Berkeley Emulation Engine

Berkeley Wireless Research Center
-Chen Chang, Kimmo Kuusilinna, Brian Richards, Kevin Camera,
Nathan Chan, Allen Chan, Robert W. Brodersen

What’s BEE?
- A real-time FPGA-based hardware emulator, with speed up to 60 MHz
- Emulation capacity of 10 Million ASIC gate-equivalents per module, corresponding to 600 Gops (16-bit adds).
- 2400 external parallel I/O providing 192 Gbps raw bandwidth.
- Automated design flow from Simulink to FPGA emulation, integrated with INSECTA ASIC design flow.

BEE Applications
- Real-time hardware emulation:
  - Novel Communication Systems with analog front-end hardware (MCMA, UWB, 60GHz)
  - Digital signal processing systems
  - Real-time control systems
  - Neuron-like network processing
- Hardware acceleration:
  - Large communication/signal processing system simulation
  - Hardware-in-the-loop cosimulation with software system
  - Complex parallel computing algorithms

The BEE Design Environment

BEE System Assembly
-Riser I/O Card
-MPB
-StrongARM Module
-Linux OS

Main Processing Board
-Local Mesh
-48 bit buses
-56 bit inter-Xbar buses
**Hardware Performance**

- Board-level Main Clock Rate: 160MHz+
- On Board connection speed:
  - FPGA to FPGA: 100MHz
  - XBAR to XBAR: 70MHz
- Off board connection speed: [3 ft SCSI cable loop back through riser card]
  - LVTTL: 40MHz
  - LVDS: 160MHz ~ 220MHz

**Hardware Capacity**

- Reference Design:
  - 10240 tap FIR filter
  - 512 taps per FPGA
- Slice utilization: 99% of 19200 slices
- Max Clock Rate: 28.5MHz
- ASIC Gate: 401K per FPGA, 8M total
- MOPS: 583,680 total (16bit add & 12bit cmult)
- Power: 2.5W per FPGA, 50W total

**Design Flow Goals**

- Fully automatic generation of FPGA and ASIC implementations from Simulink system level design
- Cycle accurate bit-true functional level equivalency between ASIC & BEE implementation
- Fast design turn-around time
  - Chip-in-a-Day
  - BEE-in-an-Hour

**Design Flow: Global Perspective**

Virtual Components $\rightarrow$ BEE Compiler (System Generator) $\rightarrow$ Performance Estimation

- Simulink Schematics
- VHDL Netlist
- CORE, VHDL Descriptions $\rightarrow$ FPGA Backend Flow $\rightarrow$ Xilinx Bitstream
- MC, VHDL Descriptions $\rightarrow$ ASIC Flow $\rightarrow$ GDSII

**Design Flow: Detailed View**

**Virtual Component Library**

- Parameterized system level blocks:
  - Bit-width
  - Pipeline stages (latency)
  - Output bits truncation
- Customizable block set library
  - Different Architecture
  - Different Technology Target
Basic Blocks

Communication & DSP Blocks

Control Logic Design

Run-time Data I/O Interface

Data I/O Interface: Hardware

Data I/O Interface: Software

- **Basic Blocks**
  - Shifter
  - VHDL
  - Concat
  - Enable
  - Const
  - Counter
  - Delay
  - Mux
  - Down
  - P to S
  - Convert
  - ReInt
  - S to P
  - Sync
  - Slice
  - Up Smp
  - Register
  - FIFO
  - DPRAM
  - ROM
  - RAM
  - Accum
  - CMult
  - AddSub
  - Inverter
  - Logical
  - Negate
  - Mult
  - Relat'n
  - Scale
  - Sin Cos
  - Shift
  - Thresh

- **Communication & DSP Blocks**
  - FIR
  - Shift
  - CIC
  - DDS
  - Puncture
  - Depuncture
  - Conv. Encoder

- **Control Logic Design**
  - Simulink level: StateFlow diagram, encapsulated in a subsystem with Xilinx gateways
  - RTL VHDL automatically generated by SF2VHD
  - Fully integrated with the BEE ISE tools

- **Run-time Data I/O Interface**
  - New and improved infrastructure for transferring data to and from the BEE
    - Control all data transfers from within a local Matlab GUI
    - Accepts standard Simulink data structures for intrinsic reuse of existing test vectors
    - Library macro contains the entire hardware interface in one fully parameterized block

- **Data I/O Interface: Hardware**
  - Pin Gateways
  - Bus Protocol Controller
  - Source RAM
  - Sink RAM

- **Data I/O Interface: Software**
  - Specify input source, BEE hostname, and data bus parameters in Matlab GUI
  - Utilizes a custom MEX socket library for network connectivity
  - Uses a simple packet header to distinguish control frames and byte streams
  - StrongARM (running embedded Linux) starts a persistent, lightweight server
  - Matlab clients connect via TCP and either send a data stream or read request
  - Incoming data is translated into the hardware protocol and broadcast to FPGA
ASIC Flow: INSECTA

- Tcl/Tk code drives the flow
- Same scripting language used by several EDA tools: First Encounter, Nanoroute, ModelSim, Synopsys
- GUI controls technology selection, parameter selection, flow sequencing
- A real “Push Button” flow...
- Users can refine flow-generated scripts

ASIC Tool Flow: Placement

- Internally developed ASIC flow:
  - First Encounter (FE)
  - Nanoroute
  - Physical Compiler
- Timing Driven!
  - FE provides accurate wire parasitic estimates
  - Placement by FE or Physical Compiler

ASIC Flow: Routing in 130nm

- Nanoroute: Ready for 130nm, 90nm designs
- Stepped metal pitches
- Minimum area rules
- Complex VIA rules
- Avoids antenna rule violations
- Cross-talk avoidance: to be evaluated

ASIC Flow: Back-end

- Using Unicad backend directly for DRC, LVS, Antenna rule checking
- Easier to track technology updates from ST.
- Critical for evaluating internally developed technology files for FE, Nanoroute

BCJR MAP Decoder

- E2PR4 Channel Encoder - Decoder
- Fully enclosed design
- Uniform RNG input vector
- Channel encoder
- AWGN filter
- Channel decoder
- BER collection mechanism
- Part of: Full 3G Turbo Decoder

BCJR As Case Study

- 13.2 MHz system clock
- SNR 14db → -1db
- 10^9 Samples
- 20 minute run-time
FPGA Implementation of a Narrow-Band Transmission System

**Purpose**
- BEE Design Flow
- Measurements for MCMA RF Front-End Specification

**Data Rate**
- 1.8 Mbit/s, 500 Kbit/s

**Carrier Freq.**
- 2.45 GHz

**Bandwidth**
- 1.8 MHz

**Modulation**
- DBPSK,OOK

**Frame Synch.**
- PN Sequence

---

2.4GHz Base-band Transmitter

**CPU time:** 57 min
- Core Utilization: 0.344418 (Pad limited)
- Size (From SoC Encounter):
  - Core Height: 565.8 u
  - Core Width: 489.54 u
  - Die Height: 1322.66 u
  - Die Width: 1242.3 u

**Synopsys estimates:**
- Total Dynamic Power = 610.5163 uW (100%)
- Cell Leakage Power = 15.9364 uW
- Critical path: 9.21 ns

---

How to get started?

- **Documentation web site**
  - [http://bwrc.eecs.berkeley.edu/Research/BEE](http://bwrc.eecs.berkeley.edu/Research/BEE)
- **Tutorials**
  - Lesson 1: Flow Basics
  - Lesson 2: Runtime Debug on BEE
  - Lesson 3: Control Logic Design
  - Lesson 4: Run-time Data I/O on BEE

---

BEE Compiler Framework

- **Increase Design Scalability**
  - High-level blocks
  - Vector Signals
- **Reduce design time**
  - Faster run time
  - Efficient/partial synthesis
  - Modular design reuse
- **Feature additions**
  - Tri-state pads/signal support
  - Global pad assignment
  - Automatic design partition
  - Script based hardware generator