The Sandbridge SDR Convergence Processor Platform

John Glossner, Ph.D., Co-Founder, CTO & EVP
Daniel Iancu, Ph.D., Chief Communications Architect
Tanuj Raja, V.P. Business Development

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

Hardware
- Low Power Core
- SB3000
- RF

Software
- IDE
- Compiler
- Simulator

Communications
- WCDMA, GPRS
- GPS
- 802.11b

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

Guenter Weinberger
CEO, Sandbridge Technologies

Guenter 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 hooking 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 …

... with multimedia

... on advanced wireless systems ...

Hanyang University
7th Annual Workshop on SDR and Smart Antennas
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

**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

**Graphs**
- % SB3000 Utilization
- MHz Utilization

**SB**, **TI C64x**, **TI C62x**, **SC140**, **ADI BlackFin**

Hanyang University
7th Annual Workshop on SDR and Smart Antennas
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

- Java Processor
  - 3G Applications Standard

- 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)
- lu r1,M(r2)
- call (De_scrambler)
- stu r3,M(r2)

Thread 7
- Monitoring CPICH
- lu r1,M(r2)
- lu r1,M(r2)
- call (De_scrambler)

Thread 6
- Handover Code
- ctsr r1,sr2
- lu r1,M(r2)
- jc cf0,(UMTS_Mode)

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

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

Thread 3
- WCDMA Handover Code
- ctsr r1,sr2
- lu r1,M(r2)
- jc cf0,(WCDMA_Mode)

Thread 2
- Exit Code
- Start Thread33

Thread 1
- WCDMA Main
- Start Thread0
- Start Thread1
- Start Thread2
- Start Thread13
- Start Thread14
- loop lc1,0xf0

Code & Data Sharing Across Threads

System Productivity Advantage

9-12+ months

Java Processor

C Programmable

Control Processor

3G Applications Standard

WCDMA, xDSL, 802.11

Control Stacks

C Programmable

Latency hiding architecture

Highly Interlocked

Fully Interlocked

Fast Cross Thread Interrupts

Highly Parallel

Sea of Threads

Multithreaded Architecture Enables C

Hanyang University
7th Annual Workshop on SDR and Smart Antennas
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:

\[
\begin{align*}
A &= 0.101 = 0.625 & A &= 1.011 = -0.625 \\
+ B &= 0.111 = 0.875 & + B &= 1.001 = -0.875 \\
= S &= 01.100 = 1.5 & = S &= 10.100 = -1.5 \\
S| &= 1.100 = -0.5 & S| &= 0.100 = +0.5 \\
<S> &= 0.111 = 0.875 & <S> &= 1.000 = -1.0
\end{align*}
\]
Saturating Arithmetic (2)

Saturating arithmetic operations are not associative

4-bit precision example

$$\langle\langle-1.0\times-1.0\rangle + \langle0.5\times0.5\rangle\rangle + \langle-0.5\times0.5\rangle$$

$$= \langle\langle0.875 + 0.25\rangle - 0.25\rangle$$

$$= 0.875 - 0.25$$

$$= 0.625$$

$$\langle\langle-1.0\times-1.0\rangle + \langle0.5\times0.5\rangle + \langle-0.5\times0.5\rangle\rangle$$

$$= \langle0.875 + \langle0.25-0.25\rangle\rangle$$

$$= 0.875 + 0$$

$$= 0.875$$
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 Dec</td>
<td>RF Read</td>
<td>Agen</td>
<td>Xfer</td>
<td>Int. Ext</td>
<td>Mem 0</td>
<td>Mem 1</td>
<td>Mem 2</td>
<td>WB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALU</td>
<td>Inst Dec</td>
<td>Wait</td>
<td>RF Read</td>
<td>Exec1</td>
<td>Exec2</td>
<td>Xfer</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I_Mul</td>
<td>Inst Dec</td>
<td>Wait</td>
<td>RF Read</td>
<td>Exec1</td>
<td>Exec2</td>
<td>Exec3</td>
<td>Xfer</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V_Mul</td>
<td>Inst Dec</td>
<td>RF Read</td>
<td>MPY1</td>
<td>MPY2</td>
<td>Add1</td>
<td>Add2</td>
<td>Xfer</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V_Mul Reduce</td>
<td>Inst Dec</td>
<td>RF Read</td>
<td>MPY1</td>
<td>MPY2</td>
<td>Add1</td>
<td>Add2 RF Rd</td>
<td>Reduce 1</td>
<td>Reduce 2</td>
<td>Reduce 3</td>
<td>Reduce 4</td>
<td>WB</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)
0.18um CMOS ASIC
Single DSP Core
SW Programmable

External Bus for L2 memory
Internal Inst/Data memory
Control Interfaces: $I^2$C, SPI, TDM, A/D, D/A
SB3000 Handset Chip

- CMOS
- Replicated SBTC core
- Low Power design
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
Audio Board
Sandbridge Software Tools
DSP Application Complexity

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

- Design Algorithms
- Map to Fixed Point C
- Write DSP Specific C
- Write DSP Assembly
- Hand Schedule Operations on DSP
- Final Product

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

Sandblaster™ Provides Dramatic Improvement
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

- C
- C++
- Java

Sandblaster Compiler

SaDL

Binary Translator

Compiled simulator

Dynamic simulator

x86 asm

C

x86 asm

sb.o

Binary Translator
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 Opt.**
- Constant Propagation
- Memory Disambiguation
- Function Inlining
- Alias Analysis
## Compiler Comparison

<table>
<thead>
<tr>
<th></th>
<th>Sandbridge</th>
<th>Intel</th>
<th>TI</th>
<th>Starcore</th>
<th>ADI</th>
<th>ZSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foundation</td>
<td>Internal</td>
<td>internal</td>
<td>Tartan</td>
<td>Metroworks</td>
<td>Edinborough</td>
<td>GNU</td>
</tr>
<tr>
<td>Dragon Book</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ILP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Inner Vectorization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Outer Vectorization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Loop rerolling</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Integer Linear Prog. Alias Analysis</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Memory Hierarchy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>OpenMP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Automatic Multithreading</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Compiler error tracking</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>DSP</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>100</td>
<td>100</td>
<td></td>
<td>700</td>
</tr>
<tr>
<td>200</td>
<td></td>
<td>200</td>
<td>200</td>
<td></td>
<td>600</td>
</tr>
<tr>
<td>300</td>
<td></td>
<td>300</td>
<td>300</td>
<td></td>
<td>550</td>
</tr>
<tr>
<td>400</td>
<td></td>
<td>400</td>
<td>400</td>
<td></td>
<td>500</td>
</tr>
<tr>
<td>500</td>
<td></td>
<td>500</td>
<td>500</td>
<td></td>
<td>450</td>
</tr>
<tr>
<td>600</td>
<td></td>
<td>600</td>
<td>600</td>
<td></td>
<td>400</td>
</tr>
<tr>
<td>700</td>
<td></td>
<td>700</td>
<td>700</td>
<td></td>
<td>350</td>
</tr>
</tbody>
</table>

---

Hanyang University
7th Annual Workshop on SDR and Smart Antennas
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
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

- **Application Software (C)**
  - **Compile**
    - **Performance Analysis**
      - **Debug**
        - **Simulate**
          - **Assemble & Link**
            - **Host Platform (Windows, Linux)**
              - **Target Platform**

- **SW Tool Kit**
- **SandBlaster™ Dev. Board**
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
Disassembly & Registers View in Debugger
Snapshot of Profile Information
Communications Systems Implementation
Integration

- MMI
- APPLICATION TASKS
- PROTOCOL STACK
- L1 CONTROL
- L1 BASEBAND SW
- DATA I/O
  - LCD, KPD ...
- IF
- RF

Hanyang University
7th Annual Workshop on SDR and Smart Antennas
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
Communications Results

% SB3010 Utilization

- **802.11b**: 1/2/5.5/11Mbps
- **GPS**: 75m .5sec xyz 5m .1sec xyz
- **AM/FM**
- **Bluetooth**
- **GPRS**: Class 10/12
- **WCDMA**: 64/384/2k Kbps
Multimedia Results

% SB3010 Utilization

<table>
<thead>
<tr>
<th>Format</th>
<th>H.264 dec</th>
<th>MPEG enc</th>
<th>MPEG dec</th>
<th>MP3 dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>QCIF</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1/2/3</td>
</tr>
<tr>
<td>QCIF/CIF</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1/2/3</td>
</tr>
<tr>
<td>11/22/32</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1/2/3</td>
</tr>
</tbody>
</table>

H.264 dec: H.264 decoding
MPEG enc: MPEG encoding
MPEG dec: MPEG decoding
MP3 dec: MP3 decoding

Hanyang University
7th Annual Workshop on SDR and Smart Antennas
Capabilities

End use integration

Design Integration Capability
- RF Integration
- Card Design

Key Sandbridge Expertise
- Software Systems
  - Compilers
  - Compiled Simulators
- Computer Architecture
  - DSP / Processor Design
  - Java Execution
- Wireless Communications
  - Transmission Systems Algorithm Design
  - 3G
- Low Power VLSI Design
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
LIVE DEMO

Stop by our booth to see a live working SDR