ESE680-002 (ESE534): Computer Organization

Day 26: April 18, 2007

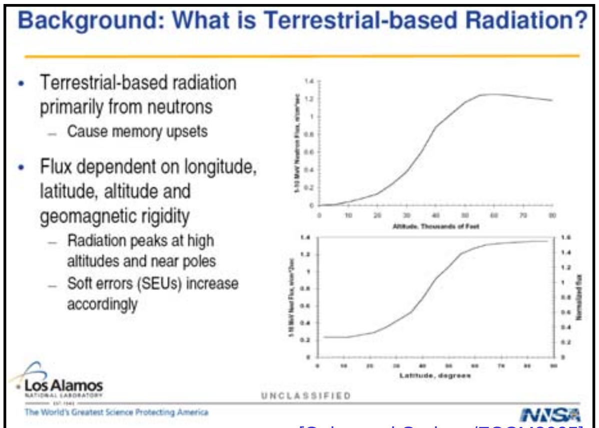
Et Cetera…

Today

- Soft Error Data
- Energy and Faults
- Defect Tolerance and FPGAs
- Big Picture Review this Course
- Things we didn’t talk about
  - Model
  - Programming
  - Mapping
- Feedback Forms
  - SEAS
  - For course

Soft Errors

- FPGA configurations stored in SRAM
- SRAM now susceptible to soft errors
- Upset configuration bit (pinst)
  - Cause errors in operation



Background: What is Terrestrial-based Radiation?

- Terrestrial-based radiation primarily from neutrons
  - Cause memory upsets
- Flux dependent on longitude, latitude, altitude and geomagnetic rigidity
  - Radiation peaks at high altitudes and near poles
  - Soft errors (SEUs) increase accordingly

Los Alamos NATIONAL LABORATORY

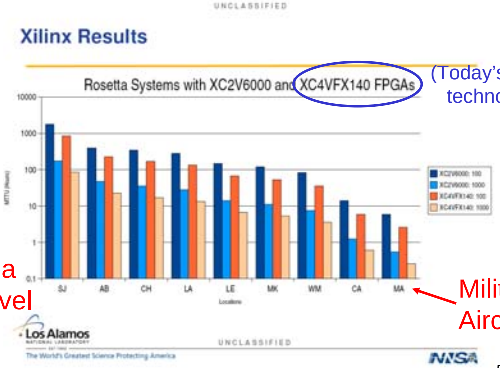
The World’s Greatest Science Protecting America

UNCLASSIFIED

NNSA

[Quinn and Graham/FCCM2005]

Soft Error Effects



UNCLASSIFIED

Xilinx Results

Rosetta Systems with XC2V6000 and XC4VFX140 FPGAs

(Today’s technology)

Sea Level

Military Aircraft

Los Alamos NATIONAL LABORATORY

The World’s Greatest Science Protecting America

UNCLASSIFIED

NNSA

[Quinn and Graham/FCCM2005]

Induced Soft Errors

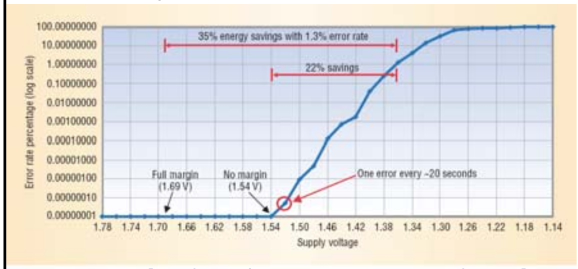
- People have deliberately induced soft errors to
  - Break crypto
  - Break JVM security
- Gives a way to change bits which security should be preventing

Margins, Energy, and Fault Rate

- Worst-case margins
  - Uncertainty tax
    - Getting larger with increasing variation
  - Costs energy and performance
- Reduce Voltage to reduce Energy
  - E∝CV^2
- For storage
  - Lower barrier to electrons hoping out of well
- Reducing energy → increasing fault rate

Margins, Energy, and Fault Rate

18x18 Multiplier on Virtex2 at 90MHz



[Austin *et al.*--*IEEE Computer*, March 2004]

Impact: Energy/Reliability

- Can trade off energy with reliability
- Reduce voltage → increase failures
- Can we get net win?
  - If check/recovery energy < energy savings

FPGA Defect Tolerance

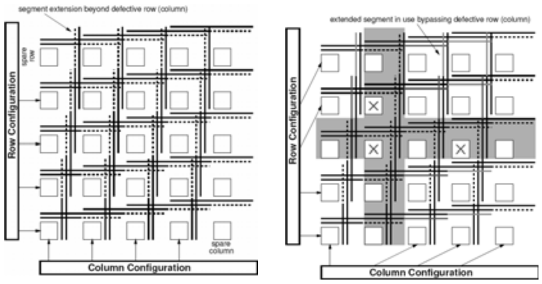
- Configuration built in → already paid for
- Three models:

1. Perfect component
2. Defect map with global (re)mapping
3. Defect map with local sparing

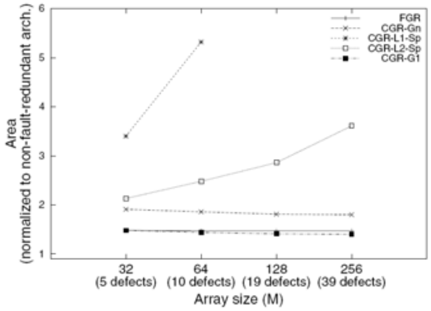
Perfect Component

- Like Memory Case
- Add extra, hidden resources
- Use to repair so looks perfect
- *E.g.* Spare Row/Column
- In use by Altera
  - Many patents
  - Claim significantly improves yield APEX 20KE

Row/Column Sparing


segment extension beyond defective row (column)
spare row
Row Configuration
spare column
Column Configuration
extended segment in use bypassing defective row (column)
Row Configuration
Column Configuration

Coarse-Grain Sparing Overheads



[Yu&Lemieux/FPT2005]

Perfect Component

- Coarse-grained – spare large units
- Expensive in area

✓Doesn't change model

✓Single mapping still works for all parts

Global Remapping

- Mark cells, wires as bad
- Make sure there are enough good cells
- Perform placement/routing per component
- Used in HP TERAMAC
  - Tolerate 3% interconnect, 10% LUT defects

[Culbertson et al. / FCCM1997]

Global Mapping

- Mapping is slow
  - Must perform for each component

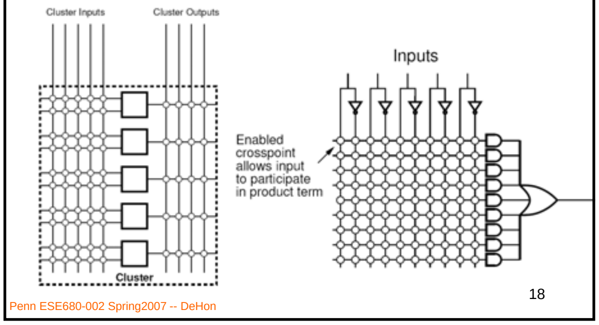
✓Minimum area overhead

- Mostly just the defective LUTs, interconnect

Local Mapping

- Organize in local pools of interchangeable resources
- Provision spares in each pool
  - Like spare rows in a memory bank
- Avoid global remap, just exchange locally

Locally Substitutable Resources

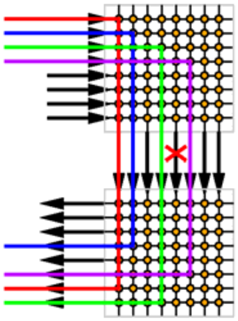

Cluster Inputs
Cluster Outputs
Cluster
Inputs
Enabled crosspoint allows input to participate in product term

Day 25

Crossbar Buses and Faults

- Two crossbars
- Wires may fail
- Switches may fail

- Provide more wires
  - Any wire fault avoidable
    - M choose N
  - Same idea

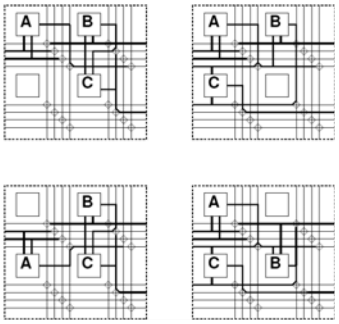


With FPGAs

- Cut into k×k tiles (at least logically)
- Provision spare within tile
- Precompute placements for all defect options within tile
- At load time
  - Lookup correct configuration based on tile defects
  - Stitch together full design from tiles

[Lach *et al.* / FPGA 1998]

2×2 Tile Example



[Lach *et al.* / FPGA 1998]

Local Sparing

- Potentially expensive to support
- Requires more spares than global mapping
  - Must have spares local

✓No global remap
  - Accommodate per component failures quickly/easily
  - fast

Review

Engineering Discipline

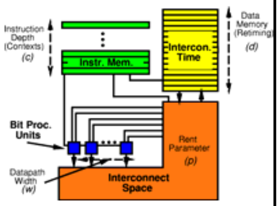
- Computations implemented in Matter
  - Require Area, Delay, Energy
  - Can Fail
- Quantify costs
- Explore how to minimize
- Approach systematically

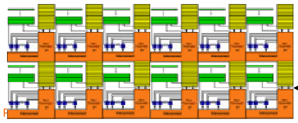
Themes

- Costs
- Change
- Design Space
- Parameterization
- Structure in Computations
  - Induces much of parameterization
    - W, c/L_{path}, Rent (c,p), d, controllers

Computing Device

- Composition
  - Bit Processing elements
  - Interconnect: space
  - Interconnect: time
  - Instruction Memory



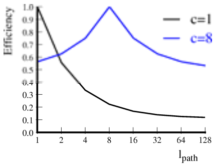


Tile together to build device

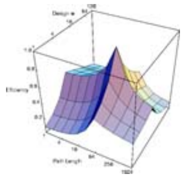
Architecture Instruction Taxonomy

Control Threads (PCs)	*pinsts* per Control Thread	Instruction Depth	Granularity	Architecture/Examples
0	0	0	n/a	Hardwired Functional Unit (*e.g.* ECC/EDC Unit, FP MPY)
	n	1	1	FPGA
			w	Reconfigurable ALUs
1	1	c	$n_v \cdot 1$	Bitwise SIMD
			w	Traditional Processors
			$n_v \cdot w$	Vector Processors
	n	c	1	DPGA
		8	16	PADDI
		c	w	VLIW
m	n	1	1	HSRA/SCORE
	1	c	$n_v \cdot w$	MSIMD
m	1	c	1	VEGA
		8	16	PADDI-2
		c	w	MIMD (traditional)

Balance



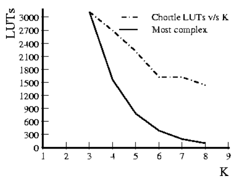
- Instructions vs Compute
- Compute vs Interconnect
- Retiming with compute and interconnect

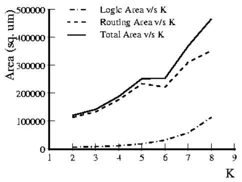


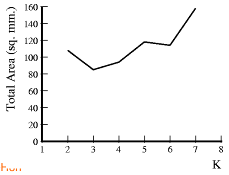
Methodology

- Architecture model (parameterized)
- Cost model
- Important task characteristics/structure
- Mapping Algorithm
  - Map to determine resources
- Apply cost model
- Digest results
  - find optimum (multiple?)
  - understand conflicts (avoidable?)

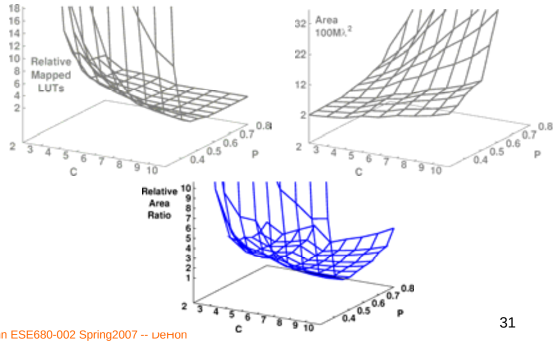
Mapped LUT Area





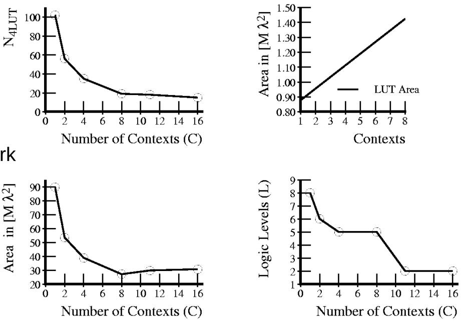


Resources × Area Model ⇒ Area



Control: Partitioning versus Contexts (Area)

CSE benchmark

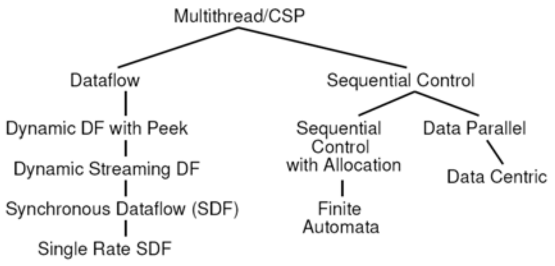


Things we didn't talk about

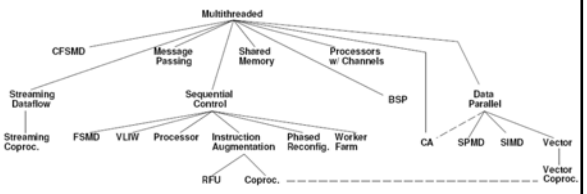
Important (not in this course)

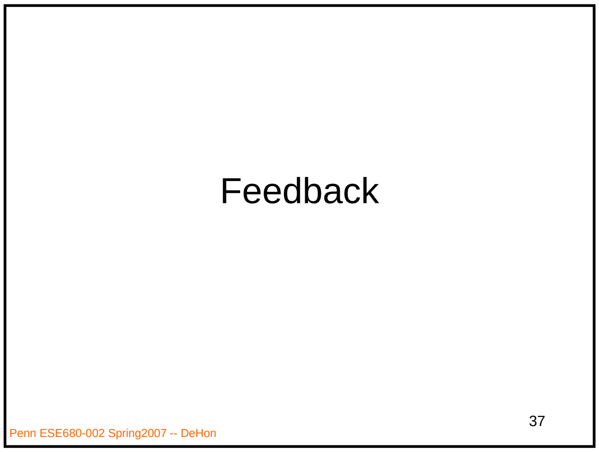
- Human time
  - Model
    - How should we reason about computation
    - How allow to scale automatically
  - Programming
    - How to capture application, freedom
- Compute/tool time
  - Algorithms to optimize

Computational Models


Multithread/CSP
Dataflow
Sequential Control
Dynamic DF with Peek
Dynamic Streaming DF
Synchronous Dataflow (SDF)
Single Rate SDF
Sequential Control with Allocation
Finite Automata
Data Parallel
Data Centric

System Architectures


Multithreaded
CFSMD
Message Passing
Shared Memory
Processors w/ Channels
Streaming Dataflow
Sequential Control
BSP
Data Parallel
Streaming Coproc.
FSMD
VLIW
Processor
Instruction Augmentation
Phased Reconfig.
Worker Farm
CA
SPMD
SIMD
Vector
RFU
Coproc.
Vector Coproc.



Feedback