

# **LEON Processor Debug Support Unit**

Jiri Gaisler

Gaisler Research

# LEON development background

- ◆ 100 MIPS SPARC V8 processor for critical space applications
- ◆ Extensively fault-tolerant, SEU-free on 'soft' processes
- ◆ Easily portable to new target technologies
- ◆ First prototype: Atmel 0.35  $\mu\text{m}$  CMOS, 50 MIPS (Q1-2001)
- ◆ Production device: Atmel 0.25  $\mu\text{m}$  CMOS, 100 MIPS (Q4-2002)

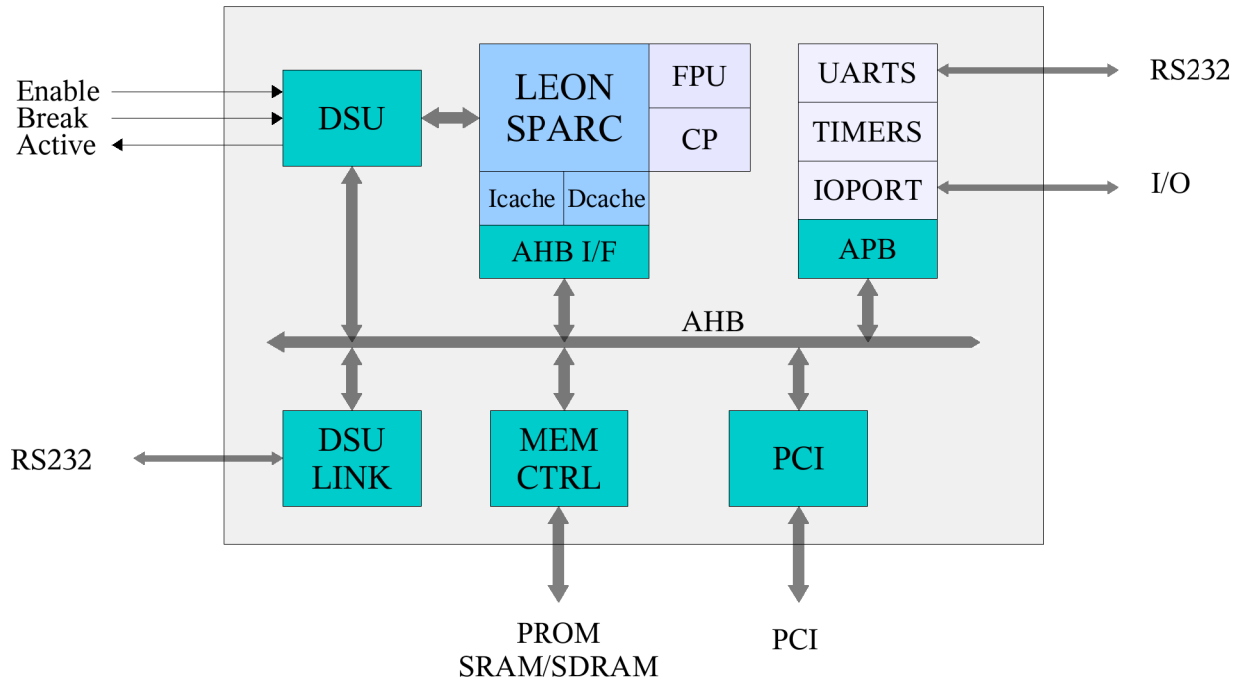
# LEON implementation details

- ◆ Full SPARC V8, on-chip I+D caches
- ◆ Meiko FPU
- ◆ Hardware MUL, DIV, MAC instructions
- ◆ On-chip peripherals: uarts, timers, interrupt ctrl
- ◆ PROM/SRAM/(SDRAM) memory controller
- ◆ Uses AMBA AHB/APB on-chip buses (ARM)
- ◆ Performance: 1 MIPS/MHz (Dhrystone 2.1)
- ◆ **On-chip debug support unit**

## Debug support in LEON

- ◆ Debug support is provided by adding a debug mode to the processor core
- ◆ In debug mode, the processor pipeline is idled while the remaining system works nominally
- ◆ During debug mode, the processor is controlled by a Debug Support Unit (DSU)
- ◆ A separate debug communication link is added to control the DSU and to access memory and on-chip peripheral units
- ◆ No clock-gating or JTAG used – fully portable also to FPGAs

# LEON with DSU



# Debug Support Unit

- ◆ 4 hardware breakpoints/watchpoints with mask
- ◆ Single-stepping
- ◆ Access to processor registers and cache memories
- ◆ 256 entry trace buffer for instructions and AHB bus traffic
- ◆ Acts as slave on the internal AHB bus

# Processor debug mode

- ◆ Processor debug mode can be entered after:
  - ◆ executing a software breakpoint instruction (ta 1)
  - ◆ hardware breakpoint/watchpoint hit
  - ◆ asserting an external break signal
  - ◆ receiving a break command from the AHB bus
  - ◆ a trap that would cause the processor to enter error mode
  - ◆ any or certain traps as programmed in the debug support unit
- ◆ The debug state is available on external pin

## DSU communication link

- ◆ Dedicated UART with auto-baud rate selection
- ◆ Communication speeds up to 1/8 of system clock frequency
- ◆ AHB master, access to all on-chip modules
- ◆ Simple and open read/write protocol with burst support
- ◆ Allows to boot system without boot-prom directly from RAM
- ◆ The DSU can also be controller by any AHB master, such as PCI



# Interfacing the DSU

```

kvt-z>
Starting
Pera Towers Queens Intex No Puzzle Quick Bubble Tree FFT
100 133 63 367 150 567 63 150 553 250

Nonfloating point composite is 318
Floating point composite is 472

Program exited normally.
tsia) per

Cycles : 33306680
Instructions : 23306849
Overall CPI : 1.43

CPU performance (14.0 Mhz) : 9.80 MFPS ( 9.63 MIPS, 0.16 MFLOPS)
Simulated time : 2379.19 ns
Processor utilisation : 100.00 %
Real-time / simulator-time : 1/1.18
Simulation performance : 8290.24 KIPS
Used time (sys + user) : 2.81 s

tsia) gdb
gdb interface: using port 1234
    
```

## DSU monitor

- ◆ Program to control the LEON debug support unit
- ◆ Stand-alone use: load programs, insert breakpoints, control trace buffer, access all registers and memory
- ◆ Gdb mode: act as gdb-gateway
  - ◆ Source-level non-intrusive debugging on target hardware
  - ◆ Re-creates stack backtrace without flushing register windows
  - ◆ Open communication protocol (gdb extended-remote)
- ◆ Runs on standard PC (linux, windows) or Sun WS

# DSUMON – probing the system

```
Jiri@venus $ ./dsumon -i
```

LEON DSU Monitor, version 1.0

Copyright (C) 2001, Gaisler Research - all rights reserved.

Comments or bug-reports to [jiri@gaisler.com](mailto:jiri@gaisler.com)

```
port /dev/ttyS0 at 115200 baud
clock frequency : 24.9 Mhz
register windows : 8
instruction cache : 4 kbytes, 16 bytes/line
data cache : 2 kbytes, 16 bytes/line
hardware breakpoints : 4
trace buffer : 128 lines
ram width : 32 bits
ram banks : 2
ram bank size : 2048 kbytes
stack pointer : 0x403ffff0
```

```
dsu>
```

# Register dump

dsu> reg

	INS	LOCALS	OUTS	GLOBALS
0:	403FFDF8	403FFE08	00000004	00000000
1:	403FFDE8	40008E74	00000003	08000000
2:	00000004	40008D38	403FFDF8	00000003
3:	40017950	00000020	00000083	50000000
4:	400178AC	00000040	403FFE00	00000001
5:	00000029	00000040	00000000	00000770
6:	403FFE20	00000000	403FFD88	00000001
7:	4000238C	00000000	40007B6C	00000000

psr: 000000E3    wim: 00000080    tbr: 40000060    y: 00000000  
pc: 40007b84    mov %i1, %l1  
npc: 40007b88    ld [%fp - 0x28], %o0

# Trace buffer

- ◆ Allows tracing of executed instructions, AHB transfers, or both:
  - ◆ Address and data/opcode
  - ◆ Instruction results
  - ◆ Instruction load/store data
  - ◆ AHB transfer parameters (master, size, response, interrupts)
- ◆ 256-entry circular buffer
- ◆ Provided with 2 hardware breakpoint registers
- ◆ Delay counter for pre-, mid-, or post-trig trace
- ◆ 30-bit time stamp for each trace entry
- ◆ Not cleared on reset – can be used for post-mortem analysis

# Tracing instructions

dsu> in

<b>time</b>	<b>address</b>	<b>instruction</b>	<b>result</b>
120828287	400096c0	sethi %hi(0x40013800), %o0	[40013800]
120828294	400096c4	ldd [%o0 + 0x220], %f2	[3ff00000 00000000]
120828304	400096c8	fcmped %f0, %f2	[3ff00000]
120828314	400096cc	nop	[00000000]
120828315	400096d0	fbule 0x40009754	[00000000]
120828316	400096d4	sethi %hi(0x40013800), %o0	[40013800]
120828320	40009754	ldd [%fp - 0x38], %f0	[bfe8ab1d 4daa6a20]
120828325	40009758	ret	[40009758]
120828328	4000975c	restore	[00000000]
120828337	40004578	ba,a 0x400045d4	[00000000]

## Tracing the AHB bus

dsu> ah

<b>time</b>	<b>address</b>	<b>type</b>	<b>data</b>	<b>trans</b>	<b>size</b>	<b>burst</b>	<b>mst</b>	<b>lock</b>	<b>resp</b>
120828317	4000975c	read	81e80000	3	2	1	0	0	0
120828324	40004578	read	30800017	2	2	1	0	0	0
120828326	4000457c	read	d21221b8	3	2	1	0	0	0
120828330	90000000	write	000045f9	2	2	0	1	0	0
120828334	400045d4	read	81c7e008	2	2	1	0	0	0
120828336	400045d8	read	81e80000	3	2	1	0	0	0
120828338	400045dc	read	9de3bf90	3	2	1	0	0	0
120828344	40006c08	read	c13fbfd0	2	2	1	0	0	0
120828346	40006c0c	read	40000928	3	2	1	0	0	0
120828349	90000000	read	000055f9	2	2	0	1	0	0

## Mixed trace

TIME	ADDRESS	OPERATION	ADDRESS/DATA
120828317		ahb read, mst=0, size=2	[4000975c 81e80000]
120828320	40009754	ldd [%fp - 0x38], %f0	[bfe8ab1d 4daa6a20]
120828324		ahb read, mst=0, size=2	[40004578 30800017]
120828325	40009758	ret	[40009758]
120828326		ahb read, mst=0, size=2	[4000457c d21221b8]
120828328	4000975c	restore	[00000000]
120828330		ahb write, mst=1, size=2	[90000000 000045f9]
120828334		ahb read, mst=0, size=2	[400045d4 81c7e008]
120828336		ahb read, mst=0, size=2	[400045d8 81e80000]
120828337	40004578	ba,a 0x400045d4	[00000000]



# Conclusions

- ◆ Non-intrusive, symbolic debugging on target hardware
- ◆ Debugging can be done over other interfaces (PCI, ...)
- ◆ PC/WS can be used as debug host without adding costly interfaces
- ◆ Trace buffer allows visibility on execution and data transfers
- ◆ High-level design allows use in both FPGA and ASIC