XtratuM Integration on Bespoke HW with TDMA-based Memory-Bus Arbitration. Assessment of Temporal Isolation and Performance Analysis.

Deliverable 6.6.2

Project acronym : MultiPARTES
Project Number: 287702
Version: v1.0
Due date of deliverable: September 2014
Submission date: 20/11/2014
Dissemination level: Confidential
Author: TUWien, UPV, fentISS

Part of the Seventh Framework Programme
Funded by the EC - DG INFSO

Table of Contents

1 Document History

Version	Status	Date
V0.1	Initial draft	13/09/2014
V0.2	Update with UPV contribution	18/10/2014
V0.3	Update with fentISS contribution	20/10/2014
V1.0	First release	13/11/2014
V1.1	Revision after internal review	18/11/2014

Approval		
	Name	Date
Prepared	TUWIEN	13 September 2014
Updated	UPV	18 October 2014
Updated	fentISS	20 October 2014
Reviewed	All Project Partners	10 November 2014
Authorised	Salvador Trujillo	20 November 2014

Circulation	
Recipient	Date of submission
Project partners	10/11/2014
European Commission	20/11/2014

2 List of Acronyms

COTS	Commercial off-the-shelf
DSM	Distributed Shared Memory
FPGA	Field-Programmable Gate Array
IPC	Inter-Process Communication
MPT	MultiPARTES Project
SPARC	Scalable Processor ARChitecture
TDMA	Time Division Multiple Access
VHDL	Very High speed hardware Description Language
XM	XtratuM hypervisor

3 Executive Summary

In addition to the initial hardware platform with a LEON3 dual-core processor evaluated in WP4 (referred to as COTS platform in this document), WP6 has consolidated two platforms. The goal was to experiment with the hardware solutions in order to avoid the temporal interference of cores when partitions run in parallel. This document provides the assessment and performance evaluation report of the multicore-platform virtualization layer developed in the frame of work package 6 based on multicore hardware with TDMA-based memory bus arbitration. Deliverable D6.6.1 ("XtratuM integration on bespoke HW with TDMA-based memory-bus arbitration. Hardware design and implementation") details the hardware modifications performed on this platform in order to avoid the temporal interference.

The main goal of this document is to analyse, evaluate and assess the hardware and software with respect to the functionalities achieved in the initial hardware platform and to provide a specific test evaluation of the expected improvements with respect to the temporal interference. The evaluation is based on the test suites used and detailed in D4.5 that have been adapted to the new execution platform.

4 Introduction

4.1 Purpose of the document

This document is one of the outputs of Task T6.5, “XtratuM integration on bespoke HW with TDMA-based memory-bus arbitration”. The goal of this document is to present the analysis and performance assessment of the MultiPARTES multicore-platform virtualization layer when executed on the hardware with TDMA-based memory-bus arbitration.

The general dependence of the task in the new reconfiguration of WP6 and the deliverables is shown on Figure 1.

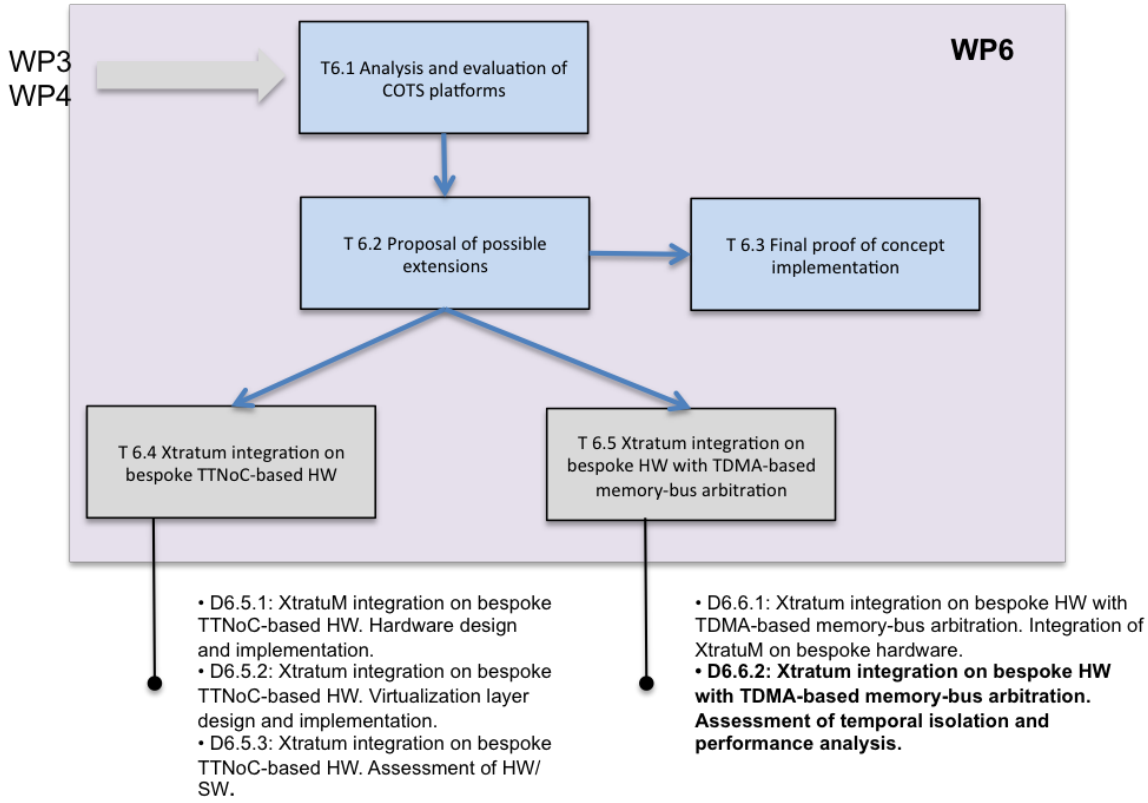


Figure 1 MultiPARTES work package and task dependences

4.2 Output of the work

The main result of this document is the assessment of the system properties, in particular the temporal interference and the analysis of the system performance of the integration of the MultiPARTES virtualization layer on the bespoke HW with TDMA-based memory-bus arbitration.

One important aspect to be pointed out is the focus on the SPARC-V8 platform that has been considered in the activities carried out in WP6. An x86 platform is out of scope of this analysis.

The output of this work will be used in order to investigate future upgrades of the MultiPARTES platform and virtualization layer with respect to system performance.

4.3 Structure of the document

This document is organised as follows:

- Chapter 4 offers the overview and structure of this document.
- Chapter 5 provides the assessment of the system properties, considering as system the hardware and the virtualization layer.
- Chapter 6 defines the metric to evaluate the platform and the set of tests adapted or designed to evaluate it.
- Chapter 7 provides the evaluation results.
- Chapter 8 performs the comparison of the results obtained with respect to the COTS SPARC.
- Chapter 8 concludes the results of this deliverable.

5 Assessment of Temporal Isolation

In D3.5, the analysis of the multi-core platform based on COTS hardware components showed that time guarantees for time-critical virtual partitions cannot be achieved because of the lack to provide temporal isolation on each level of the system, which is required for hard real-time systems. The original platform used a symmetric multiprocessor connected through an AMBA bus with shared main memory, which allowed two processor cores to work in parallel. The Hypervisor was responsible for the resource separation and the execution of the partitions in correct temporal order. This was sufficient to provide temporal isolation only at the core level. If two partitions mapped to two processors where scheduled to overlap, the time properties of each partition where jeopardized. This resulted from the arbitration scheme that forced one partition to wait for an arbitrary time till the other partition would release the shared bus. Such dynamic bus-arbitration was disrupting the scheduling plan. The extension of the COTS solution with the new TDMA-based arbiter eliminates the inter-core interference by assigning a dedicated slot in the cyclic TDMA scheme to each core. In this way, the bus access time of each core is reserved and bounded, and the temporal isolation between resource-sharing cores is guaranteed.

6 Bespoke HW with TDMA-based Memory-bus Arbitration Evaluation

6.1 Introduction

This section provides an overview of the metric used, the tests and results obtained for the evaluation of XtratuM on top of the bespoke HW with TDMA-based memory-bus arbitration developed in WP6.

The goal of the evaluation is to analyse and show that the hardware platform fulfils the same functionalities and solves the problems raised in the COTS platform. The main problem raised was the temporal interference due the parallel execution of application code on different cores.

6.2 System architecture

The hardware and software architecture has been described in D6.6.1. Figure 2 shows the main blocks of the system architecture.

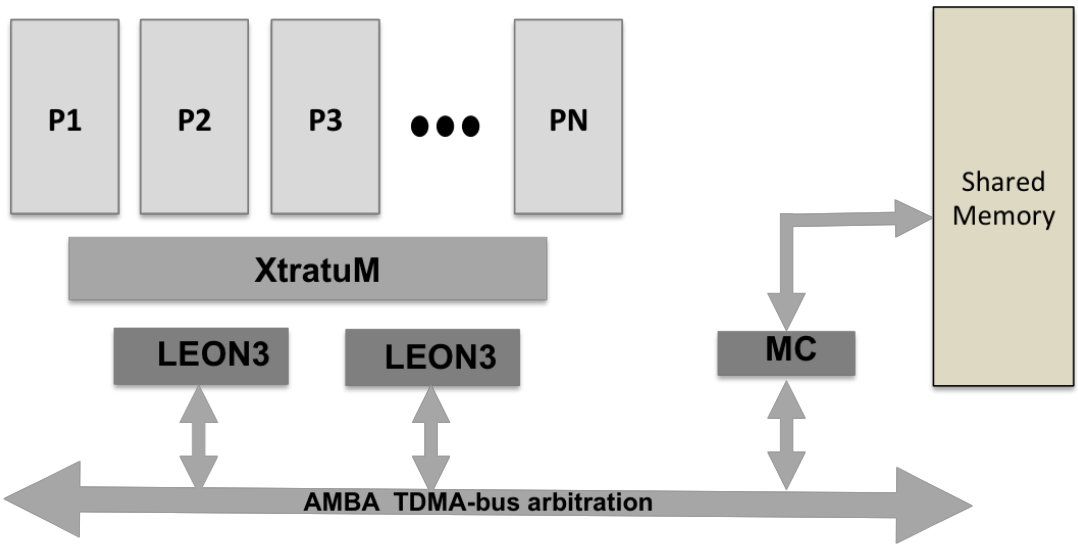


Figure 2: HW and SW architecture

The hardware architecture consists of two LEON3 processors connected through an AMBA bus with TDMA-based arbitration policy that connects through a Memory Controller (MC) to the shared memory.

The virtualization layer realizes Symmetric multiprocessing (*SMP*), which implies that all cores are controlled by one instance of the virtualization layer. The hypervisor architecture is the same that was developed for the initial hardware used in MultiPARTES (COTS solution).

6.3 Functional Evaluation

In order to evaluate the functional behaviour of the hardware solution and virtualization layer adaptation, we have selected as reference test suite the "MPT Abstraction Layer: Conformance test suite" defined in section 7 of D4.4. It defines the set of tests for:

- System Management
- Partition Management
- Inter-Partition Communication
- Time Management
- Multiple Module Schedule
- Health Monitor Management
- Trace Management

Section 7 details the results, considerations and limitations of this test-set execution on the target of the evaluation.

6.4 Performance Evaluation

In order to evaluate the bespoke HW with TDMA-based memory-bus arbitration MultiPARTES platform, the same performance metric as defined in deliverable D4.6 "Platform validation report" has been used. The metric consist on the following elements:

1. Maximum and average partition context switch time: maximum and average time for switching between two partitions in the scheduling plan.
2. Effective slot time of partition execution: time available to the partition code in a slot.
3. Partition overhead due to the virtualization layer. Performance loss of the partition due to its execution on top of the hypervisor.
4. Stability of the plan in mid and long term: time variability of the Major Frame (MAF) occurrences.
5. Inter-partition communication message transfer: time needed to send a message to resp. receive a message from another partition.
6. Service cost: evaluation of the maximum and average execution time of the hypervisor services.

The detailed specification of the test cases for the collection of the above metrics was presented in D4.6.

6.5 Temporal Interference Analysis

The design of this hardware platform aimed at avoiding the effects of temporal interference of the parallel execution of two partitions on two different cores. In order to ensure that temporal isolation has been achieved on all levels of the system, the evaluation for temporal interference

is first performed on pure hardware, without any OS, and then on the system as a whole, using the hypervisor.

6.5.1 *Evaluation of the Hardware*

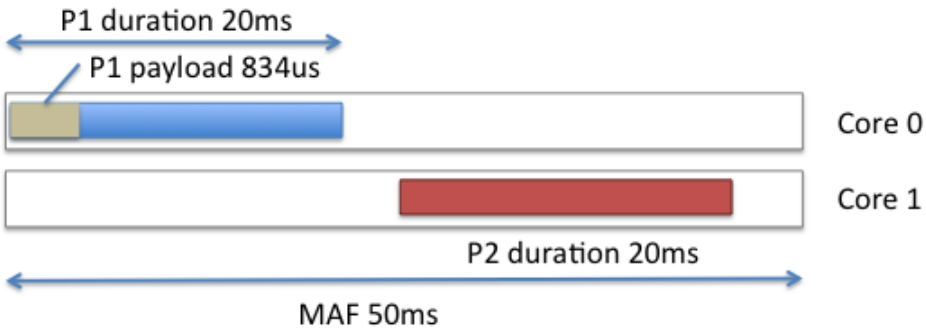
The purpose of the first evaluation is to prove that the temporal properties of the hardware are preserved. In order to avoid any timing effects (ex. OS overhead), the code is executed on a bare-metal platform. A multi-core LEON3 processor with four cores, TDMA-based arbiter and shared main memory is setup on an FPGA board. Two cores are considered as time-critical (assigned to the TDMA arbiter) and the other two as non-critical (assigned to an RR arbiter). The four cores are triggered to start at the same time. Each core executes a Bubble sort algorithm for vectors with the same contents, so that all cores have the same workload. In the experiments, the size of the vectors is varied from 100 to 1000 elements.

6.5.2 *Analysis of the System with Hypervisor*

In order to evaluate the temporal interference for the whole system, the same experiment as described in section 7.2.4 of D3.5 has been adapted to the current platform. Next, a summary of this experiment is presented.

The goal of the experiment was to analyse the effect of the parallel execution of two partitions on different cores with differing time intervals of overlapping execution.

Table 1 Test case description

Test Case Id	T-MTP-INTERF-0110-0XXX
Test purpose	To analyse the effect of the parallel execution of two partitions on different cores with different overlapping-time intervals.
Test Configuration	The scenario is built with 2 partitions. Each partition is allocated to one of the instances of the Virtualization Layer (VL). P1 is allocated to VL1 and P2 to VL2. The scheduling plan consists of one slot with a duration that is large enough to execute the partition payload (20ms). Each test is instantiated with a value for the overlap . This figure shows the example when no overlap is produced. 

Test Description	Partitions P1 and P2 execute a known predefined payload whose execution time has been measured in isolation.
Test constraints	To force the maximum impact of memory interference, the memory areas of both partitions P1 and P2 are configured as non-cacheable.
Test measurement	Time needed for the execution of the payload of P1 and P2 in the different test cases.

The following tests have been considered.

Table 2: Test cases

Test Case Id	Test Configuration
T-MTP-INTERF-0110-0000	P1 and P2 overlap in 0% of their duration
T-MTP-INTERF-0110-0025	P1 and P2 overlap in 25% of their duration
T-MTP-INTERF-0110-0050	P1 and P2 overlap in 50% of their duration
T-MTP-INTERF-0110-0075	P1 and P2 overlap in 75% of their duration
T-MTP-INTERF-0110-0100	P1 and P2 overlap in 100% of their duration

7 Evaluation results

7.1 Introduction

This section provides the results of the functional and performance tests described in the previous section.

7.2 Functional Test Results

This section provides the results obtained with the different test scenarios for the functional evaluation (see Section 6.3 “Functional Evaluation”). The term “OK” or “KO” is used to show a positive resp. negative test result. The term “PASSED” or “NOT PASSED” is used to show the fulfilment resp. non-fulfilment of a global functionality. “PASSED” is used when all tests in the respective category are “OK”.

Table 3: MPT Abstraction Layer test results

Test No	Result
System Management	PASSED
T-API-SYS-0110_0010	OK
T-API-SYS-0110_0020	OK
T-API-SYS-0110_0030	OK
T-API-SYS-0120_0010	OK
T-API-SYS-0120_0020	OK
Partition Management	PASSED
T-API-PM-0210_0010	OK
T-API-PM-0210_0020	OK
T-API-PM-0210_0030	OK
T-API-PM-0220_0010	OK
T-API-PM-0220_0020	OK
T-API-PM-0220_0030	OK
T-API-PM-0220_0040	OK
T-API-PM-0220_0050	OK
T-API-PM-0220_0060	OK
T-API-PM-0230_0010	OK
Time Management	PASSED
T-API-TM-0410_0010	OK
Health Monitor Management	PASSED

Test No	Result
Inter-Partition Communication	PASSED
T-API-IPC-0310_0010	OK
T-API-PM-0310_0020	OK
T-API-PM-0310_0030	OK
T-API-IPC-0320_0010	OK
T-API-IPC-0320_0020	OK
T-API-IPC-0320_0030	OK
T-API-IPC-0330_0010	OK
T-API-IPC-0330_0020	OK
T-API-IPC-0340_0010	OK
T-API-IPC-0340_0020	OK
T-API-IPC-0350_0010	OK
T-API-IPC-0350_0020	OK
T-API-IPC-0360_0010	OK
T-API-IPC-0360_0020	OK
T-API-IPC-0370_0010	OK
T-API-IPC-0370_0020	OK
T-API-IPC-0380_0010	OK
T-API-IPC-0380_0020	OK
T-API-IPC-0380_0030	OK

T-API-HM-0510_0010	OK
T-API-HM-0520_0010	OK
T-API-HM-0520_0020	OK
T-API-HM-0520_0030	OK
Multiple Module Schedule	PASSED
T-API-SCH-0610_0010	OK
T-API-SCH-0610_0020	OK
T-API-SCH-0620_0010	OK
T-API-SCH-0630_0010	OK
T-API-SCH-0630_0020	OK
T-API-SCH-0630_0030	OK
Traces	PASSED
T-API-TRA-000-010	OK
T-API-TRA-010-010	OK
T-API-TRA-010-012	OK
T-API-TRA-010-020	OK
T-API-TRA-020-010	OK
T-API-TRA-020-015	OK
T-API-TRA-020-020	OK
T-API-TRA-020-030	OK
T-API-TRA-020-060	OK
T-API-TRA-020-065	OK
T-API-TRA-030-010	OK
T-API-TRA-030-015	OK
T-API-TRA-030-020	OK
T-API-TRA-030-025	OK
T-API-TRA-040-010	OK
T-API-TRA-040-015	OK
T-API-TRA-040-020	OK
T-API-TRA-040-030	OK
T-API-TRA-050-010	OK
T-API-TRA-050-020	OK
T-API-TRA-050-030	OK

T-API-IPC-0390_0010	OK
T-API-IPC-03100_0010	OK
T-API-IPC-03110_0010	OK
T-API-IPC-03110_0020	OK
T-API-IPC-03110_0030	OK
T-API-IPC-03110_0040	OK
T-API-IPC-03110_0050	OK
T-API-IPC-03110_0060	OK
T-API-IPC-03120_0010	OK
T-API-IPC-03120_0020	OK
T-API-IPC-03120_0030	OK
T-API-IPC-03120_0040	OK
T-API-IPC-03120_0050	OK
T-API-IPC-03120_0060	OK
T-API-IPC-03130_0010	OK
T-API-IPC-03130_0020	OK
T-API-IPC-03130_0030	OK
T-API-IPC-03130_0040	OK

7.3 Performance Evaluation

This section presents the performance test cases and different configurations that were used in order to retrieve the performance metrics presented in Section 6.4 of this deliverable that is based on the test description detailed in D4.6.

These experiments have been carried out with a TDMA-slot duration of 20 clock cycles. This duration was determined by evaluating the intra-core interference for code with aggressive memory access. For slots equal or longer than 20 clock cycles, the execution showed no interference between cores when main memory was accessed.

7.3.1 *Partition Context Switch (PCS)*

The next table summarizes the results of the partition context switch by introducing in the code breakpoints to store time values. Analysing the time values stored, maximum, minimum and average have been determined. This corresponds to the test T-MTP-PERF-0100-0010 (D4.6).

Table 4: Overall Partition Context Switch test results

Processor	TDMA bus arbitration (in µsec)
Average	1042,67
Maximum	1041
Minimum	1056
Standard deviation	2.72

These results show that the maximum PCS time is 1041 µsec. In section 8.3 , this value is compared with the values obtained for the COTS hardware.

7.3.2 *Effective Slot Time of Partition Execution*

The effective slot time used by a partition is the time during which the partition executes its own code. The next figure shows the scheme. The black rectangle refers to the partition context switch executed by the hypervisor.

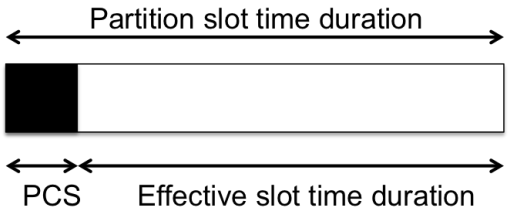


Figure 3: Effective partition execution slot time scheme

The tests defined in D4.6 (T-MTP-PERF-0110-0XXX) have been executed for different configurations of the slot duration specified. In these tests, the columns “Avg”, “Max” and “Min” show the average, maximum and minimum counter increments in a fixed period of 1 second. The “Diff” column contains the differences with respect to a reference value (average

counter with slot of 1000 ms). “Loss” obtains the difference with respect to the reference value in percent. PCS1 estimates the difference in counts with respect to the number of PCS that a partition has required. For instance, in the case of a slot duration of 100ms, the partition has required to execute 10 times the slot in 1 second and, as consequence, 9 PCS have been needed. PCS2 estimates the value of the PCS in µsec taking into account PCS1 and the number of counts that have been measured in the reference value.

Table 5: Effective slot time for TDMA bus arbitration

Slot (ms)	1000	500	100	50	10	5	1
Avg (counts)	624341	623689	618482	611971	559892	494793	0
Max (counts)	624341	623689	618482	611972	559893	494794	0
Min (counts)	624294	623689	618481	611971	559892	494791	0
Diff (counts)	0	652	5859	12370	64449	129548	-
Loss (%)	0.000	0.104	0.938	1.981	10.323	20.750	-
PCS1 (counts)	0.00	652.00	651.00	651.05	651.00	650.99	-
PCS2 (µsec)	0.00	1044.30	1042.70	1042.78	1042.70	1042.69	-

This scenario estimates the partition context switch indirectly through the performance loss of a partition when it is split into several slots. The performance loss corresponds to the external computation required to execute the partitions that are directly associated to the hypervisor scheduling. The reported results confirm the previous measurements of the PCS which were measured by instrumenting the hypervisor code.

7.3.3 *Partition Overhead*

7.3.3.1 Overhead due to the Virtualization Layer

The previous scenarios permit to estimate the performance lost due to the virtualization layer. It strongly depends on the effective time of the partition with respect to the allocated time.

From the previous table:

Table 6: Virtualisation layer overhead for SRAM configuration

Slot(ms)	1000	500	100	50	10	5	1
Avg (counts)	624341	623689	618482	611971	559892	494793	0
Diff (counts)	0	652	5859	12370	64449	129548	-
Loss (%)	0.000	0,10%	0,94%	1,98%	10,32%	20,75%	-

As the effective time of any partition slot is affected by the PCS, it can be expressed as:

$$Effective\_slot\_time_i = Slot\_time_i - PCS$$

In a MAF, a partition can have several slots with different durations. The total time of a partition is the sum of all slots; while the effective time in the MAF of a partition is the total time minus the time the PCS costs.

In general the performance loss can be computed as:

$$performance\ loss = \frac{\sum_n \mathbf{Slot\_time}_i - \mathbf{Effec\_Time\_Part\_MAF}}{\sum_n \mathbf{Slot\_time}_i}$$

7.3.4 *Overhead due to Number of Partitions*

No modifications in the previous results have been observed and consequently no specific tests have been executed.

7.3.5 *Overhead due to the Number of Slots in the Plan*

No modifications in the previous results have been observed and consequently no specific tests have been executed.

7.3.6 *Stability of the Plan*

No modifications in the previous results have been observed and consequently no specific tests have been executed.

7.3.7 *Inter-Partition Communications*

The virtualization layer provides two mechanisms for partition communication: the sampling and queuing port. These services are maintained in the new version with the constraint that partitions have to be allocated in the same node. Specific services for inter-node communications have been defined and evaluated.

The same tests, T-MTP-PERF-0140-0XXX (defined in D4.6), have been executed with the following results.

Table presents the performance results for the inter-partition communication. Time values are in µsec.

Table 7: Inter-partition communication costs

Message Size	Read sampling	Write sampling	Send queuing	Receive queuing
32	381	390	417	411

64	394	402	436	430
128	419	427	462	456
256	471	478	513	507
512	573	581	615	609
1024	778	786	815	808
2048	1189	1195	1230	1224
4096	2012	2027	2063	2039

Figure 4 shows these results in a graphical way. As it can be seen, the timing cost is linear with the number of bytes transmitted in the message.

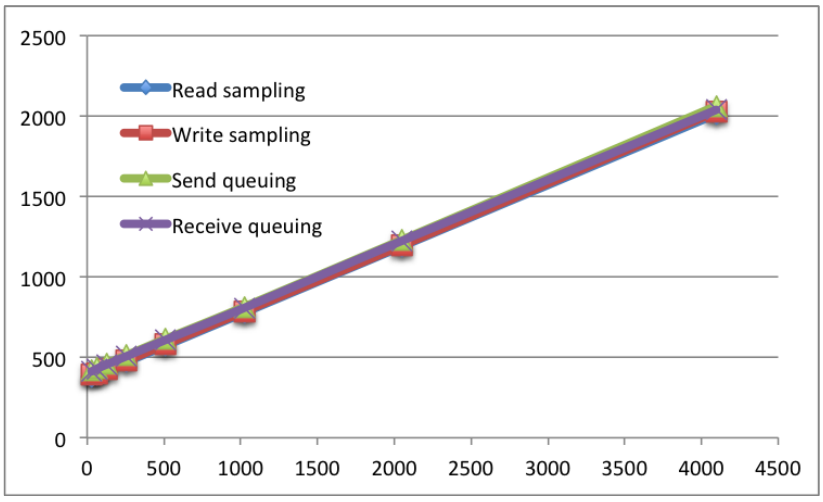


Figure 4: IPC Service costs

7.3.8 *Service Costs*

The next table summarizes the timing cost of the services offered by the virtualization layer.

Table 8: Cost of the services of the virtualization layer

Flush at the end of the slot	NO		YES	
	Min	Max	Min	Max
XM_sparc_clear_pil	38	39	126	127
XM_sparc_set_pil	18	19	111	112
XM_clear_irqmask	57	62	139	146
XM_sparc_inport	85	86	174	180
XM_sparc_outport	56	57	156	163
XM_get_time(XM_HW_CLOCK)	55	58	137	144
XM_get_time(XM_EXEC_CLOCK)	55	58	137	144
XM_set_timer(XM_HW_CLOCK)	59	62	145	153

XM_set_timer(XM_EXEC_CLOCK)	58	60	143	152
XM_hm_status	128	131	253	257
XM_trace_open	44	75	159	187
XM_trace_event	8	9	91	91
XM_trace_status	119	120	243	249
XM_create_sampling_port	180	191	335	341
XM_create_queuing_port	147	150	320	326

These results show the cost of the different services. For instance, for XM_get_time (XM_HW_CLOCK) the maximum measured cost is 58 and 144 µsecs, respectively, depending on the cache flush disable/enable at the end of the previous slot. Flushing the cache at a PCS permits to have a more predictable timing of these services.

7.4 Temporal interference analysis

7.4.1 *Temporal Interference Analysis of the Hardware*

This subsection evaluates the hardware considering the temporal isolation between cores and also the performance properties between critical and non-critical cores.

Table 9 : Execution time of bubble sort algorithm in core level

Array size	core 1 - TDMA	core 2 - TDMA	core 3 - RR	core 4 - RR
100	20660	20660	3092	3145
200	84185	84185	12600	12754
300	188480	188480	28456	28347
400	333990	333990	50135	50524
500	518660	518660	77944	78042
600	742865	742865	111326	112442
700	1017210	1017210	152698	153205
800	1344445	1344445	201361	202083
900	1709935	1709935	256150	256965
1000	2094420	2094420	317925	318722

Table 1 shows that the execution time (in clock cycles) for core one and two is the same since each memory access is strictly determined by the TDMA-based arbiter, whereas core three and core four have different times, because of the dynamic properties of RR arbitration.

7.4.2 *Temporal Interference Analysis of the System*

For the temporal interference analysis of the whole system, the tests T-MTP-INTERF-0110-0XXX as described in section 6.5 are used. The tests measure the time required by a payload to complete its execution. Different levels of interference are defined. The next table shows the results in µsec.

Table 10 Temporal interference (cache disabled)

	S0	S25	S50	S75	S100
Avg (µsec)	120099	120099	120099	120099	120099
Max (µsec)	120100	120100	120100	120100	120100
Min (µsec)	120097	120096	120098	120098	120098
Stdev	0,99	1,03	0,97	0,96	0,96
Inteference	0%	0%	0%	0%	0%

The same scenarios have been executed with cache enabled. The next table shows the obtained results.

Table 11 Temporal interference (cache enable)

	S0	S25	S50	S75	S100
Avg	48100	48100	48100	48100	48100
Max	48101	48101	48101	48101	48101
Min	48098	48099	48099	48098	48098
Stdev	0,92	0,91	0,90	0,91	0,92
Inteference	0%	0%	0%	0%	0%

The main conclusion of these results is that there is no difference among the scenarios and, as a consequence, the execution of an application on a core is not affected by the execution in other cores.

8 Comparison with COTS Solution

This section compares three aspects of the evaluation: temporal interference, computation time and partition context switch of both hardware platforms (COTS and bespoke HW with TDMA bus arbitration).

8.1 Temporal Interference

The same loads as used in the scenarios detailed in Section 5.5 have been executed on the COTS platform. Table 12 shows the results obtained for the COTS platform.

Table 12 Temporal interference COTS (cache disabled)

	S0	S25	S50	S75	S100
Avg	20645	23544,0	27181	30577,6	33418
Max	20700	23557	27326	30614	33691
Min	20698	23305	27253	30535	33596
Stdev	0,63	12,46	16,77	17,19	22,81
Inteference	0%	14%	32%	48%	62%

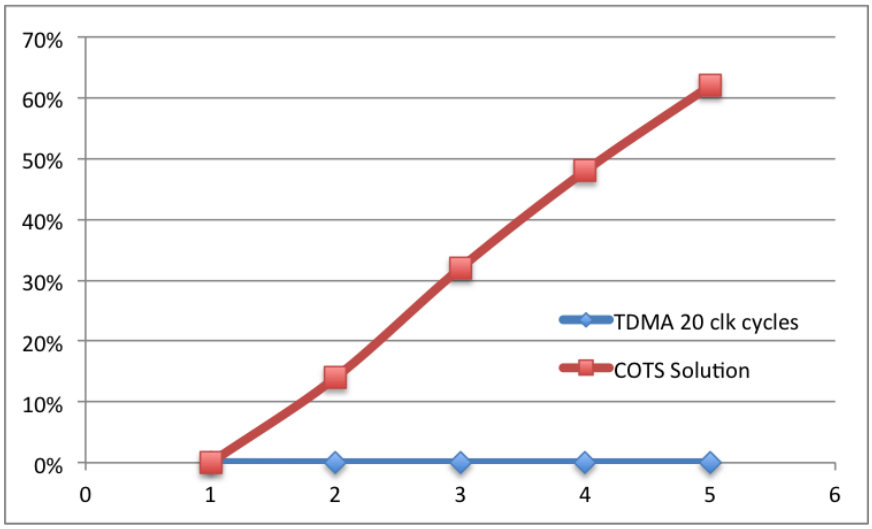
Extracting the results from both tables (10 and 12) in the same conditions (cache disabled), and comparing the interference, the results are:

Table 13 Temporal Interference comparison, bespoke TDMA-based bus arbitration vs. COTS

	S0	S25	S50	S75	S100
TDMA	0%	0%	0%	0%	0%
COTS Solution	0%	14%	32%	48%	62%

The next figure shows a graphical comparison of these results.

Figure 5 Temporal interference, bespoke TDMA-bus arbitration vs. COTS (cache disable)



8.2 Computation Time

This parameter tries to compare which is the computation time needed by each platform to execute the same payload.

Table 14 Temporal Interference comparison, computation time (cache disabled)

	Time (µsec)	MHz
TDMA 20 clk cycles	120099	50
COTS Solution	20645	50
TDMA20/COTS	5,80	1

When comparing the measured time to execute the same payload, it can be seen that the TDMA-based bus-arbitration platform is 5.8 times slower than the COTS platform.

8.3 Partition Context Switch

The same order of slowdown factor should be obtained in the PCS measurement. Extracting results from previous tables, it is observed that the increment of the PCS is 4.65.

Table 15 Temporal Interference comparison, partition context switch time (cache enabled)

	PCS (µsec)
TDMA	1042
COTS Solution	224
TDMA/COTS	4,65

As conclusion, it can be said that the TDMA platform provides full temporal isolation but is limited in its performance. While the COTS platform can operate with a reasonable overhead at partition periods higher than 1 millisecond, the TDMA can work with similar overheads at partition periods higher than 10 milliseconds. In space applications, the normal periods of activities are higher the 10 milliseconds. Thus the platform seems to be suited for these kinds of applications.

9 Conclusion

This document presented the evaluation of the bespoke TDMA-based HW and the integration of the hypervisor on this platform. This hardware platform was initially used in base experiments in WP6. Later it was decided that this platform should be used as one of the target platforms for the implementation and evaluation in the final development of MultiPARTES. The hardware design and implementation have been described in deliverable D6.4 and the virtualization-layer adaptation in D6.6.1. The evaluation described in this document is based on the developments detailed in both deliverables.

This document detailed the activities carried out to evaluate this hardware platform. These activities were:

- Assessment of the temporal isolation of the platform,
- Functional test-suite from D4.4,
- Performance evaluation to measure the impact of the hardware modifications,
- Temporal interference analysis to validate the impact of the interference of the execution of other cores,
- Result comparison between COTS solution and bespoke TTNoC based HW.

As result of all these activities, it can be concluded that this hardware platform fulfils the temporal isolation on all levels of the system.

10 References

[1] MultiPARTES FP7 project, IST-287702, D4.4 “Validation plan”.
[2] MultiPARTES FP7 project, IST-287702, D4.5 “Tool chain implementation”.
[3] MultiPARTES FP7 project, IST-287702, D4.6 “Performance Evaluation”.
[4] MultiPARTES FP7 project, IST-287702, D6.4 “Final implementation and assessment of the proposed architectural solutions”.
[5] MultiPARTES FP7 project, IST-287702, D6.6.1 “Xtratum integration on bespoke HW with TDMA-based memory-bus arbitration. Integration of XtratuM on bespoke hardware”.