Bluespec Product Status and Direction

MIT Bluespec Workshop
August 13, 2007
A long time ago…

A tool…

Content….

Templates….

Community….

Source: Arvind
Making things faster and easier

A long time ago…

Verilog
VHDL
Threads
SystemC

Content….

Templates….

Community….

AMBA AXI®

Open Content:

Designs
Tools
Templates
Where we are and where we’re going

- Bluespec status
- Bluespec product direction
- Growing the community
Product Status

• What’s relatively new:
  – Multiple clock domain (MCD) completion
  – Bluesim simulator (including MCD)
  – AzureIP Foundation Library: specifically, bus fabrics and EVE transactors

• 08-2007 Bluespec release due shortly
  – Will be posted on new Forums – please register!
    • [http://www.bluespec.com/forum/](http://www.bluespec.com/forum/) (or available off support area)
  – Highlights:
    • Bluesim
      • BSV-to-SystemC (Beta)
    • Rule scheduling attribute enhancements
    • Lots more enhancements & bug fixes
AzureIP™ foundation libraries:
ESL… only faster!

Main Categories
(Modules, Data Types & Functions):

Connectivity – transactional I/Fs
Storage – FIFOs, registers, …
Math – fixed point, complex, …
Aggregation – vectors, lists
FSM – finite state machine language for sequential, parallel, conditional, loop structures
Clock Domains (MCD) – clock synchronizers, …
Utility – completion buffer, random number, …

NEW! Bus Fabric – AMBA® AXI® & AHB and OCP
NEW! EVE – transactors, models

Just in the Aggregation Library:

Data types:

- vector – data type holding elements of one type
- list – similar to vector, but used when element numbers

Over 125 functions. A sampling:

- newVector – generates vector of undefined elements
- genVector – creates vector of its from 0 to n-1
- replicate – creates vector based on single element
- genWith – applies function to vector
- cons – adds element to vector
- append – combines two vectors into a third
- concat – combines vector of vectors into one vector
- [i] or select – extract (an) element(s)
- update – updates element in vector
- head – extracts head of vector
- last – extracts tail of vector
- take – takes number of elements from start of vector
- takeTail – create new vector by dropping head element(s)
- takeAt – take number of elements starting at any position
- rotate – move the first element to the last and shift rest
- rotateBy – shift each element n places and rotate
- shiftsA0 – shift in at head and drop tail
- shiftsA1 – shift in at index N and drop tail
- reverse – reverses element order
- transpose – matrix transposition of vector of vectors
- transposLN – matrix transposition of vector of lists
- elem – check if a value is an element of a vector
- any – test if predicate holds for any element of vector
- all – test if predicate function holds for all elements of vector
- findElem – returns index of element which equals value
- findIndex – returns index of element which satisfies predicate function
- rotateBitsBy – shifts bits left in bit-vector
- countOnes – returns number of elements equal to one in bit-vector
- countLeadingZeros – returns number of leading zeros in bit-vector
- zip – combines two vectors into vector of tuples
- unzip – separate vector of pairs into two vectors
- map – map a function over a vector, returning new vector
- zipWith – combine two vectors with a function
- fold – reduce a vector by applying function over all its elements, using binary tree
- mapPairs – map a function over a vector consuming two elements at a time
- joinActions – join number of actions together
- joinRules – join number of rules together
- scan – apply function over vector, creating new vector result, start at highest
- mapAccumL – map a function, but pass accumulator from head to tail
- mapAccumR – takes monadic function and a vector and applies function to elements
- mapM_ – mapM, but throws away vector leaving action

For example
Bluespec AzureIP™ for Bus Fabrics

Standard bus protocols, AMBA® AXI® & AHB and OCP, abstracted to…

Designers interact with simple Get/Put transactional I/Fs

…high-level transactions & data types
Accuracy and Speed Early: Accelerating Architecture, Modeling, Implementation and Verification

---

**SW**

- **C++ testbench/host SW/applications**
  - Running on host
  - Transactional Interface (e.g. OCP)
  - Transactor (SW)

**HW**

- **Transactor (HW)**
  - Running on ZeBu emulator or HDL simulator
  - Transactional or bit/cycle accurate interface (e.g. OCP)
  - HW Testbench

---

**Bluespec for synthesizable:**
- Transactors (transaction-level, high-level types)
- Testbenches (FSM, atomic stimulus, random number, ..)
- Models (e.g. traffic gen/memory/high-level customer designs)
- Implementations (e.g. AMBA® AXI®, OCP/customer block/DMA)
Bluespec Product Direction

• SoC demonstration platform initiative:
  – SoC composition
  – Synthesizable models <-> implementations
  – Legacy IP integration

• Ongoing enhancements to compiler and simulator
  – E.g. Bluesim performance

• AzureIP library: OCP next… then…

• Enhanced design experience:
  – Source level debug: high-level types & rules
  – Integrated design, debug & simulation
Growing the Community

• University program update:
  – Now providing free tools to universities
  – Forms on the website under Partnerships and Affiliations
  – Please spread the word!

• Discussion Forums: http://www.bluespec.com/forum/

Please encourage participation:
both questions and responses

• Working on:
  – A Wiki! Goal: support both Bluespec content & community content
  – Open source designs
Would love to hear ideas, suggestions and feedback:

George Harper

george.harper@bluespec.com

781-250-2207