

# A multi-core architecture for incCar digital entertainment

**Citation for published version (APA):**

Moonen, A. J. M., Berg, van den, R. M. J., Bekooij, M. J. G., Bhullar, H., & Meerbergen, van, J. (2005). A multi-core architecture for incCar digital entertainment. In *Proceedings of GSPx Conference*

**Document status and date:**

Published: 01/01/2005

**Document Version:**

Accepted manuscript including changes made at the peer-review stage

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# A Multi-Core Architecture for In-Car Digital Entertainment

Arno Moonen<sup>1,2,3</sup>, René van den Berg<sup>2</sup>, Marco Bekooij<sup>3</sup>, Harpreet Bhullar<sup>2</sup> and Jef van Meerbergen<sup>1,3</sup>

<sup>1</sup>Eindhoven University of Technology  
P.O. Box 513, 5600 MB Eindhoven, The Netherlands  
Telephone: +31 (0)40 247 3394

<sup>2</sup>Philips Semiconductors Nijmegen  
HT, Gerstweg 2, 6534 AE Nijmegen, The Netherlands  
Telephone: +31 (0)24 353 3551

<sup>3</sup>Philips Research Laboratories Eindhoven  
WDC31, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands  
Telephone: +31 (0)40 27 42310

{Arno.Moonen,Rene.van.den.Berg,Marco.Bekooij,Harpreet.Bhullar,Jef.van.Meerbergen}@philips.com

*Abstract*—This paper presents a new multi-core architecture for in-car digital entertainment. Target functions vary from terrestrial reception, digital reception, and compressed audio, up to handsfree voice with acoustic echo cancellation and USB media playback, possibly in different user modes like single versus dual media sound. In the near future, new functions like near field communication, wireless streaming, storage, digital rights management, navigation and video become important. The main challenge is that the platform must be open for future functions, which are unknown at design time. Another challenge is to reduce the design effort by maximizing reuse of hardware and software, especially from related domains like Consumer Electronics (CE). This paper describes a multi-core architecture using a network-on-chip, which provides the required flexibility and scalability. The area overhead, due to the network, is estimated to be 1.5% compared to the current architecture. Furthermore, it is shown that the latency is comparable to the current architecture.

*Categories and Subject Descriptors*—System on Chip: Multi-core architecture and the use of this in car entertainment environment.

*Keywords*—Embedded systems, multi-processor, network-on-chip, car radio, car entertainment, low cost.

## I. INTRODUCTION

Current car radio applications are serving different functions. This varies from terrestrial reception, digital reception, compressed audio, up to handsfree voice with acoustic echo cancellation and USB media playback. Furthermore, it is expected that wireless communication, storage, navigation and video will become important. These

functions typically process data streams and have real-time constraints.

The functions are currently implemented in the in-car digital entertainment platform of Philips Semiconductors. One of the chips on this platform is the multiprocessor SAF7780 [1] [2], which is used in this paper as a reference. The SAF7780 contains four EPICS DSP cores [3], four accelerators, a number of peripherals and a ARM based subsystem.

This paper discusses a new architecture based on a network-of-chip. The pros en cons of the new architecture are quantitatively compared to the reference architecture. The solution is optimized for functions that process streams of data.

The paper is organized as follows: Before going to a new architecture, first the overlap and differences from the CE and the automotive domain are explained in section II. The next section will give the characteristics of the current application in SAF7780. Section IV explains the bottlenecks in the current architecture and the expected problems in next generations. Based on these results a new multiprocessor template is proposed in section V. The template uses new concepts like network-on-chips that provide a scalable and flexible communication infrastructure. Section VI proposes a network based architecture, which is open for introducing new features from the other domains like video and digital TV. The network is analyzed in terms of area and timing in section VII. It is

shown that the timing constraints of existing algorithms can be met, despite the use of a network. This is important to overcome new field and quality tests for existing algorithms. Finally, in the last section, there is a conclusion.

## II. CE VERSUS AUTOMOTIVE DOMAIN

Entertainment functions are present in 3 different user domains, namely Consumer Electronics (at home), Automotive (on the road) and the PC world (at work). The portable devices are key to transfer data between these domains. Typical functions in the *at work* domain are email access, calendar entry, wireless phone access based on near field communication. In the *at home* domain users are interested in functions as music playback, radio of all kinds of formats and supporting media, video and TV. In the *on the road* domain functions like navigation, road access services and eSafety are important. Users expect the same user interface in the different environments. They want to experience the same look and feel. Therefore, it is expected that those domains will converge towards one platform. Furthermore this helps to reduce the cost by increase reusability.

At the other hand there are also differences, like time for introduction of new features, life time of features and devices, quality and cost. For automotive products the planning and development of a new device takes about 3 years and the life cycle is about 8 to 10 years. For the CE products the planning and development time is 6 to 9 months and the life cycle is about 1.5 years. For the PC world this is even faster (several months). A longer life cycle requires a higher product quality, which brings the topic of migrat-

ing the different domains into one platform to the biggest challenges of today and the next generations of products. To survive the automotive life cycle, all functions should have a high quality, flexibility and upgrade possibility.

## III. APPLICATION CHARACTERISTICS

In this section the characteristics of the application are investigated and it is shown that these functions typically process data streams and have real-time constraints. In this paper, these functions are called streaming functions.

The functions can be started and stopped by the user. Each possible set of simultaneously activated functions is called a use-case. The number of use-cases is increasing rapidly. Functions can be started or stopped while others continue. This switching between use-cases is a dynamic process, which must be handled at run-time. Obviously, functions can only be started if enough resources are available.

Streaming functions can be presented by a graph that consists of tasks, which communicate via a point-to-point connection. The granularity for communication is a token. An example of a token is an audio stereo sample, MP3 frame, pixel, video line or a video frame. Synchronization between tokens is done via a First In First Out (FIFO) buffer. In the streaming model it is made explicit which data is private (state) and which data is shared (communication). The tasks have random access to their private state memory as well as random access within a token, despite the FIFO synchronization between tokens.

Functions have timing constraints that can be classified as Hard Real-Time (HRT), Soft Real-Time (SRT) and Best Effort (BE). In the case of a HRT function it is not allowed to miss any deadline. An example of a HRT function is an audio stream that is connected to a digital to analog converter. In this case missing a deadline will result in an unacceptable audio quality. The functions in SAF7780 from Philips Semiconductors are typical examples of HRT function. In the case of SRT functions, for example video, it is allowed to miss some deadlines but the miss rate should be preferably low. BE functions like access services don't have deadlines, they finish as soon as possible.

## IV. ARCHITECTURE CHALLENGES

The bottlenecks in the current architecture as well as the expected problems in the future are described here. Based on these results and the requirements from the application



Fig. 1. Connected planet experiences

it is possible to explore new architectures.

Architectures for Embedded Systems implemented in Silicon are quickly evolving towards reconfigurable multiprocessor architectures combining programmable CPU and DSP cores with application specific cores. This way platforms can be carefully optimized for the target domain and still be programmable. The SAF7780, for example, contains an ARM7 micro-controller subsystem, four EPICS DSP cores, four accelerators and a number of peripherals that connect the chip with the outside world. There are different solutions for communication in different parts of the SAF7780, as shown in Figure 2. One is the Digital In/Out (DIO) switch with four masters (EPICS cores) and a number of slaves (peripherals and accelerators). Another one is the Inter Tile Communication (ITC) channels between the four EPICS DSP cores. The DIO switch is in the critical path and is not scalable for next generations. The ITC channels support *remote write* to another tile but not *remote read*. ITC channels are sufficient for streaming communication but defining the size of the memories is becoming extremely difficult. Therefore in the future remote read is needed as well.

In next generations it is expected that the multiprocessor architecture will become communication centric. Getting the right data to the right place at the right time will dominate the architecture. In order to master the complexity in the deep sub-micron processes, it is necessary to come up with a generally applicable and scalable communication infrastructure. Networks-on-silicon can deliver a scalable and flexible communication architecture and serve as the basis for next generation platforms.

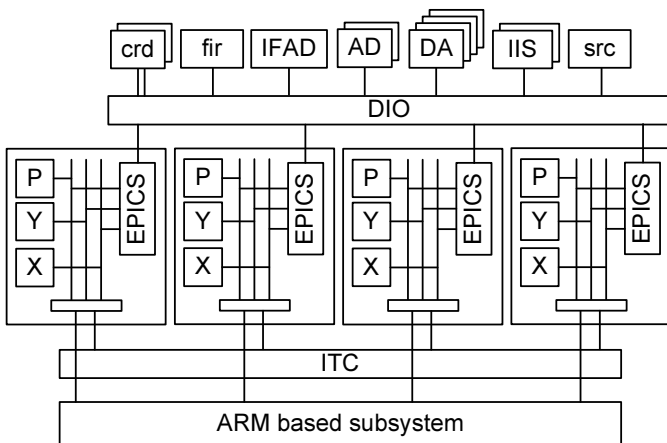


Fig. 2. The architecture of the SAF7780

The aim of this paper is to replace both the DIO switch and the ITC channels from the SAF7780 with a network such that the architecture is scalable and flexible. Two aspects are studied in detail: the latency and the cost. The latency of the network will be analyzed to verify that the timing requirements are met. Cost is defined in terms of silicon area.

## V. ARCHITECTURE FOR IN-CAR ENTERTAINMENT

The architecture that is explained here is optimized for streaming functions and is based on the Hydra template [4], which is shown in Figure 3. The architecture consist of tiles that communicate via a packet switched network. The tiles contain a programmable core, a memory, a Communication Assist (CA) [5] and a network interface (NI). The NI connects the tile to the network. The CA has 2 tasks. The first task is copying data between the NI and the memory such that the core is decoupled from communication. The second task is the arbitration between the CA and the core for accessing the memory.

For this paper the  $\mathcal{A}$ ethereal network [6] is chosen. The  $\mathcal{A}$ ethereal network consists of Network Interfaces (NI) that are connected via a link to one Router (R). Any topology of the routers in the network is possible, like a mesh, tree or ring. The NI can have multiple ports. Each port of the NI is linked to a number of FIFOs in the NI. The number of FIFOs and the FIFO capacities can be chosen at design time. The network is able to set-up a point-to-point connection between different FIFOs at run-time. Each point-to-point connection can support Guaranteed Throughput (GT) service or Best Effort (BE) service. The bandwidth of a GT connection is configurable at run-time and is fixed. The latency of a GT connection is bounded. In the case of a BE connection there are no guarantees for the provided bandwidth and the worst-case latency. Tiles connected to the  $\mathcal{A}$ ethereal network can have different clocks because the

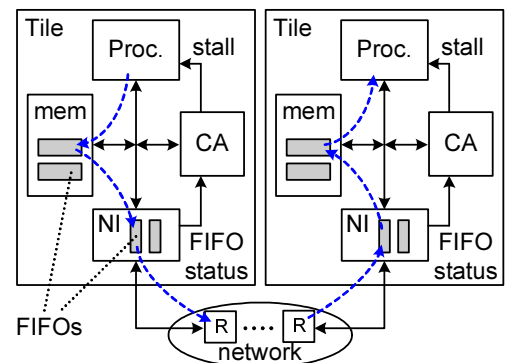


Fig. 3. Streaming multiprocessor template

NI supports clock domain crossing.

Streaming communication between two tasks takes place via a communication channel at the function level. Given our architecture there are two options for implementing a communication channel. The first option is to implement the communication channel as a dedicated point-to-point connection in the network. The second option is to implement communication channels onto one network connection by transferring an address and data using the shared address space.

The implementation of the first option is shown in Figure 3 by the arrows with the broken line. There is a dedicated network connection, which consists of a FIFO in the producing NI and a FIFO in the consuming NI. The producing task, which is executed on the processor, writes its token to a logical FIFO in its local memory. The CA copies the data into a FIFO of the NI. The data is transported over the dedicated point-to-point connection to the NI at the receiving tile. As soon as the data arrives in the FIFO of the consuming NI, it is copied by the CA into the logical FIFO in the local memory of the receiving tile. The data is read from this FIFO after the consuming task has detected that there is sufficient data in this logical FIFO. This option makes use of local and fine grain synchronization. Flow control is achieved by making sure that data is not written into a FIFO before it is checked that there is space available. The advantage of this option is that the bandwidth can be defined for each communication channel. The disadvantage is that each communication channel needs a dedicated point-to-point connection in the network and the maximum number of network connections, which can be supported by a NI, is fixed at design time. Therefore, the flexibility of this option is limited. Furthermore, the area of the network is determined by the maximum number of network connections, that are supported by the network. Using one network connection for multiple communication channels would increase the flexibility and decrease the network area.

In the second option there is only one connection that support a shared address space between two tiles. There can go multiple messages over the connection, which contain data as well as an addresses. The latency of this option is higher because the data and address is both send over the network and there are multiple communication channels send over one connection. Another advantage is that there are less network connections needed than in the first option.

## VI. NETWORK BASED ARCHITECTURE

In this section the SAF7780 in Figure 2 is taken as a reference for the network based architecture. The SAF7780 contains four EPICS DSP cores, two CORDIC (crd) hardware accelerators, one FIR accelerator, one SRC accelerator, a number of peripherals and an ARM based subsystem. The ARM based subsystem is more control related and is therefore seen as a single tile.

The first decision is to define the number of tiles and their content in the new network based architecture. The proposed architecture is shown in Figure 4. There are four EPICS tiles that contain an EPICS DSP core, a CA and X, Y and P memory. The accelerators, which have a tight latency constraint with the EPICS DSP core, are connected as a separate tile to the network. A NI can have multiple ports to multiple tiles, which is done in the case of the accelerators in Figure 4. The peripherals are grouped in one tile, which is connected to the network via the NI. The peripherals are passive IPs that are connected to a local bus. In the peripheral tile there is a CA that transfers data between the peripherals and the NI. The ARM based subsystem is connected as one tile to the network via a NI.

The communication channels from and to the peripherals are all implemented via the shared address space. This is possible because in our application communication with the peripherals is not latency critical. The communication channels between the EPICS DSP cores and the accelerators are mapped to a dedicated point-to-point connection to reduce the latency. In section VII the maximum round trip latency between the EPICS and the CORDIC is analyzed. The communication channels between the EPICS DSP cores can be implemented using the shared address

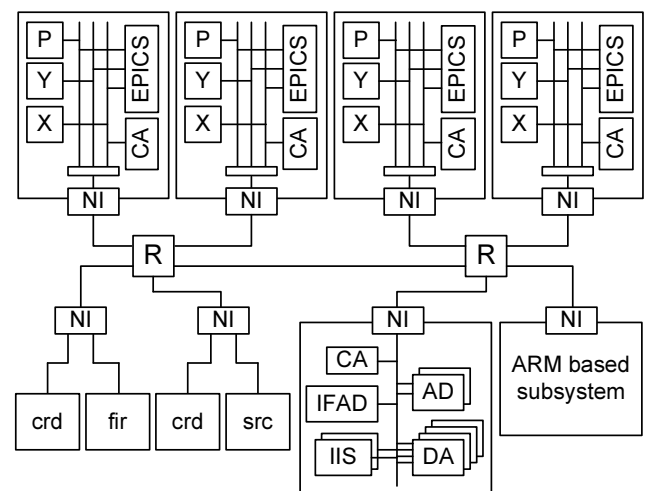


Fig. 4. Proposed multiprocessor architecture

space or the dedicated point-to-point connection depending on the latency, performance and costs. The controller of the system is mapped to the ARM micro controller, which configures the DSP domain using the shared address space.

## VII. NETWORK ANALYSIS

Now the network is analyzed in terms of timing and cost. The current radio implementation is used as the driver for timing analysis because it has tight timing constraints. The network cost is derived with the models of  $\text{\AE}thernet$  and compared with the current architecture.

The current radio implementation contains an adaptive filter that needs to calculate and update new filter coefficients for every sample. This updating of filter coefficients creates a cycle with tight timing constraint. The calculation of the filter coefficient is done by the EPICS DSP core in cooperation with the CORDIC accelerator. The challenge is to come to an architecture with a low round trip latency from the DSP to the accelerator and back, as shown in Figure 5. The architecture should still be flexible such that each DSP is able to use these accelerators.

In the SAF7780 the EPICS and CORDIC are synchronous and the clock frequency is 125MHz in  $0.18\mu\text{m}$  technology. The round trip latency is the execution time of the CORDIC, which is 36 clock cycles.

For analysis it is assumed that one CORDIC execution will consume one input token of 4 words and produce one output token of 2 words. For the round trip latency in Figure 5 all the numbers are translated to  $0.13\mu\text{m}$  technology for comparing the latency. The clock frequency of the EPICS is taken 125MHz, which is the same as in the SAF7780. The EPICS can run at a higher clock frequency in  $0.13\mu\text{m}$  technology but for comparison it is ok to assume 125MHz. The network can run at a clock frequency up to 500MHz in  $0.13\mu\text{m}$  technology. In this technology it is expected that the CORDIC can run at a clock frequency of 250MHz. The network in Figure 5 has a dedicated point-to-point connection for the forward path and the a

dedicated point-to-point connection for the return path. Both connections are GT with a bandwidth allocation of 50%. The flow control of the forward connection is piggy-backed on the data of the return connection. The flow control of the return connection is piggy-backed on the data of the forward connection. The worst-case round trip latency is 37 clock cycles for the EPICS at 125MHz. The execution time of the CORDIC is 18 cycles at 125MHz, because it runs at twice the speed compared to the current architecture. The latency of the clock domain boundaries is 4 cycles at 125MHz. The rest of the latency is introduced by the network. By comparing both round trip latencies it is shown that they are very close. Therefore, the radio implementation is able to meet the timing requirements in the new architecture.

Second aspect to analyze is cost in terms of area. The  $\text{\AE}thernet$  network has a parametric model for calculating the area of a router and a network interface [6]. The model is based on a  $0.13\mu\text{m}$  technology with a clock frequency of 500MHz and a data path width of 32 bits for each link/port. The resulting cost model of a router, given in terms of arity  $a$  is:

$$A_R(a) = 0.808a^2 + 23a(10^{-3}\text{mm}^2) \quad (1)$$

Similarly, let  $p$  be the number of ports of the NI,  $c$  be the number of connections per port and  $q$  be the depths of the queues in the NI. Then the cost model for a network interface is:

$$A_{NI}(p, c, q) = 19.6pc + 0.72pcq + 4.8(10^{-3}\text{mm}^2) \quad (2)$$

These two models are used for estimating the network area in Figure 4. The network consists of two routers and eight network interfaces. The area of the two routers is  $0.27\text{mm}^2$  and the area of the eight network interfaces is  $0.73\text{mm}^2$ . Therefore, the total area of the network instance is around  $1\text{mm}^2$  in a  $0.13\mu\text{m}$  technology. When comparing the network area with the area of the DIO and ITC infrastructure then the area increase is 1.5% on the SAF7780.

## VIII. CONCLUSION

Convergence of the CE and automotive domains leads to new requirements for a new platform. The platform needs to be flexible and upgradable such that functions from the CE domain can be implemented. Platform architectures are quickly evolving towards communication centric multiprocessor architectures. We believe that networks-on-silicon can deliver a scalable and flexible communication

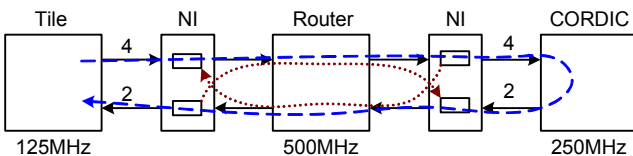


Fig. 5. Round trip latency between EPICS and CORDIC

architecture and serve as the basis for next generation platforms.

This paper evaluates the pros en cons of a new architecture based on a network and compares it with the current architecture. The packet switched network of Æthereal supports point-to-point communication as well as a shared address space. The proposed multiprocessor architecture is optimized for streaming communication. The streaming communication is implemented via a point-to-point communication in the case of tight latency constraints. It is shown for the radio implementation that the latency between a DSP core and an accelerator is within the timing constraints. If cost is important then the shared address space is used for implementing streaming communication to reduce the number of network connections, since this parameter dominates the cost. The cost of the proposed multiprocessor architecture is 1.5% higher than the current architecture due to the network.

#### REFERENCES

- [1] R. van den Berg and H.S. Bhullar, "Next generation philips digital car radios, based on a sea-of-dsp concept", IEEE ISPC GSPx, 2004.
- [2] H.S. Bhullar, R. van den Berg, J. Josten, and F. Zegers, "Serving digital radio and audio processing requirements with sea-of-dsps for automotive applications the philips way", IEEE ISPC GSPx, 2004.
- [3] R. Schiffelers, R. van den Berg, J. van den Braak, H.S. Bhullar, S. de Feber, and M. Klarwater, "Epics7b - a learn and mean concept", IEEE ISPC GSPx, 2003.
- [4] M. Bekooij, O. Moreira, P. Poplavko, B. Mesman, M. Pastrnak, and J. van Meerbergen, "Predictable embedded multiprocessor system design", Proceedings International Workshop on Software and Compilers for Embedded Systems (SCOPEs), September 2004.
- [5] D.E. Culler, J.P. Singh, and A. Gupta, *Parallel computer architecture: a hardware/software approach*, Morgan Kaufmann Publishers, Inc., 1999.
- [6] S.G. Pestana, E. Rijpkema, A. Radulescu, K. Goossens, and O.P. Gangwal, "Cost-performance trade-offs in networks on chip: A simulation based approach", Paris, February 2004, Proceedings Design, Automation and Test in Europe Conference and Exhibition (DATE).