



HPEC 2007: Multicore Processors and Their Impact on  
DoD HPEC Systems

# Gedae Portability: From Simulation to DSPs to the Cell Broadband Engine

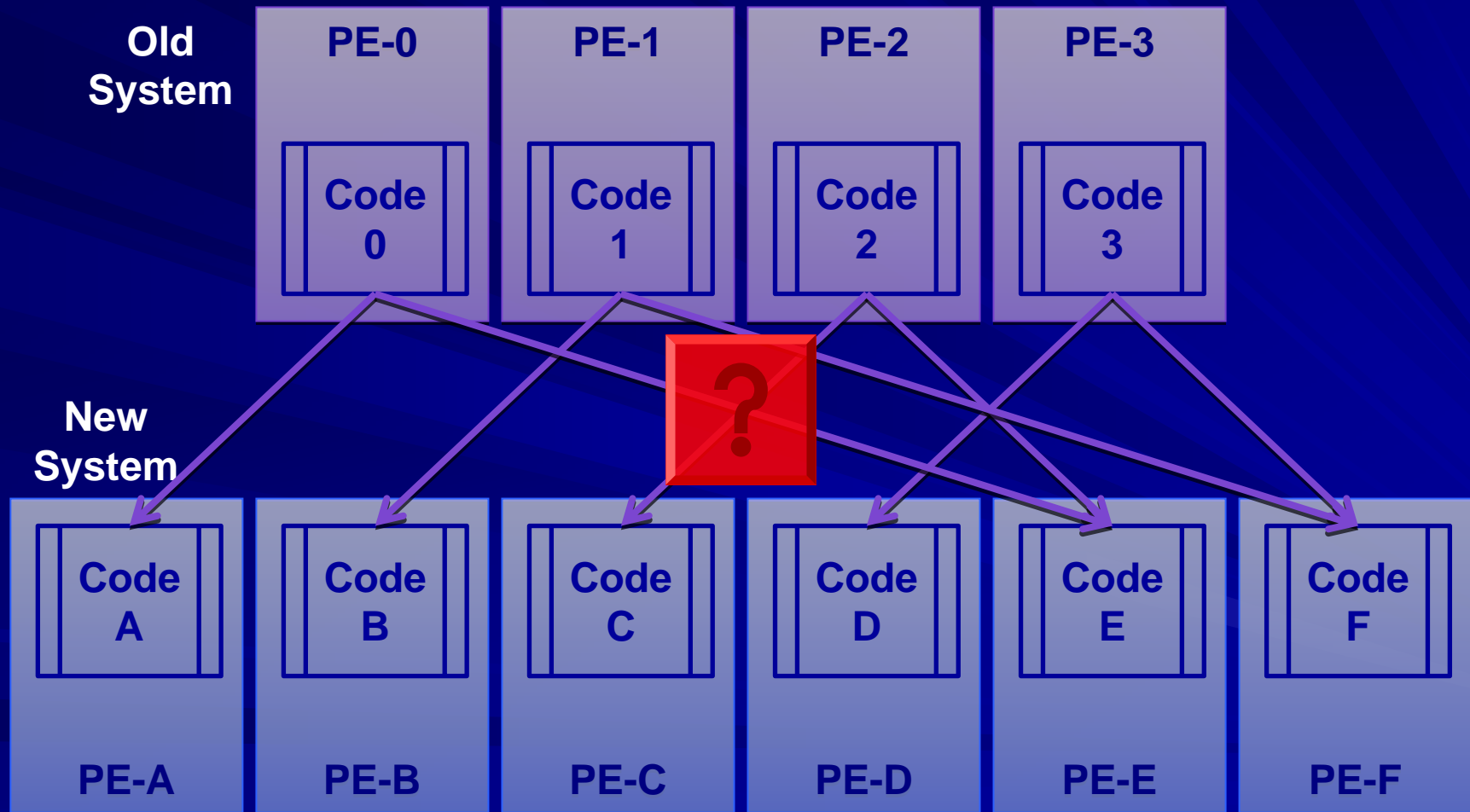
James Steed, William Lundgren, Kerry Barnes

Gedae, Inc.

[www.gedae.com](http://www.gedae.com)

856 - 231- 4458

# The Software Architecture Makes Hardware Refreshes Difficult



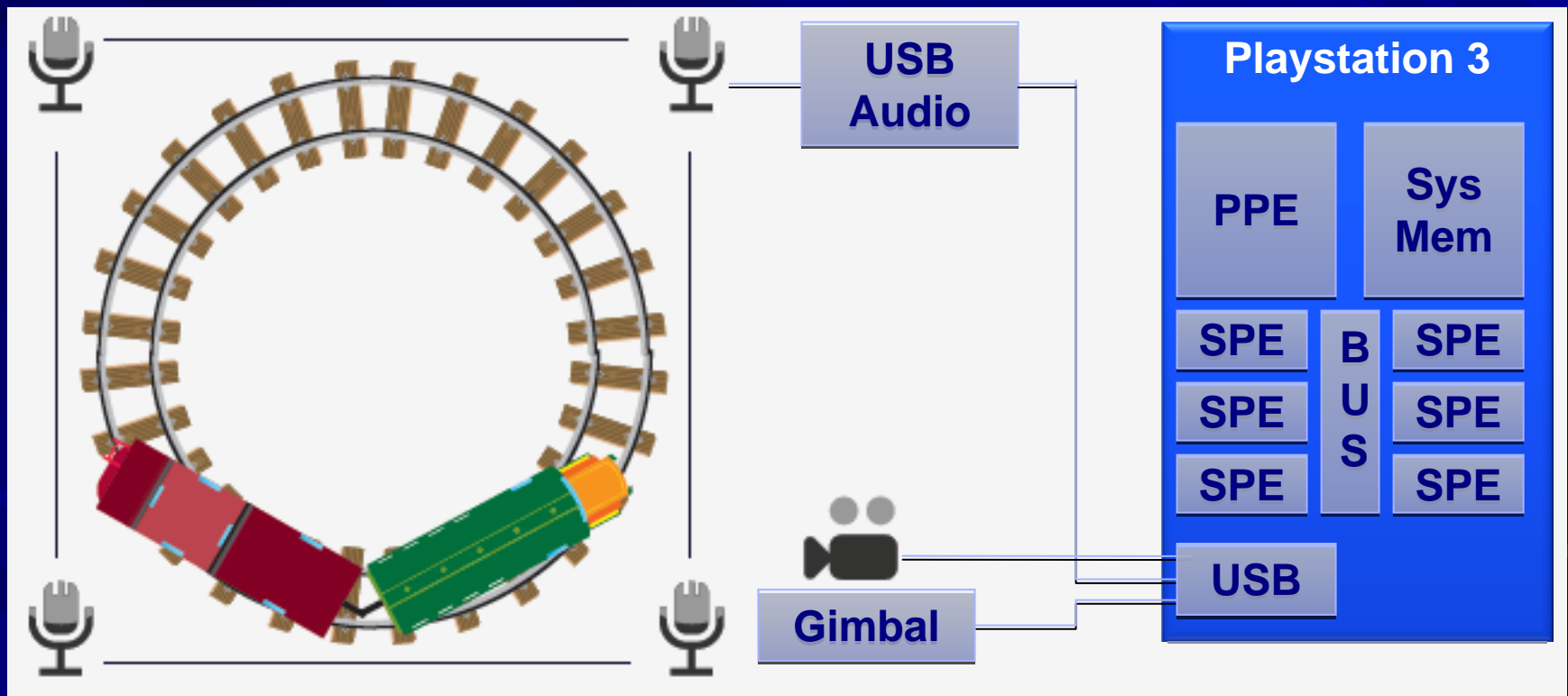


# Problems that Reduce Software Portability

- Languages and compilers are based on serial processors
- Software architecture is buried in the code
- Differences in multiprocessor/multicore hardware necessitate changes to the software architecture
  - Number of processors
  - Interconnection
  - Bandwidth
  - Processor speed
  - Memory size
  - Memory structure
- Gedae mitigates the risk of porting software by automating the incorporation of the software architecture

# Application Environment

- Search and track using four audio channels
- Display using camera directed by pan-tilt unit



# Stages in Development



Developed as  
simulation with file  
input and rendered  
output

Deployed on quad  
PowerPC board,  
processing in real  
time at limited  
frame rate

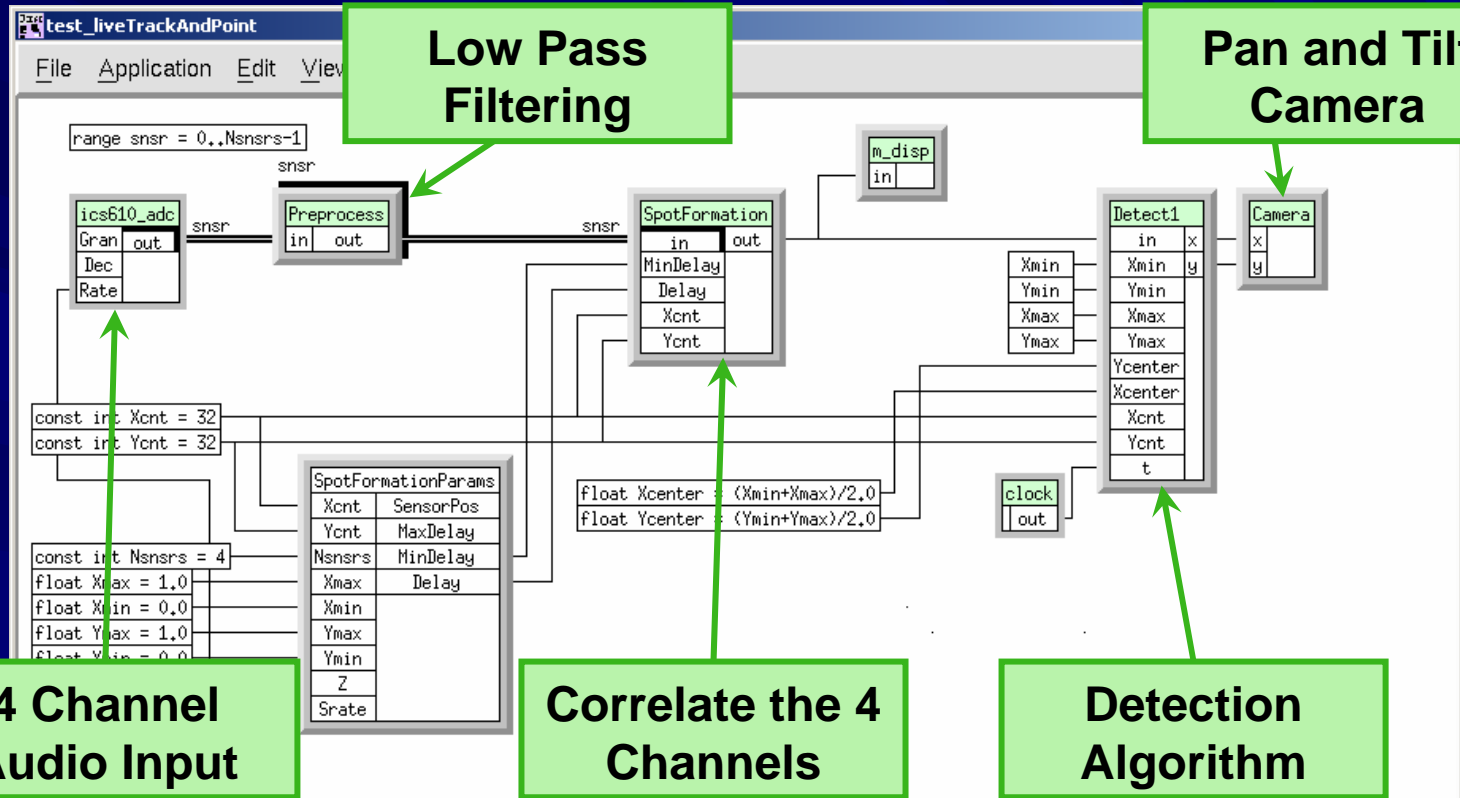
Hardware refresh  
to Cell Broadband  
Engine processor,  
processing more  
frames per second

# System Specifications



	Simulation	Mercury AdapDev	Playstation 3
Processors	1	4 PowerPC AltiVec (500 MHz), 1 Pentium	1 PPE, 6 SPEs
Sensors	Datafile of 4 recorded channels	ICS 610 ADC PCI Board, 4 microphones	M-Audio Quattro USB Device, 4 microphones
Output	Constellation display	Directed Perception D46-17 Pan-Tilt Unit	Directed Perception D46-17 Pan-Tilt Unit
UI	Rendered scene	Matrix Vision BlueFOX USB Camera displayed using Video for Gedae	Matrix Vision BlueFOX USB Camera displayed using Video for Gedae

# Algorithm Specified in Gedae

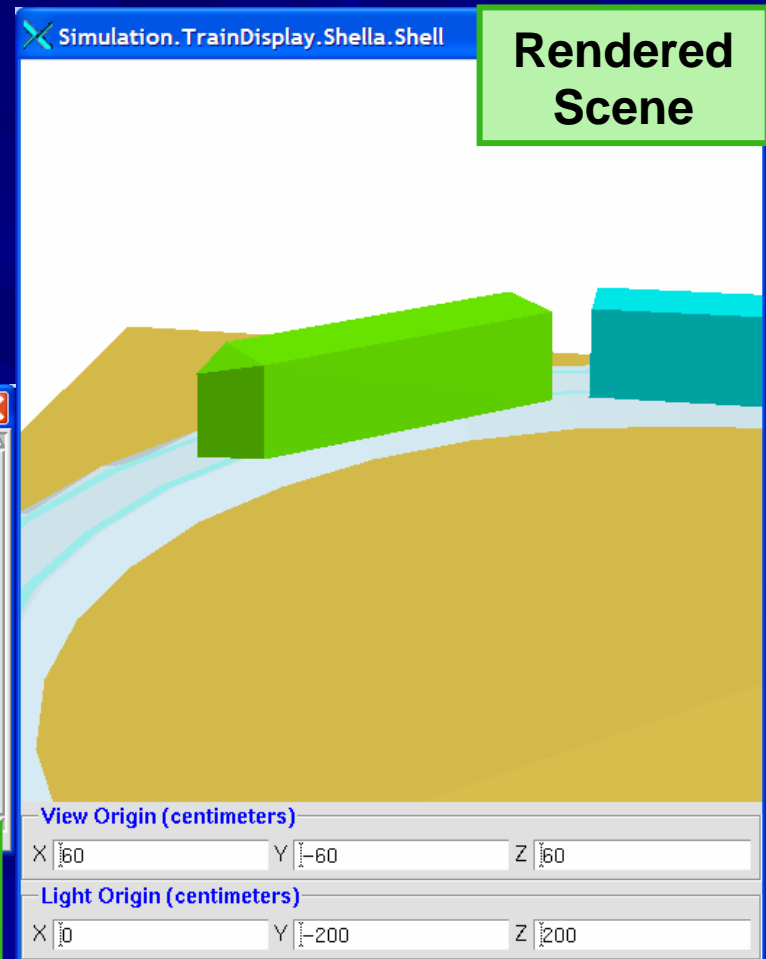
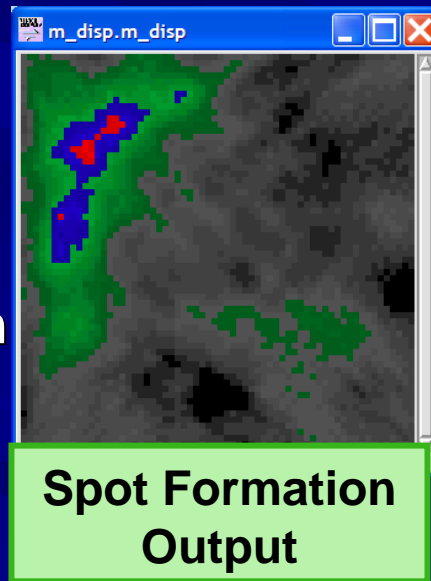


Same flow graph used for simulation, quad DSP board, and Cell/B.E. processor

# Simulation Using Gedae-Sim



- Audio data captured from actual model train, recorded to file
- Simulated 3-d environment created with train, track, camera, and light
- Scene rendered from vantage-point of camera
- Gedae-Sim used to verify algorithm and run on multiple virtual processors



# Mercury AdapDev



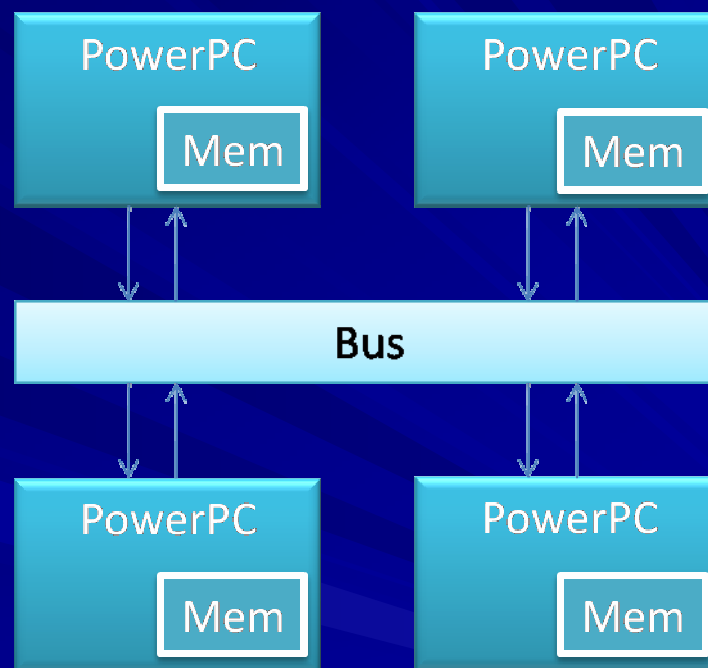
- **Pentium III development host**

- 1.26 GHz
- 1 GB SDRAM

- **Quad PowerPC 500MHz (MCP7410)**

- AltiVec instruction set
- 2 MB L2 cache
- 256 MB SDRAM
- DMA engines

- **RACE++ switched-fabric architecture**



# Mercury AdapDev Implementation



**Map partitions to 4 PowerPCs**

Name	CP ProcNum	System Name	Trace Size	Trace Mer
0	100=\$1+100	mcos_altivec	100000 *	default
1	101=\$1+100	mcos_altivec	100000 *	default
2	102=\$1+100	mcos_altivec	100000 *	default
3	103=\$1+100	mcos_altivec	100000 *	default
default	host	host	100000 *	default

**Put Preprocessing of 4 channels in 4 partitions**

ics610_adc	default	
[0]si_s	0=\$1	1 *
[1]si_s	1=\$1	1 *
[2]si_s	2=\$1	1 *
[3]si_s	3=\$1	1 *
[0]lpf	0=\$1	1 *
[1]lpf	1=\$1	1 *
	2=\$1	1 *
	3=\$1	1 *
	0=\$1	1 *
	1=\$1	1 *
	2=\$1	1 *
	3=\$1	1 *
[0]sub	0=\$1	1 *
[1]sub	1=\$1	1 *
[2]sub	2=\$1	1 *
[3]sub	3=\$1	1 *
SpotFormation	0 *	
m_disp	default	
clock	default	
Detect1	default	
cmplx		
MkViewParams		
neg		
PtuController		

**Nonblocking transfer of audio data from host to PowerPCs**

Name	Id	Source	Dest	Xfer Type	NBsize
[0]si_s<in		host	100	host>mcos_altivec	nb *
[1]si_s<in		host	101	host>mcos_altivec	nb *
[2]si_s<in		host	102	host>mcos_altivec	nb *
[3]si_s<in		host	103	host>mcos_altivec	nb *
Detect1					
m_trackpeak<in		100	host	mcos_altivec>host	0
SpotFormation					
v_sum<[1]in		101	100	dsa_dx *	
v_sum<[2]in		102	100	dsa_dx *	
v_sum<[3]in		103	100	dsa_dx *	

**Strip mine for cache performance**

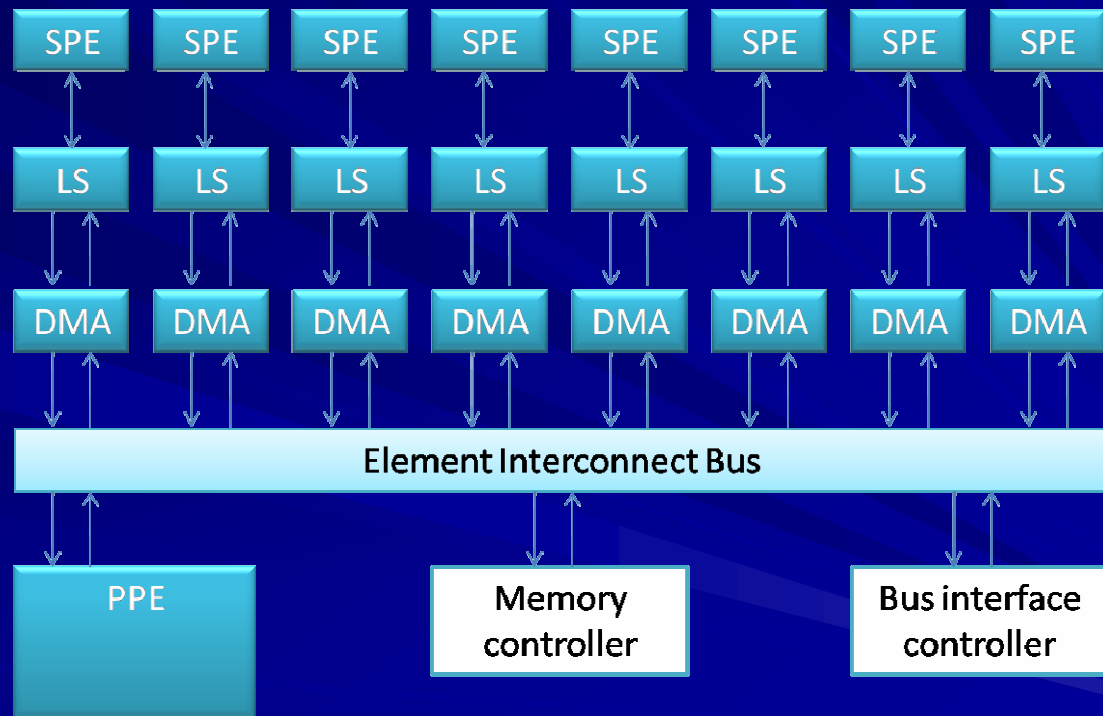
**DMA between processors**

**Add Correlation to 0-th partition**

# Cell/B.E. Architecture



- Power Processing Element (PPE)
- 8 Synergistic Processing Elements (SPE)
  - VMX SIMD instruction set
  - DMA engines
  - 256 kB Local Storage (LS)
- System Memory
- Element Interconnect Bus (EIB)
  - Over 200 GB/s



# Cell/B.E. Implementation

- Alter implementation to use 6 SPEs
- Alter implementation to fit in the SPEs' 256KB Local Storage
- Maximize use of SPEs

**Put Preprocessing of 4 channels in 4 partitions**

Name	Part	subSched
quattro	default	
[0]Preprocess	0=\$1	1
[1]Preprocess	1=\$1	1
[2]Preprocess	2=\$1	1
[3]Preprocess	3=\$1	1
SpotFormation		
[0]s_ovr1_v	0=\$1	1 *
[1]s_ovr1_v	1=\$1	1 *
[2]s_ovr1_v	2=\$1	1 *
[3]s_ovr1_v	3=\$1	1 *
[0]v_selWinterpolateV_v	0=\$1	1 *
[1]v_selWinterpolateV_v	1=\$1	1 *
[2]v_selWinterpolateV_v	2=\$1	1 *
[3]v_selWinterpolateV_v	3=\$1	1 *
v_sum	4 *	1 *
v_sqr	4 *	1 *
v_integrate	5 *	1 *
v_m1	default	

**Strip mine to reduce memory footprint**

**Map partitions to 6 SPEs**

Name	CP ProcNum	System Name	Trace Size
0	100=100+\$1	spu	1000 *
1	101=100+\$1	spu	1000 *
2	102=100+\$1	spu	1000 *
3	103=100+\$1	spu	1000 *
4	104=100+\$1	spu	1000 *
5	105=100+\$1	spu	1000 *
default	host	elinuxppc	10000

**Use 2 SPEs to perform 1st stage of correlation**

# Results



- Gedae was used to easily move the application to new hardware
- Changes to the implementation were handled by automation and simple GUIs, not changes to code
- High performance gains were realized with minimal effort

Target	Programmer Hours	Performance
Simulation	4 weeks	-
Mercury AdapDev	6 hours	3 Hz
Cell/B.E.	2 hours	15 Hz