RAMP Blue

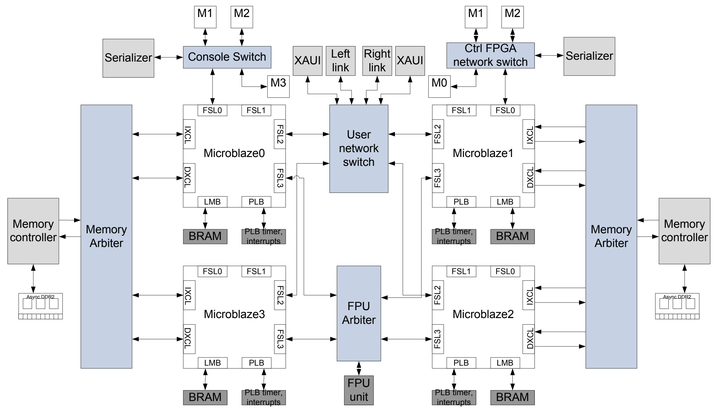
Architecture and RDL implementation

RAMP Blue Architecture and its RDLC implementation
Presented 1/11/2007
RAMP Retreat, Berkeley, CA

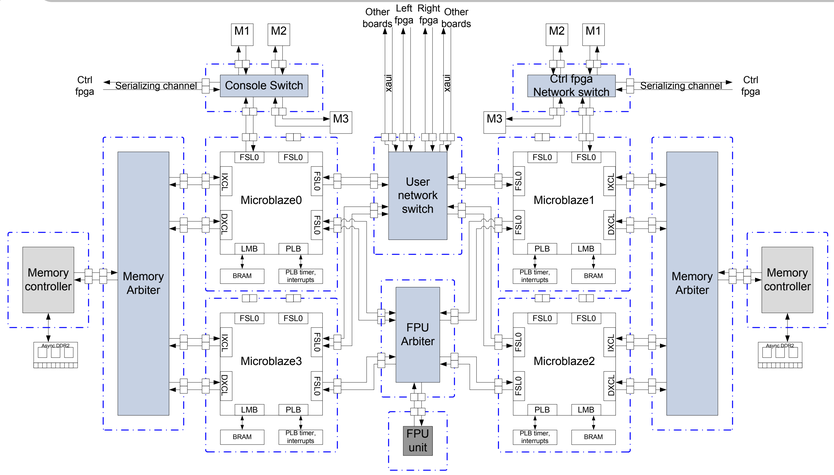
Jue Sun (juebejue@gmail.com)
http://ramp.eecs.berkeley.edu
Greg Gibeling, Alex Krasnov, Professor Wawrzynek
Special thanks tothe entire UC Berkeley RAMP Group

RAMP Blue Architecture

Note
- **Implemented in EDK**
- **Currently RAMP blue can have up to eight Microblazes (only four shown)**

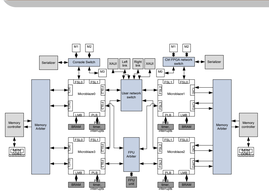


Ideal RDL Implementation of Ramp Blue

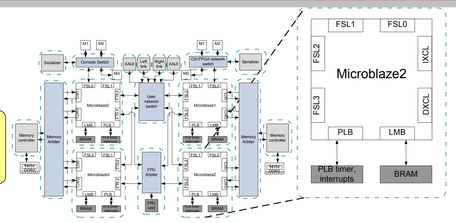


- **Communication modules e.g. XAUI, serializer) replaced & implemented by RDL channels or links**
- **Standard port interface - different units can be swapped in place of each other easily**
- **Changing timing of channels can be used to do performance research**

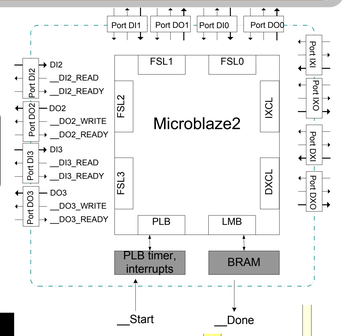
Tool Flow for RDL design of Ramp Blue



divide system into RDL units



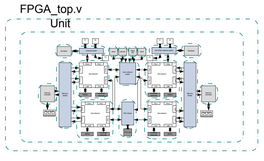
Write RDL wrapper for each unit to comply with the RDL channel interface, example:



Write RDL files that instantiate each units

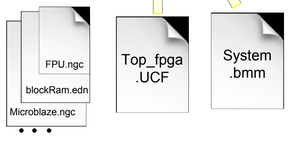
generate Verilog files that instantiate all units, RDL wrappers for each unit & port using RDLC compiler

Write a top_fpga file that instantiate the top unit in unit.v to add non-unit modules and connect to FPGA pins



Create a new ISE project and add all the generated and copied files

Copy all black boxes used by units (can be synthesized by EDK) into ise project folder



set "-bm system.bmm" option in ngdbuild option; this generates system_bd.bmm that indicates BRam location for each MB

Generate a bitfile in ISE

Compile bootloader

Run data2mem generate a bit file that initialize bram for each MB with bootload code

Current implementation

(per processor centric)