It is not just about the cores
Overview

• The picoArray concept
  – architecture overview
  – picoBus structure
• Tool flow
  – Probes
• Conclusions
The picoArray concept

- Targeted at wireless communications applications
- Why highly parallel hardware?
  - Wireless systems have great deal of parallelism
  - Software design simpler than hardware
  - Software defined gives flexibility
  - Scalable solution
- Replacement for DSP, ASIC, FPGA combinations
  - Single architecture
  - Single development environment
- Inter-process communications is fixed at compile time
  - No run-time arbitration
  - Deterministic
The picoArray concept: Architecture overview
The picoArray concept: Interconnected picoArrays

- Several tens of picoArrays may be connected together and programmed as one group.
The picoArray concept: picoBus
Tool chain

- VHDL parser
- ANSI C Compiler (gcc)
- Cycle accurate simulator
- Design partitioning
- Place and Switch (Plastic)
- Network checking
- Debugging
  - Simulation
  - Hardware acceleration
Tool chain cont.
Programming picoArrays

Input language is a mixture of:

- C/ASM – Used to program individual processes
- VHDL – Used to connect processes together using signals

All processes and signals are allocated at compile time.

Processes communicate over signals using PUT/GET in ASM, and builtin (intrinsic) functions in C.
VHDL example

entity Producer is
  port (decodedData:out integer32 @8);
end entity Producer;

architecture ASM of Producer is
begin
  MEM
    CODE
      COPY.0 0,R0 \ COPY.1 1,R1
    loopStart:
      PUT R[0,1],decodedData \ ADD.0 R0,1,R0
      BRA loopStart
    ENDCODE;
end;

Entity Consumer is
  port (decodedData:in integer32 @8);
end;

architecture C of Consumer is
begin
  STAN
    CODE
      long array[10];

    int main() {
      int i = 0;
      while (1) {
        array[i] = getdecodedData();
        i = (i + 1) % 10;
      }
      return 0;
    }
ENDCODE;
end Consumer;
VHDL example cont.

use work.all;

entity Example is
end;

architecture STRUCTURAL of Example is
  signal dataChannel: integer32 @8;
begin
  producerObject: entity Producer
    port map
    (decodedData=>dataChannel);
  consumerObject: entity Consumer
    port map
    (decodedData=>dataChannel);
end;

• Work.all is rest of current file
• STRUCTURAL defines connectivity
• Port map actually assigns signals to ports
C Compiler

- Based on GNU Compiler Collection (GCC)
- ANSI/ISO standard C
- Scheduling of LIW uses DFA scheduling algorithm
- Can be used to generate stand-alone libraries
- Communications supported using special functions
- Supports intrinsics e.g. BREV
Simulation

- Models entire systems including peripherals
- Cycle accuracy possible with back annotation from Plastic
  - Signal timings
  - Inter picoArray timings
- Used to verify rtl design to ensure working silicon
Design Partitioning

- Partitions design between multiple chips (manual)
- Automatically splits signals which cross chip boundaries
- Peripherals must be placed on specific processors
- Output provided for Plastic in the form of
  - Tcl command file (one per chip)
  - Segmented design file (one per chip)
Plastic (**Place and Switch to IC**)  
- Works on a single chip at a time  
- Automatically places processes on processors  
- Automatically switches (routes) signals between processors  
- Attempts to minimise overall bus usage  
- Manual operation is possible but difficult  
- Output is a load file
Network checker

- Analogous to an LVS tool
- Recreates netlist from port and switch configurations
- Works at chip level
  - Checks the connectivity of load file produced by Plastic
  - Checks for bus clashes.
  - Check signals are mapped onto valid ports.
  - Check switch settings are sensible
- Works at system level
  - Checks inter PC102 connections
Debugging and verifying picoArray systems

Differs to debugging and verifying sequential, or small scale parallel systems in the following ways.

• Scale
  - Thousands of processes and signals
  - System-wide debug and verification rather than process-centric

• H/W support
  - Silicon area best used for computation. Keep support to a minimum to allow more processors to be fitted onto a device.
  - System-wide debug and verification, rather than processor-centric

• Embedded environment

• Communications and synchronisation
  - Deterministic interconnect fabric – the picoBus
  - No runtime arbitration – removes source of possible bugs
Probes

- AEs are used to “spy” on communications in order to gather useful data.
- This approach is non-intrusive and has no impact on the performance of signal processing blocks.
- Relies on signals ability to be one-to-many.
- The term “Probe AE” is used to describe an AE whose sole function is to gather data.
- These are not special purpose pieces of hardware.
Probes : In action
Conclusions

• picoArray concept gives scalable, software defined systems
• Rapid development due to
  - deterministic communications
  - single programming environment
• Integrated tool set
• Probes provide non-invasive debugging and monitoring