The Sandbridge Sandblaster SB3000 Multithreaded CMP Platform

John Glossner, Ph.D., Co-Founder, CTO & EVP

jglossner@SandbridgeTech.com

1 North Lexington Ave, 10th Floor
White Plains, New York 10601
914-287-8500
Agenda

Sandbridge Introduction
- Company Background
- Motivation

Sandblaster Platform
- ISA
- Vector Architecture
- Multithreading
- Saturating arithmetic

Software
- IDE
- Compiler
- Simulator

Communications
- WCDMA, GPRS
- GPS
- 802.11b

Hardware
- Low Power Core
- SB3000
- RF

Applications
- H.264 / MPEG4
- MP3

Summary
Sandbridge Technologies

Fabless Semiconductor Company
developing...

Software reconfigurable

Wireless chipsets for low power applications

E-Gang
Bridging The Gaps
Scott Woolley, 09.01.03

Guenther Weinberger
CEO, Sandbridge Technologies

Guenther Weinberger runs a company that is barely two years old, has only 40 employees and has yet to earn a dime. But, boy, does this guy have big dreams. "I don't want to appear too far away from reality," he says, "but we have the technology to become the next Intel."

What 78,000-employee Intel is to the PC industry, Weinberger thinks his shop, Sandbridge Technologies, can be to the cell phone industry—which spent $20 billion on chips last year.

He aims to achieve this audacious feat by solving one of the most glaring—and annoying—problems for cell customers: the alphabet soup of incompatible standards that hinders a multitude of networks from easily looking up with one another. Today's global traveler must lug different phones for different countries (and endure incompatible voice mail boxes and different phone numbers and bills).
A Whole Industry’s approach failed …

… on advanced wireless systems …

… with multimedia

MPSoC
July 2005
What is it that we eventually carry with us?

It has a color screen, camera, audio and antenna ... … but all features need high computing performance and ultra low power consumption

- **Wireless communication** 2G – 2.5G – 3G – WLAN – BT – etc.
  - GSM/IS-95a,b/IS-136/PDC/ iDEN – CDMA2k/GPRS/EDGE – FDD/TDD/TD-SCDMA/Jap.WCDMA/CDMA2k-3x– 802.11a,b,g
- **Radio broadcast** GPS – radio – TV – etc.
  - Location based services/911/tracking services – AM/FM/DAB – Sat./Terr.TV
- **Encryption – decryption – media encode – media decode**
- **Games – speech to text – natural language processing**
Handset design at C level in min. time

![Bar chart showing simulation speed in millions of instructions per second for different processors.]

**Simulation Speed (1GHz Laptop)**

- SB 24.639
- TI C64x (Code Composer) 0.114
- TI C62x (Code Composer) 0.106
- SC140 (Metrowerks) 0.002
- ADI Blackfin (Visual DSP) 0.013

**Complete system design in C**
- Cycle accurate simulator delivers immediate feedback
- Design for conformance in C

**AMR Encoder**
(out-of-the-box C code)

**System performance by design**
- Parallelizing compiler generates production assembly code
- Multithreading (sea-of-threads) ensures concurrent execution

![Graph showing utilization of different communication protocols and DSPs.]

**Graph showing utilization of different communication protocols and DSPs:**
- SB3000 Utilization
- 802.11b 1/2/5/5.5/11Mbps
- GPS 5m
- Bluetooth Class 10/12
- GPRS 64/384/2k Kbps
- WCDMA

**DSPs**
- SB
- TI C64x
- TI C62x
- SC140
- ADI BlackFin
The Sandbridge Approach ...

SandBlaster™ DSP

- Programmable
- Ultra-low power
- High-performance
- Multithreaded

SandBlaster™ Tools

- Improved productivity C compiler
- 70% reduction in time-to-market
- User-friendly

DSP Platform

- Scalable & Programmable
- Integrated Sandblaster cores
- Up to 2Mbit/sec data rate
- 40,000 RISC MIPS
- Low Cost 0.13um CMOS
- Integrated protocol stack

DSP Ref Design

- Low Cost
- Power Efficient
- Ultra-high performance
- Fully tested / validated
- Dedicated Customer Support
- Flexible and upgradeable

Core technology equally applicable to Networking, Storage, Automotive, GP-DSP, etc.
Architecture
Sandblaster Architecture Performs

- Compilable DSP
  - C programmed
  - Latency hiding architecture
  - System Productivity Advantage
    - 9-12+ months
  - 3G Applications Standard
  - Java Processor
- Control Processor
  - 3G, xDSL, 802.11
  - Control Stacks
Multithreaded Architecture Enables C

Key to Low Power Implementation

Sea of Threads

Thread 0
# Monitoring FCCH&SCH
lu r1,M(r2)
lv r1,M(r2)
call (De_scrambler)
stu r2,M(r2)

Thread 1
# WCDMA Main
Start Thread0
Start Thread1
Start Thread2

Thread 2
# Exit Code
.
.
Start Thread3
.

Thread 3
# WCDMA Handover Code
ctsr r1,sr2
lu r1,M(r2)
.
.
Start Thread4
.

Thread 4
# Pulse shaping Code
lvu vr1,M(r4)
vmacs
vr3,vr1,vr2,wr0
loop lcr0,label(4x)

Thread 5
# FIR Filter
lvu vr1,M(r4)
vmacs
vr3,vr1,vr2,wr0
loop lcr0,label(4x)

Thread 6
# Handover Code
cctsr r1,sr2
lu r1,M(r2)
jc cf0,(UMTS_Mode)

Thread 7
# Monitoring CPICH
lu r1,M(r2)
lv r1,M(r2)
call (De_scrambler)
stu r3,M(r2)

9-12+ months

System Productivity Advantage

Code&Data Sharing Across Threads

Fast Cross Thread Interrupts

Fully Interlocked

Highly Parallel

Hardware Scheduled

Compilable DSP

C programmed

Latency hiding architecture

Java Processor

3G Applications Standard

Control Stacks

Control Processor

C programmed

3G, xDSL, 802.11

9-12+ months

System Productivity Advantage

Code&Data Sharing Across Threads

Fast Cross Thread Interrupts

Fully Interlocked

Highly Parallel

Hardware Scheduled

Compilable DSP

C programmed

Latency hiding architecture

Java Processor

3G Applications Standard

Control Stacks

Control Processor

C programmed

3G, xDSL, 802.11

Key to Low Power Implementation
Parallelism

**Multiple cores (MP)**
- 4 cores

**Multithreaded (TLP)**
- 8 threads/core

**Compound Instructions (ILP)**
- 3 operations out of
  - Integer
  - Load/Store
  - Branch
  - Vector

**Vector (DLP)**
- 4 data parallel operations
Performance

Peak
- 3 operations/cycle
- 16 RISC-ops/cycle
- 4 MACS/cycle

Example
L0: lvu %vr0,%r3,8
|| vmulreds %ac0,%vr0,%vr0,%ac0
|| loop %lc0,L0

- load vector update: 4 16-bit loads + address update
- vector multiply and reduce: 4-16 bit saturating multiplies + 4 32-bit saturating adds
- loop: decrement, compare against zero and branch

20 tap FIR
- 3.92 taps/cycle sustained including automatic multithreading
- ~16 RISC-ops/cycle sustained
Saturating Arithmetic

Many DSP applications require saturating arithmetic

Saturation means

- Results greater than largest representable number are saturated to the largest representable number
- Results less than the smallest representable number are saturated to the smallest representable number

• 4-bit precision example:

\[
A = 0.101 = 0.625 \\
+ B = 0.111 = 0.875 = S = 01.100 = 1.5 \\
\]

\[
A = 1.011 = -0.625 \\
+ B = 1.001 = -0.875 = S = 10.100 = -1.5 \\
\]

\[
S| = 1.100 = -.5 \\
\]

\[
< S > = 0.111 = 0.875 \\
\]

\[
< S > = 1.000 = -1.0 \\
\]
Saturating arithmetic operations are not associative

4-bit precision example

\[ \langle\langle-1.0*-1.0\rangle + \langle0.5*0.5\rangle\rangle + \langle-0.5*0.5\rangle\rangle \]
\[ = \langle\langle0.875 + 0.25\rangle - 0.25\rangle \]
\[ = 0.875 - 0.25 \]
\[ = 0.625 \]

\[ \langle\langle-1.0*-1.0\rangle + \langle0.5*0.5\rangle \rangle + \langle-0.5*0.5\rangle\rangle \]
\[ = \langle0.875 + \langle0.25-0.25\rangle\rangle \]
\[ = 0.875 + 0 \]
\[ = 0.875 \]
Sandbridge Low Power Hardware
Low Power IDLE Instructions

Architecturally possible to

- **Turn off 1 or more processors**
  - All clocks disabled
  - Instruction fetch disabled
  - Memory disabled
  - Memory state not preserved

- **Turn off 1 or more threads within a processor**
  - Clocks disabled on a per thread basis
  - Instruction fetch disabled on a per thread basis
  - Memory state preserved (including registers)
  - Threads awaken via interrupt
Using Multithreading to Optimize for Power

- Multiple cycles to access memory
- Slow memory accesses hidden by multithreading

Multithreading decouples CPU from MEM, combining best power/performance for CPU AND MEM

- Optimized CPU power
- Optimized memory power
- Optimized overall power/performance
## SandBlaster Pipeline

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ld/St</td>
<td>Inst</td>
<td>Dec</td>
<td>RF</td>
<td>Read</td>
<td>Agen</td>
<td>Xfer</td>
<td>Int.</td>
<td>Ext</td>
<td>Mem</td>
<td>Mem</td>
<td>Mem</td>
</tr>
<tr>
<td>ALU</td>
<td>Inst</td>
<td>Dec</td>
<td>Wait</td>
<td>RF</td>
<td>Read</td>
<td>Exec1</td>
<td>Exec2</td>
<td>Xfer</td>
<td>WB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I_Mul</td>
<td>Inst</td>
<td>Dec</td>
<td>Wait</td>
<td>RF</td>
<td>Read</td>
<td>Exec1</td>
<td>Exec2</td>
<td>Exec3</td>
<td>Xfer</td>
<td>WB</td>
<td></td>
</tr>
<tr>
<td>V_Mul</td>
<td>Inst</td>
<td>Dec</td>
<td>RF</td>
<td>Read</td>
<td>MPY1</td>
<td>MPY2</td>
<td>Add1</td>
<td>Add2</td>
<td>Xfer</td>
<td>WB</td>
<td></td>
</tr>
<tr>
<td>V_Mul</td>
<td>Reduce</td>
<td>Dec</td>
<td>RF</td>
<td>Read</td>
<td>MPY1</td>
<td>MPY2</td>
<td>Add1</td>
<td>Add2</td>
<td>RF Rd</td>
<td>Reduce</td>
<td>Reduce</td>
</tr>
</tbody>
</table>

### Staggered Read/Write
- Allows single write-port register files

### Very Long Reduce
- Shifted/Offset against vector pipe
Interlock Checking Hardware

The multithreaded implementation is transparent
  - No interrupt restrictions
  - Load / Branch delay slots not visible

Multithreading hides instruction execution latencies

No interlock checking hardware is required
  - One exception – long loads
A single compound instruction can contain a vector load or store and a vector operation.

```
stvu %vr1, r0, 8 || vmac %vr5, %vr4, %vr3, %ac0
```

- Our design requires a single write port per space
  - int 1R/1R, vec 2R/1W, and acc 1R
- VLIW’s may require 14R/5W for the same computation
- If load, up to 9W simultaneous write ports may be required

Our design staggers Load/Stores in multiple ways:

- **Time staggered**
  - different pipeline stages
- **Spatially staggered**
  - Banked register files
- **Architecturally staggered**
  - Separate architected register spaces (e.g. Integer, Vector, Accumulator)
**Description**

- **0.18um CMOS ASIC**
- **Single DSP Core**
- **SW Programmable**

**Features**

- **External Bus for L2 memory**
- **Internal Inst/Data memory**
- **Control Interfaces: I^2^C, SPI, TDM, A/D, D/A**

**Abbreviations**

- **DSP**
- **Ins & Data Memory**
- **EXT INT**
- **GPIO**
- **SPI**
- **XMTR I/O**
- **RCVR**
- **I/O**
- **I^2^C**
- **SPI**
- **TDM**
- **A/D Data**
- **To/From External Memory**
- **A/D Data**
- **RF Device Control**
- **TDM**
- **Memory Bus**
- **GPIO**
- **System Clk/Cntl**
- **External Bus for L2 memory**
- **Internal Inst/Data memory**
- **Control Interfaces: I^2^C, SPI, TDM, A/D, D/A**
SBTC Digital Card
SB3000 Handset Chip

- General Purpose I/O
- Smart Card Interface
- Sync Serial Port
- Keyboard Interface
- Keypad Interface
- UART/IrDA
- Audio Codec Interface
- General Purpose I/O
- ARM Processor
- DMA Controller
- AHB/APB Bridge
- Peripheral Device Control
- Multi Port Memory Controller
- USB Interface
- LCD Interface
- DSP Complex
  - Ins & Data Mem (32KB / 64KB)
  - EXT INT
  - L2 Mem (256KB)
  - EXT INT
  - DSP
- DSP ⇔ ARM Bridge
- Vector Interrupt Controller
- TAP (JTAG Port)
- Multimedia Card Interface
- Smart Card Interface
- Sync Serial Port
- Keypad Interface
- UART/IrDA
- General Purpose I/O
- Timers
- RTC
- Memory Interface (Synchronous and Asynchronous)
- DSP Local Peripherals
- General Purpose I/O
- Serial Interfaces (SPI, I2C)
- Prog. Timers/Gens
- Parallel Streaming Data Interface
- RF Control
- Timer I/O
- TX Data
- RX Data

- CMOS
- Replicated SBTC core
- Low Power design

MPSoC
July 2005
SB3000 Digital Card
WCDMA 2Mbps Front End Card
CDMA-2k Front End Board
GSM/GPRS Front End Card
802.11b WLAN Front End
Multiplexer Board Block Diagram
Sandbridge Software Tools
DSP Application Complexity

10x Complexity every 10 years
Compiler saves R&D and time-to-market

1. Design Algorithms
2. Map to Fixed Point C
3. Write DSP Specific C
4. Write DSP Assembly
5. Hand Schedule Operations on DSP
6. Final Product

6-9 Months!  6-9 Months!
Compiler saves R&D and time-to-market ...

Sandblaster™ Provides Dramatic Improvement

Design Algorithms

Map to Fixed Point C

Write DSP Specific C

Write DSP Assembly

Compile

Hand Schedule Operations on DSP

Final Product

6-9 Months!

Final Product

6-9 Months!
System Software

Compilation Tools
- Compiler
- Assembler
- Linker
- Loader

Library
- C library
  - Standard C & Math
- Device drivers

Simulator
- Just-in-time
  - Models peripherals
- Cycle-accurate C
- Cycle-accurate VHDL

IDE
- Netbeans based
- Integrated S/W debug

RTOS
- Light-weight kernel
  - Based on POSIX API
  - Filesystem

Test-Cases
- DSP kernels
- DSP applications
- Commercial test-suites
  - Plum-Hall, Perennial, Nullstone, CosY
- Nightly builds

H/W Debugger
- Breakpoint/profile
- JTAG
Sandblaster Tools

- SaDL
- C
- C++
- Java
- Sandblaster Compiler
- Binary Translator
- sb.o
- x86asm
- C
- x86asm
- compiled simulator
- dynamic simulator

MPSoC
July 2005
Compiler Optimizations – Dragon Book +

**DSP Optimizations**
- Saturation Arithmetic
- Fixed Point Semantic Analysis
- Bit-exact ETSI compliance

**Vector Optimizations**
- Vector Loads
- Vector Stores
- Vector Arithmetic
- Vector Reduction
- Saturating Vector Operations

**Multithreaded Optimizations**
- OpenMP
- Automatic Parallelization
- Automatic Multithreading

**Loop Optimizations**
- Loop Invariant Code Motion
- Strength Reduction
- Induction Variable Elimination
- Loop Splitting
- Software Pipelining

**Interprocedural Opts.**
- Constant Propagation
- Memory Disambiguation
- Function Inlining
- Alias Analysis
Sandblaster AMR Results

Programmed in C or Java

- Super-computer class compiler
  - Vectorization
  - DSP instruction generation
- Standard Library
  - `Printf();`
- POSIX pthreads or Java threads
- 50k+ testcases used for validation
  - Industry standards: Plum-Hall, perennial, nullstone

AMR Encoder
(out-of-the-box C code)

<table>
<thead>
<tr>
<th>DSP's</th>
<th>SB</th>
<th>TI C64x</th>
<th>TI C62x</th>
<th>SC140</th>
<th>ADI BlackFin</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHz</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>10 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sandbridge AMR Simulation Results

Compiled Simulator
- JIT “Flash” compilation
- Up to 100 MHz on high end x86
- Multi-threaded supported

Up to 4 orders of magnitude faster
- Dramatic development time reduction
- Significant productivity improvement

Simulation Speed (1GHz Laptop)

<table>
<thead>
<tr>
<th>Processor</th>
<th>Millions of Instructions Per Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandbridge</td>
<td>24.639</td>
</tr>
<tr>
<td>TI C64x (Code Composer)</td>
<td>0.114</td>
</tr>
<tr>
<td>TI C62x (Code Composer)</td>
<td>0.106</td>
</tr>
<tr>
<td>SC140 (MetroWerks)</td>
<td>0.002</td>
</tr>
<tr>
<td>ADI Blackfin (Visual DSP)</td>
<td>0.013</td>
</tr>
</tbody>
</table>

MPSoC
July 2005
Multithreaded Programming

Automatic Multithreading of DSP Kernels
- Compiler can vectorize and multithread
- Uses pthreads as underlying infrastructure

Multiple threads usage via pthreads library
- POSIX API
- Complete support for thread management, synchronization, and communication
- Thread-safe version of the C library
- Entire coding is done in C

Applications multithreading in Java
- Inherently a multithreaded language

Multithreaded H/W with multithreaded S/W
- Automatic mapping of parallelism
- Easy parallel programming methodology
Java Support

Java J2ME implementation
- KVM 1.0 bytecode engine
- CLDC 1.0
- MIDP 1.0 support provided
  - MIDP 2.0 in process
- Multiple Java threads execute on multiple H/W thread units
  - First known hardware multithreaded KVM
  - Sandblaster tools compile KVM with Java-specific optimizations
  - Java is another application on the Sandblaster processor
  - A java thread is scheduled on any available hw thread unit
    - Dynamic number of hardware thread units may be used
  - Synchronization mechanisms fully supported
  - Multithreaded garbage collection supported
Development Environment

Host Platform (Windows, Linux)

Application Software (C)

Compile

Assemble & Link

Performance Analysis

Debug

Simulate

SW Tool Kit

Target Platform

SandBlaster™ Dev. Board

Integrated Dev. Environment
Integrated Development Environment (IDE)

Based on Java open source netbeans

- **Enhanced with**
  - C compilation and editing tools
  - Source debugger
  - Project management
  - Scripting languages

Automatic Error recognition
Works in multiple languages too!
Variables, Threads and Memory View in Debugger
Communications Systems Implementation
Integration

- MMI
- PROTOCOL STACK
- L1 CONTROL
- L1 BASEBAND SW
- IF
- RF
- APPLICATION TASKS
- DATA I/O
- LCD, KPD …
Real-time WCDMA Performance

Real-time chip, bit, and symbol rate processing
- 1 SB9600 chip for 2Mbps Rx concurrently with 768kbps Tx
- <75% utilization for 384kbps Rx / 384kbps Tx

Includes functions traditionally implemented in H/W
- Turbo Decoder
- Rake Receiver
- Tx/Rx Filters
Summary

Multithreaded baseband processor
- High-performance and low-power
- DSP, Java, and Control processing

Sophisticated compiler technology
- Automatically generates DSP operations
- Automatically multithreads applications
- Hand coded performance

Reconfigurable Communications Protocols
- WCDMA, GSM, GPRS, etc.
- 802.11b, Bluetooth, etc.

Multimedia capability
- MP3
- MPEG4