
V8uC: Sparc V8 micro-controller derived from LEON2-FT

Software and Data System Divisions
Final Presentation Days
ESTEC, 25+26 April 2012

Walter Errico

SITAEL S.p.A.

Phone: +39 050 9912116

E-mail: walter.errico@sitael.com

URL - <http://www.sitael.com>



Introduction

V8uC is an ESA project to realize a 32 bit micro-controller from LEON2-FT core data-base

From CPU to micro-controller a new simplified memory data path had to be designed

Design approach of reuse of large parts of the LEON2-FT core:

- to reduce design timing and risk
- to inherit its SW tools and libraries

Changes minimized to not reintroduce errors in debugged components

Topics

- Target applications
- MC Architecture
- Peripherals
- SW development tools
- Core numbers
- Conclusions

Target applications

Embedded sub-systems where LEON features/costs exceed, but not cover effective requirements.

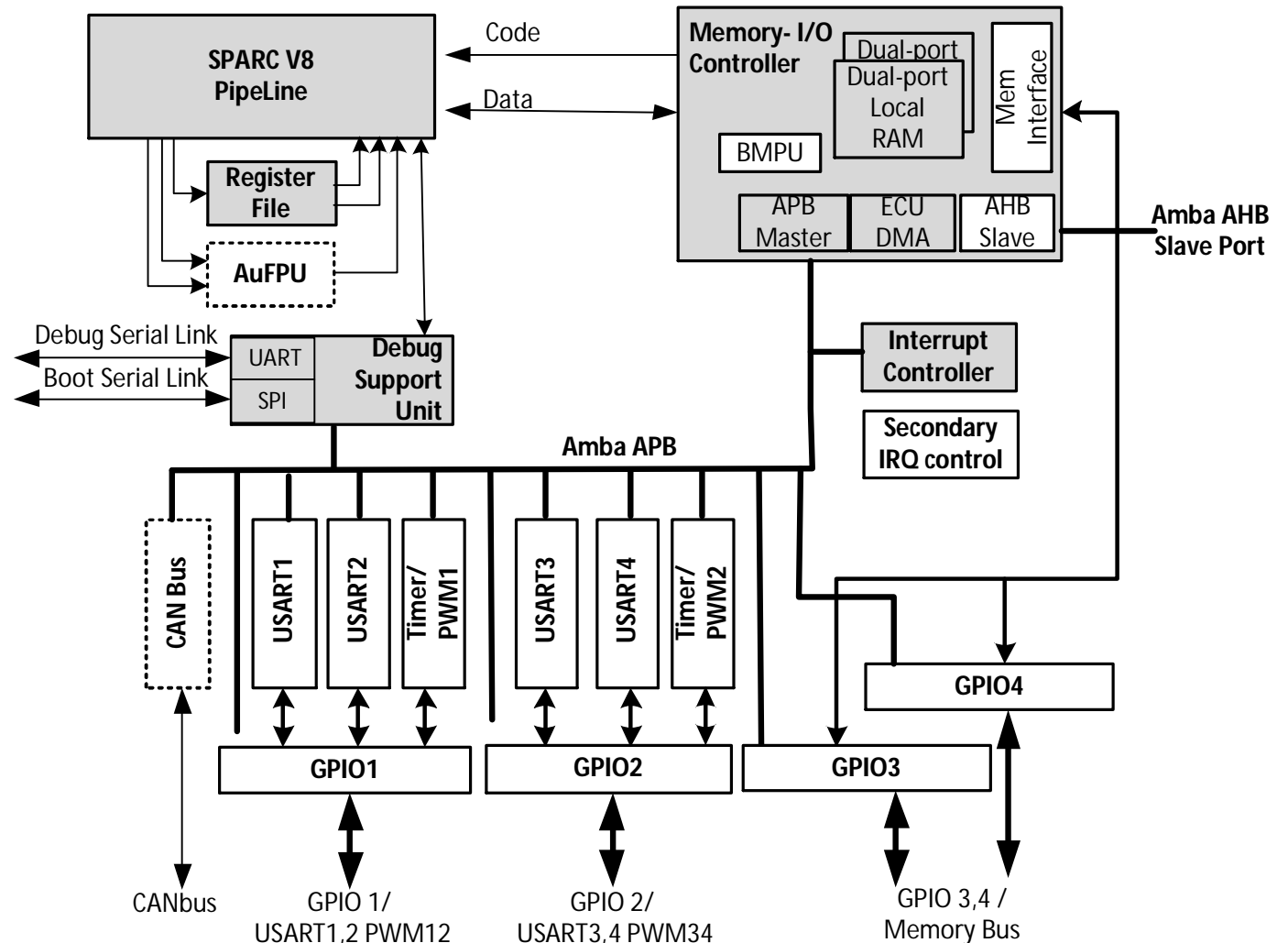
- Closed control loop applications
- Acquisition and serialization of multiple discrete or analogue signals
- In-line data elaboration

- Small SW: no real Operative System but only scheduler loop
- Memory data path simplified with on chip local memory (64/128 Kbyte) sufficient to keep entire application
- Remote boot via serial link as option to save non volatile memory on board
- Several I/O lines and serial peripheral interfaces

The objective is to cover I/O or control boards requirements without need of additional FPGA design

MC Architecture

- No Caches
- New Memory Controller with DMA engine
- Programmable number of USARTs, GPIOs and PWMs
- Simplified APB/AHB BUS structure
- AuFPU and CANbus are optional plug-ins



From cache to addressable local RAM

Cache tries to keep frequently accessed information close to the processor for low latency accesses

Cache is an expansive mechanism that may be not effective in specific embedded application:

- Data are not always reused, or have limited reuse
- Data are loaded only when requested the first time
- Cache line may be not a good match for data size
- In Real-Time applications, Cache makes the system behavior not easily predictable

In micro-controller the complexity is moved from HW to SW.

Micro-controller is expected to execute few programs heavily optimized by hand

The programmer is in charge to move information close to the processor when it is needed

The V8uC data path

The design of the V8uC data path required the major effort of the Project.

It includes the only modules not derived from LEON2FT but redesigned from scratch.

- Local memory is directly connected to the MP pipeLine with two buses to manage ideally 0-latency parallel accesses to data and code segments
- Data path control has to resolve conflicts between program and data cross memory accesses
- Data path control has to allow the data transfer between the local and remote memory areas without blocking the MP activity
- Local memory and external memory buses have to be protected with EDAC

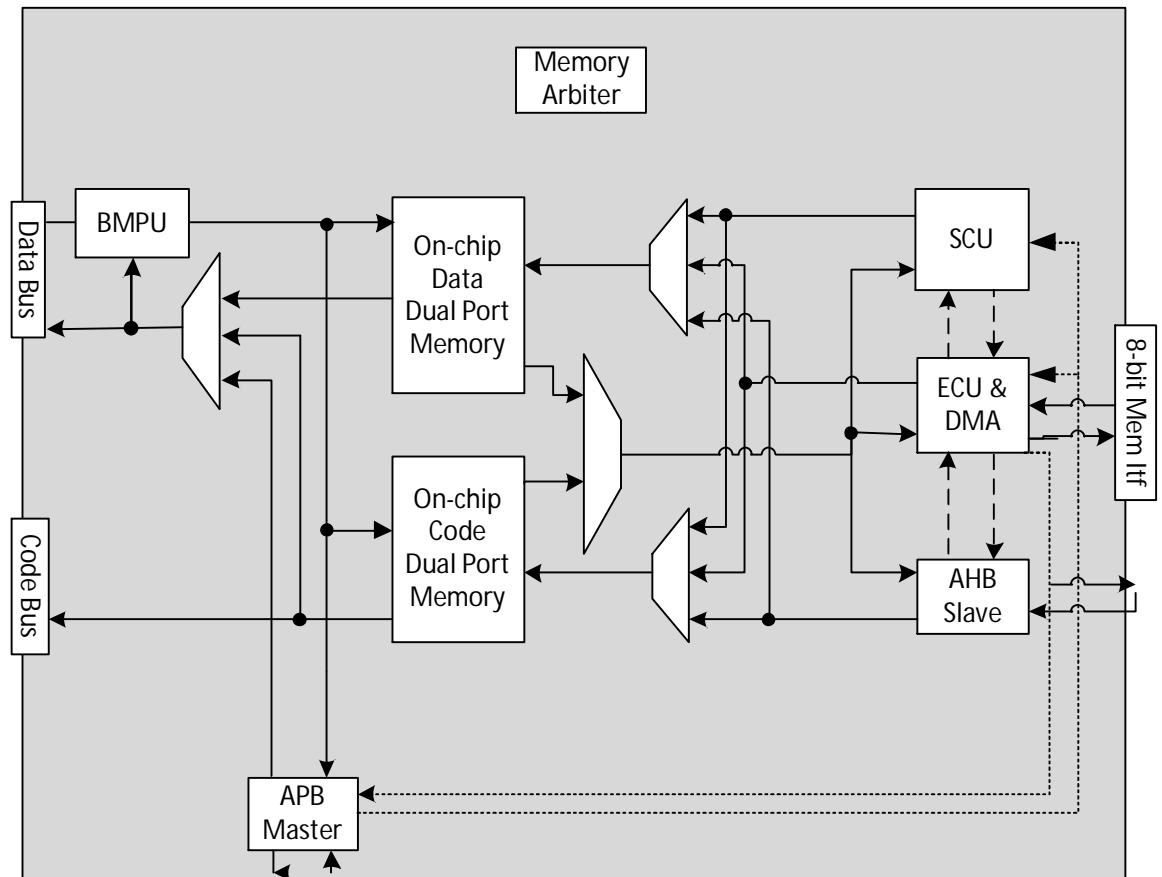
The V8uC memory controller

The V8uC data path memory controller architecture is based on dual port memories:

The 1st port is dedicated to the MP pipeline. On this side the MP can reach Data, Program memory and its APB peripheral bus

The 2nd port is shared among DMA controller, Scrubber unit and AMBA AHB port.

A memory arbiter handles lock requests



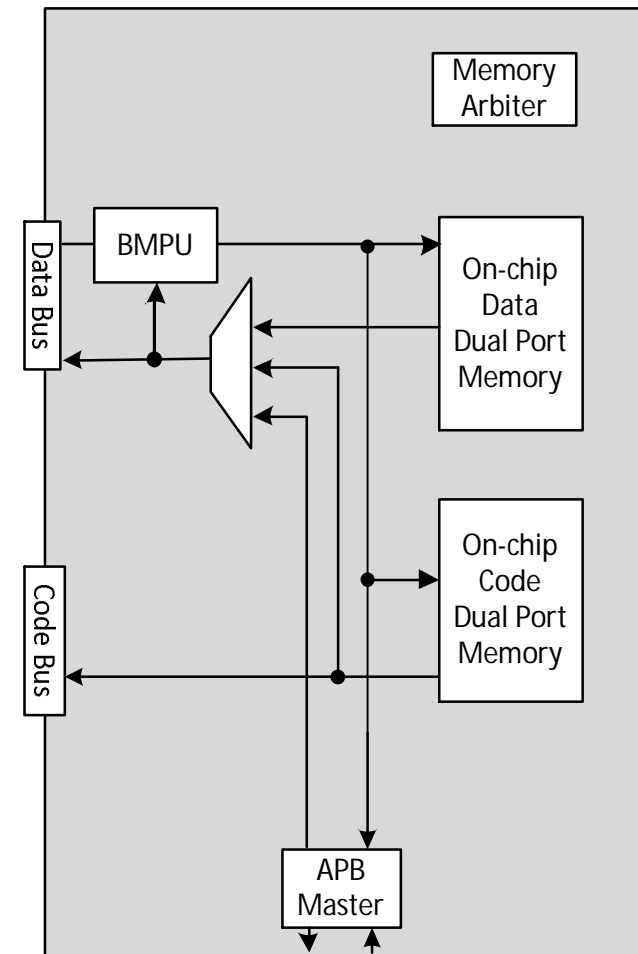
The V8uC memory controller

- On the MP side a Bit Manipulation & Protection Unit (BMPU) control all the data request coming from the pipeline and manages the memory locked accesses.

Memory lock requests can be explicit or implicit. A single operation is executed as the atomic read-modify-write back sequences

- Explicit: SparcV8 LDSTUB and SWAP instructions
- Implicit: Byte or Half-word write with EDAC
- Implicit: Read-Modify on I/O space for bit/field manipulation (triggered by STD in aliased memory area of Maskbit+NewVal sequences)

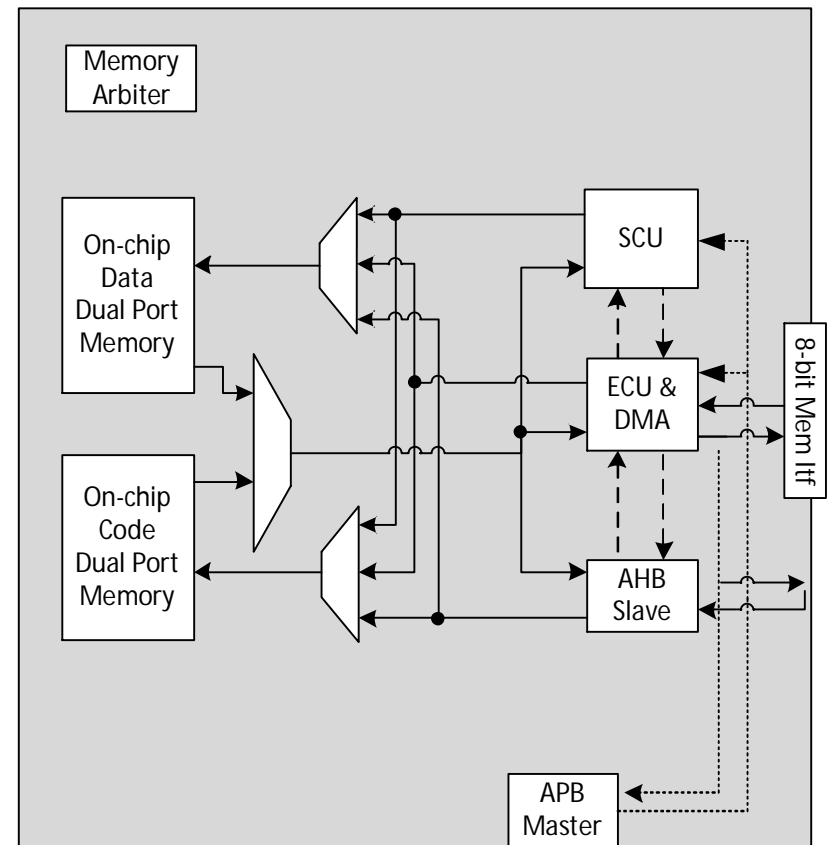
BMPU includes access control to program 2 or 3 separated protection areas (like LEON2FT)



The V8uC memory controller

On the other side the local memory ports are shared among 3 modules

- The DMA controller can transfer data between, local memory, external memory interface and APB peripheral bus.
- The external memory interface supports SRAM/PROM/EEPROM with 8 bit interface and serial 32 + 7 bit EDAC
- The memory Scrubber Unit performs the scrubbing and washing of the on-chip local RAMs.
- The AMBA AHB port makes the core compliant with external network modules (SpW-RMAP or MIL1553) with AHB master interface



Event Controller and DMA unit

DMA engine capable to manage up to 8 channels between any combination of on/off-chip memories and peripherals

- Two different triggers for DMA channel activation.
 - Program controlled activation (for example for data pre-fetch before elaboration and data store after elaboration)
 - Interrupt request (for example incoming message or end of transmission, timers)
- Possibility of chaining multiple channels

The Event Controller and DMA unit manages in parallel to the processing pipeline all “interrupts” that can be served with a memory transfer.

Peripherals: Timer/PWM, USART

New functionalities are added to the LEON2 peripherals to address the largest range of devices without need of external logic (FPGA)

Timers/PWM

- New option to chain up to 4 x 16 bit timers into a 64 bit resulting timer (RTC)
- PWM for BLDC motors control: 6 outputs to drive a 6-MOS inverter; with programmable LUT for the direct encoding of 3 Hall sensors feedbacks;
- Complementary mode for half-bridge driving with programmable dead time
- Windowed watchdog

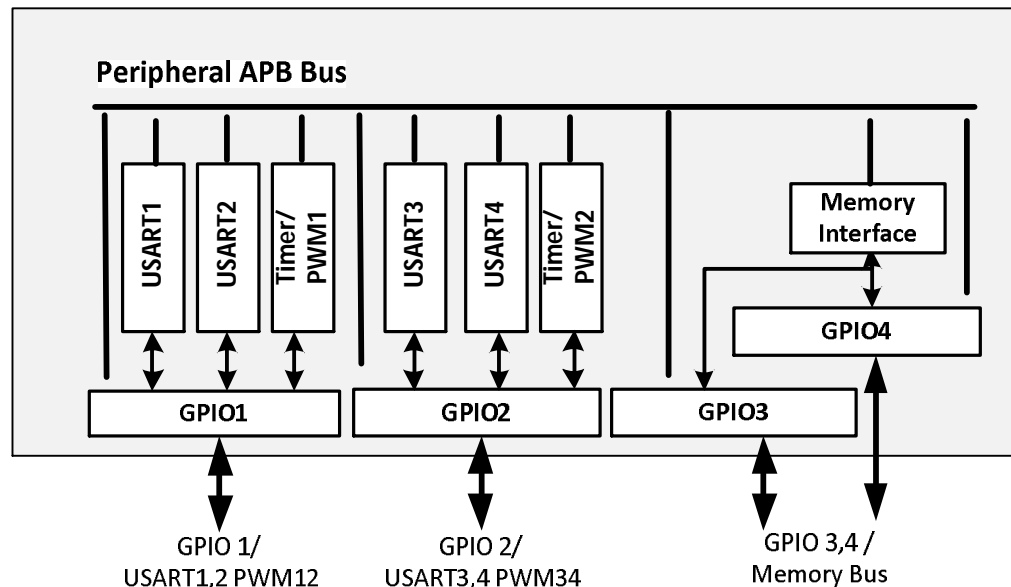
USART

- New master/slave SPI 8 or 16 bit function
- New control signals for Map(rx) and PacketWire(tx) support
- Up to 4 ports

Peripherals: GPIO

GPIO

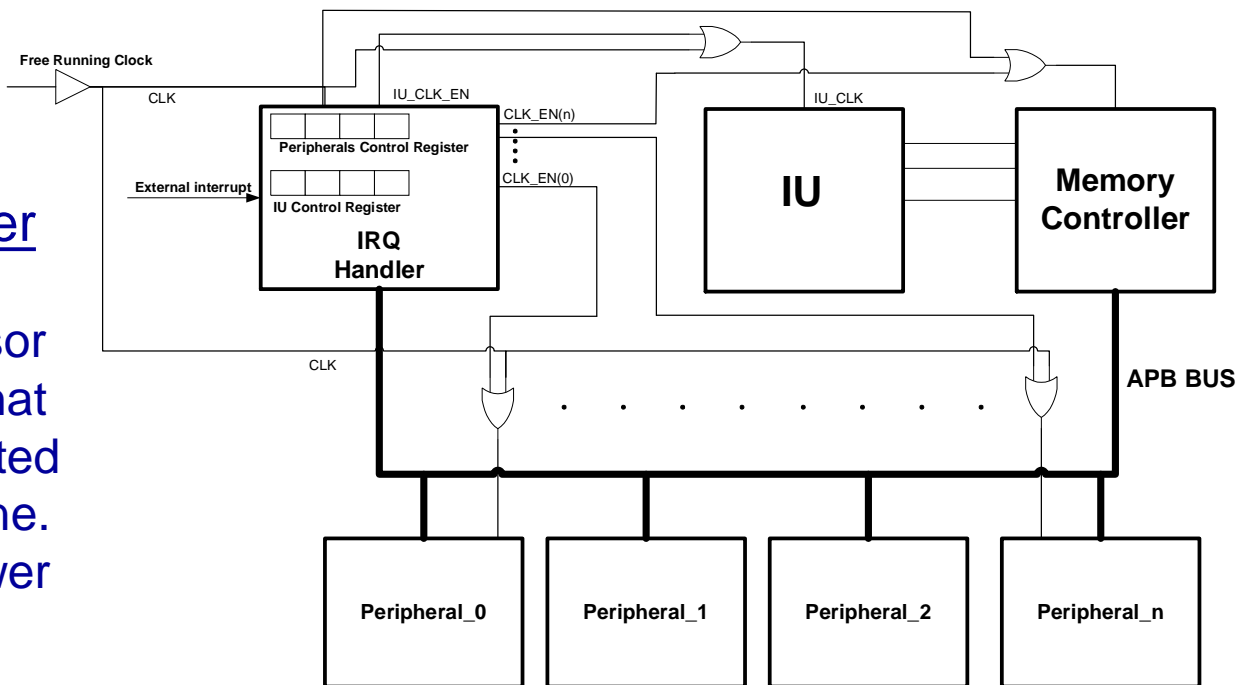
- Line size enlarged to 24 bit bitwise programmable
- “S&H” buffer added to interface parallel ADC/DAC devices (AD774/AD667/..)
- Up to 4 x 24 lines
- GPIO pins shared with all others peripherals and memory controller



Interrupts and Sleep controller

IRQ controller changed in Interrupts and Sleep Controller

Each peripheral and the processor pipeline itself have an 'Enable' that in HW (ASIC) may be implemented as stretch control for the clock line. Disabled blocks go into low-power mode with no switching activity



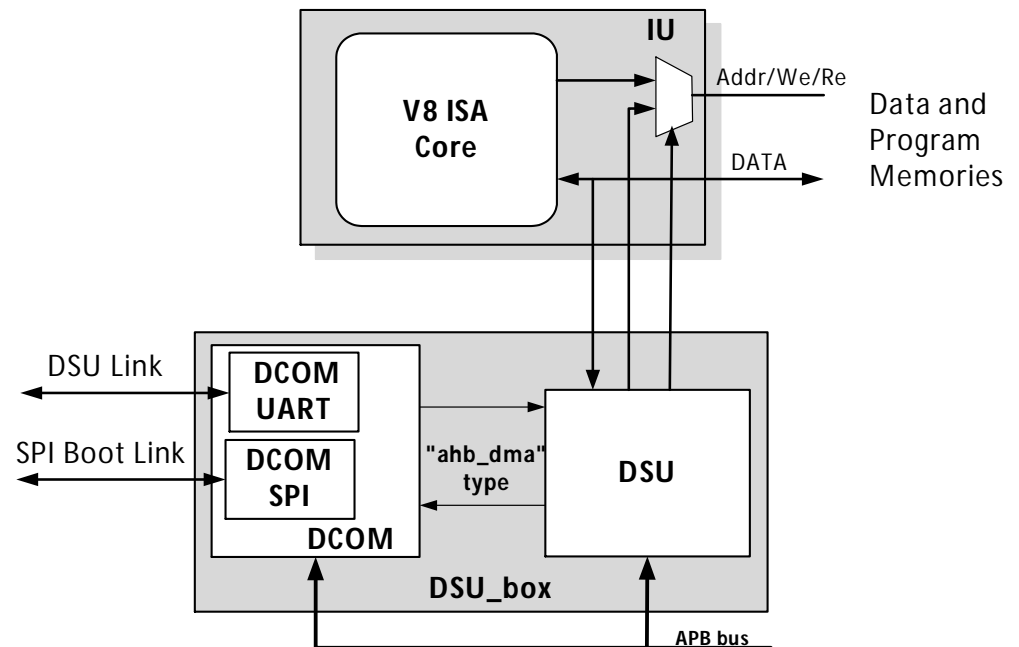
- The programmer can disable any peripheral units (not all of them !) and put to sleep also the processor pipeline.
- Any Interrupt request wakes up the processor. Examples of wake up may be interrupt for timer expiration, incoming message on serial or trigger on external wake-up port.
- At reset processor pipeline and all peripherals re-start enabled

Debug Support Unit

V8uC partially reuses DSU modules of the LEON2-FT.

The DSU box module allows to know the internal state of V8uC microcontroller through its Debug Serial Link.

- A new SPI module is included to allow the fetch of DSL commands from an external Serial Memory as boot option.
- As alternative boot option program image can be directly fetched into local RAM by the DMA engine



Optional Plug-ins: AuFPU, CANBUS

- AuFPU is a pipelined floating point unit implementing sub-set of SPARC V8 FP instructions
- The effective precision is a configurable parameter from the IEEE 754 standard single precision down to custom solutions with reduced mantissa sizes to save resource occupation.

SPARC V8 Instruction	Latency (1)	Sub-module
FADDS	3	FPadder
FSUBS	3	
FCOMPS	3	
FDIVS	8	FPdiv
FSQRT	NA	NA
FMULS	2	Fpmul
FITOS	2	Fpunary
FNEGS	2	
FABS	2	

(1) Latency applies for 20 MHz clock frequency on an Actel ProAsic3 device

- A CAN APB interface working with ESA Hurricane Core is available as communication module

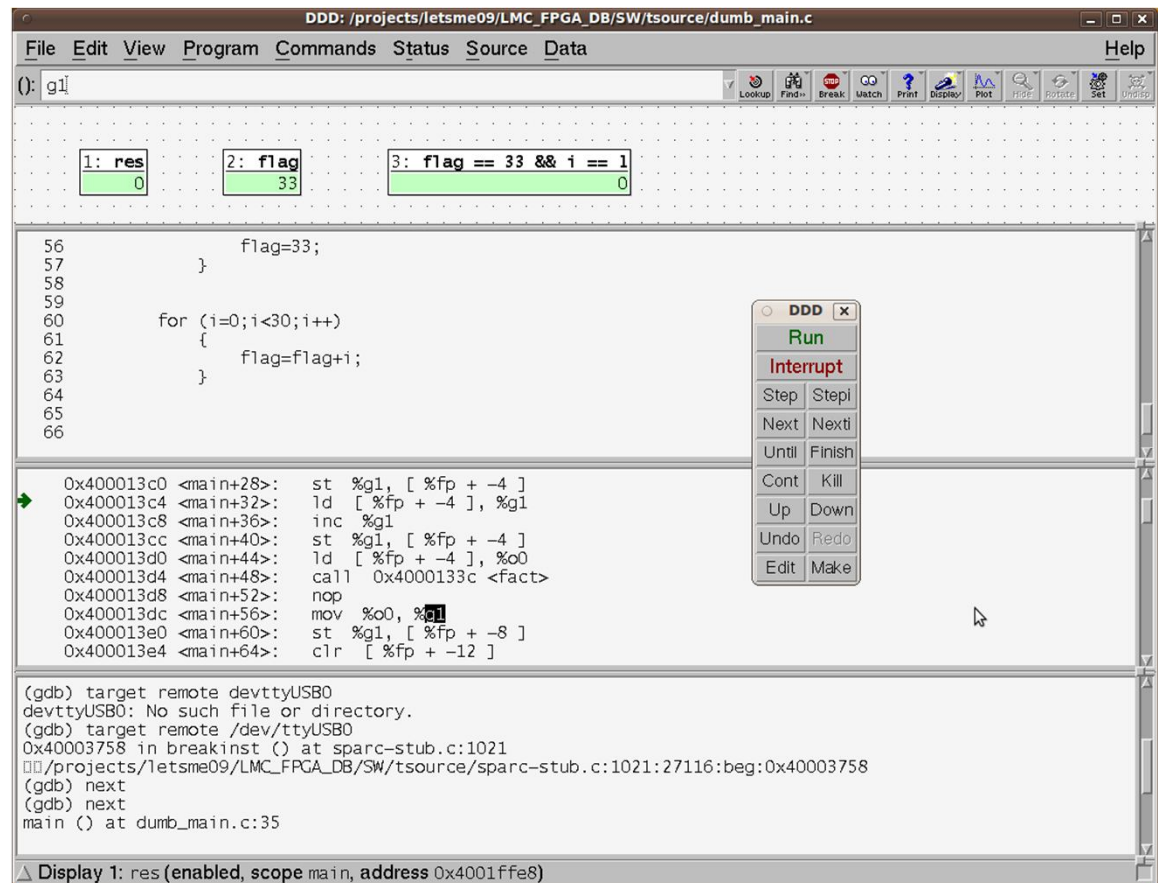
SW development tools : compiler and debugger

V8uC is a SparcV8 processor compliant with the GNU **sparc-elf-gcc** cross compilers.

The core is equipped with: boot code, linker scripts and V8uc-GDB stub file for the easy 'C' programs compiling and debugging with GNU tools.

The **DSU** module has been maintained for compatibility with LEON2-FT Tools.

Except for memory controller and cache settings, V8uC can execute LEON2-FT codes without changes.



```

DDD: /projects/letsme09/LMC_FPGA_DB/SW/tsource/dumb_main.c
File Edit View Program Commands Status Source Data Help
(): gll
1: res 0
2: flag 33
3: flag == 33 && i == 1 0
56      flag=33;
57      }
58
59
60      for (i=0;i<30;i++)
61      {
62          flag=flag+i;
63      }
64
65
66
0x400013c0 <main+28>: st %g1, [ %fp + -4 ]
0x400013c4 <main+32>: ld [ %fp + -4 ], %g1
0x400013c8 <main+36>: inc %g1
0x400013cc <main+40>: st %g1, [ %fp + -4 ]
0x400013d0 <main+44>: ld [ %fp + -4 ], %o0
0x400013d4 <main+48>: call 0x4000133c <fact>
0x400013d8 <main+52>: nop
0x400013dc <main+56>: mov %o0, %o1
0x400013e0 <main+60>: st %g1, [ %fp + -8 ]
0x400013e4 <main+64>: clr [ %fp + -12 ]
(gdb) target remote devttyUSB0
devttyUSB0: No such file or directory.
(gdb) target remote /dev/ttyUSB0
0x40003758 in breakinst () at sparc-stub.c:1021
[]/projects/letsme09/LMC_FPGA_DB/SW/tsource/sparc-stub.c:1021:27116:beg:0x40003758
(gdb) next
(gdb) next
main () at dumb_main.c:35
Display 1: res (enabled, scope main, address 0x4001ffe8)
  
```

Core Numbers: resource budget

Synthesis result of the V8uC core (Configuration: V8uC pipeline, Register File with 8 register windows, V8uc Memory controller with 40 Kbyte +EDAC local RAM, IRQ/Sleep controller, GPIO 24 bit x 3, USART, TIMER-PWM and DSU Box)

FPGA	Total Cells	Utilization
Altera StratixII EP2S60	5473	21%
Actel Proasic3E A3PE3000	25044	33%
Actel RTAX 2000S	16949	53%
Xilinx Virtex4 xc4vlx60	11139	20%

Core Numbers: Dhrystone benchmark

The Dhrystone benchmark for integer computation and strings manipulation has been executed on a V8uC, LEON2-FT and M8051

V8uC: 8 Register windows; 64Kbyte program + 64Kbyte data memories on chip.

LEON2: Instruction Cache = 4 x 8Kbyte x 8byte ; Data Cache = 2 x 8Kbyte x 4byte

FPGA	V8uC	LEON2-FT	MC8051
Clock Frequency	32 MHz	32 MHz	16 MHz
Dhrystone/s	49230	44076	595
MIPS/MHz	0.876	0.780	0.021

V8uC is 10% faster than LEON (Local Memory Vs Caches advantage) and 40 times faster than M8051 (8 bit too few to manage Dhrystone bench !?)

Core Numbers: Dhrystones Memory budgeted

The memory size of the Dhrystones test-bench compiled for V8uC with the Debug option for GDB

FPGA	Text Segment (Kbyte)	Data Segment + BSS (Kbyte)
STD Lib (no printf (*))	8	2
V8uC Lib (Scprintf)	7	
Sparc-Stub	8	8
Dhrystone	10	13
Total	33	23
(*)STD Lib (including printf) → 42 Kbyte Text + 3 Kbyte Data		

The reuse of standard libraries doesn't optimize the program and data memory resources utilization. Specific micro-programmer library are needed

Conclusions

- V8uC core passed the SPARCV8 Compliance Test Suite and additional test procedures for its peripherals
- A specific stress test produced all the possible sequences of 5 consecutive memory instructions also in condition of fault injection and simultaneous memory washing by the scrubber unit.
- The core database has been delivered to ESA, to be subject of further demo and evaluation activity

- Further optimizations are possible
 - on HW: Trap-Table, Code compression and IRQ management
 - and SW: Microcontroller optimized libraries, Instruction Set Simulator ..



Thank you for your attention!

Walter Errico

SITAEEL S.p.A.

Phone: +39 050 9912116

E-mail: walter.errico@sitael.com

SITAEEL S.p.A.

S.P. 231, KM. 97.900 – 70026 Modugno (BA)

Via Livornese 1019 – 56122 Pisa (PI)

<http://www.sitael.com>