



Introduction to the Bertacco Lab

Bertacco Lab
graduate students
Computer Science and Engineering
University of Michigan – Ann Arbor

Spring 2015

Introductions



Rawan Abdel-Khalek



Biruk Mammo



Doowon Lee



Dong-Hyeon Park



Abraham Addisie



Vaibhav Gogte



Helen Hagos

introduce
yourself

Agenda

- 1. Group's research overview** – with featured projects
 - Heterogeneous and embedded system architectures
 - Viable chips
- 2. The Ph.D. program at UofM** – and all its secrets
- 3. Questions and answers**

Bertacco Lab Research Areas

**1. HETEROGENEOUS AND EMBEDDED
SYSTEM ARCHITECTURE**

Part 1: Topics

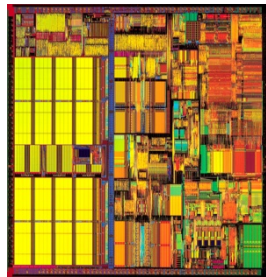
- Heterogeneous Architectures – Goals:
 - Design for reliability
 - Design for energy-efficiency
- Network-on-chip – Goals:
 - Design for reliability
 - Design for low-power
 - Design for high bandwidth
 - Design for correctness

Featured projects:

- HiROIC (Vaibhav)
- FATE (Doowon)
- SeRVe (Rawan)
- ReDEEM (Biruk)
- SCMR (Abraham)

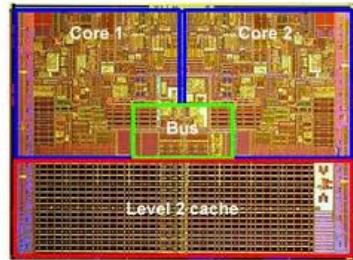
Processor and system-on-chip design trends

Pentium3 -2000



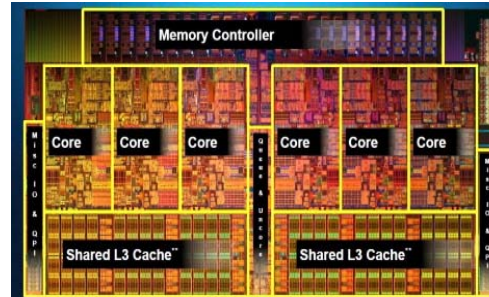
1 core

Core2Duo - 2007



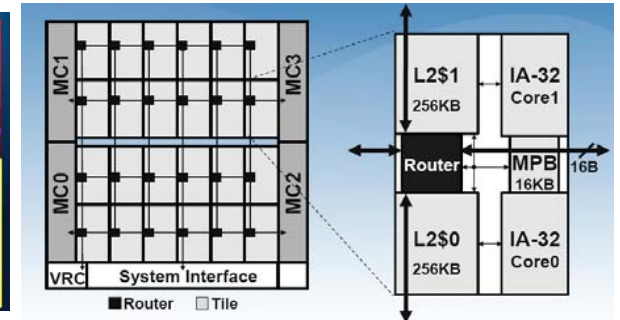
2 core; simple bus

Intel i7 980X - 2010



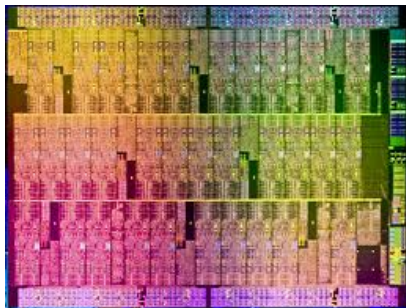
6 core; point-to-point bus

Intel SCC - 2010



48 core; complex network on-chip

Intel Xeon Phi – 2010-2013



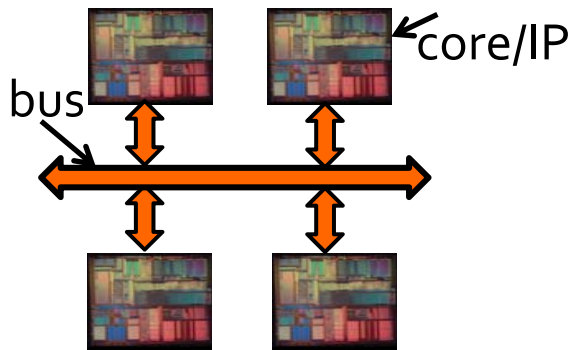
>50 cores; 2D mesh

TI OMAP5-2012

Dynamic memory manager L2 cache		ARM Cortex-M4		ARM Cortex-M4
ARM® Cortex™-A15 MPCore (up to 2 GHz)	ARM Cortex-A15 MPCore (up to 2 GHz)	POWERVR™ SGX544-MPx 3D graphics	mini-C64x DSP	IVA-HD video accelerator
		TI 2D graphics	Image signal processor	
L3 Network-on-chip interconnect				
Timers, Int Controller, Mailboxes, System DMA			Audio processor	
Boot/Secure ROM, L3 RAM				
M-Shield™ system security technology: SHA-1/SHA-2/MD5, DES/3DES, RNG, AES, PKA, secure WDT, keys, crypto DMA				
L4 peripherals			Multi-pipe Display Sub-System (DSS)	

SoC; custom network on chip

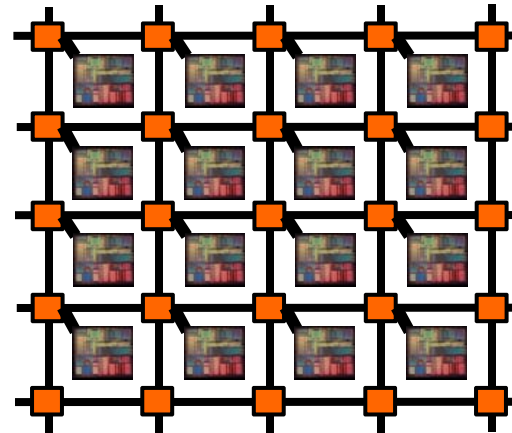
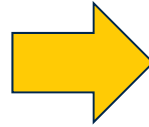
Interconnect design trends



Bus-based multicore/SoC

Bus: shared channel

- (-) limited bandwidth/scalability
- (+) low complexity



NoC-based multicore/SoC

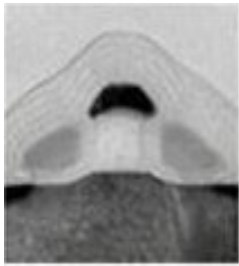
Network-on-chip (NoC)

- Routers connected via links
- Cores connected via network interface (NI)

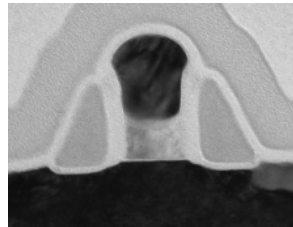
Distributed communication

- (+) higher bandwidth/scalability
- (-) high complexity

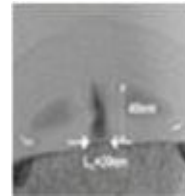
Power and energy constraints



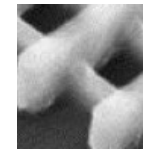
65nm



45nm



32nm



22nm

Excessive Power

due to increasing device densities and activity

**The
New York
Times**

**Progress Hits Snag: Tiny
Chips Use Outsize Power**

Many integrated
components

Not all can be switched
on at the same time

Heterogeneous
designs for
energy efficiency

Complex
interconnects

NoC consumes up to
30% of chip power

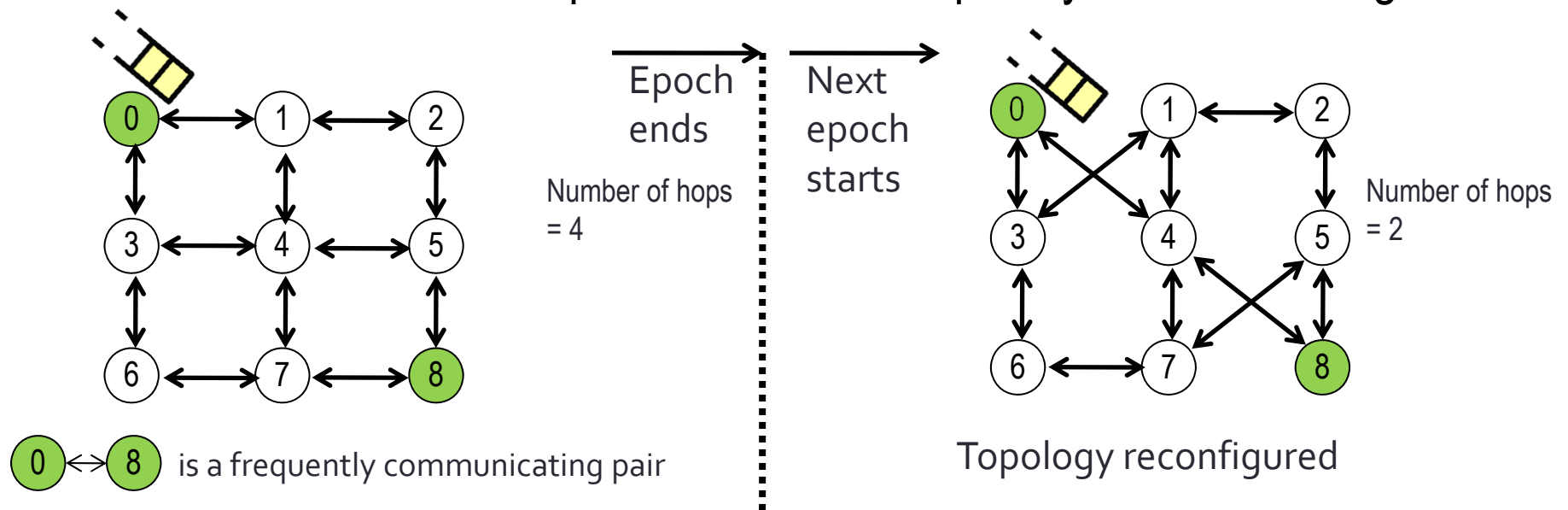
Heterogeneous and Embedded System Architecture

FEATURED PROJECTS

HiROIC | FATE | SeRve | ReDEEM | SCMR

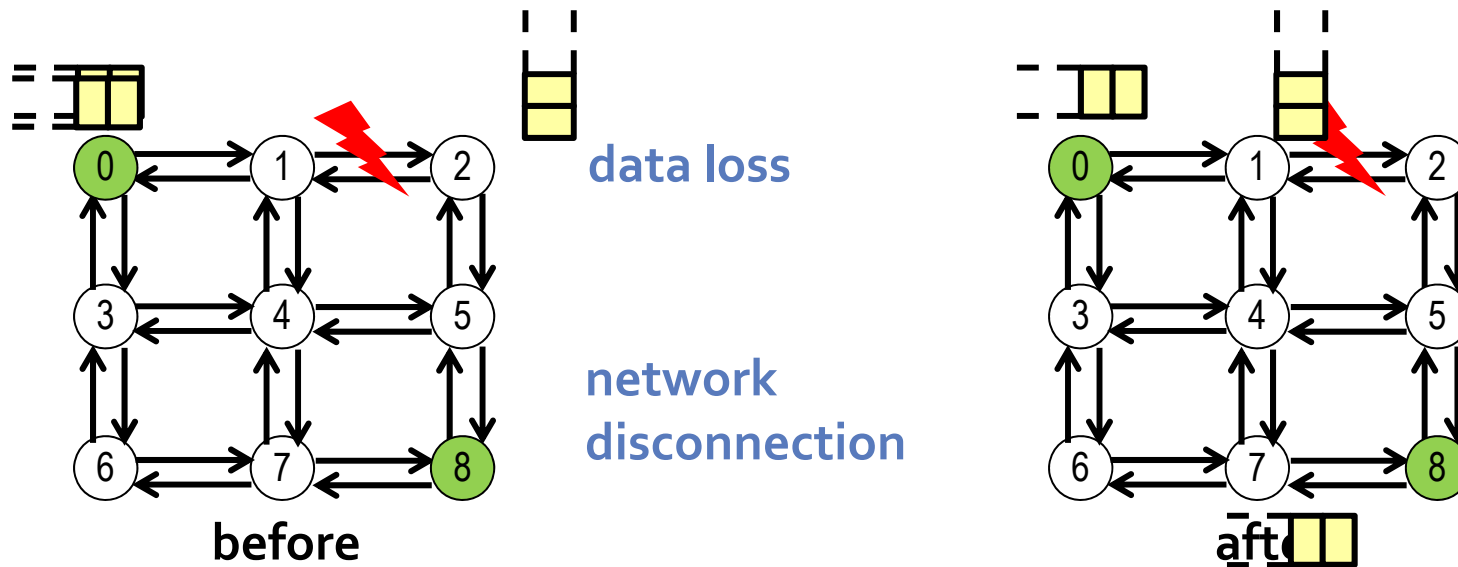
HiROIC

- Architectures and applications are heterogeneous
 - 10% of source-destination pairs share more than 60% of the traffic
- Optimize for the common case
- Change the routing and topology at runtime
 - Reduce the number of hops between the frequently communicating nodes



Fault-tolerant routing for NoCs

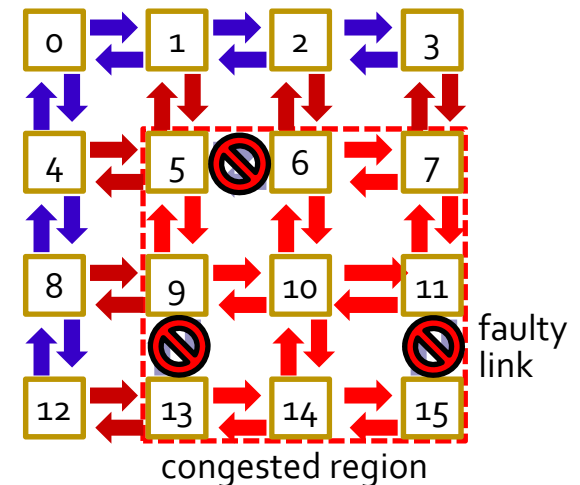
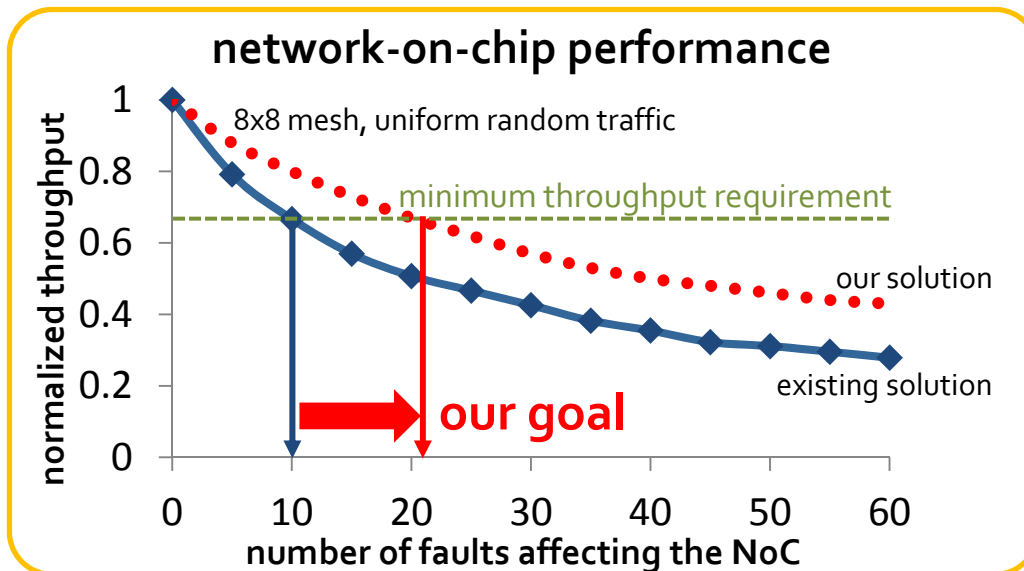
- Routing reconfiguration upon a fault occurrence



- Save as many chip components as possible (connectivity)
- Keep reconfiguration impact small (non-intrusive)

Low-performance in faulty NoCs

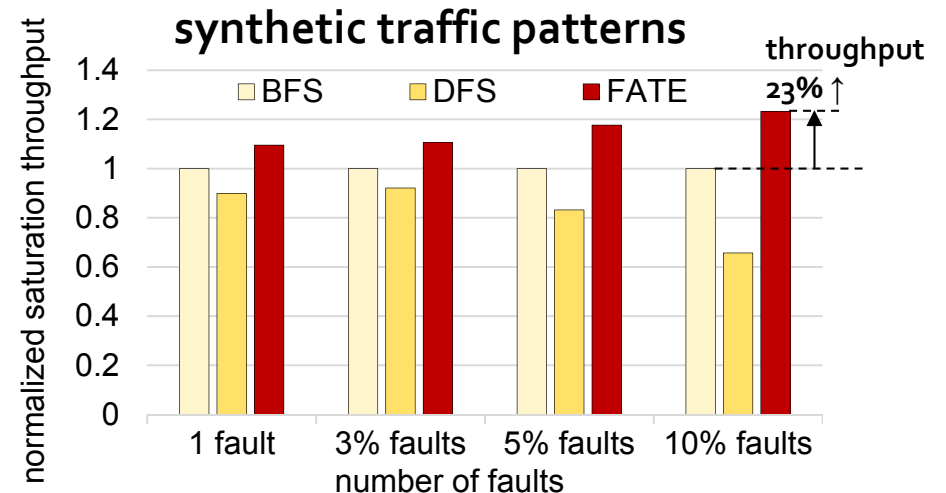
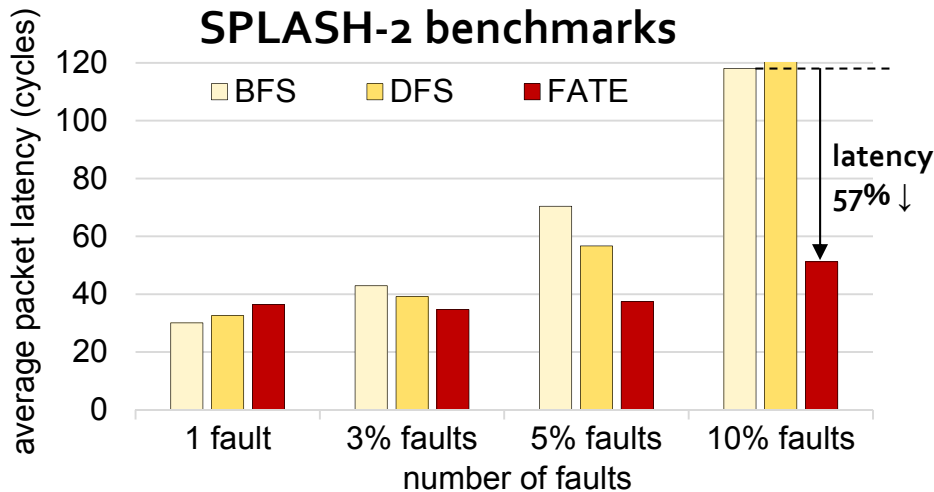
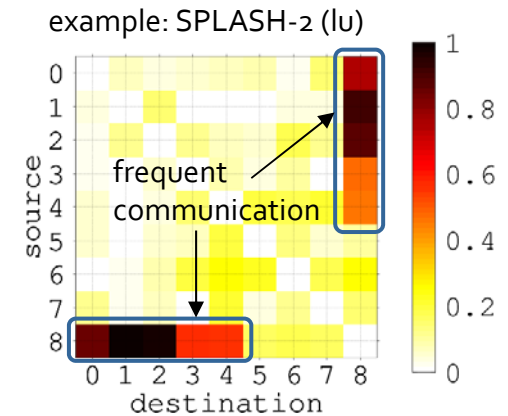
- Network-on-chip performance can degrade significantly upon permanent fault occurrences
 - Due to **traffic congestion** in faulty regions
 - Existing state-of-the-art fault-tolerant routing
 - Provides fast recovery, maximal connectivity, deadlock-freedom
 - Handles traffic congestions poorly



Underperforming networks-on-chip are becoming a concern for faulty CMPs and SoCs

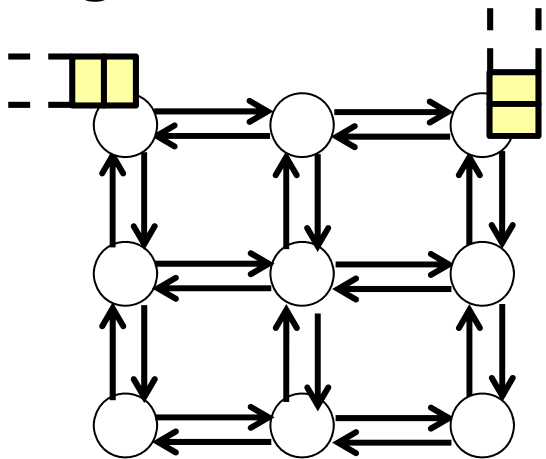
Routing optimization for applications

- Our approach leverages **communication patterns** in CMP/SoC applications
- **FATE**: Fault- and Application-aware routing
 - Heuristic solution to quickly compute a **near-optimal routing function** for application
 - **Balances traffic loads** through available network resources
 - Still provides maximal connectivity, deadlock-freedom

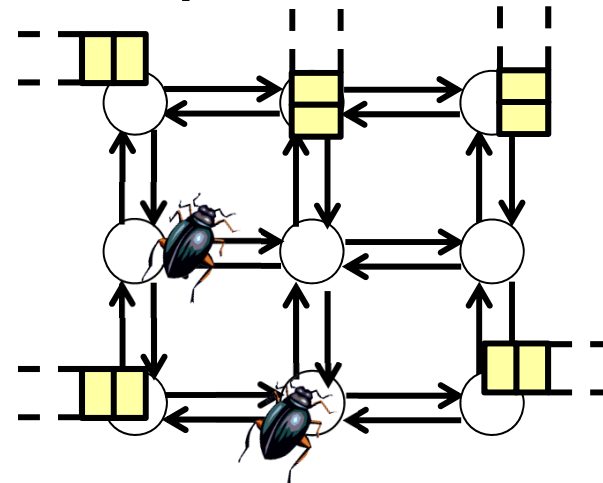


SeRVe: NoC correctness

Design time verification of network-on-chip interconnects



- basic operations
- most common execution scenarios



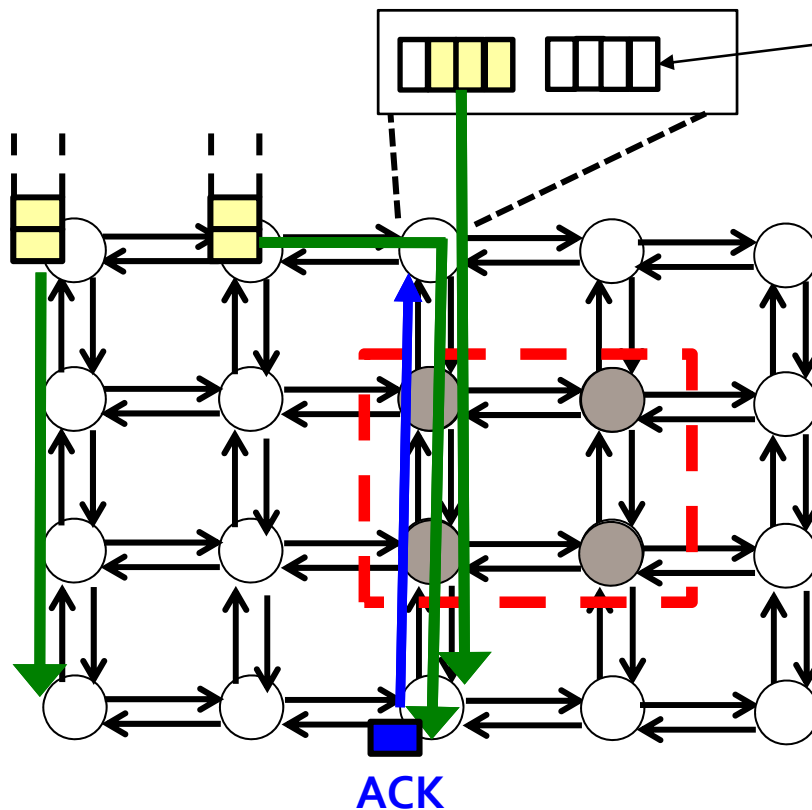
- complex execution
- corner cases
- unverified operations

Design bugs escape into final product

Compromise correctness of communication and entire system

SeRVe: Selective retransmission for verification

Goal: High traffic regions are prone to escaped bugs
=> Protect packets passing through such regions



retransmission buffers

1- Find congested regions

2- Packets passing through non-congested areas proceed normally

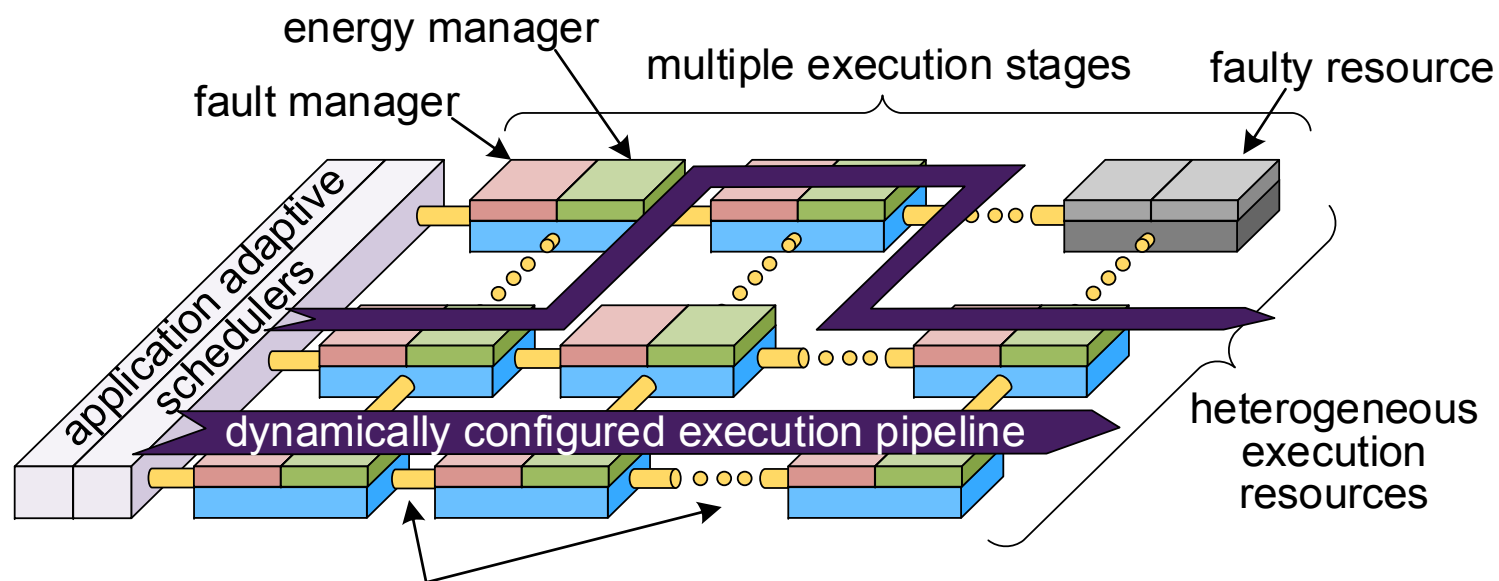
3- Packets passing through congested region

- Copy packet into retransmission buffer
- Upon correct delivery, destination sends back an ACK
- In case of errors, packet is retransmitted

ReDEEM

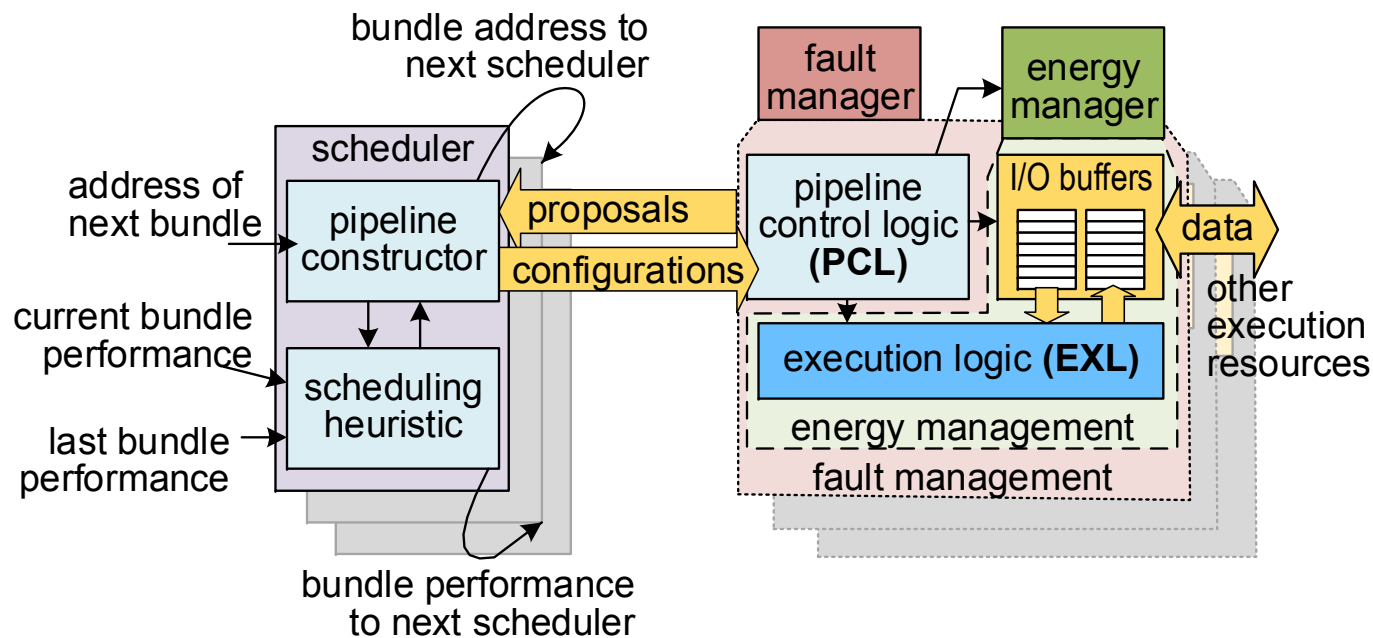
An energy efficient, reliable microarchitecture designed from the ground up

- **Distributed and decoupled execution resources** for high reliability
 - Execution pipelines constructed dynamically
- **Heterogeneous execution resources** for performance-power diversity
- **Application-adaptive schedulers** for creating energy-efficient pipelines



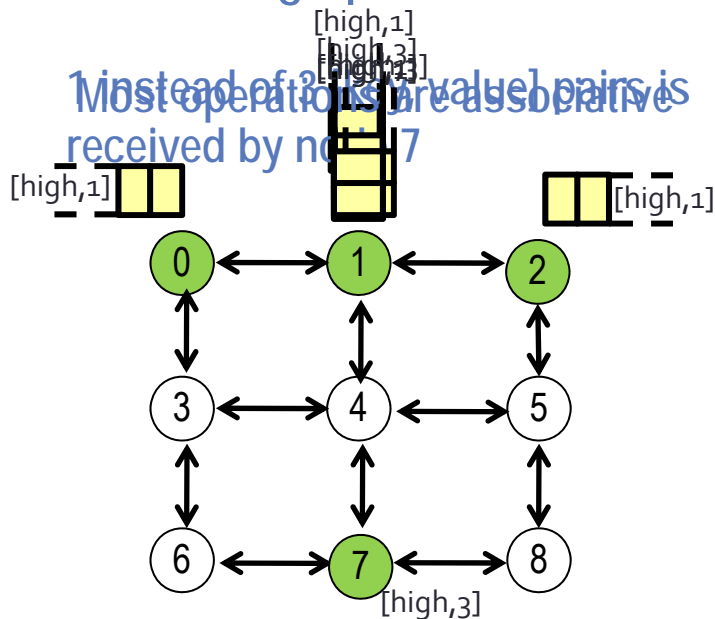
Microarchitecture details

- Schedulers and execution resources on a robust interconnect
- Energy-enhanced execution resources
 - Heterogeneity via *synthesis for diverse voltage-frequency targets*
 - *Intelligent power-gating* to eliminate static dissipation
 - *Dynamic frequency adjustment* to create single-frequency pipelines



Accelerating MapReduce on a single chip

- Emerging big data applications and many-core chips require MapReduce-like programming model
 - During its shuffling phase, MapReduce involves an all-to-all exchange of [key, value] pairs leading to network congestion
- Reduce network congestion by doing in-network aggregation of [key, value] pairs
- Leads to high-performance and low power consumption



- ❖ Processing nodes attached to routers 0, 1, and 2 send the same [key, value] pairs to node 7
- ❖ Router 1 does partial aggregation (one packet to router 7 packet), freeing downstream network
- ❖ Router 1 sends the aggregated [key, value] pairs to router 4 which will pass it to 7

Bertacco Lab Research Areas

2. VIABLE CHIPS

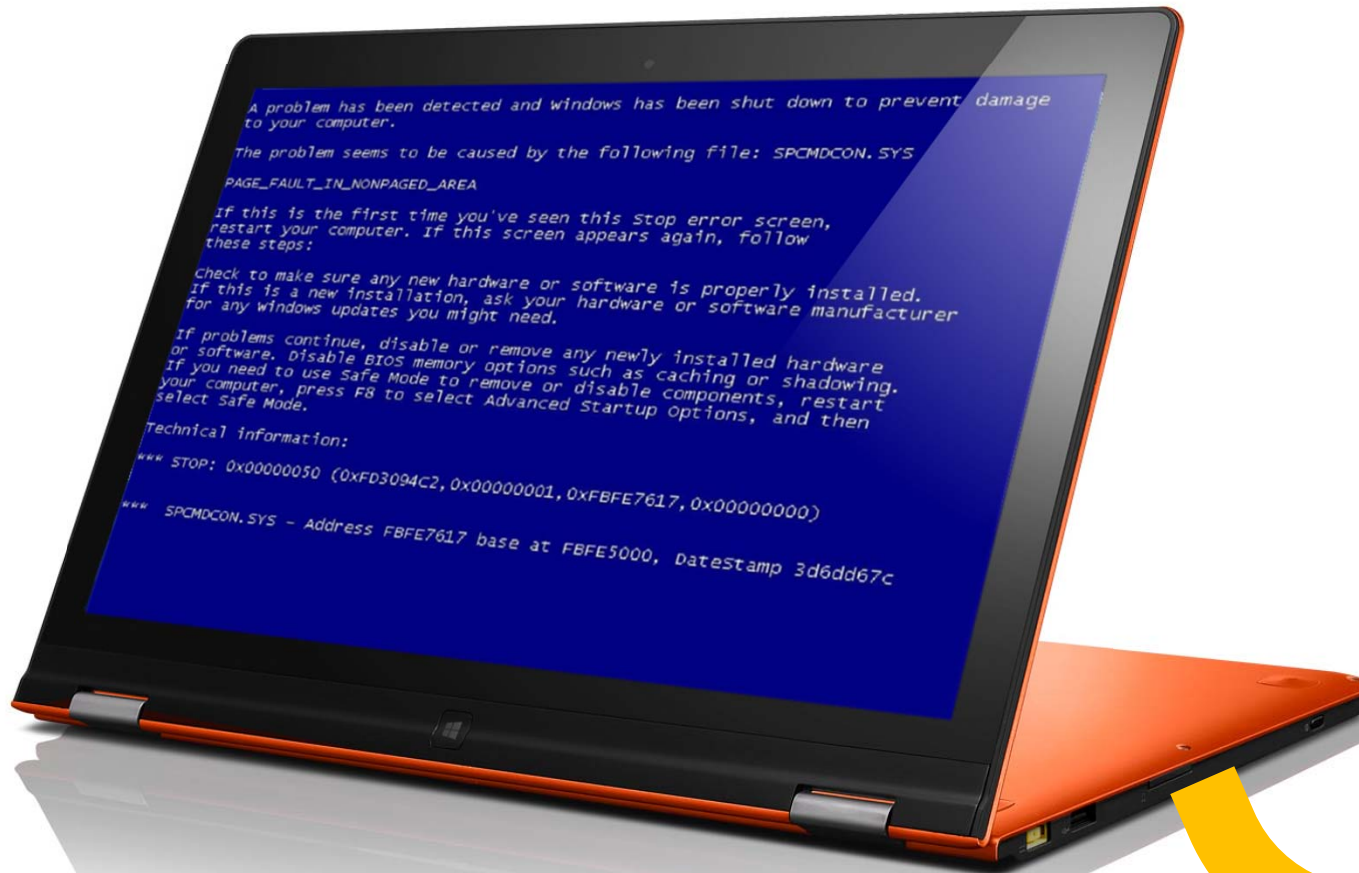
Part 2: Topics

- Enhancing verification
 - Tools for generating and visualizing NoC traffic
 - Intelligent test generation for verification
 - Machine-learning based solutions for post-silicon verification
- Runtime correctness
 - Designing runtime validation features
 - Guaranteeing runtime correctness for NoCs
 - Enhancing design reliability
- Security
 - Machine-learning based malware detection

Featured projects

- PacketGenie (Dong-Hyeon)
- NocVision (Vaibhav)
- PhoenixTest(Doowon)
- ItHELPS (Wade via Doowon)
- SNIFFER (Jocelyn via Biruk)

Does your chip work correctly?



A) Rovio



B) Microsoft



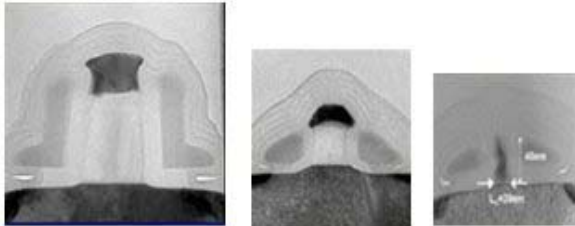
C) Graphics Driver

D) Hardware

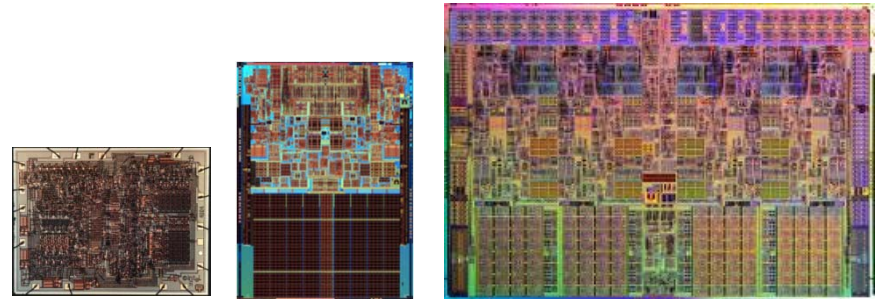


- Incorrect design (bugs)
- Transistor failures

What can go wrong?



Shrinking transistor size
"Moore's Law"



Increasing design complexity



More and more fragile transistors

- **Transient faults**
- **Permanent transistor failures (wearout)**

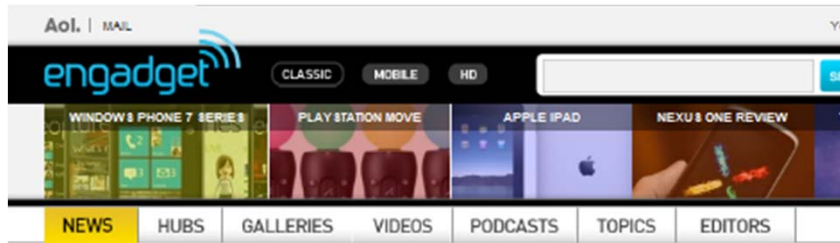
Difficult to :

- implement correctly,
- verify implementation



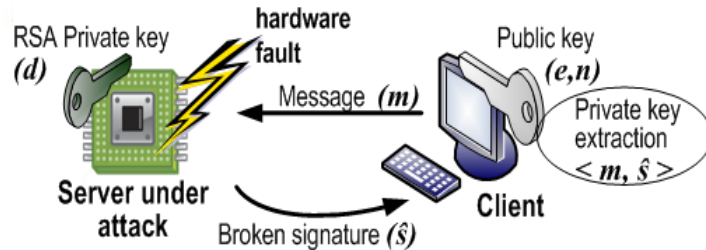
- **More verification engineers than designers in industry**
- **Design bugs escape into products**

Is that all? ... Can you trust your system?

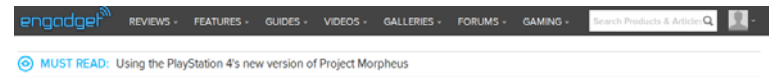


FILED UNDER [Misc. Gadgets](#)

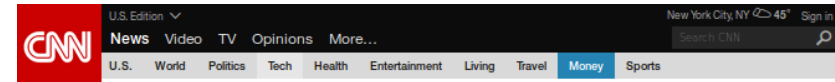
1024-bit RSA encryption cracked by carefully starving CPU of electricity



Since 1977, RSA public-key encryption has protected privacy and verified authenticity when using computers, gadgets and web browsers around the globe, with only the most brutish of brute force efforts (and 1,500 years of processing time) felling its 768-bit variety earlier this year. Now, three eggheads (or Wolverines, as it were) at the [University of Michigan](#) claim they can



Flame malware snoops on PCs across the Middle East, makes Stuxnet look small-time



The 'Heartbleed' security flaw that affects most of the Internet

By Heather Kelly, CNN
Updated 5:11 PM ET, Wed April 9, 2014



Enhancing computer security:

- Minimize vulnerabilities by giving **hardware support** to software
- Malware **detection and prevention**
- Eliminate **hardware security vulnerabilities**

Viable Chips

FEATURED PROJECTS

PacketGenie | NoCVision | PhoenixTest | ItHELPS | SNIFFER

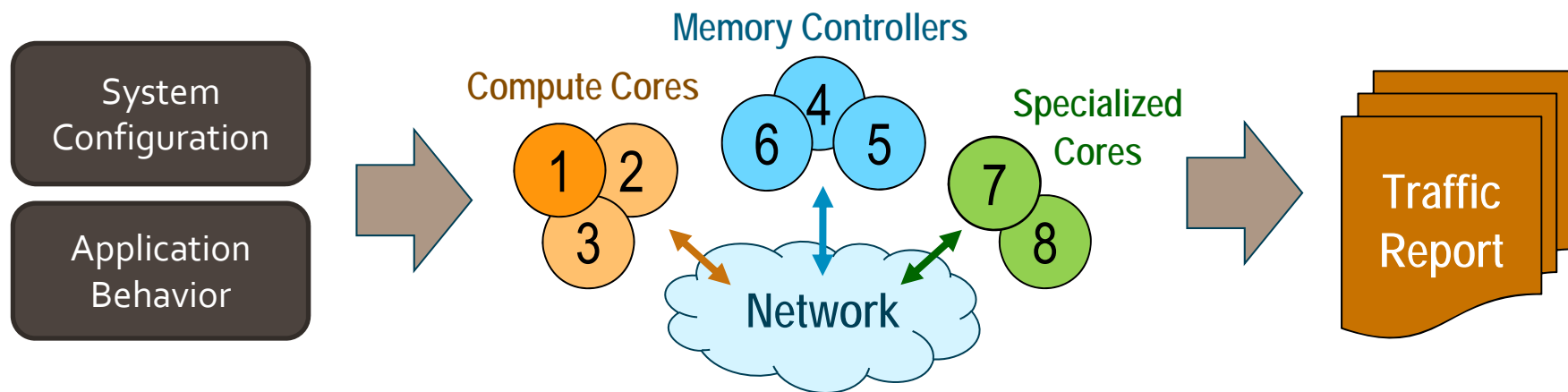


- **PacketGenie Purpose:**

- Allow **quick design-space exploration** of heterogeneous NoC interconnect
- Extract application behavior and generate traffic based on system model

- **PacketGenie Goals:**

- **Quick, light-weight simulation** of NoC traffic behavior
- **Flexible and configurable** to accommodate heterogeneous architectures
- **Easy to integrate** with existing network simulators



NoCVision: A graphical visualization tool for NoCs

Issues with traditional debug of complex NoCs

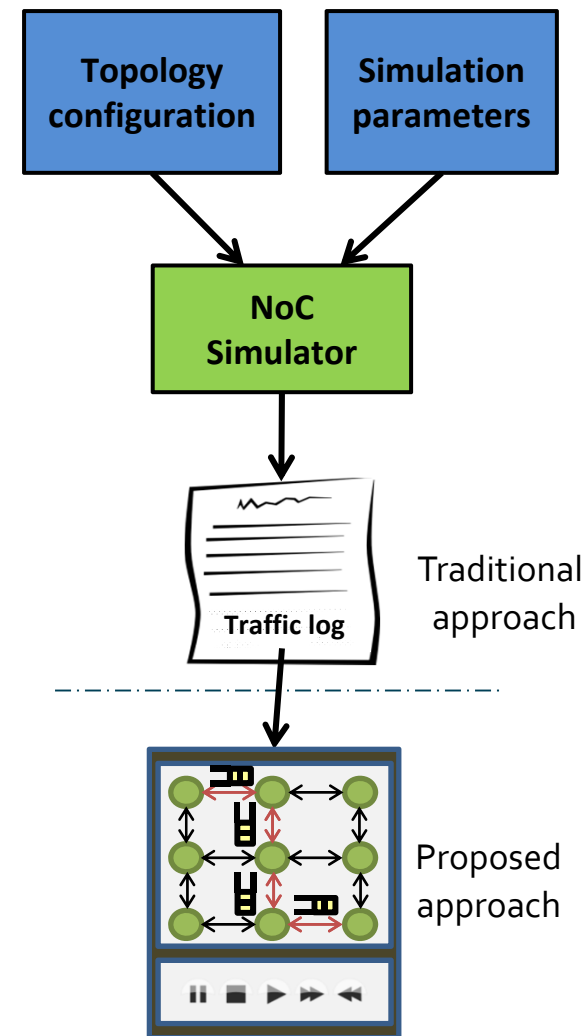
- Analyze millions of cycles
- Debug large logs

Proposed debug approach

- Graphical representation of traffic flow
- Extraction of relevant traffic details e.g. network congestion, bottlenecks etc.

Implementation

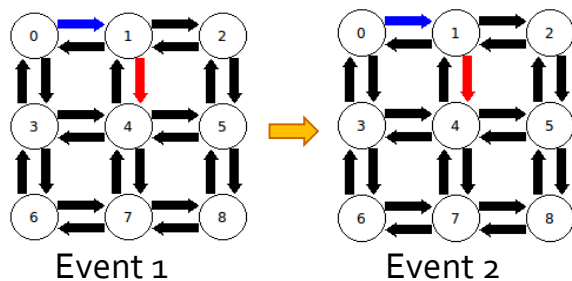
- Capturing interval-based and event-based packet flow
- Visualization on a network topology: router, link or VC level information



NoCVision: Modes of operation

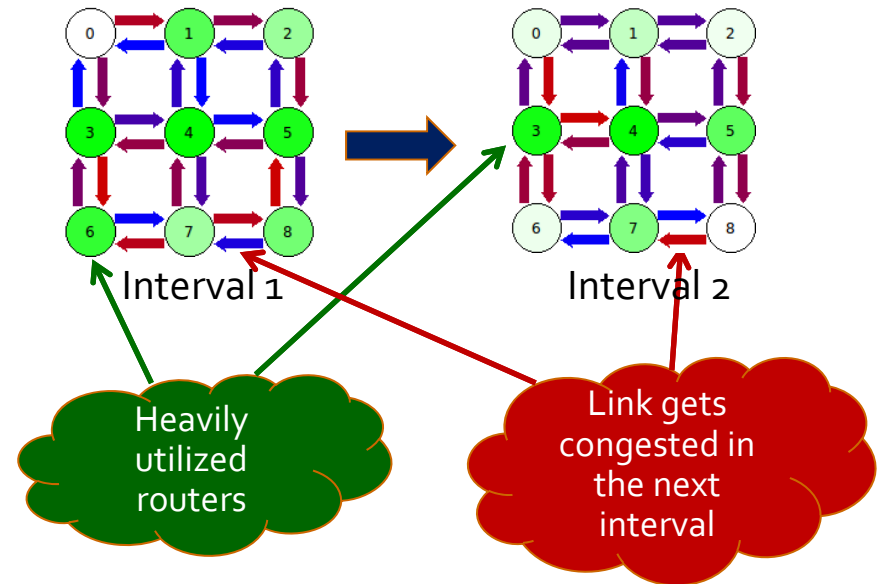
Interval Mode

- Interval-based packet flow
- Link, router and VC utilization across time windows
- Color intensity to represent parameters e.g. congestion, utilization etc.



■ Packet id 4
■ Packet id 6

Events plotted –
Traversal of packet
IDs 4 & 6



Event mode

- Determine region-of-interest and log data for specific events
- Parse through the data across events e.g. packet traversal.

Post-silicon random test generation

- Various, complex functionalities in microprocessors
 - Random tests verify unexpected corner cases (extremely rare bugs)
- Fast execution speed in post-silicon validation
 - However, checking the execution results can be bottleneck
- How can we generate high-quality test-cases for post-silicon validation to overcome the limited checking capability?

Applying static analysis on random test

- Data-flow analysis: Which buggy instructions cannot be detected at the end of test-case (check point)?

Example of random test-case (POWER ISA)

```
Inst 1: fcpsgn f14,f6,f15
Inst 2: mfctr r27
Inst 3: mulldo r25,r23,r15
Inst 4: li r27,0
Inst 5: lwaux r26,r25,r27
Inst 6: frsqrtes. f2,f11
Inst 7: add r25,r24,r16
```

...
<Check register values here>

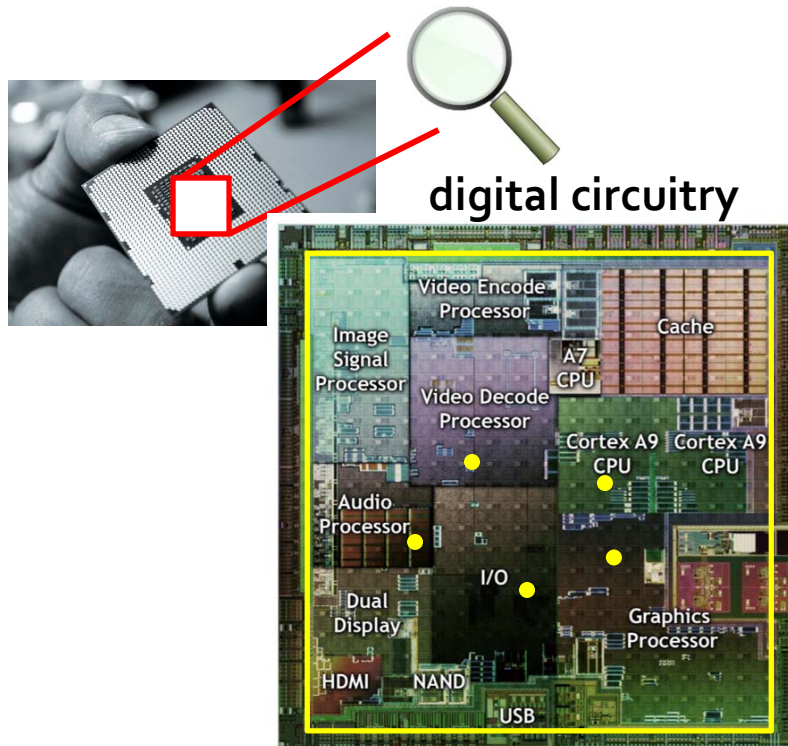
register usage table

Inst.	Target operands	Source operands
1	f14	f6, f15
2	r27	CTR
3	r25	r23, r15
4	r27	bug undetected
5	r26	r25, r27
6	f2	f11
7	r25	r24, r16

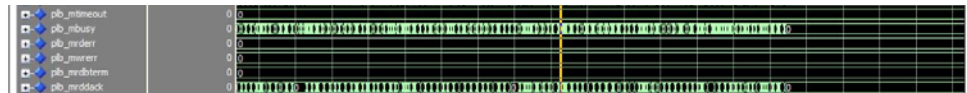
bug detected at check point

- Early-stage research “PhoenixTest”
 - Register bookkeeping, test-case mutation, mathematical model

Post-silicon error localization



post-silicon debugging: limited signals

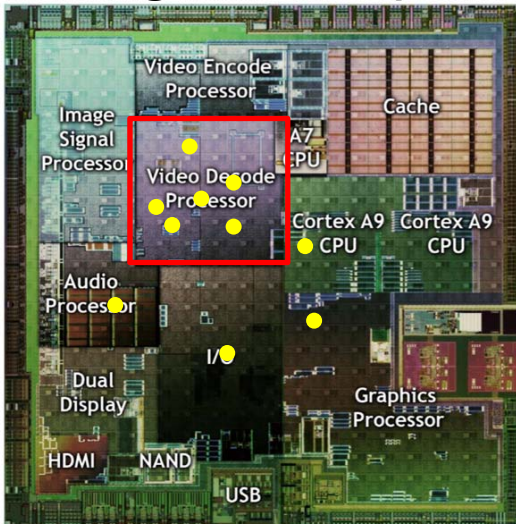


- Post-silicon debugging targets
 - Rare functional bugs
 - Intermittent electrical errors
- But, it is **difficult** due to limited capability of observing signals

Iterative high-accuracy error localization

- Applying machine learning technique to diagnose erroneous behaviors

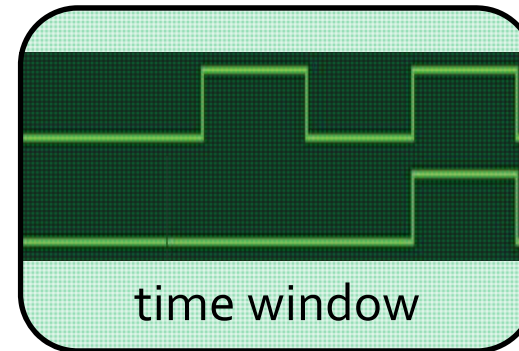
digital circuitry



- : golden feature value
- : post-si feature value
- : cluster

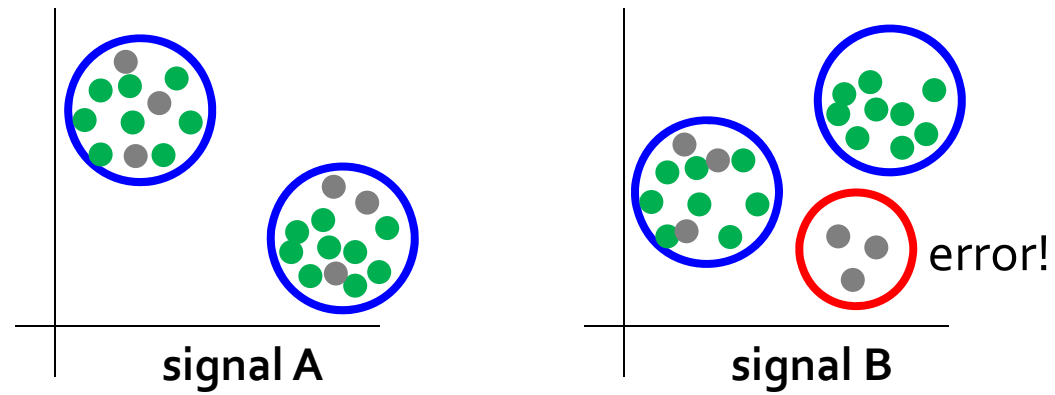
signal A

signal B

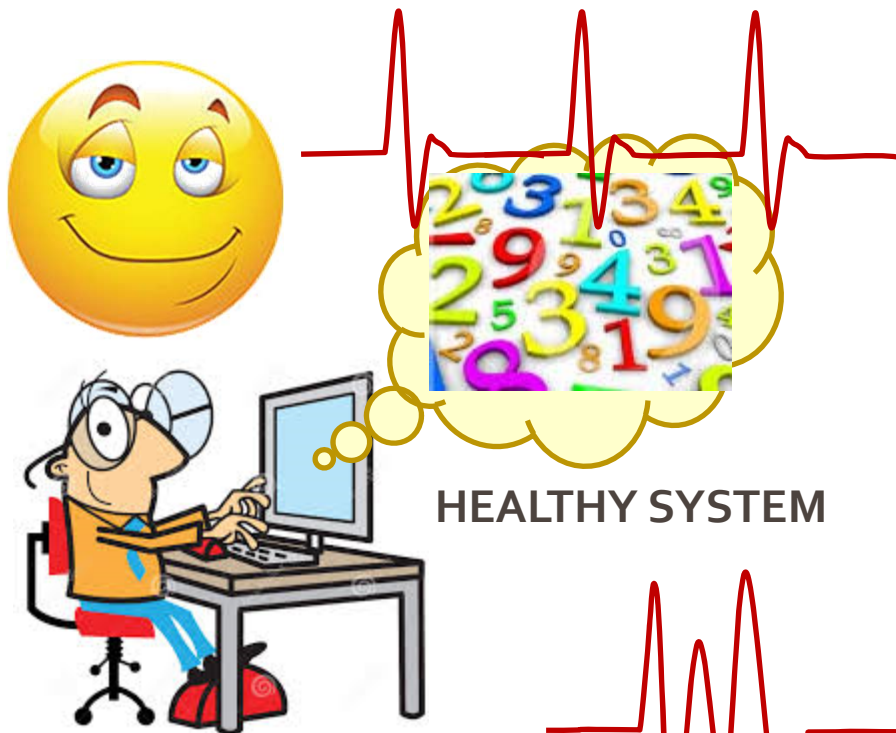


feature value A: 2

feature value B: 1

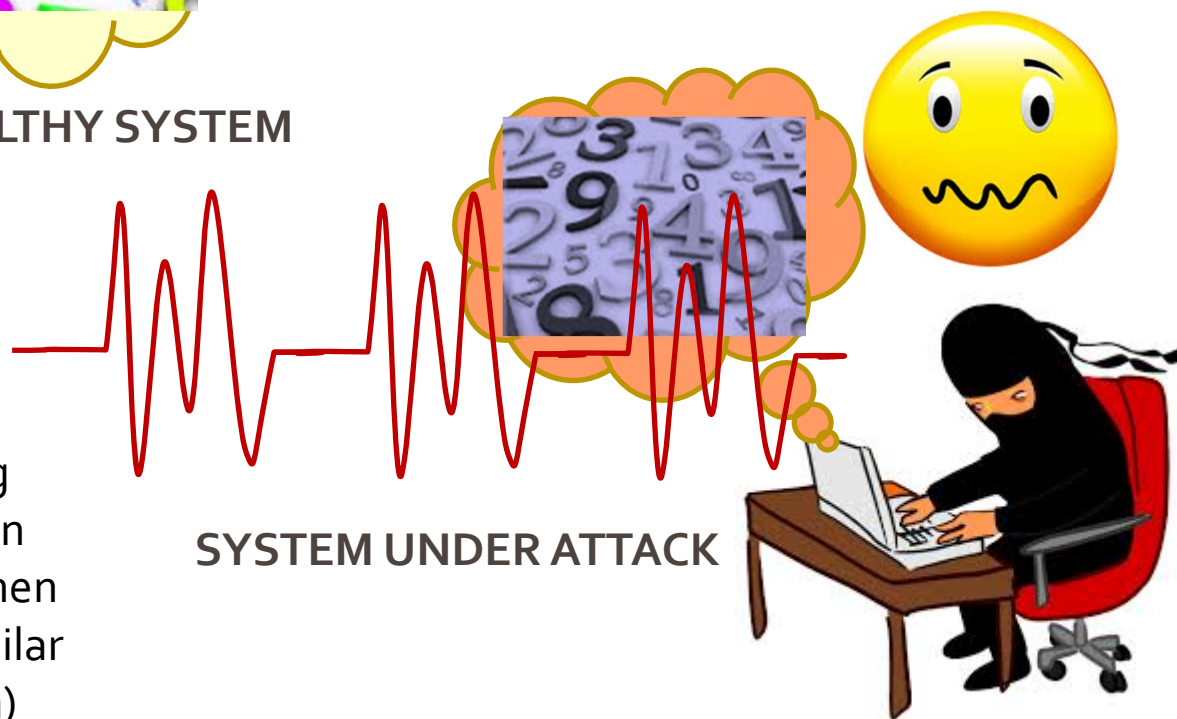


The SNIFFER - Sniffing out security attacks



Monitors a system's activity to detect signs of a security attack. Examples: email-send frequency, IP range for outgoing packets, etc.

Detects attacks by leveraging machine learning – training on typical good behaviors and then noticing atypical activity (similar to credit card fraud detection)



Lab funding

- C-FAR center on future architectures
 - <http://futurearchs.org>
- NSF
- IBM
- Intel
- Cisco

THE CSE PH.D. PROGRAM AT THE UNIVERSITY OF MICHIGAN

Timeline (w/o incoming MS)

- 1st year: Coursework/research – choose an adviser early
- 1st summer – research on campus
- 2nd year: light coursework and mostly research

- **Qualification:** Research presentation + coursework (1.5~2 yrs)

- More research (2-3 years)

- **Proposal** – 60-80% of the work is done. You have a story for your thesis

- About 1 year later: **Defense**/thesis filing

Choosing an adviser

- Earlier is better
- Explore interesting faculty at CSE
 - Visit faculty pages
 - Read their research papers/presentations
 - Contact graduate students – faculty is far too busy to answer all your questions
 - Work with them on graduate course projects
 - Make contact
- Work on a mutually interesting research topic
 - Professors typically cover broad areas under their research umbrella

Goals and advice

- Hardware courses first – preliminary research experience
- Start early on research – that's your critical path to graduation
- Collaborate – with your personal goals in mind
- YOUR Ph.D. – goal: advance the field in your area of expertise
- Research direction – Flexible at start; focused towards the end
- Constant progress (even if little) is essential

Life @CSE, UofM

- Expect to work long hours during the first year
 - Courses take time; assignments/projects are fun
- After the first year (or two) – flexible work schedule
 - Regular days 6-10 hours of work
 - 12-16 hours approaching deadlines (2-3 deadlines a year)
- Reading groups; social hours; student interactions; faculty counselling; academic lectures; workshops; career-fairs
- Discounted/free student tickets to virtually everything – theatre, movies, sports, transport, etc.

Getting started in Ann Arbor

- Housing – university housing or off-campus
- Getting around – public/private, buses, cars, airplanes
- Funding and expenses – GSI, GSRA, or fellowship
- Leisure – activities, university clubs, sports, restaurants, bars
- and the Government – driver's license, social security, taxes

QUESTIONS?