



OpenSPARC on FPGAs

Durgam.Vahia@Sun.Com

OpenSPARC Engineering & Partnership Development
Sun Microsystems, Inc.

DV Club – July 2009



OpenSPARC



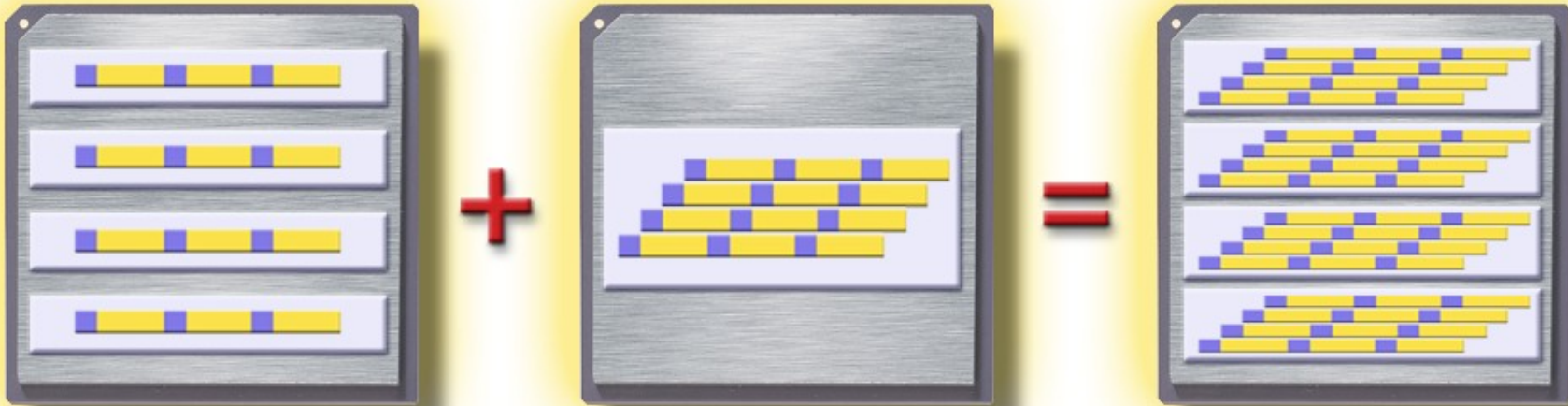
OpenSPARC.net

- Open source variants of Sun's CMT microprocessors
 - UltraSPARC T1
 - UltraSPARC T2
- Governed by GPL version2
 - Widely used in Linux distribution
- US export compliant for world-wide distribution

Agenda

- Chip Multi-threading
- OpenSPARC
- Motivation for the FPGA port
- Implementation
- Results
- Resources
- Q & A

Chip Multi-threading (CMT)



CMP
(Chip MultiProcessing,
a.k.a. “multicore”)

n cores per processor

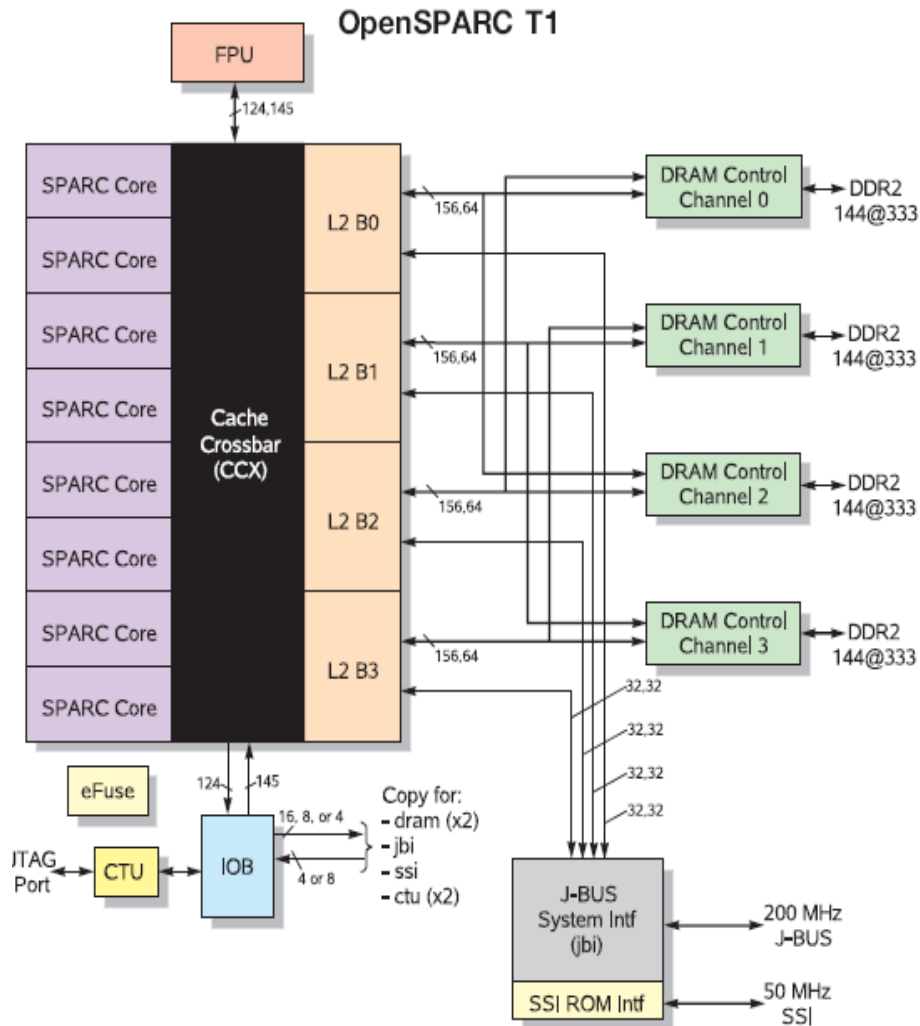
HMT
(Hardware
Multithreading)

m threads per core

CMT
(Chip
MultiThreading)

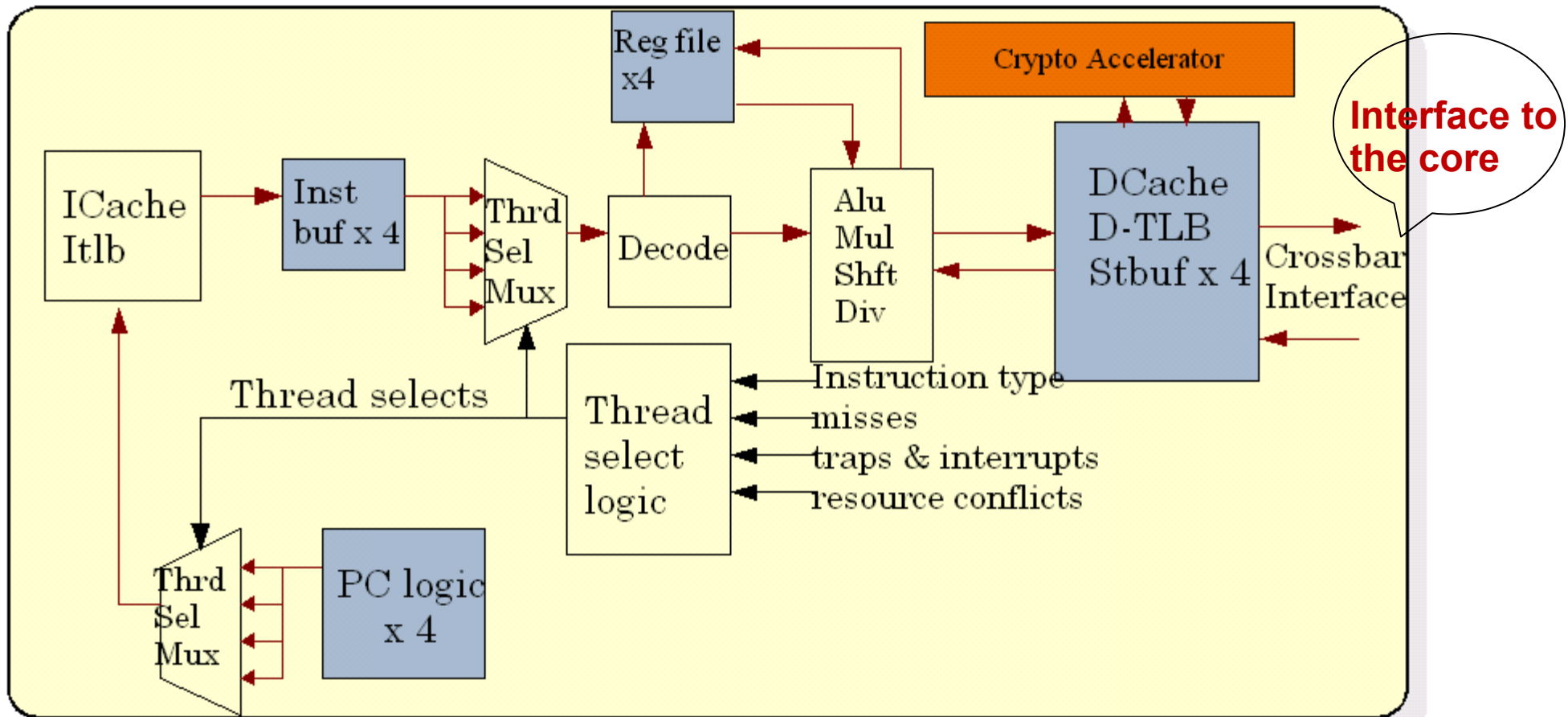
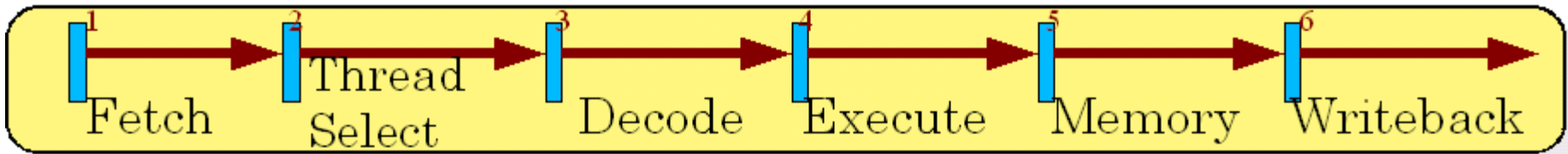
n x m threads per processor

UltraSPARC T1

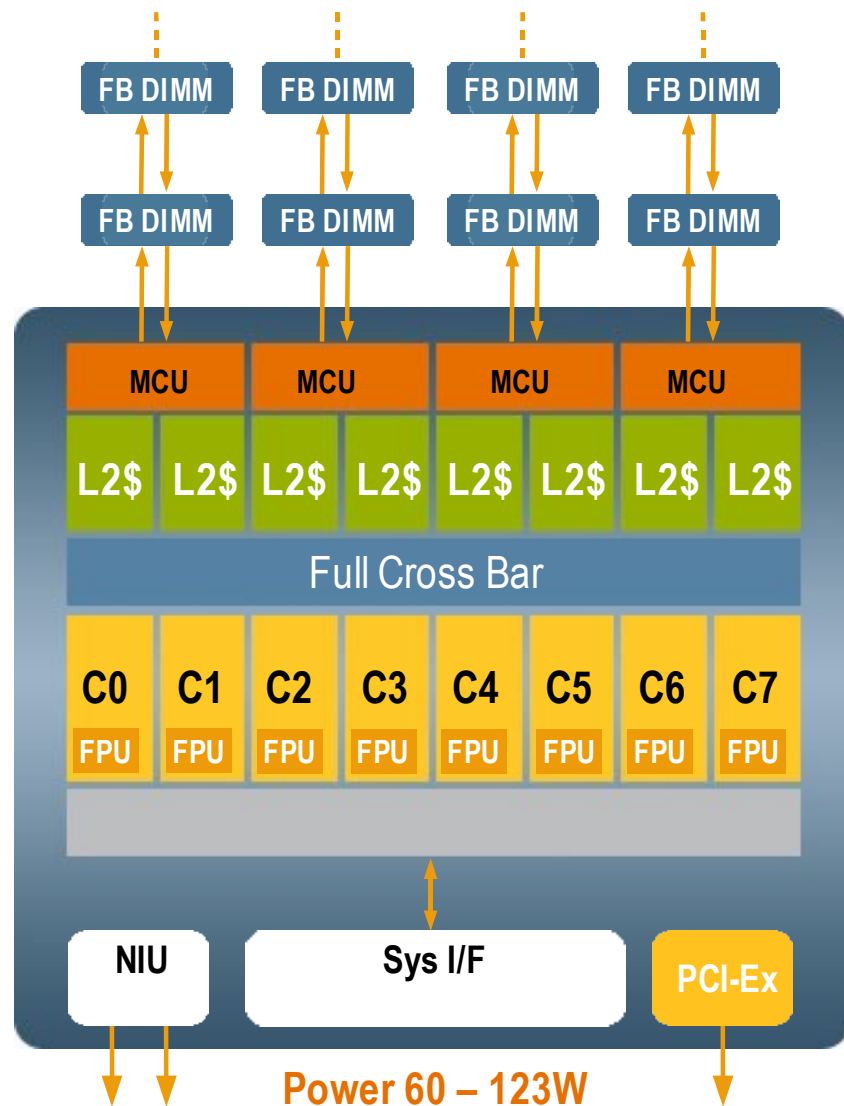


- SPARC V9 ISA, 64-bit
- Eight cores, four threads each
- Single issue 6-stage pipe
- High BW 12-way associative 3 MB on-chip L2 cache
- 4 DDR2 channels
- Shared on-chip FPU
- Chip IO through JBUS

SPARC Core Pipeline



UltraSPARC T2



2x 10GE Ethernet

www.OpenSPARC.net

x8 @2.5GHz

DV Club – Silicon Valley

- Eight cores, eight threads each
- 8-stage single issue pipe
- One FPU per core
- 16-way associative 4 MB on-chip L2 cache
- Four dual channel FBDIMM
- Two 10G Ethernet
- PCIe and Crypto excluded

OpenSPARC Bundles

● Hardware

- HDL design files written in Verilog
- Verification test-bench, diagnostics, scripts
- Synthesis scripts for ASIC and FPGA
- Lots of documents

● Software

- SPARC Architecture Model (SAM) source
- Full-system simulator (Legion) source
- Hypervisor, Open Boot PROM (OBP) source
- Solaris10 disk image
- Scripts to build all the components

Source browser at www.opensparc.net

verilog view: cmp_top - Web Browser

File Edit View Go Bookmarks Tools Window Help

Home Bookmarks Internet Lookup New&Cool Netcaster

OpenSPARC

Hide All Show All Files
 Modules Signals Tasks
 Functions Search Help

OpenSPARC T1

Hierarchy for cmp_top

- cmp_top
 - o OpenSPARCT1
 - o bw_sys
 - o cmp_clk
 - o cmp_dram
 - o cmp_mem
 - o cpx_stall
 - o dbg_port_chk
 - o dffrl_async x 4
 - o err_inject
 - o jbus_monitor
 - o jp_sjm x 2
 - o monitor
 - o one_hot_mux_mon
 - o pcx_stall
 - o sas_intf
 - o sas_tasks
 - o slam_init
 - o sparc_pipe_flow
 - o tap_stub

This page:
 Created: Thu Mar 9 11:04:17 2006

```

module OpenSPARCT1: (/*AUTOARG*/
// Outputs
DRAM0_RAS_L, DRAM0_CAS_L, DRAM0_WE_L, DRAM0_CS_L, DRAM0_CKE,
DRAM0_ADDR, DRAM0_BA, DRAM0_CK_P, DRAM0_CK_N, DRAM1_RAS_L,
DRAM1_CAS_L, DRAM1_WE_L, DRAM1_CS_L, DRAM1_CKE, DRAM1_ADDR,
DRAM1_BA, DRAM1_CK_P, DRAM1_CK_N, DRAM2_RAS_L, DRAM2_CAS_L,
DRAM2_WE_L, DRAM2_CS_L, DRAM2_CKE, DRAM2_ADDR, DRAM2_BA,
DRAM2_CK_P, DRAM2_CK_N, DRAM3_RAS_L, DRAM3_CAS_L, DRAM3_WE_L,
DRAM3_CS_L, DRAM3_CKE, DRAM3_ADDR, DRAM3_BA, DRAM3_CK_P,
DRAM3_CK_N, J_PACK0, J_PACK1, J_REQ0_OUT_L, J_REQ1_OUT_L, J_ERR,
TSR_TESTIO, TDO,
// Inouts
DRAM0_DQ, DRAM0_CE, DRAM0_DQS, SPARE_DDR0_PIN, DRAM1_DQ, DRAM1_CE,
DRAM1_DQS, CLK0BS, SPARE_DDR1_PIN, DRAM2_DQ, DRAM2_CE, DRAM2_DQS,
SPARE_DDR2_PIN, DRAM3_DQ, DRAM3_CE, DRAM3_DQS, SPARE_DDR3_PIN,
J_AD, J_ADE, J_ADTYPE, SPARE_JBUSR_PIN, DBG_DQ, DBG_CK_P,
DBG_CK_N, VDDA, VDDB0, VDDB0, VDDC0, VDDT0, VDDL18, VDDR18, DIODE_TOP,
DIODE_BOT, VPP, SSI_MISO, SSI_MOSI, SSI_SCK, PMO, VDD_SENSE,
VSS_SENSE, SPARE_MISC_PIN, SPARE_MISC_PAD, SPARE_DDR0_PAD,
SPARE_DDR1_PAD, SPARE_DDR2_PAD, SPARE_DDR3_PAD, SPARE_DBG_PAD,
// Inputs
DRAM01_P_REF_RES, DRAM01_N_REF_RES, DRAM23_P_REF_RES,
DRAM23_N_REF_RES, J_PAR, J_PACK4, J_PACK5, J_REQ4_IN_L,
J_REQ5_IN_L, J_RST_L, DTL_L_VREF, DTL_R_VREF, JBUS_P_REF_RES,
JBUS_N_REF_RES, DBG_VREF, J_CLK, TCK, TCK2, TRST_L, TDI, TMS,
TEST_MODE, TEMP_TRIG, PWRON_RST_L, CLK_STRETCH, DO_BIST,
EXT_INT_L, BURNNIN, PMI, PGRM_EN, PLL_CHAR_IN, VREG_SELBG_L,
TRIGIN, HSTL_VREF
);

output          DRAM0_RAS_L; // From pad_ddr0 of pad_ddr0.v
output          DRAM0_CAS_L; // From pad_ddr0 of pad_ddr0.v
output          DRAM0_WE_L; // From pad_ddr0 of pad_ddr0.v
output [3:0]    DRAM0_CS_L; // From pad_ddr0 of pad_ddr0.v
output          DRAM0_CKE; // From pad_ddr0 of pad_ddr0.v
output [14:0]   DRAM0_ADDR; // From pad_ddr0 of pad_ddr0.v
output [2:0]    DRAM0_BA; // From pad_ddr0 of pad_ddr0.v
output [127:0]  DRAM0_DQ; // To/From pad_ddr0 of pad_ddr0.v
input [15:0]    DRAM0_CE; // To/From pad_ddr0 of pad_ddr0.v
input [35:0]    DRAM0_DQS; // To/From pad_ddr0 of pad_ddr0.v
output [3:0]    DRAM0_CK_P; // From pad_ddr0 of pad_ddr0.v
output [3:0]    DRAM0_CK_N; // From pad_ddr0 of pad_ddr0.v
input           DRAM01_P_REF_RES; // To pad_ddr0 of pad_ddr0.v
input           DRAM01_N_REF_RES; // To pad_ddr0 of pad_ddr0.v
input           SPARE_DDR0_PIN; // To/From pad_ddr0 of pad_ddr0.v

output          DRAM1_RAS_L; // From pad_ddr1 of pad_ddr1.v
output          DRAM1_CAS_L; // From pad_ddr1 of pad_ddr1.v
output [3:0]    DRAM1_WE_L; // From pad_ddr1 of pad_ddr1.v
output          DRAM1_CS_L; // From pad_ddr1 of pad_ddr1.v
output          DRAM1_CKE; // From pad_ddr1 of pad_ddr1.v
output [14:0]   DRAM1_ADDR; // From pad_ddr1 of pad_ddr1.v
output [2:0]    DRAM1_BA; // From pad_ddr1 of pad_ddr1.v
input [127:0]  DRAM1_DQ; // To/From pad_ddr1 of pad_ddr1.v
input [15:0]    DRAM1_CE; // To/From pad_ddr1 of pad_ddr1.v
input [35:0]    DRAM1_DQS; // To/From pad_ddr1 of pad_ddr1.v
output [3:0]    DRAM1_CK_P; // From pad_ddr1 of pad_ddr1.v
output [3:0]    DRAM1_CK_N; // From pad_ddr1 of pad_ddr1.v
input           DRAM23_P_REF_RES; // To pad_ddr1 of pad_ddr1.v
input           DRAM23_N_REF_RES; // To pad_ddr1 of pad_ddr1.v

```

OpenSPARC Ecosystem

Area	Community	Universities	Partners
Tool Kit	<ul style="list-style-type: none"> ● OpenSPARC.net ● Contests ● Blogs ● Conferences ● Tutorials ● Operating System Port 	<ul style="list-style-type: none"> ● Center of Excellence ● Collaborations ● Curriculum ● Research ● Tutorials 	<ul style="list-style-type: none"> ● FPGA implementation ● ASIC implementation ● EDA ● SoC implementation ● Foundry relationship
Focus	<ul style="list-style-type: none"> ● Evangelize OpenSPARC ● Encourage contributions ● Build knowledgebase 	<ul style="list-style-type: none"> ● Research publications ● Textbooks ● Shared coursework 	<ul style="list-style-type: none"> ● Chip spins ● Coprocessors ● CMT EDA tools

Encourage community innovation

OpenSPARC momentum



More than 11,000 downloads
Includes academic and commercial interests

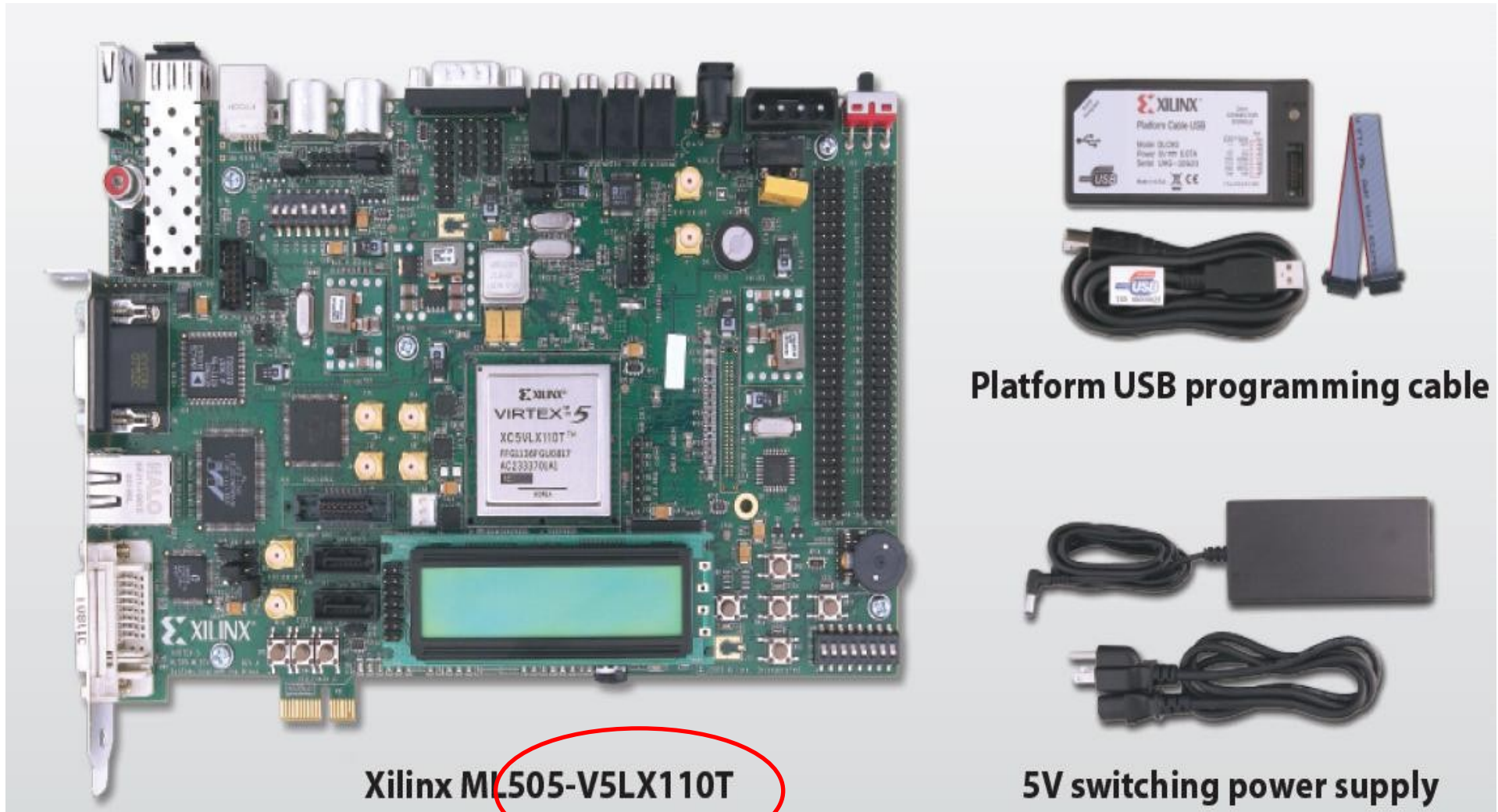
Agenda

- Chip Multi-threading
- OpenSPARC
- Motivation for the FPGA port
- Implementation
- Results
- Resources
- Q & A

Why FPGAs?

- Community requested it
 - As a platform for experimentation
 - Gentler introduction to server class processor design
 - Basic building block for more interesting/complex designs
- Need off-the-shelf FPGA board that is
 - Large enough for “good-sized” design points
 - At the “right” price point
 - Available world-wide

OpenSPARC/Xilinx FPGA Evaluation Kit



Available from <http://www.digilentinc.com/v5osdk>

FPGA implementation – Key objectives

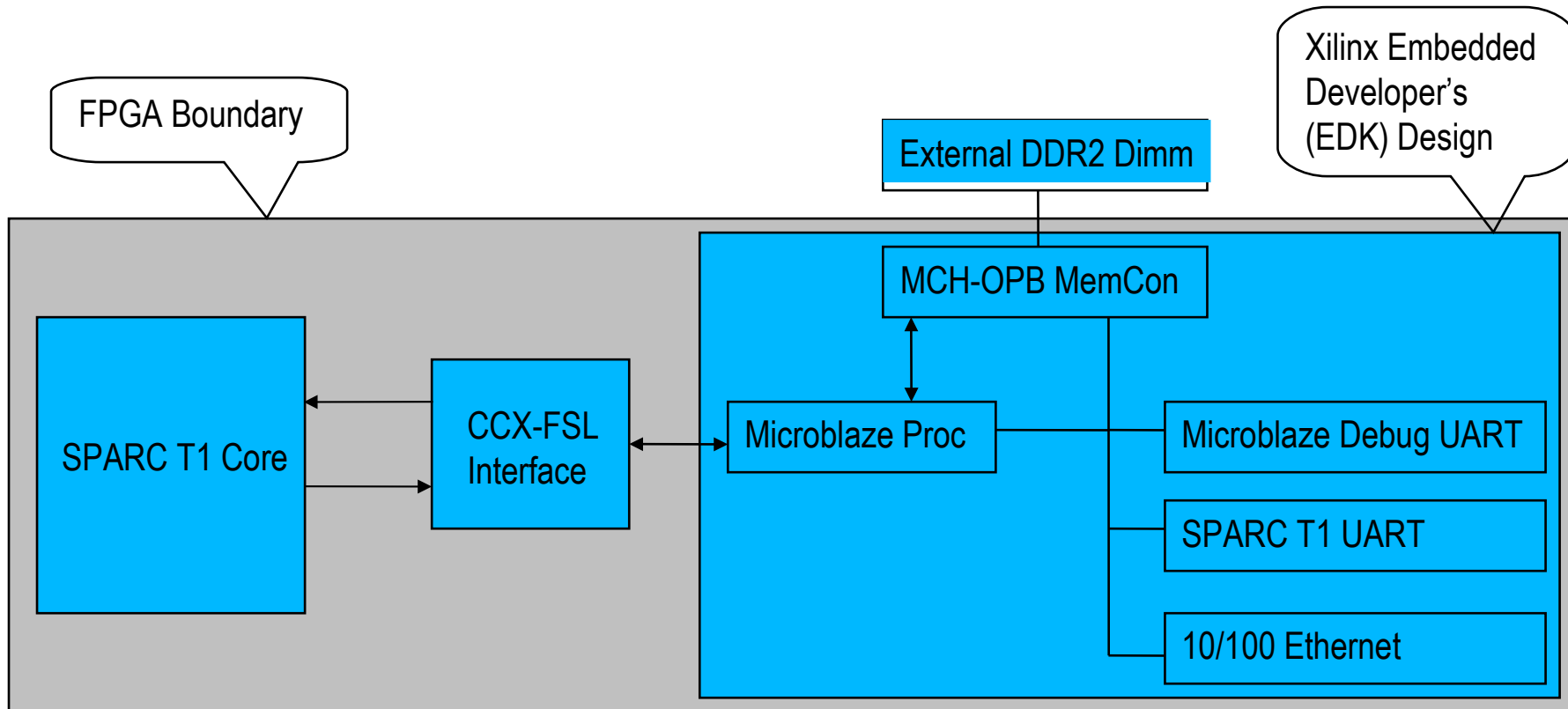
- Proliferation of OpenSPARC & Xilinx FPGA Technology
 - Make OpenSPARC FPGA-Friendly
 - Create reference design with complete system functionality
 - Boot Solaris and Linux
 - Open it up ..
 - Seed ideas in the community

Processor core changes

- Primarily to reduce the area foot-print of the multi-threaded SPARC core
 - Recode custom SRAM cells for better resource utilization
 - Parameterizable single- and multi-thread options
 - Removal of all the asynchronous logic – no clock gating
 - Simplified reset
 - Only flops, no latches
 - Reduce multi-port SRAM arrays
 - Removable logic – Crypto accelerator, Floating-point

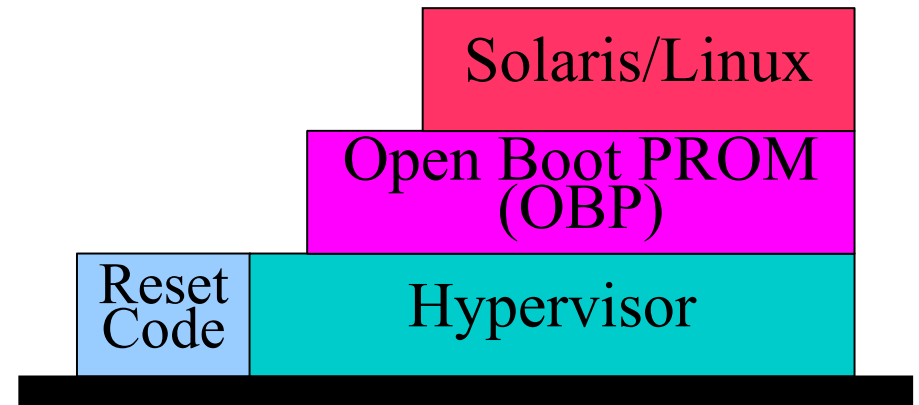
Overall 50% reduction in area

OpenSPARC FPGA System



Software Stack

- Out-of-the-box operating system installation
- Boots from a virtual disk in RAM which holds the Solaris binaries
- Able to boot either Linux or OpenSolaris
- Entire software stack is open source



Results – Capacity Utilization

- Using Xilinx XC5VLX110T device
- Synthesis results (no SPU, 16 TLB entries)
 - 1-thread core: 31475 LUT (45%), 115 BRAM (78%)
 - 4-thread core: 51558 LUT (74%), 115 BRAM (78%)

(synthesized with Synplicity Synplify Pro)

- Complete system:

SPARC core, MicroBlaze, 2 UARTs,
Ethernet, and DDR2 controller:

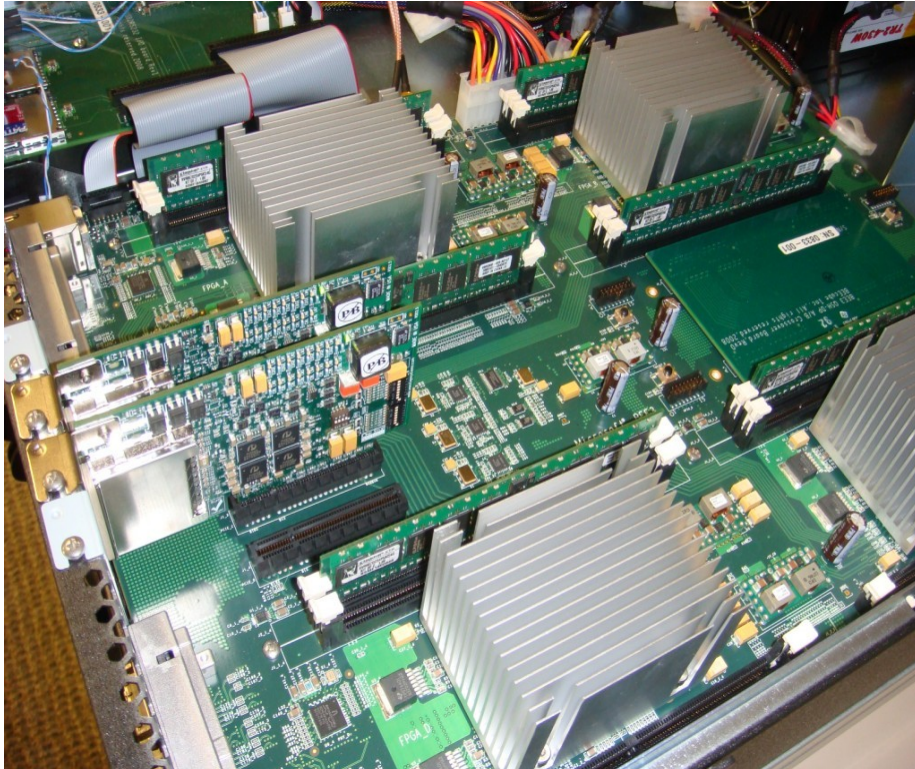
1-thread core: 38271 LUT (55%), 128
BRAM (86%)

Results – Community building

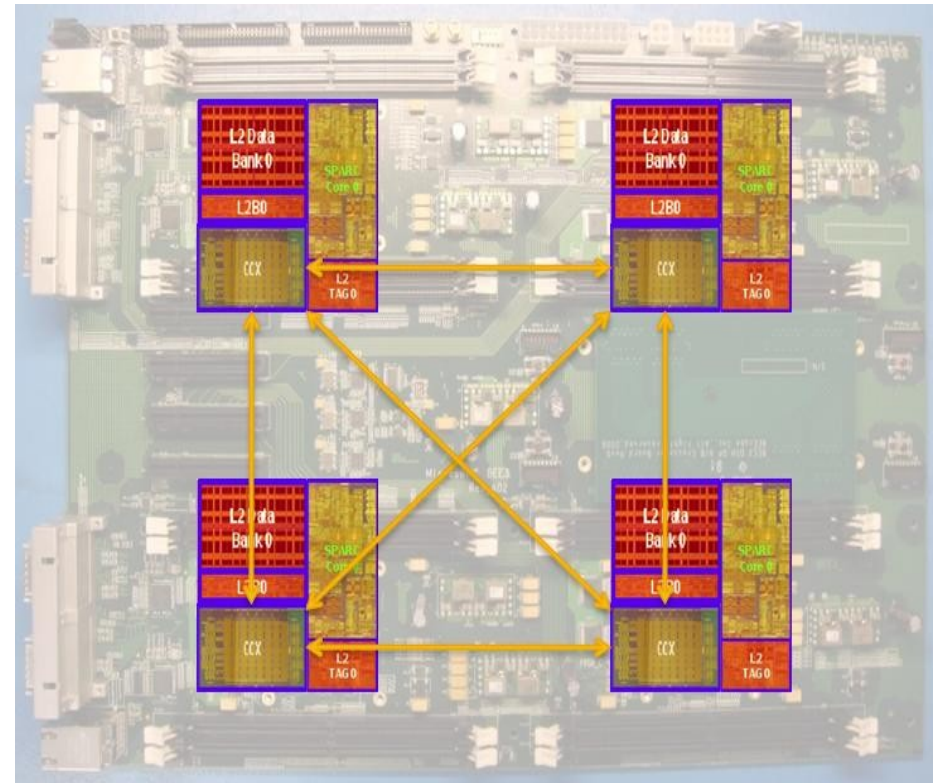
- 50+MHz OpenSPARC FPGA system
 - Platform for experimentation for HW developers
 - Building blocks to design truly multi-core, multi-thread processors
 - Based on open-source technology
 - Costs ~\$700 for academics (\$2,000 for others)
 - Higher-end boards available for larger design points
- Used in 200+ Universities world-wide
 - Architectural exploration, fault tolerance, reconfigurable computing, low power architectures

Free download, visit <http://www.opensparc.net>

Results - Partner



<http://www.beecube.com>



Key Learnings

- FPGA capacity and performance are increasing each successive generation, however
 - Optimal use of these resources is a big challenge
- FPGA design/coding is significantly different from ASIC style
 - E.g. pipeline bypass, latches, clock gating creates both functional and P&R issues
 - For prototyping, create special FPGA models
- FPGA tools are full of quirks
 - Invest in verification at each level – RTL, gate, layout, system

Web pointers

- Program

- <http://www.opensparc.net>

- Publications

- <http://www.opensparc.net/publications/>

- Downloads

- <http://www.opensparc.net/opensparc-t1/downloads.html>

- <http://www.opensparc.net/opensparc-t2/downloads.html>

- Participation (Forums)

- <http://forums.sun.com/category.jspa?categoryID=120>

- FPGA development boards

- <http://www.digilentinc.com/v5osdk>



OpenSPARC on FPGAs

Durgam.Vahia@Sun.Com

OpenSPARC Engineering & Partnership Development
Sun Microsystems, Inc.

www.OpenSPARC.net

