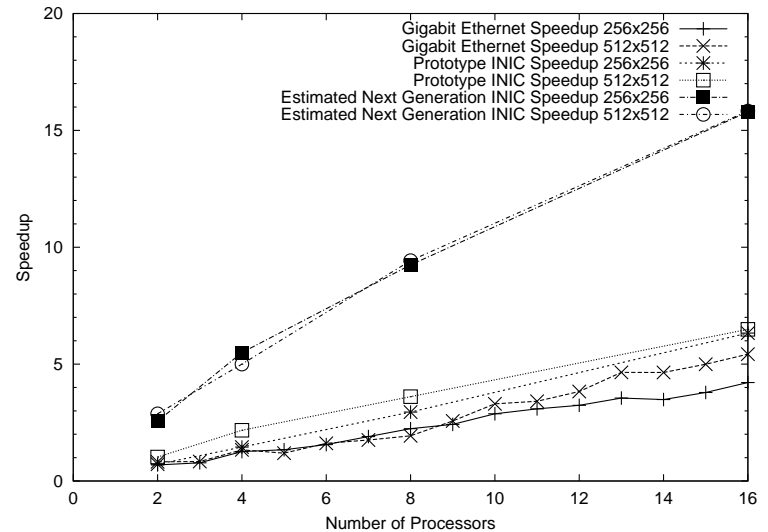
# Adaptable Computing Cluster



**(a)** standard parallel transpose algorithm



**(b)** INIC implementation



**(c)** 2D-FFT speedup

## RCADE, Take Two

- by implementing RC-specific engineering expertise into *agents*, a RC problem solving environment can be used to make
  - ❑ RC more accessible to software programmers
  - ❑ automate menial dataflow-tasks for hardware engineers
  - ❑ foster collaboration between hardware and software folks
- Previous RCADE had agents in development: pipeline balancing, throughput analysis, spatial partitioning, precision analysis, placement, and code generation – concept established but not extensible enough and branched from original development environment

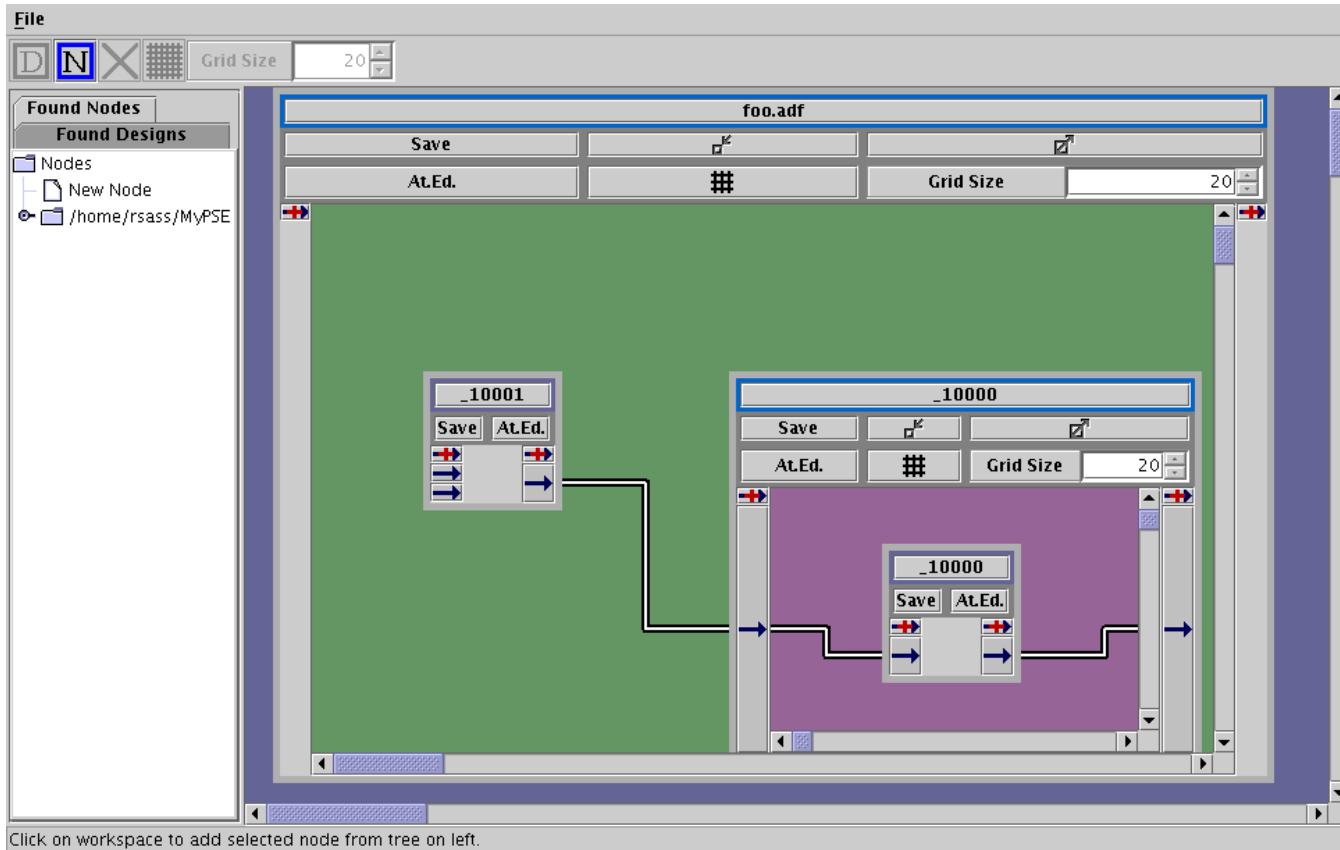
## RCADE, Take Two (cont'd)

⇒ new/ported agents:

- ❑ Andy: porting precision tool
- ❑ Pradeep: reconfigurable memory-access and cache controller
- ❑ Ranjesh: configurable network protocols
- ❑ Shyam: arbitrary mapping function units



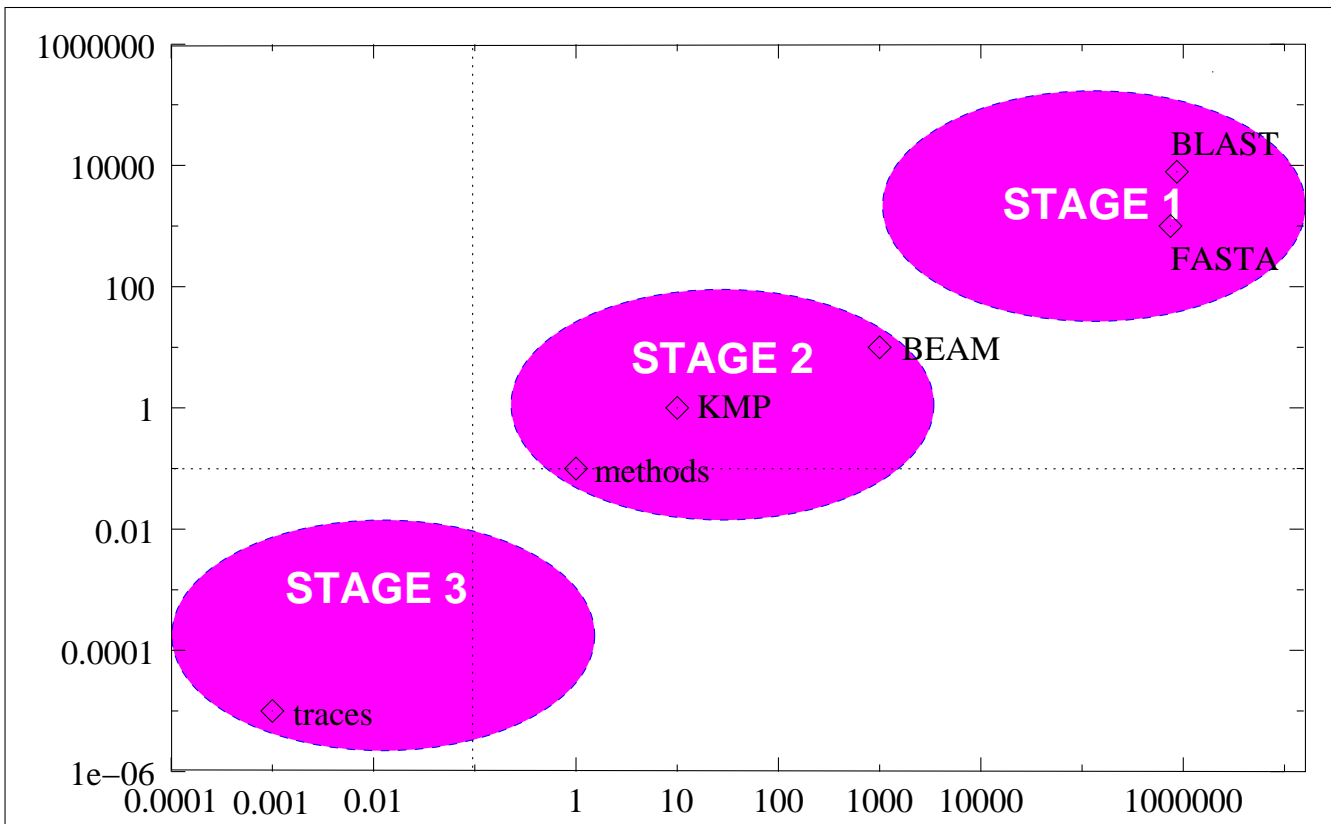
# RCADe, Take Two (cont'd)



## Online Reconfiguration

- ⇒ late specialization is hardware feature unique to RC; by adapting the hardware over time gives a size, performance, and power advantage over existing systems while addressing the programming issue
- ⇒ largely “in development”, we have not established any solid evidence
- ⇒ Shyam: arbitrary mapping function units
- ⇒ Chinmay: accelerating BLAST
- ⇒ Srinivas: Forge in a Java environment
- ⇒ Brian, Matt, Srinivas: interfacing (changing) RC hardware to an application

# Stages



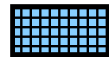
## Project Summaries

⇒ title, link to detailed pages

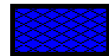
⇒ key to project status



idea development



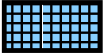





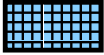
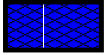
building infrastructure



conducting experiments

⇒ students

## Project Summaries (cont'd)

- Reconfigurable MMU  Pradeep, Andy
- Configurable Network Protocols  Ranjesh
- Broadcast-Based Barrier  Kenneth
- Video over Ad Hoc Wireless Networks  Ryan
- Accelerating BLAST with RC  *unmanned*
- Polymorphous Stable Interface  /  Srinivas, Matt, Brian
- Arbitrary Mappings in RC  Shyam

Fall 2002 Schedule

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
7:00am-					
8:00am-	class prep		class prep		class prep
9:00am-					
10:00am-		RC Mem. Controler		Config. Net. Proto.	
11:00am-					
noon-	ECE 493/693 RIGGS 227		ECE 493/693 RIGGS 227		ECE 493/693 RIGGS 227
1:00pm-	ECE 852 RIGGS 227	lunch	ECE 852 RIGGS 227	lunch	ECE 852 RIGGS 227
2:00pm-	lunch	ACR/RCADE	lunch	RC Interface	lunch
3:00pm-	Office Hours RIGGS 313-C		Office Hours RIGGS 313-C		
4:00pm-	Shyam	Ryan			
5:00pm-					
6:00pm-					
7:00pm-					Happy Hour
8:00pm-	Outing Club				
9:00pm-					

## Scheduling

- ⇒ weekly meetings – start August 26th
  - Monday 4:00pm – Shyam
  - Tuesday 10:00am – Pradeep, Andy
  - Tuesday 1:30pm – RCADE/ACR(Nasa)  
Ranjesh, Pradeep, Alan
  - Tuesday 4:00pm – Ryan
  - Thursday 10:00am – Ranjesh, Kenneth
  - Thursday 1:30am – PCA Interfacing  
(Srinivas, Brian, Matt)
- ⇒ Supercomputing 11/15-11/22

## /projects Organization

- ⇒ all machines should have automount working
- ⇒ currently five projects defined...
  - ❑ /projects/rcc
  - ❑ /projects/pse
  - ❑ /projects/syssoft
  - ❑ /projects/fusion
  - ❑ /projects/video
- ⇒ note on linux systems, 'unseen until mounted...'



## Riggs 309 Lab Status

- workstations: rajah, rabbit, kanga, harthey, tigger, baggy, kah, mowgli, plunder
- login servers:
  - ❑ riggs.parl.clemson.edu
  - ❑ eib.parl.clemson.edu
- merge between EIB/Riggs in near complete
  - ❑ login to any machine
  - ❑ public\_html works
  - ❑ .forward works
- plunder has a disk again
- wireless available — need to register your MAC address

## Goals

- ⇒ NASA Alternative Computing Roadmap (ACR) project
  - September 17th is new launch date
  - December — computation model
- ⇒ RCADE
  - demo at Supercomputing
  - FCCM submission
- ⇒ ROAR
  - Shyam's results — this fall
  - BLAST?

## Goals (cont'd)

### ⇒ Proposals

- ❑ 11/05 – NSF/CISE proposal deadline for distributed systems and compilers: RCADE as a mechanism for programming INICs in a heterogeneous ACC
- ❑ 12/04 – NSF/CISE proposal deadline for computer systems architecture: using late binding in object-oriented systems to as stable interface to polymorphous architectures
- ❑ ???? – NASA/AIST proposal: open source/configware collaborative design environment

## Upcoming Conferences

- ⇒ FCCM - Decemeber
- ⇒ FPT - July
- ⇒ ICS - Feb 15th
- ⇒ FPL - March 15th
- ⇒ MAPLD - May 28th
- ⇒ FPGA - September
- ⇒ Workshop on Communication Architecture for Clusters - October
- ⇒ IPPS/IPDS

◀ back

forw ▶

*Fini*

PARL

21

RCC

# Reconfigurable MMU



⇒ goals:

- ❑ automate common task
- ❑ improve application performance
- ❑ online adaptation ★
- ❑ an RCADE tool for programming INICs

⇒ Pradeep K. Nalabalapu, Andy Chester

# Reconfigurable Cache for Image Processing Applications

## ⇒ Main Idea

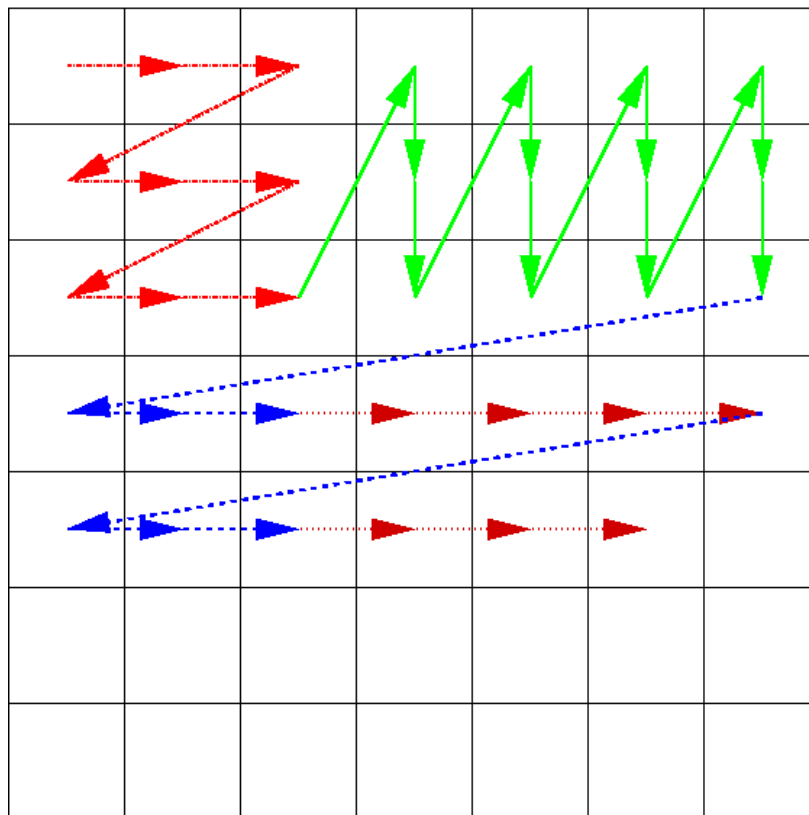
- ❑ Image Processing applications have high spacial and temporal locality
- ❑ Reconfigurable Cache custom built for an application exploits locality to max
- ❑ Prefetching if done can improve speed by hiding memory latency
- ❑ Compiler can derive memory access information from program

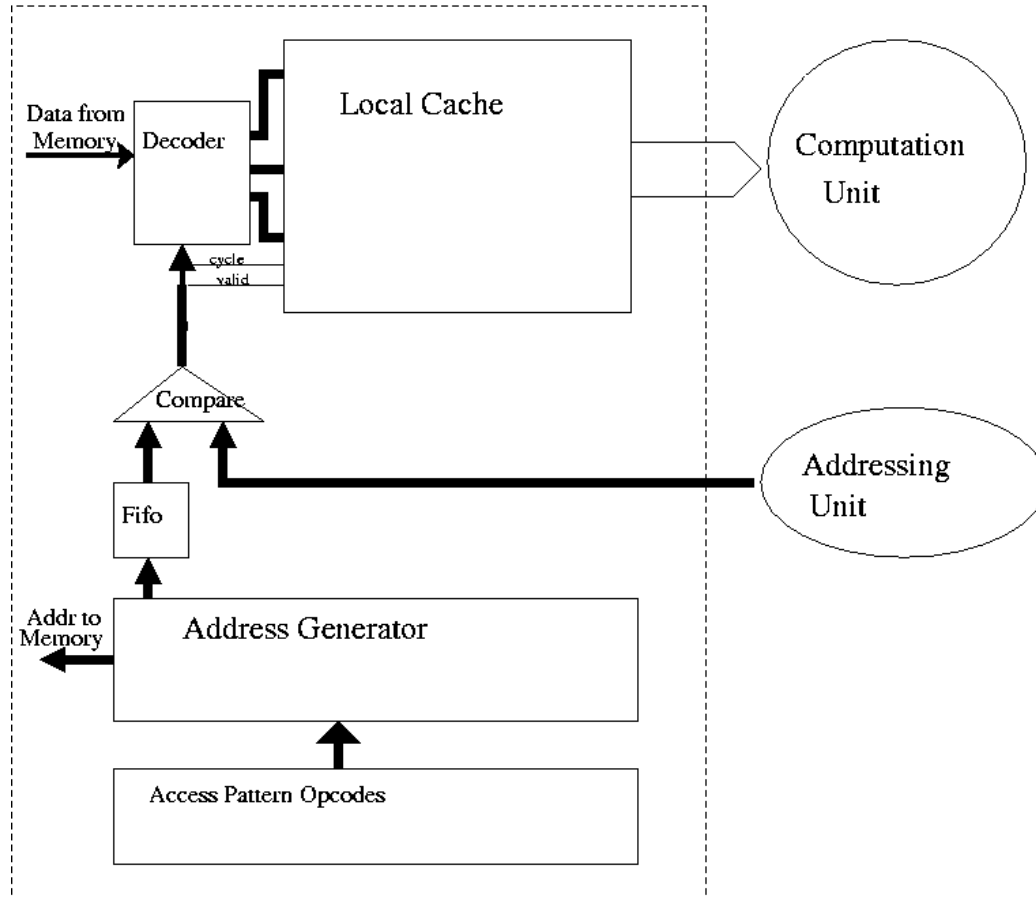
## Prefetch and Opcodes

- ⇒ Information content of addresses is less. Can be encoded
- ⇒ Compiler encodes access patterns in the form of opcodes
- ⇒ Compiler communicates with cache module to get cache size available
- ⇒ Cache dimensions are used to build custom opcodes



Access Patterns





## Parts of Reconfigurable Image Cache

- AddressGenerator generates addresses using opcodes
- 2 types of addresses
  - ❑ Addresses not in cache are read from memory
  - ❑ Addresses for comparing with those from addressing unit
- Comparator checks if address generator matches addressing unit
- Decoder routes the data to its place in the cache memory

# Configurable Network Protocols



⇒ goals:

- ❑ make major mode of INIC operations accessible to software programmers
- ❑ balance RC resources and required functionality
- ❑ an RCADE tool for programming INICs

⇒ Ranjesh G. Jaganathan

## Broadcast-Based Barrier



⇒ goal:

- improve application performance via new barrier technique

⇒ Kenneth Russell (with help from Keith Underwood)

## Video over Ad Hoc Wireless Networks



⇒ goal:

- ❑ show CBR models of VBR data are inaccurate
  - ❑ show VBR data requires features not present in existing protocols
  - ❑ propose features
- ⇒ Ryan Hanks (co-advising with Harlan Russell)

## Accelerating BLAST with RC



➤ goals:

- ❑ use command-line parameters to do last minute specialization
- ❑ 5× to 30× performance improvement
- ❑ viability of Stage 1 online reconfiguration

➤ *unmanned*

## Polymorphous Stable Interface/Online Forge

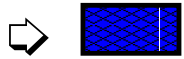


⇒ goals:

- ❑ synthesize hardware “as needed” via Forge and Black-down JVM
  - ❑ provide a stable application interface to architecture with changing features
  - ❑ viability of Stage 2 online reconfiguration
  - ❑ device driver/library artifact
- ⇒ Srinivas R Beeravolu, Matt Simpsons, Brian Greskamp



## Arbitrary Mappings in RC



⇒ goals:

- ❑ compare the RC resources used between three techniques for arbitrarily mapping function units
  - ❑ show the ‘programming time’ to realize changes in the mapping units
  - ❑ an RCADE tool for mapping units
- ⇒ Shyamnath Harinath

# List of Slides

		Online Reconfiguration .....	10
Reconfigurable Computing — Spring 2002	1	Stages .....	11
Thrusts .....	2	Project Summaries .....	12
Big Picture .....	3	Project Summaries (cont'd) .....	13
Supported Students .....	4	Scheduling .....	15
New Students .....	4	/projects Organization .....	16
Adaptable Computing Cluster .....	5	Riggs 309 Lab Status .....	17
Adaptable Computing Cluster .....	6	Goals .....	18
RCADE, Take Two .....	7	Goals (cont'd) .....	19
RCADE, Take Two (cont'd) .....	8	Upcoming Conferences .....	20
RCADE, Take Two (cont'd) .....	9	<i>Fini</i> .....	21

Reconfigurable MMU .....	22	Video over Ad Hoc Wireless Networks ....	30
Reconfigurable Cache for Image Processing Applications .....	23	Accelerating BLAST with RC .....	31
Prefetch and Opcodes .....	24	Polymorphous Stable Interface/Online Forge	
Parts of Reconfigurable Image Cache .....	27	32	
Configurable Network Protocols .....	28		
Broadcast-Based Barrier .....	29	Arbitrary Mappings in RC .....	33