Implementation of a Target Recognition Application Using Pipelined Reconfigurable Hardware

Benjamin Levine* and Herman Schmit
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA
blevine@ee.pitt.edu, herman@ece.cmu.edu

**Benjamin Levine is now an Assistant Professor in the Electrical Engineering Department of the University of Pittsburgh*

Electrical & Computer ENGINEERING

Outline

- PipeRench
- SAR ATR application
- Results from 2000 project
- PipeRench prototype devices
- Hardware test platform
- Application implementation
- Results
- Future Work

Reconfigurable Computing

- Design “hardware” for each application.
  - Custom logic design gives high performance, like an ASIC.
- Map it onto **programmable** hardware.
  - We can change the hardware, like software on a CPU.

- Advantages:
  - Performance approaching that of an ASIC.
  - Flexibility approaching a general purpose processor.
  - Quick design cycles and low design costs for each application, like software design.

PipeRench

- PipeRench is a reconfigurable computing chip developed at Carnegie Mellon University.
- Has complete tool set, including high-level entry language.
- The PipeRench architecture, chip design, simulation tools, compiler, and assembler were all produced by faculty and students in the ECE Department and the School of Computer Science.
- Part of the DARPA ACS program

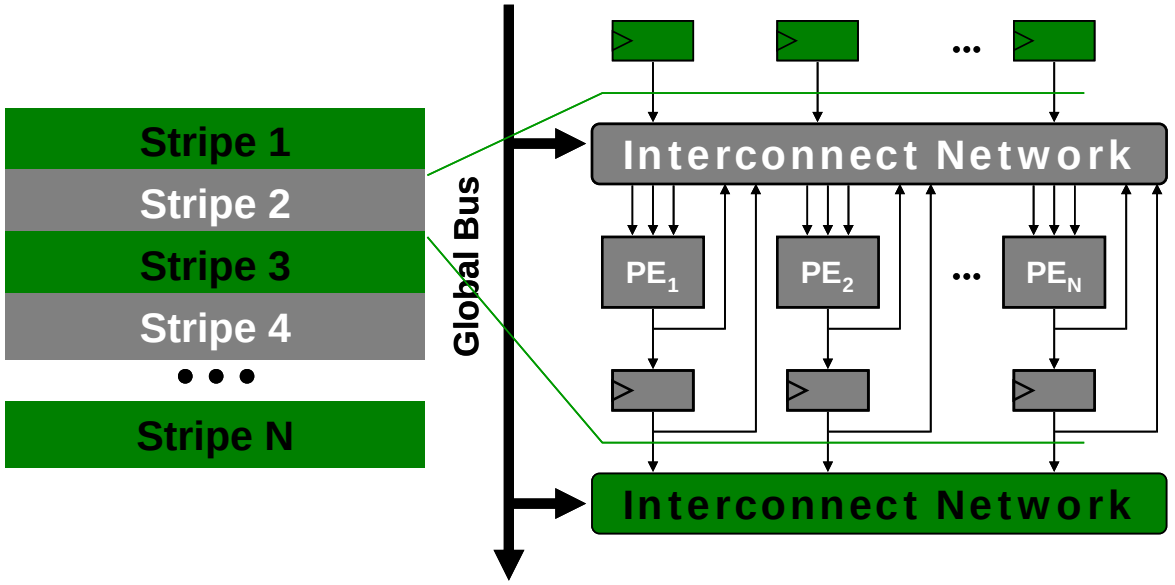
More information on PipeRench at:

http://www.ece.cmu.edu/research/piperench

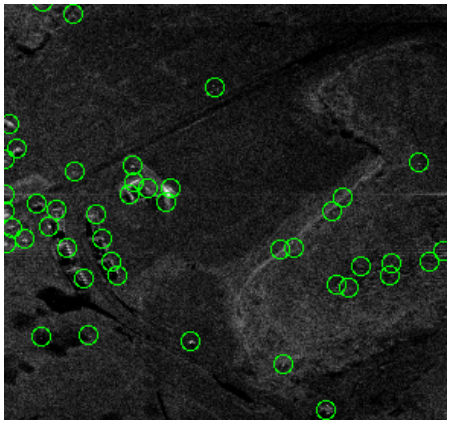
PipeRench Architecture

Reconfigurable fabric of 8-bit processing elements.

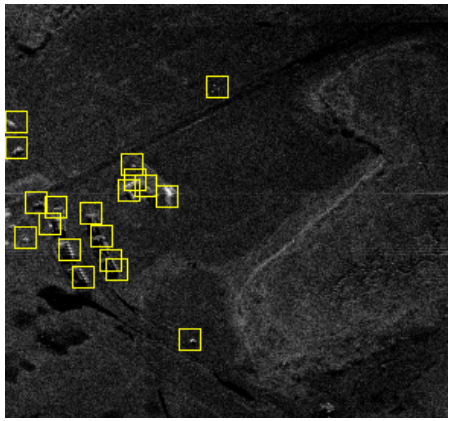
- Compiler targets unbounded virtual architecture.
- On-chip, run-time reconfiguration allows virtual application to run on bounded physical hardware.



SAR ATR Application



- Designates Regions of Interest (ROI)
- Processes entire image
- Detect areas of high contrast
- Pixel Clustering



- Eliminates Natural Clutter
- Processes only ROI's
- Computes several features including pose estimate
- Uses linear classifier

- Finds target class
- Processes only ROI outputs from discriminator
- Generate correlation planes using composite filters
- Computes scalar features from correlation plane

- Identifies specific target type from target class
- Processes with refined filters
- Computes detail features from correlation planes

Screener → **Discriminator** → **Classifier** → **Identifier**

2000 Project

- Project only worked with first two stages, the screener and the discriminator.
  - Widest variety of algorithms
  - Largest amounts of data
- Analyzed program to identify computational kernels.
- Implemented kernels on PipeRench simulator.
- Compared to workstation and signal processor platform implementations.

2000 Program

I&IR SAR ATR Front End – Speedup of Computational Kernels

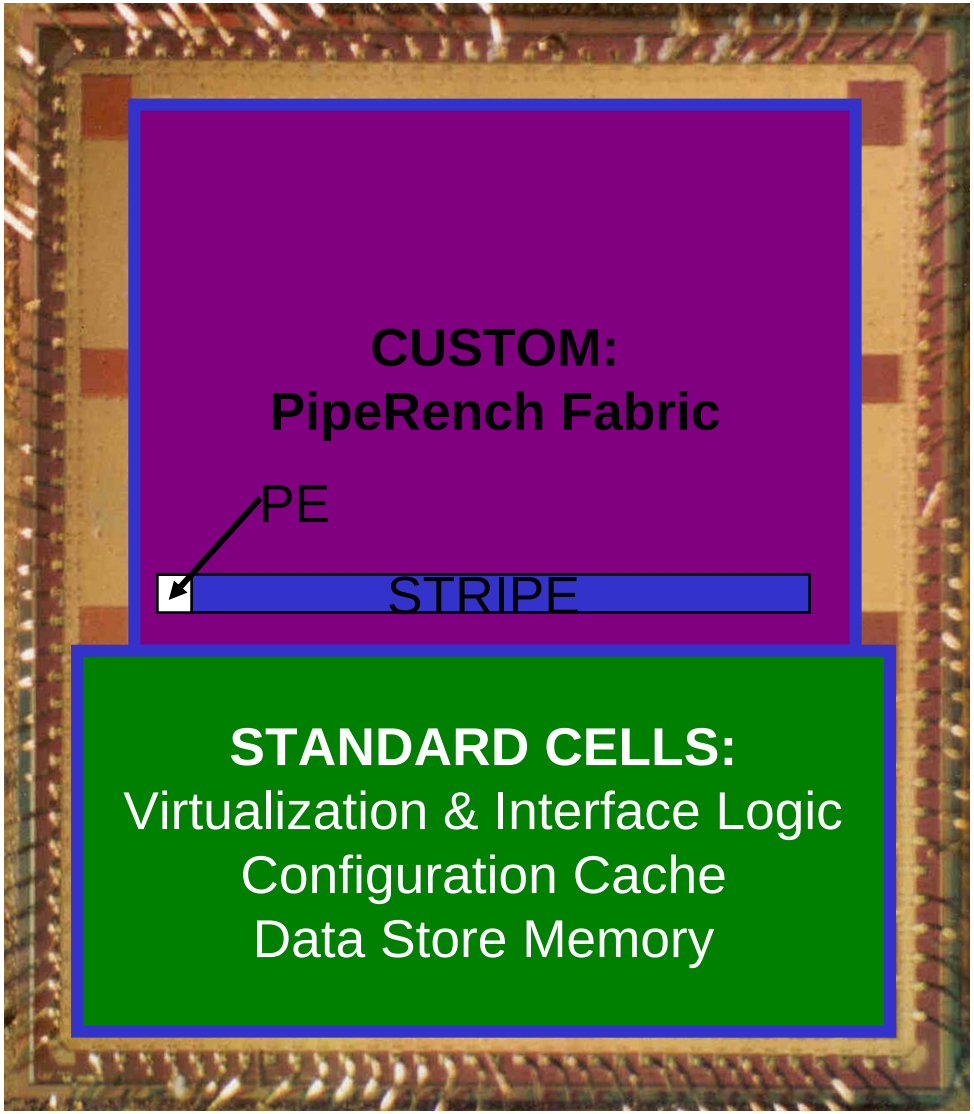
	Kernel	Sun Time* (ms)	Mercury Time** (ms)	PipeRench Time (ms)	PipeRench vs. Sun	PipeRench vs. Mercury
A	Decimate/Correlate	38852	31393	971	40.01	32.33
B	STD Image	34055	54978	430	79.13	127.74
C	Image Stats	2362	144	107	22.03	1.34
D	Image Threshold	1703	586	107	15.88	5.46
E	Clump Binary Image	1524	5450	189	8.08	28.88
F	Clump Maxima	15510	35257	-	-	-
G	Inv. Dis. Filter	56307	26142	1179	47.75	22.17
H	Cue Stats	27849	2125	315	88.55	6.76
I	Cue Thresh	4995	8986	315	15.88	28.57
J	Cue Density/PBC	5935	545	19	311.50	28.61
K	Bright Cue Stats	13844	668	152	90.83	4.38
L	Comp Fractal	13126	25940	152	86.12	170.19
M	Comp Neighbor	4740	6656	610	7.77	10.92
	Aggregate Speedup				**45.16**	**35.99**

* Sun Ultra 5 Workstation w/ 360 MHz UltraSPARC II
** Mercury Computer PPC-G4 AltiVec with 133MHz ASIC

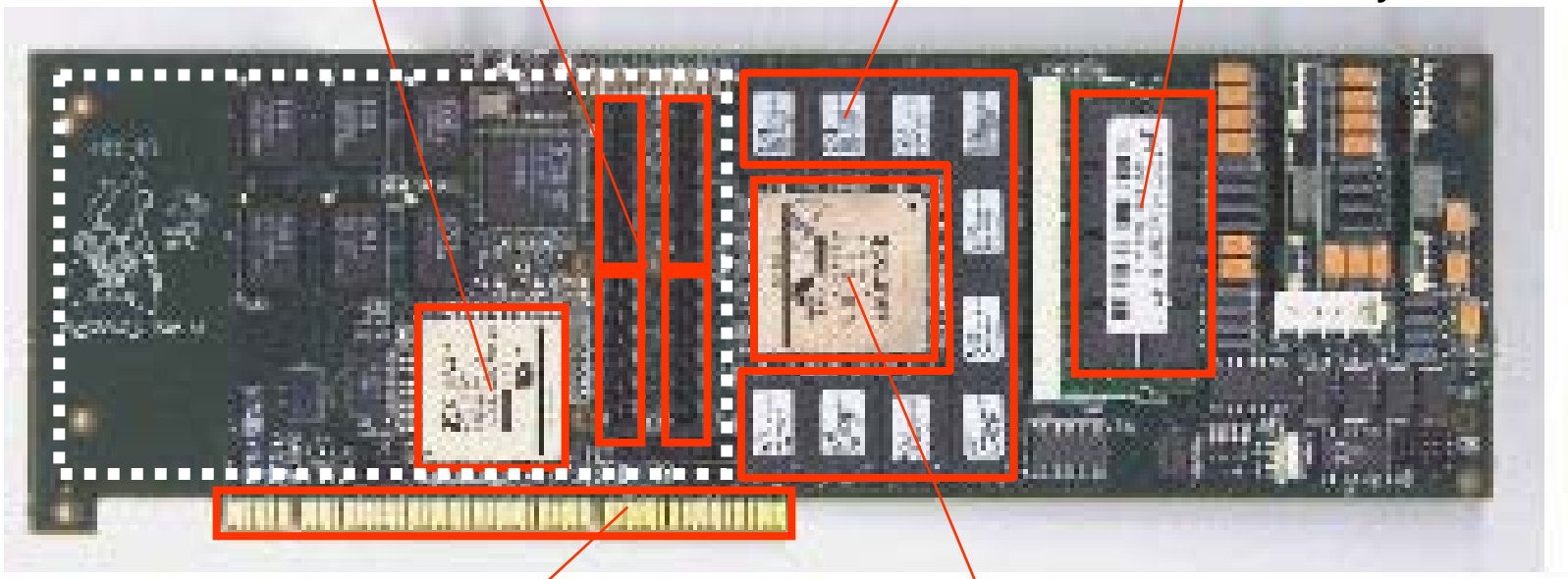
< 10 X Faster
10-100 X Faster
> 100 X faster

PipeRench prototypes

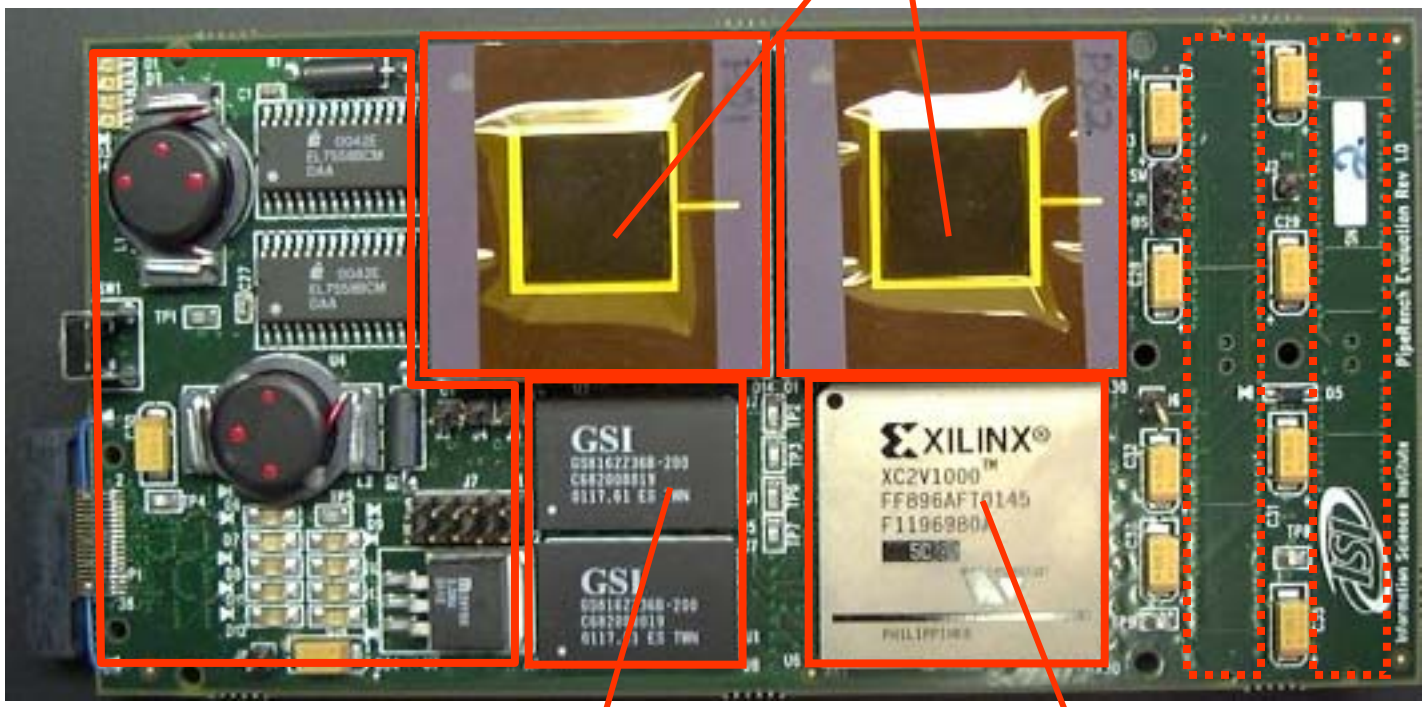
- 3.6M transistors
- Implemented in a commercial 0.18 µ, 6 metal layer technology
- 125 MHz core speed (limited by control logic)
- 66 MHz I/O Speed
- 1.5V core, 3.3V I/O



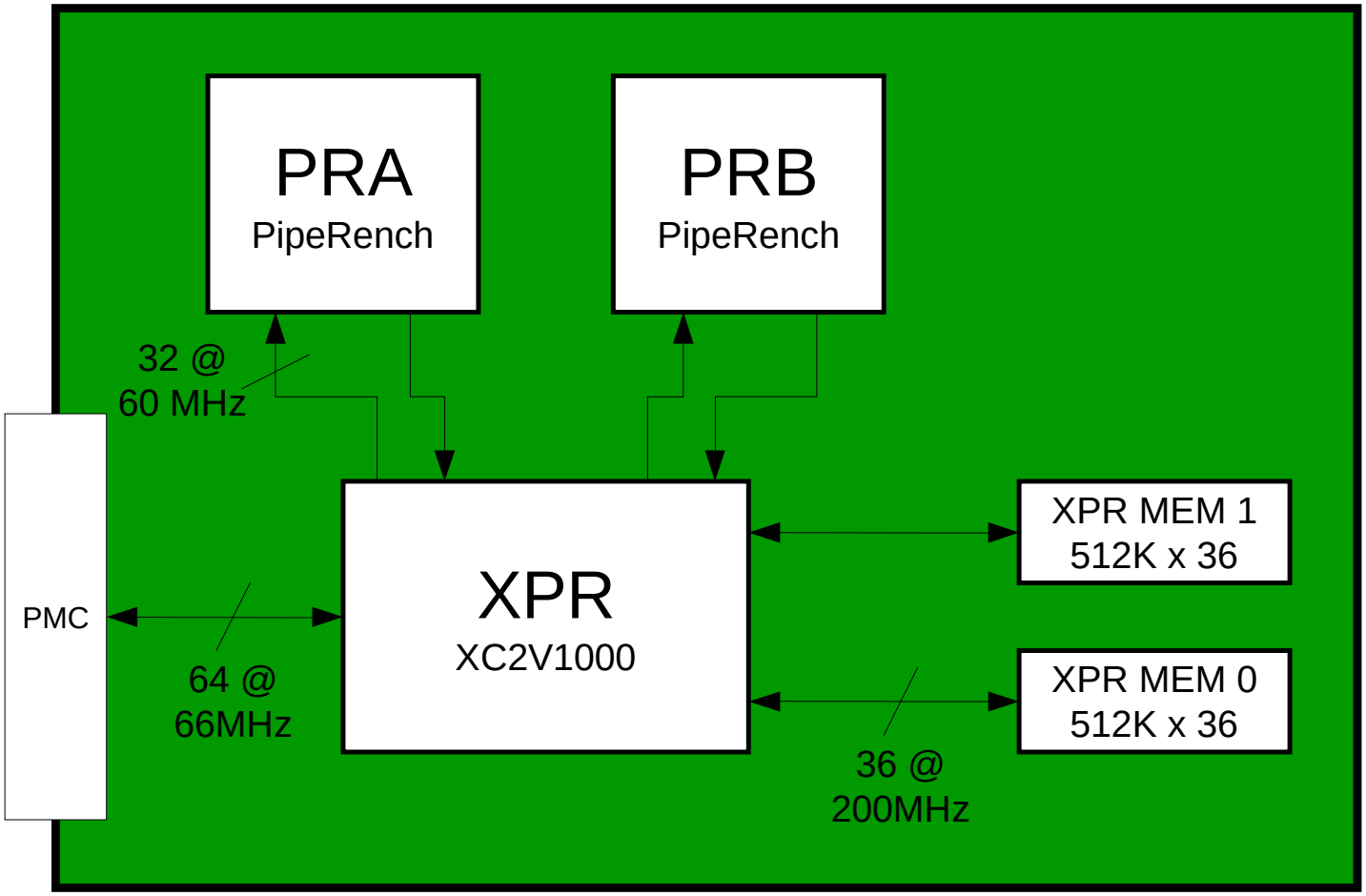
USC ISI Osiris Board



PipeRench Mezzanine Card

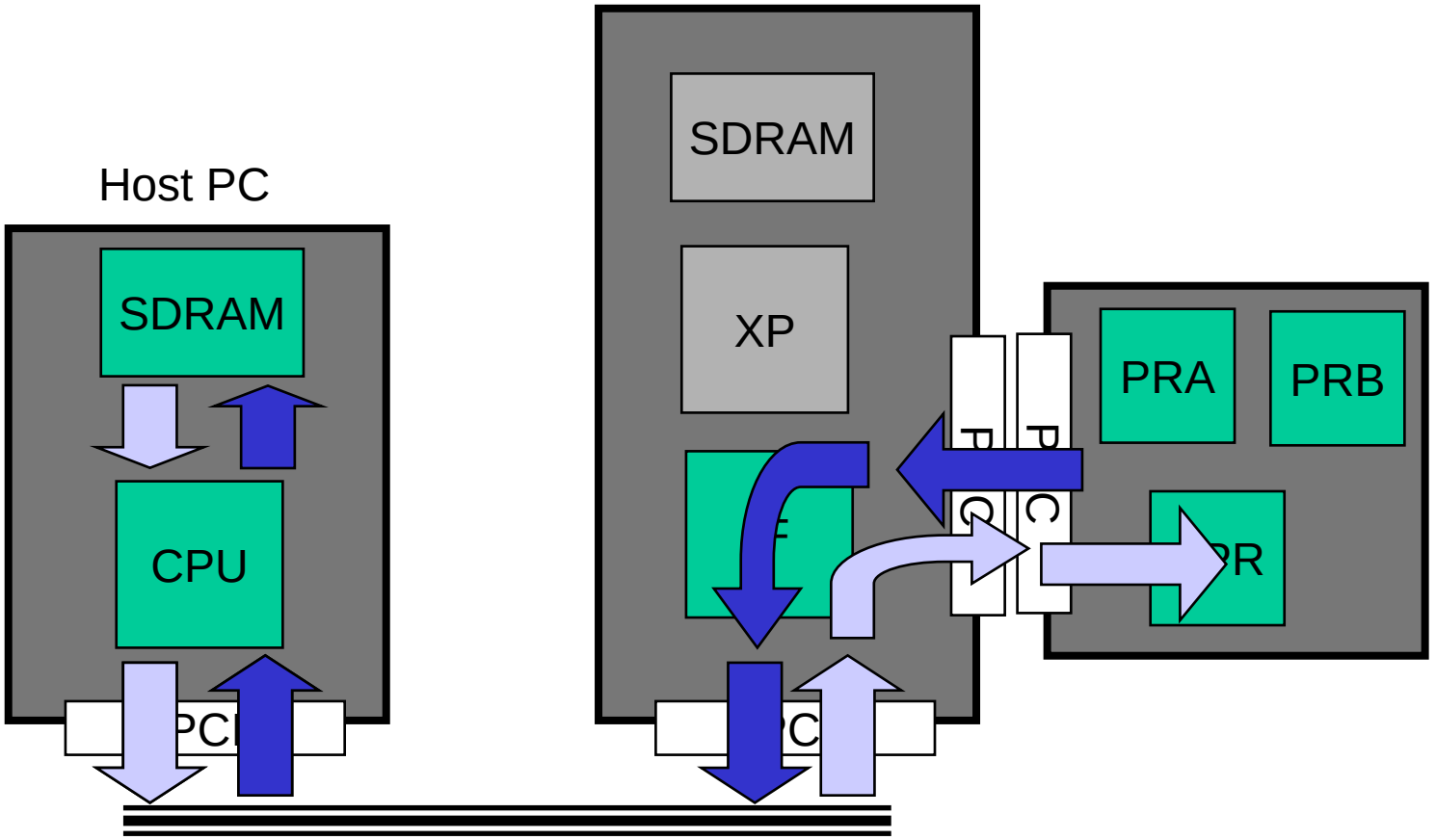


PipeRench Mezzanine Card



Mezzanine Card

Host Driven Operation



Input Image

8 bits/pixel, ~70 MB

Decimation & Correlation

D/C Image

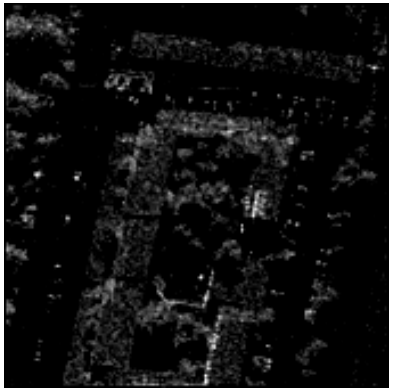
8 bits/pixel, ~8 MB

Local Statistics

Image Statistics

Image Mean and STD

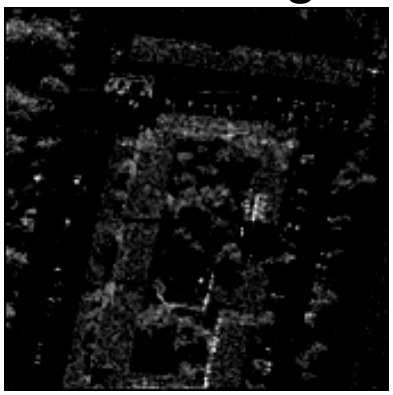
STD Image



8 bits/pixel, ~8 MB

Screener, Stage I

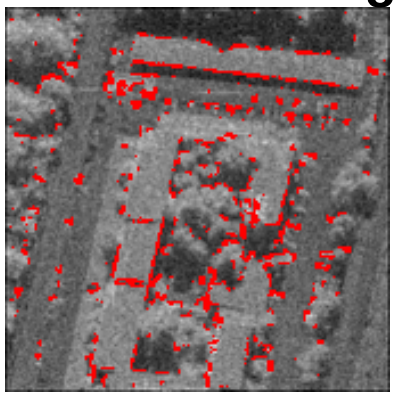
STD Image



8 bits/pixel, ~8 MB

Threshold

Thresholded Image

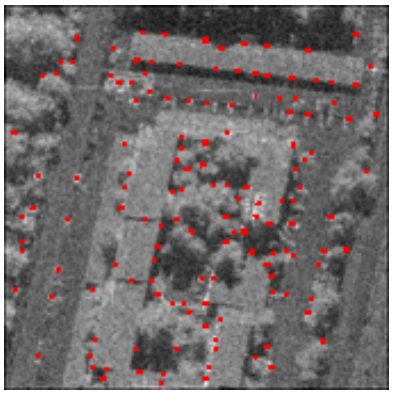


1 bit/pixel, ~1 MB

Image Mean and STD

Clustering

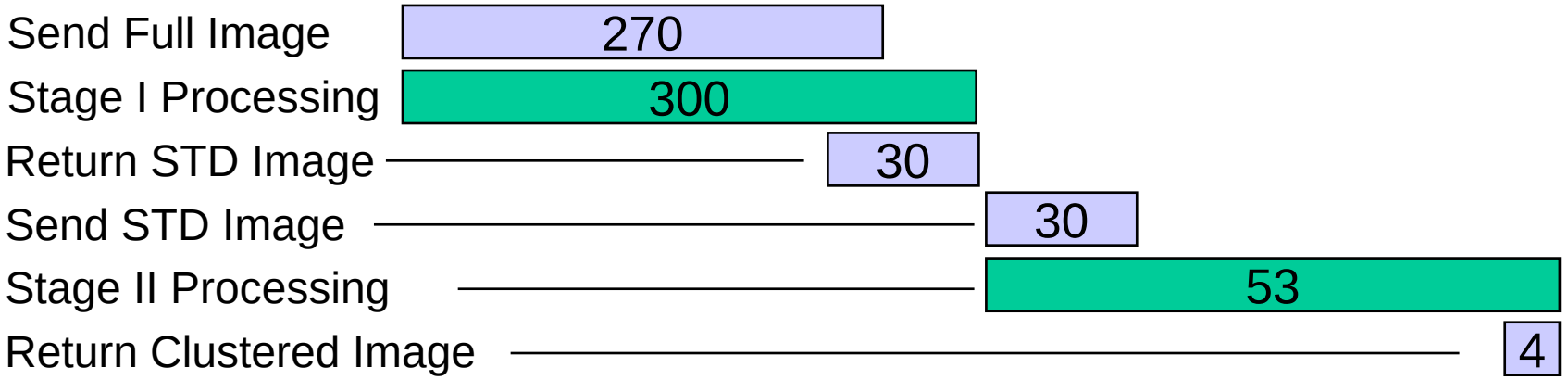
Clustered Image



1 bit/pixel, ~1 MB

Screener, Stage II

Performance



- Hardware Processing Time = 353 ms
  71x speedup vs. Sun Workstation
  15x speedup vs. 1.8 GHz P4 Xeon

Lessons Learned

- Many, if not most, applications map well to PipeRench, even when not specifically designed for it.
- Entering designs in high-level language much easier and preferable to HDL.
- Interfacing reconfigurable computer to host time consuming and error-prone.
- I/O limits PipeRench performance.

Performance Improvements

PipeRench I/O limits performance
(32 bits in / 32 bits out @ 66 MHz)

- 128 bit wide I/O @ 150 MHz I/O speed gives **9x** speedup for total ~**135X** vs 1.8 GHz CPU

Fabric speed and size limit performance:

- 500 MHz fabric clock gives another ~**3x**, for total ~**400x** speedup
- Doubling fabric to 32 stripes gives another 70% speedup, for total ~**680x** speedup.

Future Work

- Expand PipeRench programming model
- Integrate pipelined fabrics with processors :
  HASTE – Hybrid Architectures with A Single, Transformable Executable