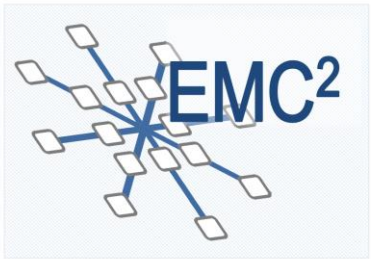


Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments

Project Acronym:

EMC²

Grant agreement no: 621429

Deliverable no. and title	**D4.14 – Recommendations for the certification of mixed-criticality applications on multi-core systems**	
Work package	WP4	SP_Multi-core Hardware Architectures and Concepts
Task / Use Case	T4.2, T4.3	Advanced processor concepts and architecture definition Core and chip interconnect, peripheral devices
Lead contractor	Infineon Technologies AG Dr. Werner Weber, mailto:werner.weber@infineon.com	
Deliverable responsible	NXPGE Michael Philipp, michael.philipp@nxp.com	
Version number	v1.0	
Date	12/04/2017	
Status	Final	
Dissemination level	Public (PU)	

Authors

Partici-pant no.	Part. short name	Author name	Chapter(s)
18H	UoBr	Steve Kerrison	3. - University of Bristol (18H UoBR)
01R	NXPGE	Michael Philipp	4. – NXP Germany (01R NXPGE)

Document History

Version	Date	Author name	Reason
0.1	08/03/2017	Michael Philipp	First Draft
0.2	31/03/2017	Michael Philipp	Incorporated review comments
0.9	12/04/2017	Haris Isakovic	Final review
1.0	12/04/2017	Alfred Hoess	Final editing and formatting, deliverable submission

Publishable Executive Summary

The deliverable D4.14 is intended to give recommendations for the certification of mixed-criticality applications on multi-core systems based on the concepts described in deliverables D4.11, D4.12 and D4.13.

Two approaches to the certification topics are presented based on different backgrounds. One approach focuses on exploiting predictable network structures with the example of the MCENoC, developed by UoBR. The other approach proposes a certification process for a car radio application on a multi-core system.

The two different proposals show that a generalized recommendation of a certification process is not possible and highly dependent on the application, but the strategies how a certification procedure can be described are being presented.

Table of Contents

List of figures

1. Introduction

1.1 Objective and scope of the document

The document provides the results of the work performed by the University of Bristol and NXP in WP4, specifically tasks T4.2 and T4.3. It includes a short description of the motivation behind the activity of each partner, general considerations of the certification process of mixed-criticality applications in multi-core system and an example related to the individual work of each partner.

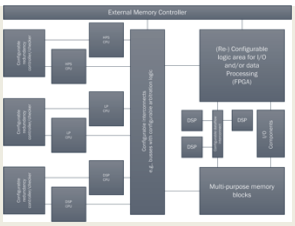
1.2 Structure of the deliverable report

The document is organized in three parts: a general part with the introduction and executive summary followed by the contributions of the both partners.

2. University of Bristol (18H UoBR)

Technology lanes covered:

- **Heterogeneous Multiprocessor SoC architectures**
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- Virtualization and verification technologies



UoBR's focus has been on a predictable NoC architecture for EMC2 type systems. As such, recommendations stem from the innovations made in this area, focusing on hardware capabilities and how these are significant for software applications running upon them. This follows the discussion of formal verification methods and non-blocking network architectures previously presented in deliverables D4.11 and D4.12. Each recommendation is described in turn in the following subsections.

2.1 Ensure all essential hardware components behave predictably

In a multi-core system, time-deterministic predictable processors can guarantee timing behaviour local to each core. However, there is a need for the communication infrastructure to be similarly predictable. Traffic must be schedulable and prioritised on an EMC2 network-on-chip (NoC) to ensure that tasks of different criticality cannot interfere with each other's operation.

This can be equated to choosing processors that are cache-less for real-time embedded use. In order to accurately predict execution time, the memory access time must be tightly bounded. The hit vs. miss latency of a cache reduces predictability, and the performance benefit of a cache may need to be ignored in such systems, to account for the possibility of a miss adversely affecting execution time. In a communication network, varying latency and contention can introduce similar unpredictability, unless the network infrastructure is designed appropriately.

2.2 Enable the addition of workloads non-disruptively

The below figure gives a simple illustration of a multi-core system with a 2D mesh network and a set of possible communication dependencies. Same-coloured arrows represent communications between parts of the same task-group (for example data-forwarding in a pipeline of computation), whilst different-coloured arrows represent communication between distinct tasks or groups. An extra task is added to the system, using spare resource, but creating a new communication dependency, both depicted with dashed lined objects. This creates contention existing communication along its route through the mesh. Under ideal circumstances, adding further tasks could be done in a way that provably avoids disrupting the behaviour of other tasks, reducing the effort required to achieved renewed certification.

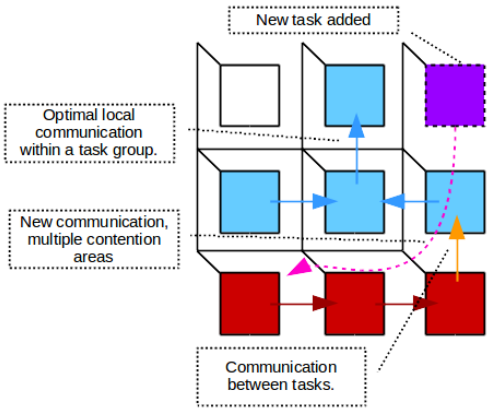


Figure 2: 3x3 mesh network example with potential communication contention

Potential solutions include routing the contentious traffic differently, prioritising traffic appropriately, re-locating tasks or scheduling communication to avoid the contention. However, providing these may require more complex hardware, or may change the timing and behaviour of existing behaviour, requiring the full system to be re-examined in depth for certification.

We demonstrate a simplification that can be achieved by using the MCENoC network structure presented in [1] and discussed in deliverables D4.4 and D4.5. The MCENoC uses a Benes structure to allow N concurrent 1-to-1 communications in a network of N nodes. This is achieved through a fixable number of switching stages. An eight node example, is depicted in the following two figures, first with a single route highlighted through a five-stage network, then as an example where the network is "folded" to connect its inputs and outputs to nodes (i.e. processors).

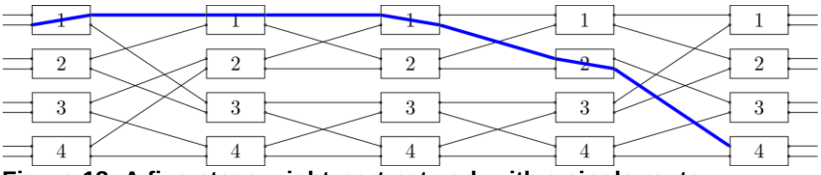


Figure 18: A five-stage, eight-port network with a single route highlighted

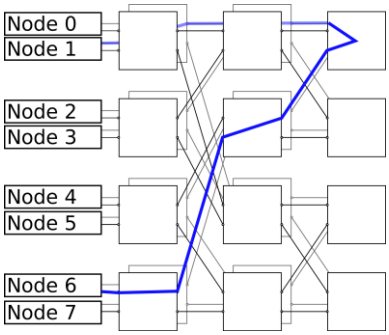


Figure 34: The same network, folded and depicting node connections

The structure of the MCENoC reduces the impact of additional traffic by removing any indirect contention. A re-imagining of the task communication scenario is shown in the below figure. The MCENoC structure is abstracted away, but the capabilities described in the previous paragraph are assumed. Traffic passing through the MCENoC is not contented, except where a common destination is required. In this case, only one such case exists - the communication from two blue nodes to another blue node. This contention also existed in the mesh, and in either case, is resolvable by scheduling these communications separately, through an appropriate method such as time-triggered networking or flow control.

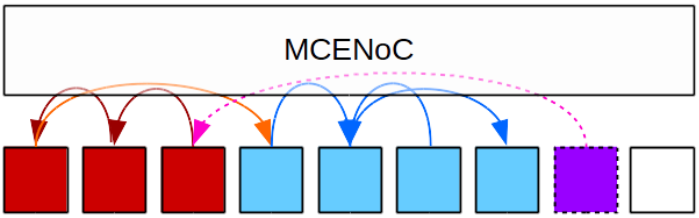


Figure 50: Traffic through the MCENoC has no contention, except at egress points

With the MCENoC, the effect of the new communication has been minimized, because its traffic cannot interfere with other communications, each logically appearing to follow an exclusive path from source to destination, progressing without blocking each other.

2.3 Simplify task allocation by equalising communication overheads

In many network structures, the communication latency between nodes will depend on the locations of the respective nodes. Thus, upper and lower bounds on a communication may exist, which must be considered in the system implementation. First, it may require pessimism in the allocation of resources, where for example worst case execution time must be considered. Secondly, it motivates the allocation of tasks for spatial locality, putting communicating activities physically close to each other to maximise performance, reduce latency, and reduce risk of interference.

In the MCENoC, a Benes structure places all nodes equidistant in terms of latency, and the width of the network is equal to the number of nodes, reducing contention only to nodes that are sought by multiple sources simultaneously. By using this type of network or similar, the task allocation problem is now simpler. Further, the equal latency between all nodes tightens the best- and worst-case bounds, potentially fixing them to a single value in an appropriately synchronised system.

Communication structures such as MCENoC do not remove all challenges of scheduling and allocation. For example, allocating time and bandwidth on the network must still be calculated to ensure throughput and response times per-task are preserved, which remains non-trivial. However, at least one dimension of uncertainty is removed.

2.4 Partition the network where stronger separation is desired

In scenarios where the level of criticality dictates strong separation between portions of the system, the network should be partitioned. This can theoretically be achieved in any network with appropriate routing schemes or link configuration. In the MCENoC, partitioning can be achieved by fixing the routing configuration within particular network stages, to produce any number of sub-networks of size 2^N.

The following figures depict two examples of 4 port partitions created in an 8 port networks. Example routes for each partition are displayed in a unique line-style and colour, and the switches that have their configurations fixed are highlighted in grey boxes. All other switches are used to enable routing within each partition. As a result of this capability present in the structure of MCENoC, it provides good flexibility separating activities, even allowing alternate routing and scheduling strategies to be used within sub-networks. If TDM phases are also strongly enforced, then multiple partitioning schemes can be overlaid onto the same network over time.

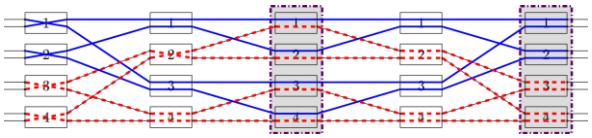


Figure 66: Partitioning ports 0,1,2,3 and 4,5,6,7

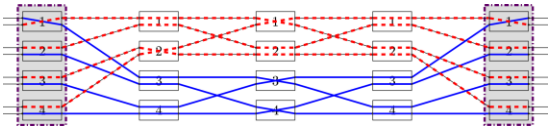


Figure 79: Partitioning ports 0,3,5,7 and 1,2,4,6

2.5 Summary

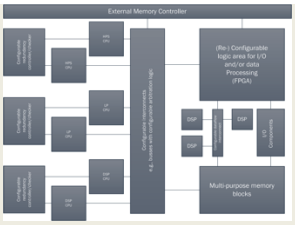
The approaches recommended herein focus on exploiting predictable network structures. The MCENoC, developed by UoBR during this project, is used to frame these recommendations. The primary objective of the MCENoC is to deliver a network with improved predictability, allowing more precise calculations to made in activities such as task scheduling and reducing the impact of adding new workloads to systems that increase in size and complexity over time.

A more in-depth discussion of components of this work, which extends that presented in [1], is currently being prepared for submission to a journal, with a target submission date of April 2017.

3. NXP Germany (01R NXPGE)

Technology lanes covered:

- **Heterogeneous Multiprocessor SoC architectures**
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- Virtualization and verification technologies



3.1 Motivation

With the steadily increasing performance of semiconductor products, the complexity of applications is growing as well, offering a wide set of features. These products need to satisfy different requirements. Besides the sheer performance of an application, the most important aspects are safety and security. Safety critical applications are found in products where failure could have a negative effect on the health of people or significant risk for the environment, security critical applications are sensitive to loss of personal data or vulnerability to hacking.

For end users, it is hard to judge, if a product supports the standard performance requirements and if it guarantees a high level of safety and security. Certification of products helps the consumer to make a first choice by guaranteeing a minimum level of safety, security, functionality and performance.

Due to the complexity of multi-core mixed criticality systems, they are prone for failures. One important aspect for the development of multi-core mixed criticality systems is a suitable debugging environment, as was discussed in D4.13 with the example of the NXP software defined radio (SDR) concept for car radio applications. Based on these applications, a certification approach will be proposed.

3.2 General considerations

Certification can be defined as the process of confirming, that a certain product has passed performance tests and quality assurance tests, and meets qualification criteria defined by experts in this field.

There are general certifications based on, for example, the IEC 61508 standard (Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems) and also certifications for specific applications like DAB radio, for example the Digital Radio UK TickMark[1]. Among the many aspects of certification, some will be discussed in the following with the focus on multi-core systems.

In most mixed-criticality applications on multi-core systems, high priority functionalities co-exist with less critical software. High priority functionalities can be either safety and security related but also include functionalities for real time applications like augmented reality.

Certification of these systems should focus on two aspects:

1. High priority critical functionalities must be guaranteed to have a very high level of reliability
2. Less critical software parts, covering mostly the feature set of a product, must be guaranteed to offer a minimum functionality to the user for the designated application

The reliability is defined as '1-Probability of failure'. The required reliability level is defined by the certification body and depends on the application/product to be certified.

With an increasing feature set, more resources are required in a multi-core system with shared resources and therefore the risk of interference with the security-critical functions is increasing. In many safety-critical applications non-interference of safety-critical components by non-critical ones is ensured by strict isolation of components with different criticalities. This approach has the disadvantage that the utilization of resources might be low and inefficient. Efficient multi-core systems overcome this disadvantage with smart scheduling algorithms for maximum reuse of resources.

[1] www.getdigitalradio.com

Certification of applications with mixed criticalities must take both aspects into account, but the focus must clearly be on aspect one, the critical functionalities. The optimal approach for a certification would be to validate the different subsystems on various confidence levels. But as certification is intended to be used for many products from different suppliers with different architectures, the certification object must be seen as a black box and the test scenarios must be more abstract.

Depending on the application, worst case scenarios must be defined. These must include all possible situation that might lead to a failure of the critical function. Example use cases are:

- corner cases of the application
- maximum load on the system
- maximum number of parallel processing tasks
- duration tests with test times significantly above the typical usage time of the application
- harsh environmental conditions (e.g. high and low temperature)

There a many standardized performance criteria to measure the robustness, a certification procedure must consider. One is the reliability factor to give an example. It's typically defined as the mean time between failures (MBTF). MTBF is calculated with the following equation:

MTBF = (total elapsed time – sum of downtime) / number of failures

Another one is the worst-case execution time (WCET) that must be considered in real-time systems.

Aspect two, the certification of the less critical software parts, is not in the main focus of certification but is still being addressed, as these functions typical represent the feature set of convenience functions of the application. This part of the certification defines a minimum set of requirements a product must provide to make use of the basic functionality of an application.

3.3 Example of the NXP multi standard SDR platform

The NXP multi standard SDR platform we described in D4.13 is used for several applications, which are subject to certification. For all supported digital radio standards HD-Radio, DAB+ and DRM a certification procedure is defined by independent organizations. The aim of the certification process is to guarantee the end user of the product, that it supports the basic functions of the application. None of these applications is safety critical but each contains parts with high criticality levels and low criticality levels.

A proposal for a certification process of DAB+ digital radio for automotive use will be briefly described in the following section. The certification tests can be divided into several groups:

- Real time behavior
- Robustness
- Compatibility
- Feature set

A car radio does not have security- and safety-critical functions. But it contains multiple parts with different levels of criticality. The real-time processing of the received audio stream is the high priority function of the application thus representing the high criticality part of the application. Other functions like data service reception and background scanning are of low criticality as the user would not recognize a failure immediately.

The above defined test groups are described in the following section with the focus on the criticality aspect:

Real time behavior

Delayed processing of data services might not be crucial to users or even not recognizable, but continuous audio reception must be guaranteed under all circumstances. Tests in this groups must cover scenarios with maximum load of the system, e.g. with the maximum allowed audio data rate and data service reception in parallel under difficult reception conditions.

The picture in Figure 7: simplified structure of a DAB+ receiver depicts a simplified architecture of a

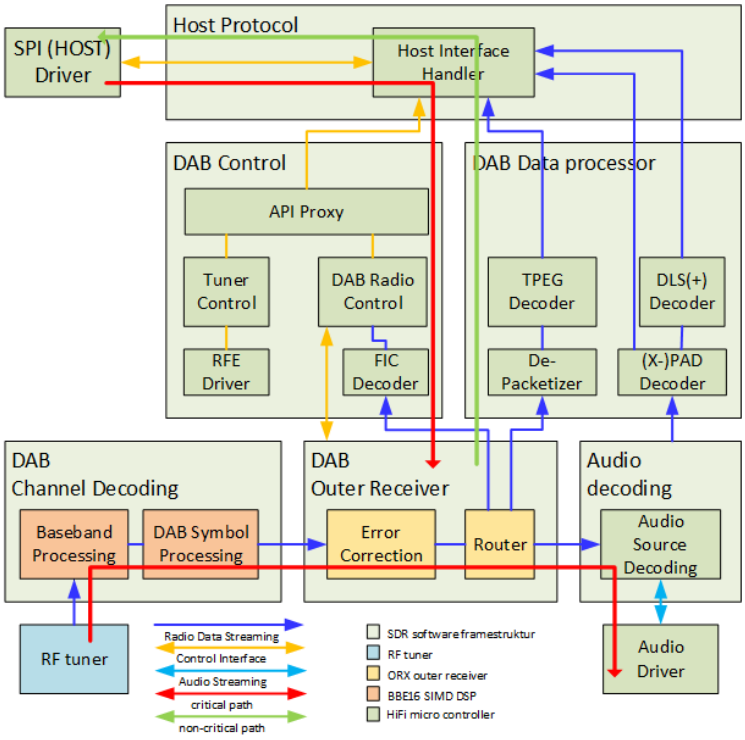


Figure 80: simplified structure of a DAB+ receiver

DAB+ receiver.

As can be seen, there are two critical paths in the system and one non-critical. One critical path coming from the host system must guarantee the real-time processing of host commands to the receiver, the other critical path must guarantee the reception and decoding of audio data in real time. In parallel there is a non-critical path for the reception of data services. On all paths, resources are shared between the critical and non-critical parts of the multi-core system.

The certification must test that the scheduling algorithm of the multi-core system is prioritizing the critical tasks.

Robustness

Also this group of tests targets the highly critical parts of the system and is intended to verify that the product is robust under severe conditions. It includes high and low temperature tests, as well as duration tests for up to 24h. Especially in automotive applications, very high and low temperatures can affect the behavior of the hardware.

Compatibility

This group of certification items checks the compatibility of the application to the standard, as well as to other devices in case there is a communication between two or more devices. Particularly inter-operability of different manufactures must be guaranteed. Most of these functions are non-critical.

Feature set

Next to the basic functionality of an application, an end user expects a number of additional features. Certification typically defines some of these features that are seen as must haves for a good user experience. In the DAB+ context this might be for example switching to another frequency in case the service is lost without user interference. These features are always non-critical.

3.4 Summary

Certification of mixed-criticality applications in multi-core systems is highly dependent on the type of application. It should consist of two parts: validation of critical functions and non-critical functions of the application. The focus of the certification is clearly on the critical functions, not only if they are safety critical. As the certification object is a black box, the certification must define tests that validate the certification object under worst case conditions without having detailed knowledge of the products architecture.

4. References

[1] S. Kerrisin, D. May and K. Eder, "A Benes Based NoC Switching Architecture for Mixed Criticality Embedded Systems," in *IEEE 10th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC 2016)*, Lyon, France, 2016.

5. Abbreviations

Table 1: Abbreviations

Abbreviation	Meaning
µC	Micro-Controller
SDR	Software Defined Radio
EMC^2	Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments
DAB	Digital Audio Broadcast
DRM	Digital Radio Mondial
MPSoC	Multiple-processor System-on-chip