EUROPEAN COMMISSION

Thematic Priority:
SIXTH FRAMEWORK PROGRAM

Priority 2.5.3
INFORMATION SOCIETY TECHNOLOGIES
Unit G3 Embedded Systems

Project Acronym:

SOCRADES

Project Full Title:

Service-Oriented Cross-layer infRAstructure for Distributed smart Embedded devices

Proposal/Contract No: EU FP6 IST-5-034116 IP SOCRADES

Deliverable D5.6.2

Microcontroller with Integrated DSP and SIMD Capability

Status:	**Final**
Version:	**V1.0**[1]
Dissemination Level[2]**:**	**CONFIDENTIAL**
Date:	**15.04.2009**

Organization Name of the Lead Contractor for this Deliverable**: ARM**

[1] V0.x before peer-review approval, V1.0 at the approval, V1.x minor revisions, V2.0 major revision
[2] See p.3 for explanation of Dissemination Levels, as defined in the DoW

Status Description:

Scheduled completion date[3]:	28.02.2009	Actual completion date[4]:	15.04.2009
Short document description:	Deliverable D5.6.2 describes the purpose and activities of a project that has the objective of creating a microcontroller with DSP and SIMD capabilities with focus on character data processing.		
Author(s) deliverable:	Paul Marston, Jürgen Jagst (ARM/ARM)	**Report/deliverable classification:** [x] Deliverable [ ] Three-Monthly Activity Report [ ] Six-Monthly Activity Report	
[ ][ ] Partner ↓ Contributions ↓ Peer reviews	[x][x] Schneider Electric Automation [ ][ ] ABB [ ][ ] APS GmbH [ ][ ] Boliden AB [ ][ ] FlexLink Automation Oy. [ ][ ] Institut f. Automation und Kommunikation e.V. Magdeburg [ ][ ] Kungliga Tekniska Högskolan	[ ][ ] Loughborough University [ ][ ] Luleå University of Technology [ ][ ] Politecnico di Milano [ ][ ] SAP AG [ ][ ] Siemens AG [ ][ ] Tampere University of Technology [ ][ ] Jaguar Cars Ltd. [x][x] ARM Ltd. [x][x] Schneider Electric Industries	
Peer review approval :	[x] Approved [ ] Rejected (improve as specified hereunder)	Date:	08.04.2009
Suggested improvements:			

Version History:

Version:	Date:	Comments, Changes, Status:	Person(s)[5]:
0.1	23.02.2009	Draft revision	ARM
0.2	24.02.2009	Provisional diagrams and further detail added	PM/ARM
0.21	24.02.2009	Review session 1, minor changes	JJ/ARM
0.22	25.02.2009	Review session2, minor changes in formatting	JJ/ARM
0.23	27.02.2009	Added chapter 3.4 and 3.5.	JJ/ARM
0.24	09.03.2009	Typo Fixing	JJ/ARM
1.0	08.04.2009	Final	

[3] As defined in the DoW
[4] Scheduled date for approval
[5] A list of company short tags can be found in DoW

Table of Contents:

List of Figures:

Dissemination Level:

PUBLIC	Public
PP	Restricted to other programme participants (including the Commission Services)
RESTRICTED	Restricted to a group specified by the consortium (including the Commission
CONFIDENTIAL	Confidential, only for members of the consortium (including the Commission Services)

1. Introduction

Designers seeking solutions to signal processing and data transmission in low frequency applications have a wide choice of microcontrollers for system control, and a selection of dedicated signal processor units on the other hand. The processing performance of a 32-bit RISC architecture offers the potential for extensive DSP functions to be integrated within the MCU as well as support for single instructions multiple date (SIMD).

Both control and signal processing capability can be managed by one processor with sufficient bandwidth for many low power applications. In particular a wide range of high quality audio functions can be handled in this way, although opportunities extend to applications where MCU is not a common processor platform, such as dedicated communication and data transmission in the industrial, automotive and medical market sectors.

In the Industrial sector a trend can be identified – a move to standard communication interfaces. In the past there have been a wide variety of proprietary hardware communication interfaces, like differential RS485 or 4.20mA current loops, to interface sensors and actuators for e.g. temperature measurement or BLDC-motor control on long distance cable lines in industrial applications.

Data transmission on those industrial lines has never been standardized from a global software & hardware point of view. Demand to decrease cost per unit while increasing communication bandwidth, along with improving performance in the microcontroller sector has lead to the increasing use of multifunctional microcontroller units (MCU) in the industrial communication sector as well, providing of the shelf standard interfaces for long distance data transmission such as Ethernet/MAC. Related to this, there is a trend for moving into standard communication software and stacks like TCP/IP and web services like Data Profile for WebServices (DPWS) even for small microcontroller units (MCU) in the industrial sector.

The high layered software derived from the personal computer (PC) tends to communicate data package payloads in character formats or even in Hypertext Mark-up Language (HTML/XML), which is very well known from common PC-driven internet browsers.

Certain characteristics of those higher layer communication formats can be supported by special instructions on MCU side to accelerate data processing even on very small applications (e.g. sensors). This leads to a new ARM development in the MCU sector which is suitable for those industrial application needs.

2. Architectural and Design

2.1. Motivation

Comprehensive DSP and SIMD capability exists in the ARM processor Instruction Set Architecture. The opportunity to redefine and merge this powerful capability into the mid-range 32-bit MCU architecture is the motivation for this project. The development project is known as 'Merlin'. This is also used as the codename for the processor under development, which will have a similar micro architecture to the existing ARM Cortex-M3 MCU [5].

2.2. Functional Overview

While the open-access architecture level documentation [2] is being revised for MCU class processors, the proposed DSP and SIMD extensions are defined in confidential documents that specify the MCU micro-architecture [1], [3]. The extensions comprise 51 DSP instructions and 38 SIMD instructions, optimised for compact code space by using 16-bit encodings.

The decoding of the DSP and SIMD instructions will be incorporated into the MCU core of a processor with the following top level structure:

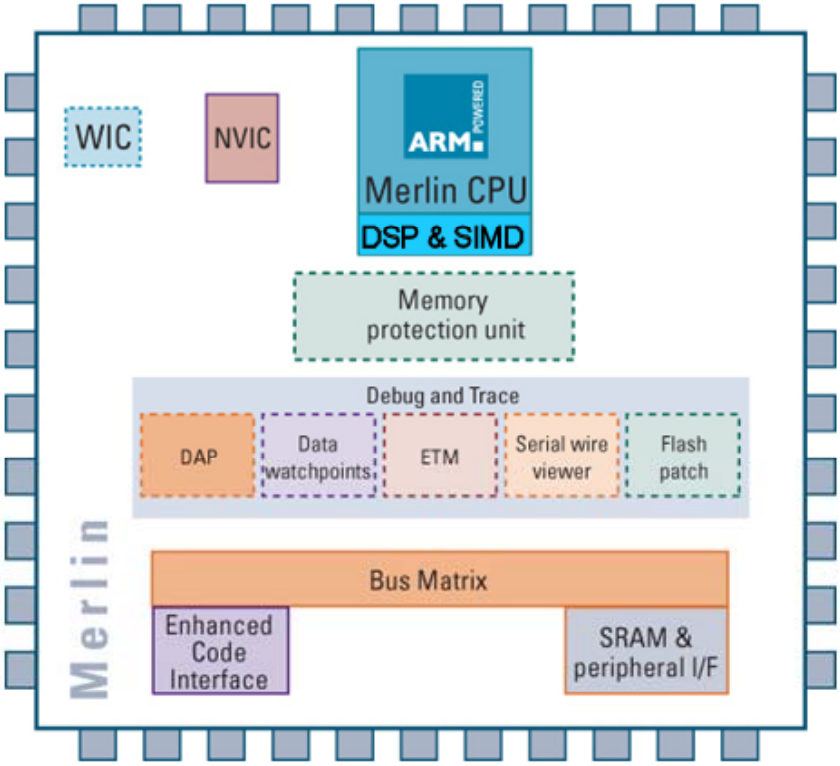

Figure 1: Abstract top level block diagram of 'Merlin' MCU

2.3. Main components

The main components of the 'Merlin' MCU, as identified in Figure 1 are outlined below.

CPU – The central processing unit, or processor core, controls the flow of instruction execution and data processing.

Bus Matrix - Instruction and data memory is external to the MCU. The bus matrix arbitrates the flow of data from memory and peripheral components, as determined by the CPU.

Memory Protection Unit - The MPU ensures that accesses to the defined regions of external memories and peripherals are valid.

Flash Patch - The Flash Patch component provides the ability to replace erroneous code in the fixed instruction ROM with corrected code in changeable memory such as RAM or Flash ROM.

DAP - The Debug Access Port provides and controls access to the MCU memories and registers for debugging purposes. An external debugging tool communicates with the DAP via a serial debug interface.

ETM - The Embedded Trace Macrocell provides a stream of information at real-time, that allows the instruction execution flow of the MCU to be monitored externally.

Data Watchpoints - This component, under debug control, monitors data transfers for specified data items. Upon detection of such data during program execution, this block triggers or halts trace execution by the ETM.

Serial Wire Viewer - This component formats trace information from the ETM as standard ASCII data, sent serially to trace analysis tools through a single pin.

NVIC - The Nested Vectored Interrupt Controller provides a configurable number of interrupts to the MCU, with selectable priority levels. This allows high speed transfer from normal code execution to an Interrupt Service Routine (ISR), and from one ISR to another of higher priority.

3. Software Development

3.1. Development Tools Support

Software development for the 'Merlin' processor, in both C and assembler code, will be supported by standard ARM processor and microcontroller development tools. The complete set of DSP and SIMD instructions for MCU, as defined in [2], will be supported. These tools provide compilation, assembly, simulation, trace and debugging facilities. See references [6] and [7]. A number of independent development tools vendor companies will also add the 'Merlin' MCU to their portfolio of supported processors.

3.2. Library of pre-coded functions

The DSP and SIMD capability of the 'Merlin' MCU will provide the opportunity for a wide variety of existing library applications to be optimized for performance, taking advantage of the available instructions. The 'Merlin' project will accommodate optimizations of popular applications to demonstrate potential improvements over existing processors and MCUs. An example of such an application is the Speex audio codec.
Speex is an Open Source/Free Software patent-free audio compression format designed for speech. Speex aims to lower the barrier of entry for voice applications by providing a free alternative to proprietary speech codecs. Speex is well-adapted to internet applications and provides useful features that are not present in most other codecs. Speex is part of the GNU Project and is available under the revised BSD license.

3.3. MCU benchmarking

In addition to measuring performance improvements in applications, widely recognised proprietary benchmark suites include EEMBC and BDTI. The nature of the EEMBC suites provides little opportunity to measure performance improvements by the use of DSP and SIMD instructions. However, the BDTIbenchmark™ suites [8] are heavily oriented towards DSP functions. BDTI provides independent benchmarking of processors, processor cores, and other processing devices and is recognized as an industry leader and vendor-independent source for signal processing benchmarks.

The BDTI DSP Kernel Benchmarks measure only the performance of the processor core; they do not evaluate the effects of I/O, peripherals, or external memory. These suites will be used to demonstrate the typical improvements achievable by the 'Merlin' MCU in comparison to existing products, with and without DSP and SIMD capabilities, such as ARM966-E [4] and ARM Cortex-M3 [5] respectively.

3.4. XML Text and String Processing

On characteristics in XML is the transmission of information or data in text and/or string format. The receiving unit therefore has to store a string based transmission and has to parse the data to indentify relevant information from it. Based on this it is quite common that a string comparison takes place against an expected string of characters. Other operations will be finding a character or character sequence in a text string, checking characters against an upper and lower limit and finding a small substring within a larger one.

Those operations usually require a complex set of algorithms especially in the case of expression checking and searching parts of a text string in all thinkable cases including transmission errors. Usually a text string

is parsed by a program loop which surrounds a character comparison operation algorithm. This may occupy most MCU performance under worst case conditions (e.g. branch miss prediction, cache squeeze outs).

Therefore the arising popularity of XML-based content transmission leads to arising ECU performance requirements which can be solved by special MCU operations and opcodes. Further information on ARM-SIMD can be found at [1].

3.5. SIMD Example for String Parsing

Further algorithm development directions lead to a parallelism of character processing to accelerate string parsing and comparison. By this algorithm a transposition of character bits takes place like seen in the example below.

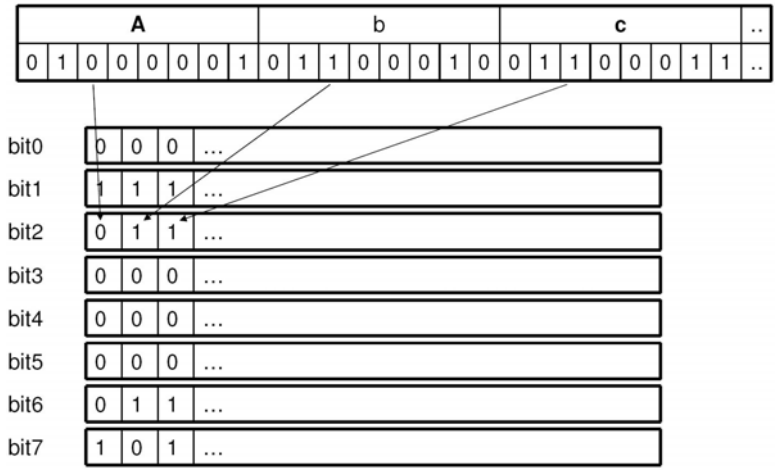

Figure 2: Serial to parallel Transposition

Wide local register sets (e.g. 8x128bit for 128 characters) plus fast manipulation operations on bit-logic shall accelerate operations like this. Bitwise logic and shift operations, bit scans, population counts and other bit-based operations will carry out the desired results. They can easily implemented in fast hardware and provide huge performance improvements as compared to conventional byte-sized character scanning loops.

A detailed theoretical description of string processing algorithms will exceed the focus of this document, further information can be found at [10][11].

3.6. 'Merlin' SIMD Example

Typically, to take advantage of SIMD instructions, an algorithm would need to be restructured. This will involve manual intervention, because compilers cannot easily be directed to identify when SIMD instructions can be applied. As an example of this, the following code represents part of a Viterbi algorithm within a repeating loop.

The original assembler code, that contains no DSP or SIMD instructions, comprises four identical operations, executed sequentially, in the form shown by the first code sequence. However, in the case where SIMD instructions can be used, the whole four repetitions of the original code can be replaced by the second code sequence.

The replacement sequence is less than half the length of the original. As a consequence, the instruction cycle count for the whole algorithm reduces from approximately 26,000 cycles for the original to 15,000 cycles for the SIMD based code.

Note that in these examples, descriptive comments have been removed because of the proprietary nature of the subject matter.

Original code sequence, containing no SIMD instructions, repeated four times per algorithm loop.

```
LDMIA  olds!, {st10, st32}
ADD    tmp, st10, met1
CMP    tmp, tmp, LSL#16
MOVCS  tmp, tmp, LSR#16
ORRCS  trans, trans, #1<<6
STRH   tmp, [news], #8*2
ADD    tmp, st10, met1, ROR#16
CMP    tmp, tmp, LSL#16
MOVCS  tmp, tmp, LSR#16
ORRCS  trans, trans, #1<<14
STRH   tmp, [news], #-7*2
ADD    tmp, st32, met1, ROR#16
CMP    tmp, tmp, LSL#16
MOVCS  tmp, tmp, LSR#16
ORRCS  trans, trans, #1<<7
STRH   tmp, [news], #8*2
ADD    tmp, st32, met1
CMP    tmp, tmp, LSL#16
MOVCS  tmp, tmp, LSR#16
ORRCS  trans, trans, #1<<15
STRH   tmp, [news], #-7*2
```

Replacement code sequence, taking advantage of SIMD instructions, executed once per algorithm loop.

```
;; CREATE BRANCH METRIC VECTORS
PKHBT  r6, r6, r6, LSL #16
PKHBT  r7, r7, r7, LSL #16
SADD8  r4, r6, r7
SSUB8  r5, r6, r7
MOV    r6, r4, LSL #8
MOV    r7, r5, LSL #8
SSUB8  r4, r4, r6
SSUB8  r5, r5, r7
;; GENERATE NEXT STATE CANDIDATES
SADD8  r6, r0, r4
SSUB8  r10, r0, r4
SSUB8  r8, r1, r4
SADD8  r4, r1, r4
SADD8  r7, r2, r5
SSUB8  r11, r2, r5
SSUB8  r9, r3, r5
SADD8  r5, r3, r5
;; GENERATE CANDIDATE DIFFERENCES
SSUB8  r0, r8, r6
SSUB8  r2, r4, r10
SSUB8  r1, r9, r7
SSUB8  r3, r5, r11
;; GENERATE TRANSISITON BITS
SSUB8  r0, r0, r12
SEL    r0, r8, r6
MRS    r8, apsr
SSUB8  r1, r1, r12
SEL    r1, r9, r7
MRS    r9, apsr
;; SELECT NEXT STATE
SSUB8  r2, r2, r12
SEL    r2, r4, r10
MRS    r10, apsr
ORR    r9, r8, r9, LSL #4
AND    r10, r10, #0xF0000
LDMIA  sp!, {r8}
SSUB8  r3, r3, r12
SEL    r3, r5, r11
MRS    r11, apsr
AND    r9, r9, #0xff0000
AND    r11, r11, #0xF0000
```

4. System Development

4.1. Prototyping System

System development based on the 'Merlin' MCU can be undertaken using the FPGA-based Microcontroller Prototyping System (MPS). This comprises the MCU coupled with a customisable example system of common peripheral components and software routines [9]. The 'Merlin' project includes the extension of the MPS for ARM Cortex-M3 to make it suitable for system development based on the features of the 'Merlin' MCU.

Figure 3 shows a conceptual block diagram of the MPS, where the CPU block is an FPGA containing the 'Merlin' design, the DUT FPGA contains the customised system components under development, and the human interface to the MPS is pictorially represented.

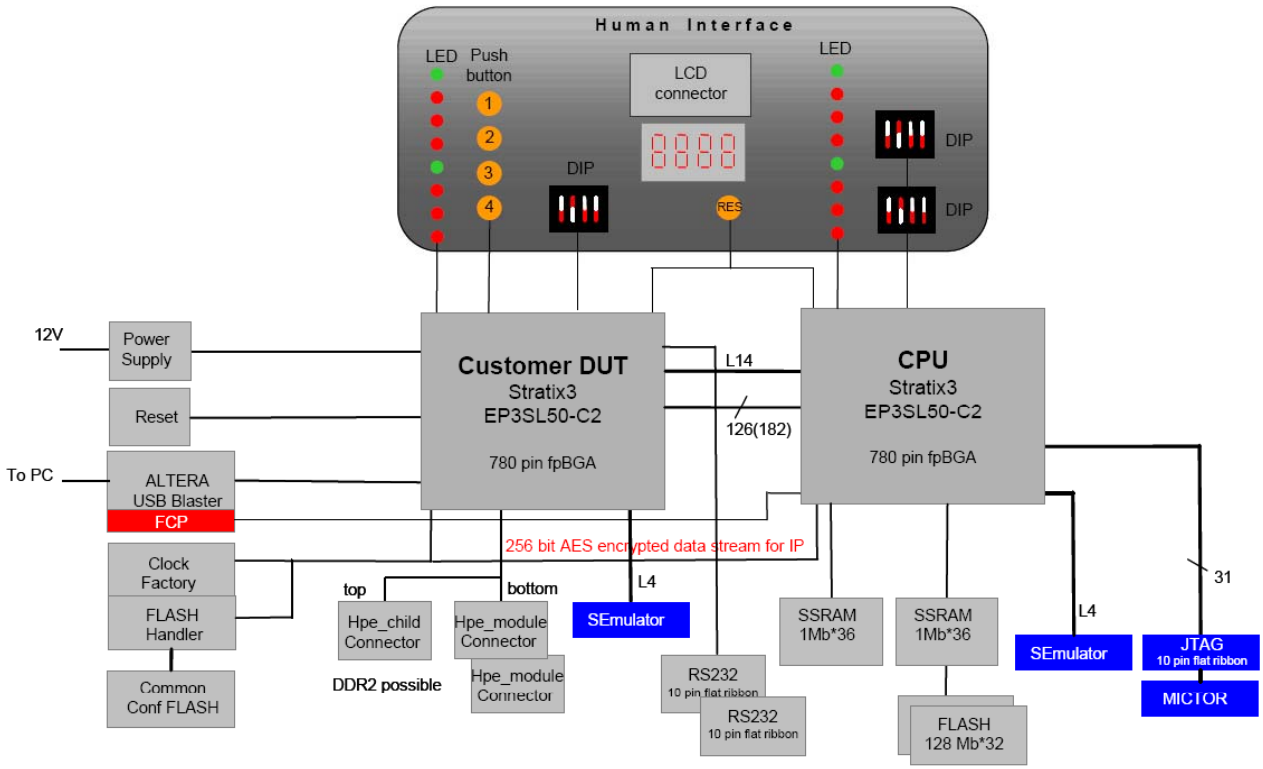

Figure 3: Diagram of the Microcontroller Prototyping System

5. Confidentiality

5.1. Protection of ARM Intellectual Property (IP)

Due to the fact that this new intellectual property (IP) has been developed by ARM Ltd under guidance of EU-funded project 'SOCRADES' which will lead to commercial available products later on, a public technical information which contains all detailed items cannot be provided within this document. Furthermore it will exceed the concept of a single document far to much.

Please refer to chapter 6 or contact ARM for further information.

5.2. Availability of Detailed Technical Information

Detailed technical information for development as seen above is available from ARM Ltd. under certain premises e.g. a Non Disclosure Agreement (NDA).

6. References

6.1. Input Documents

[1]
'Merlin' Engineering Specification (confidential), 2008, ARM Limited, Cambridge, UK

[2]
ARMv7M Architecture Reference Manual, 2006-2008, ARM Limited, Cambridge, UK

[3]
ARMv7M Extensions Proposal (confidential) , 2008, ARM Limited, Cambridge, UK

[4]
ARM Information Center, ARM966-E Technical Reference Manual, 2008, ARM Limited, Cambridge, UK

[5]
ARM Information Center, ARM Cortex-M3 Technical Reference Manual, 2008, ARM Limited, Cambridge, UK

[6]
ARM Information Center, ARM RealView™ Development Suite, 2006-2008, ARM Limited, Cambridge, UK

[7]
ARM Information Center, Keil™ Embedded Development, 2007-2008, ARM Limited, Cambridge, UK

[8]
The BDTI DSP Kernel Benchmarks™, 1999-2009, Berkeley Design Technology, Inc.

[9]
Application Note: Using the ARM Cortex-M3 on the Microcontroller Prototyping System, 2006-2008, ARM Limited, Cambridge, UK

[10]
"Architectual Support for SWAR Text Processing with Parallel Bit Streams: The Inductive Doubling Principle", 2009, Robert D. Cameron, Dan Lin, School of Computing Science, Simon Fraser University, Canada

[11]
"High Performance XML Parsing Using Parallel Bit Stream Technology", 2008, Robert D. Cameron, Kenneth S. Herdy, Dan Lin, School of Computing Science, Simon Fraser University, Canada

6.2. Related standards and norms

- None -

7. Acronyms and Abbreviations

Abbreviation / Acronym:	*Description:*
MCU	Microcontroller (Unit)
SIMD	Single Instruction Multiple Data
RISC	Reduced Instruction Set Computer
BDTI	Berkeley Design Technology Inc.
ETM	Embedded Trace Macrocell
NVIC	Nested Vector Interrupt Controller
WIC	Wake-up Interrupt Controller
DAP	Debug Access Port
DSP	Digital Signal Processor / Processing
IP	Intellectual Property
CPU	Central Processing Unit
EEMBC	The Embedded Microprocessor Benchmark Consortium

As this is a document from professionals for professionals, all other terms are expected to be known.