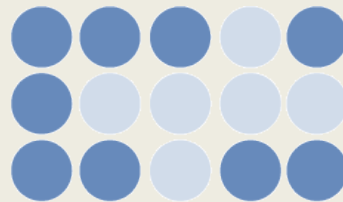


HPEC 2011

Implementation of 2DFFT Processing and Large Dense Matrix Multiply on the TMS320C6678 Processor



Gedae

WWW.GEDAE.COM

KERRY BARNES (KBARNES@GEDAE.COM),
WILLIAM LUNDGREN (WLUNDGREN@GEDAE.COM),
JAMES STEED (JSTEED@GEDAE.COM),
AMON ARNON FRIEDMANN, PH.D. (TI, ARNON@TI.COM),
HECTOR RIVERA (TI, HRIVERA@TI.COM)

HPEC Presentation History



1

- Graphical Programming and Autocoding for Multiprocessor Systems - 1998
- The Expressiveness of the GEDAE Graph Language – 1999
- Implementation of the Intelligent Detector-Tracker Algorithm on Embedded Hardware Connected to a Local Area Network – 2002
- Gedae Runtime Kernel Performance Characterization – 2004
- Gedae: Auto Coding to a Virtual Machine – 2005
- How Code Generation Will Save Moore’s Law – 2006
- The Impact of Programming Difficulty on Hardware Obsolescence – 2007
- Programming Examples that Expose Efficiency Issues for the Cell Broadband Engine Architecture – 2008
- Simple, Efficient, Portable Decomposition of Large Data Sets – 2009
- Implementation of 2-D FFT on the Cell Broadband Engine Architecture – 2010
- Automated Software Cache Management – 2010

"When we start talking about parallelism and ease of use of truly parallel computers, we're talking about a problem that's as hard as any that computer science has faced. I would be panicked if I were in industry."

John Hennessy, Stanford President, Computing Pioneer

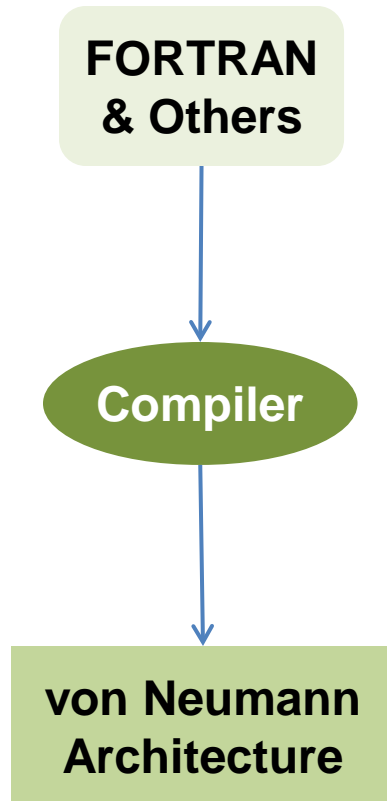
"A Conversation with Hennessy & Patterson," ACM Queue Magazine, 4:10, 1/07

Gedae's technology was developed from first principles to target parallel computers. The effort spanned a quarter century.

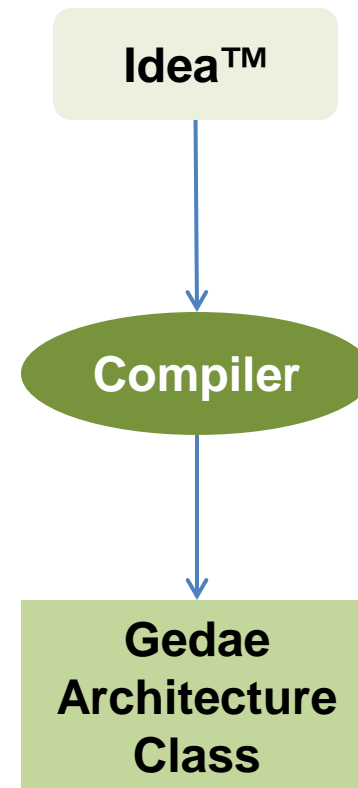
The Concept is Familiar

3

von Neumann



Gedae



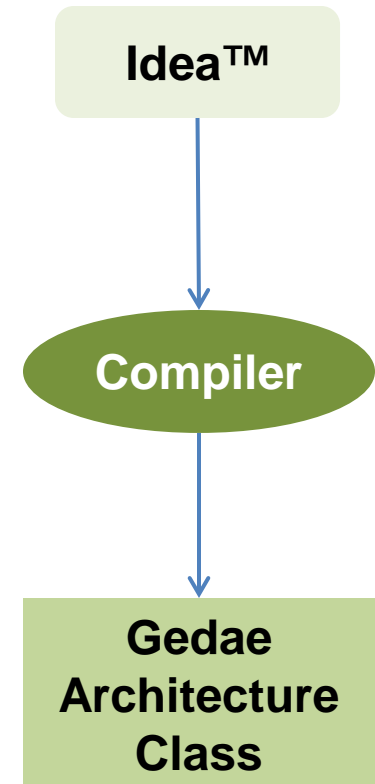
The Language Enables the Compiler



4

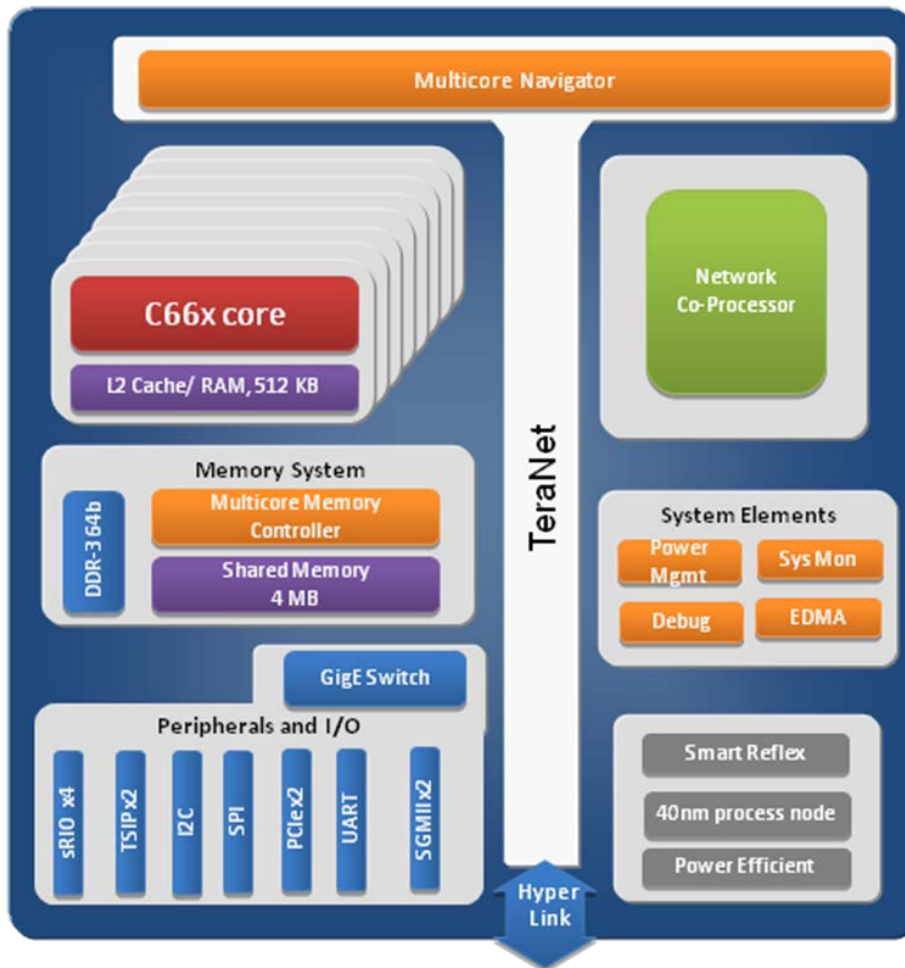
The schedule of distributed execution is important because it dramatically effects resource use and ultimately throughput and power consumption.

Gedae's Idea language allows the compiler to reason about concurrency, resource use and the schedule of execution to produce efficient parallel code that runs on architectures with N CPUs, M memories, K interconnects



TMS320C6678 Architecture

5



- Key Characteristics:**
- *Maximum of 160 gflops*
 - *Maximum memory IO of 16 gB/s*
 - *Power consumption about 10 watts*

Summary of Results



6

Summary of Processing and IO Times

	Theoretical†		Practical†		Measured	
	CPU	Memory	CPU	Memory	CPU	Memory
2D FFT (512x512)	1.48e-4	6.55e-4	3.58e-4	7.28e-4	TBD	TBD
Matrix Mult (4K x 4K)	8.56e-1	1.57e-2	1.64	1.75e-2	TBD	TBD

Summary of GFLOP Sustained

	Theoretical	Practical	Measured
2D FFT (512x512)	100	36.0	TBD
Matrix Mult (4K x 4K)	160	82.5	TBD

†*practical is the rate achievable based on available optimized vector functions*
†*theoretical is the rate achievable based on the balance between adds and multiplies*