Design and Verification Methodology of Modern High-Speed Switches

- A C++ Modeling Approach -

F. Abel · IBM Zurich Research Lab.

Outline

- Packet-Routing Switch Overview
- Design Methodology
  - C++ Behavioral Model
- Principles of Validation
  - Unified Simulation Environment
- Results
- Conclusions
Packet-Routing Switch - Overview

Outline

- Packet-Routing Switch Overview
  - Design Methodology
    - C++ Behavioral Model
  - Principles of Validation
    - Unified Simulation Environment
- Results
- Conclusions
Design Complexity

- High degree of parallelism
- Short clock cycle (4ns)
- OEM oriented
  - Four expansion modes
  - Multiple configurations
- Time to market, Cost ...

Requirements

Design & Validation Environment

- Fast design exploration
- Continuous refinement
- Fast simulation/validation

C++ Based Design Methodology

Disadvantages
- Switch design
  - Manual translation in HDL
- Testcases/Testbenches
  - Manual conversion
- Maintain 2 models

Decisions
- Automated translation
  - "stylized VHDL" coding
- Unified environment
  - Used at all levels
- C++ clock accurate model
C++ Behavior Model - Entity Declaration

```cpp
entity(InpCtrl)
    // Itf with the Application Registers (ARG)
    input ( boolean, ARG_StandbyReset );
    input ( boolean, ARG_MasterPort );
    input ( boolean, ARG_ControllerEnable );
    inputArray ( boolean, ARG_OQueueEnable, 32 );
    inputArray ( boolean, ARG_BMFilterMask, 32 );
    ...
    // Itf with Input Frame Processing (IFRAM)
    input ( boolean, IFRAM_DiscardThisCell );
    input ( int, IFRAM_RXCountIn );
    input ( int, IFRAM_RowCountIn );
    ...
    // Itf with the On Chip Monitor Registers (ORG)
    output ( boolean, ICTRL_NoAddrError );
    ...
    // Itf with Packet Memory (PktMem)
    output ( boolean, ICTRL_INSAddr );
    ...
    // Itf with On Chip Monitor (OCM)
    // For beh model only. Used to gather some stat.
    input ( p_oGlobalChipData, ARG_GlobData );
};
```

architecture(default, InpCtrl)

```cpp
begin
    ... boolean Receiving;
    boolean IdleCell;
    byte Qualifier;
    boolean ASAInvalid;
    uLong BMHeader;
    ...
    void ClockTick(int itfnr) { wait until clock'event and clock='1';
        int ByteCounter;
        ...
        if (sread(ARG_StandbyReset) == TRUE) { // Reset Sequence
            return;
        } ...
        ByteCounter = sread(IFRAM_RXCountIn);
        if ((RowCounter == 0) && sread(ARG_MasterPort)) {
            ...
            swrite(ICTRL_NoAddrError) = TRUE;
            ...
        }
    }
end_architecture(default, InpCtrl);
select_architecture(default, InpCtrl);
```

SAME2000, Oct.
else if (Qualifier != 0xCC) {
    if (ASAInvalid == TRUE) {
        if (DataCell == TRUE) {
            write(ICTRL_NoAddrError) = TRUE;
            Error("No ASA Address available!");
        }
    }
}

// "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" 

/ / B e h a v i o r  m o d e l  o n l y :  p e r f o r m  s o m e  s t a t .
/ / - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
if ((ByteCounter==7) && (Receiving==TRUE)) {
    sread(ARG_GlobData)->TotNrOfPktsReceived++;
    sread(ARG_GlobData)->TimeStamps[ASA|RowCnt] = (g_SEClockCycle -
    ByteCounter);
} ... 
... 
end_architecture(default, InpCtrl);
select_architecture(default, InpCtrl);

Outline

► Packet-Routing Switch Overview
► Design Methodology
  ■ C++ Behavioral Model
► Principles of Validation
  ■ Unified Simulation Environment
► Results
► Conclusions
Validation Environment
- Black-box simulations
- Randomized testcases

Outline
- Packet-Routing Switch Overview
- Design Methodology
  - C++ Behavioral Model
- Principles of Validation
  - Unified Simulation Environment
- Results
- Conclusions

SAME2000, Oct.
Results (1/3)

- Simulation Time: RS/6000 · 375MHz processor · 2GB of memory

<table>
<thead>
<tr>
<th></th>
<th>C++ Beh.</th>
<th>VHDL/RTL</th>
<th>Gate Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>te_connect</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>time</td>
<td>43s</td>
<td>12'52s</td>
<td>9H55'47s</td>
</tr>
<tr>
<td>speed</td>
<td>277</td>
<td>15.5</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Simulation time and speed (in clock cycles/second)

- Validation Environment
  - 160,000 lines of C/C++ ~ 5,000 lines of Shell scripts
    - Resources = 1 Architect + 1 Software Engineer
  - 100 testcases
    - Resources = 5 to 8 Testcase writers
    - Regression = 5000 testcase runs

Results (2/3)

- Test coverage & Randomized testcases
- Distributed Regression
  - 50 workstations => 5000 testcase runs in 12 hours
- Simulation Processes

<table>
<thead>
<tr>
<th></th>
<th>C++ Beh.</th>
<th>VHDL/RTL</th>
<th>Gate Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 PSR64G chip</td>
<td>6.8</td>
<td>150</td>
<td>650</td>
</tr>
<tr>
<td>2 PSR64G chips</td>
<td>10</td>
<td>200</td>
<td>913</td>
</tr>
</tbody>
</table>

Size of the simulation process (in MB)
Results (3/3)

- 2 Successful ASIC designs: PRS28,4G & PRS64G
  - Fast simulation time
  - Pin-out simulation & Randomized testcases
  - Quality of the C++ to VHDL translation
- Design cycle # 18 months

Outline

- Packet-Routing Switch Overview
- Design Methodology
  - C++ Behavioral Model
- Principles of Validation
  - Unified Simulation Environment
- Results
- Conclusions
Conclusions

- Most of the HW design and verification of a packet-routing switch can be handled at a C++ level
- C++ to model hardware
  - High design productivity and fast simulation
  - Faster simulation => Significant reduction of the design cycle
  - More time for architectural investigations and optimizations
- Unified Validation Environment
  - C++ based => Faster to develop and execute
  - Faster VHDL validation => Env. is ready and debugged
  - More time for synthesis and real error corrections
- Directions for Future Research
  - Hardware synthesis directly from C++

Design and Verification Methodology of Modern High-Speed Switches

- PRS28.4G:
- PRS64G:

F. Abel - IBM Zurich Research Lab.
http://www.zurich.ibm.com