

CS/CoE 536

Reconfigurable System On Chip Design

Lecture 17 : Final Project Integration

(Revision 2)

Washington University
Fall 2002

<http://www.arl.wustl.edu/~lockwood/class/cs536/>

Chris Neely, Chris Zuver, John Lockwood

Copyright 2002
Lockwood@arl.wustl.edu

Project Testing

- You should test your project in hardware
- test server
 - <http://fpx2.arl.wustl.edu/cs536>
 - On-line now
- Testing procedure
 - Upload your bitfile
 - Upload your test traffic
 - Your INPUT_CELLS.DAT will be converted to suitable format for hardware tests.
 - Wait statements will be ignored
 - Cells sent in back to back

Revised Project Files

- *If your project needs to projects packets in both directions, you need to download revised circuits*
 - FlowBuffer with bidirectional support
 - http://www.arl.wustl.edu/arl/projects/fpx/cs536/FlowBuffer_11-20-02.tar.gz
 - Note: Your makefile needs one small change:
 - my_request_fifo.vhd was replaced with my_tail_fifo.vhd
 - Wrappers with bidirectional support
 - Available by request from Tas

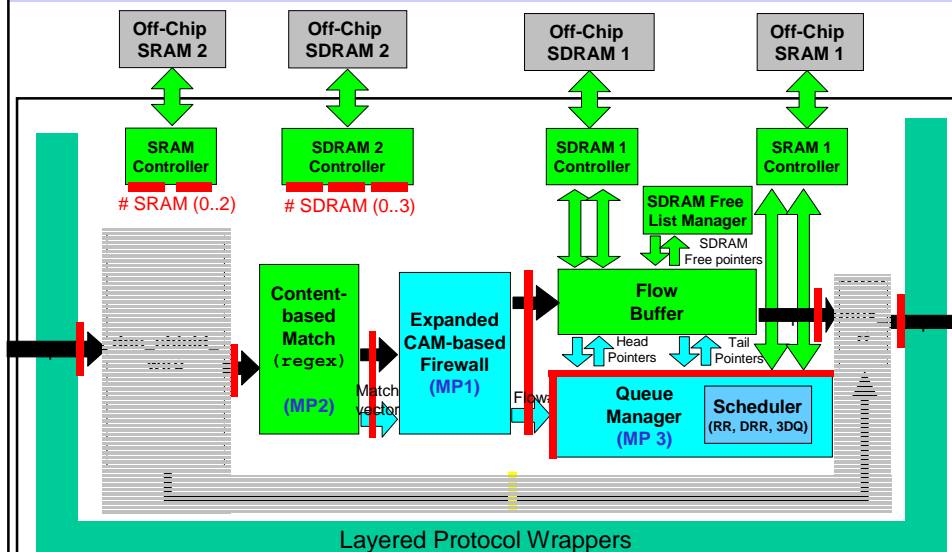
Using VPIs for bi-directional traffic flows

- **If you need to simulate a project with bi-directional traffic flows, note that:**
 - Bidirectional packets must be edited manually
 - `make_input_cells` does not handle VPI
 - Adding VPI will invalidate the header checksum
 - Must replace both atm header words (first two words of each cell).
 - PT bit in the cell header specifies the end of a packet.
 - The last atm cell of every packet must have the PT bit set
 - Default Headers
 - VPI/VCI = 0x00000320 ; Default (Current Make Cells)
 - VPI/VCI = 0x00000322 ; Default + PT (Current Make Cells)
 - Reverse-Direction Headers (and HECs)
 - VPI/VCI = 0x08000320 ; VPI=128 (Manual change)
 - HEC IS 3a000000
 - VPI/VCI = 0x08000322 ; VPI=128 + PT (Manual change)
 - HEC IS 34000000

Project Integration into the SOC

- Your top-level vhdl code will be parsed to determine your entity's signal naming convention
 - After submitting, you will be presented with a web document to map your signal names to the SOC interface (red bars in SOC diagram)
 - Your design will automatically become available for distribution upon a complete submission.
- Integration server
 - <http://cad4.arl.wustl.edu/>

Block Diagram (Example: DoS Shield)



┆ = Available Interface

▨ = New Component

↔ = New Connectivity

Project Integration Naming Rules

- *To integrate projects into the SOC, a few rules have been developed:*
 - You must provide one top-level entity per project
 - Multiple top-level entities can be combined to create a wrapper
 - As shown in the diagram on the next slide
 - Top-level entity names must be unique
 - Your entity name should be unique
 - » top.vhd → top_11111.vhd
 - » Replace “11111” with your project ID
 - If your project interacts with another groups project
 - Those groups are responsible for submitting one Top-level entity that contains both projects.

Project Integration of COREgen Components

- You need to include your COREgen files
 - COREgen components generate their .edn
 - Component's must have unique names
- Naming Convention: Rename fifo to fifo_11111
 - Will create:
 - vhdl file called fifo_11111.vhd
 - edn file called fifo_11111.edn
 - Only the .edn of CoreGen files needs to be submitted for integration.

Project Submission of Integrated Project

- **For now, submit the following files to the integration server to add your design to SOC**
 - `top_11111.vhd` (your top-level vhdl code)
 - **.edn file for your design.**
 - This only includes your project, not the wrappers and MPs
 - Create a separate Simplicity project
 - Synplicity project name should be identical to entity name (example: `ddos_11111.prj`).
 - **.edf files generated by COREgen**
 - Project ID in names (example: `fifo_11111.edn`)
 - **.par file with your circuits resource usage.**
 - Used to determine which projects can be built on a single chip without resource contention.

Control Port Assignment

- To unify control of the SOC, each team has been assigned unique control ports.
 - Control cells are sent to Destination IP 192.168.30.13
 - Projects should not alter controls cells.
 - Projects do not have to use every (or any) control cell port assigned.

PROJECT	DESTINATION PORT
Advanced Queue Module	801 - 805
Bloom Filter	806 -810
Image Filter	811 - 815
Control Packet Security	816 - 820
DNS Caching	821 - 825
Denial of Service Classifier	826 - 830
Intelligent SPAM Filter	831 - 835
NAT	836 - 840
RSVP	841 - 850
DOS Shields	851-855
CAMS	800

Final Submission of Source Code

- *For the final submission, you will submit all source code and a working build script to the gradebot server.*
 - <http://www.gradebot.com/cs536/submit/>
 - Select 'Final Project'
 - Create a tar file that contains
 - All source code
 - A working build script to simulate and synthesize
 - Simulation testbench
 - Synthesis scripts
 - Synthesis results
 - Do not include
 - Bitfiles
 - EDIF files
 - Other large data files