An Application-Specific Network-on-Chip for Control Architectures in RF Transceivers

Siegfried Brandstätter
DMCE GmbH & Co KG
Freistädter Straße 400, 4040 Linz, Austria
Email: SiegfriedX.Brandstaetter@intel.com

Mario Huemer
Klagenfurt University
Universitätsstraße 65-67, 9020 Klagenfurt, Austria
Email: Mario.Huemer@ieee.org

Abstract—This paper focuses on the design of an on-chip communication system for control architectures used in RF (Radio Frequency) transceivers. Continuous developments and enhancements of RF transceivers, especially of smart transceivers supporting multi-mode standards, led to new and complex SoC (System-on-Chip) designs. These designs are defined by a distributed controlling concept using several processing modules which are connected over an advanced communication system. Based on the requirements and restrictions of this communication system an application-specific NoC (Network-on-Chip) is presented and analyzed in this work.

I. INTRODUCTION

Control architectures for RF (Radio Frequency) transceivers are used to control the internal and external, analog and digital modules of the RF IC (Integrated Circuit). Previous control architectures were implemented as centralized systems using a simple bus system or even dedicated control wires to distribute the control information. The implementation of transceivers for new communication standards like UMTS (Universal Mobile Telecommunication System) led to advanced timing requirements for the distribution of the control information and respectively to the implementation of additional synchronization concepts. Detailed information about the evolution of these control architectures can be found in [1].

A new generation of control architectures, described in [1] [2], will implement a distributed controlling concept. This means that the RF transceiver is partitioned into standalone units, e.g. an RX (Receive) unit, an RXPLL (Receive Phase-Looked-Loop) unit, a TX (Transmit) unit, etc., which are capable of controlling themselves after providing essential parameters and triggering the startup. A reduced block diagram of the proposed distributed control architecture can be seen in Figure 1. Even though the units are self-controlling, there are several issues which require communication between them. This work presents an application-specific NoC (Network-on-Chip) which enables the required on-chip communication.

A considerable number of proposals and implementations for NoCs can be found in literature. A survey of them is presented in [3]. Since the requirements and restrictions of NoCs differ widely, the defined NoC architecture had to be tailored towards the requirements and restrictions of the proposed control architecture.

II. APPLICATION-SPECIFIC NOC

A. Requirements and Restrictions

The control flow for a reconfiguration within the proposed distributed control architecture is divided into two phases, the pre-configuration phase and the reconfiguration phase.

During the pre-configuration phase, the CPU (Central Processing Unit) subsystem decodes configuration macros received from the baseband and writes the generated control information (parameters) to the memory system of each unit controller. Since the generation of control information also depends on unit specific parameters, a random read/write access to the memory systems is necessary. Furthermore, the baseband interface, as a second master, is capable of accessing the memory systems, too.

During the reconfiguration phase, the controllers within each unit apply the pre-processed parameters to their corresponding analog and digital modules. This phase, including its startup, is very timing critical allowing only a few clock cycles of jitter. The baseband controls the startup by transmitting TAS (Time Accurate Strobe) macros which are decoded by the interface unit and broadcasted to each unit over the NoC. The maximum timing jitter of ±1/4 chip (±65 ns) is specified by the customers to fulfill the 3GPP (3rd Generation Partnership Project) specification of UMTS [4]. This budget must be arranged over the whole signal path including the interface unit, the NoC and...
the unit controllers. Since synchronization elements within the interface unit and the jitter of unit controllers consume most of this budget, the package latency for transferring the broadcast over the NoC has to be constant.

Besides the TAS distribution, there are several other tasks which require communication between the unit controllers. They have to exchange and gather status information and, in some cases, this information is timing critical. Therefore, the control architecture includes a common time base, well known in literature [5], which is needed to synchronize linked tasks within the transceiver. In order to implement a common time base, the NoC must provide means to keep the standalone units synchronized to the common time base and to enable time stamp based communication.

Table I shows an overview of the above described communication services including their essential restrictions and requirements. It can be seen that the TAS and time distribution services are only used for broadcasting by a single source, i.e. the interface unit and the global timer. The number of sources for the messaging service is giving by the number of unit controllers plus the CPU subsystem, the interface unit and the global timer. As stated above, only the CPU subsystem and the interface unit make use of the random read/write access service during the pre-configuration phase. The priorities of the communication services is given by their priority number whereat a low priority number represents a high priority.

Apart from the timing latency and timing jitter requirements, there are frequency restrictions of the NoC, too. Current RF transceivers are manufactured using the C65LP CMOS (Complementary Metal Oxide Semiconductor) technology, a 65 nm technology especially designed for low power and RF design. The frequency of the digital architecture, including the control architecture and several DSP (Digital Signal Processing) structures, is limited to 104 MHz. The reason for this limitation is (i) to reduce the silicon area effort, (ii) to achieve low power consumption and (iii) to reduce the noise coupled into the analog RF circuits of the transceiver.

**B. Design Decisions and Implementation**

When looking at the stated requirements and restrictions of the NoC, it appears that using a common NoC architecture, which provides QoS (Quality of Service) [6] [7], is likely the best approach. But the following analysis will show that a common network based architecture of the NoC does not lead to an optimal result.

Most NoC architectures, which provide QoS, implement this feature by introducing virtual channels which share the physical channels between the NoC routers. As shown in [6], the implementation of virtual channels multiplies the number of input buffers within the router microarchitecture. In case of the proposed control architecture, which requires four different communication services with different priorities, the number of input buffers would be quadrupled. When looking at the silicon area effort of a single NoC router, the input buffers are the part of the architecture which consumes most of the area. Therefore, using QoS in a NoC architecture has a great impact on the silicon area as well as on the power consumption.

Another drawback of using a QoS based NoC for implementing the required communication services is that the TAS and time distribution services can not be broadcasted with a constant package latency. Of course, the latency of these service can be reduced to a minimum by applying high priorities but additional hardware structures are required to assure a constant latency from sending to receiving the packages.

A general drawback of common NoC architectures, compared to other on-chip communication systems, is a higher latency when transferring data from one node to another over the pipelined routers. This has no impact on random write accesses, but when performing random read accesses, a larger number of wait states has to be expected. As stated in [8], common NoC architectures can overcome this downside by operating at higher clock frequencies. But the clock frequency of the NoC within the proposed control architecture is limited to 104 MHz.

Based on the partitioning of the RF transceiver into standalone units which include one or several processing elements (e.g. a unit controller, a CPU subsystem, etc.), using a tree-based topology for the NoC is likely the best approach. The minimum number of routers can be achieved by implementing a MinRoot architecture [9]. As seen in Figure 2, the proposed architecture uses a router microarchitecture which always provides one interface to the higher level and four interfaces to

### Table I

**Required Communication Services.**

<table>
<thead>
<tr>
<th>Communication Service</th>
<th>Priority</th>
<th>Broadcast</th>
<th>Point-to-Point</th>
<th>Number of Sources</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAS Distribution</td>
<td>0</td>
<td>*</td>
<td></td>
<td>1</td>
<td>Low and constant latency required</td>
</tr>
<tr>
<td>Time Distribution</td>
<td>1</td>
<td>*</td>
<td></td>
<td>1</td>
<td>Constant latency required</td>
</tr>
<tr>
<td>Messaging</td>
<td>2</td>
<td>*</td>
<td>*</td>
<td>(N_{UnitController} + 3)</td>
<td>Time stamp based communication</td>
</tr>
<tr>
<td>Random Read/Write Access</td>
<td>3</td>
<td>*</td>
<td></td>
<td>2</td>
<td>Higher latency jitter allowed, low latency favored</td>
</tr>
</tbody>
</table>

![Fig. 2. MinRoot topology.](image-url)
the lower level of the topology. There is no reason to restrict the number of lower level interfaces to four. Therefore, in a more general approach, the number of routers in a level $l$ can be calculated by

$$N_{\text{Level Router}} = \text{ceil} \left( \frac{N_{\text{Client}}}{N_{\text{LL Interface}}} \right)$$  \hspace{1cm} (1)

where $N_{\text{Client}}$ is the number of clients and $N_{\text{LL Interface}}$ represents the maximum number of lower level interfaces per router. The number of levels is given by

$$N_{\text{Level}} = \text{ceil} \left( \frac{\log_{10} N_{\text{Client}}}{\log_{10} N_{\text{LL Interface}}} \right).$$  \hspace{1cm} (2)

The total number of routers in a MinRoot architecture can be expressed by

$$N_{\text{Router}} = \sum_{i=1}^{N_{\text{Level}}} \text{ceil} \left( \frac{N_{\text{Client}}}{N_{\text{LL Interface}}} \right).$$  \hspace{1cm} (3)

When looking at the stated communication services, it appears that several requirements and restrictions can lead to a reduction of the complexity within the NoC. The following are the key points:

- The TAS and time distribution services only require a broadcast mechanism.
- The TAS and time distribution services are only used by a single source each.
- The random read/write access service is only used by two sources.

Since the TAS and time distribution services are only used by a single source, a special package injection point at the root of the tree-based topology is chosen for these services. We call this technique "root injection". By applying root injection, the path from the leaf to the root of the system can be eliminated. This leads to a lower package latency and savings of silicon area. Because only a broadcast mechanism is required, two dedicated distribution systems are used in parallel to the NoC instead of using QoS for these services. Even though there is an additional logic and routing overhead, which consumes silicon area, the expected area effort is less compared to using QoS. This especially takes effect when increasing the number of unit controllers and therefore the number of routers. Furthermore, using additional distribution systems also decouples the communication services. Therefore, the TAS and time distribution services can provide a low and constant latency for transferring packages.

A further enhancement belongs to the random read/write access service. The BeNoc (Bus-Enhanced Network-on-Chip) architecture [10] makes use of a bus system which operates in parallel to a common NoC. The bus system concurrently functions as a low latency broadcast/multicast capable media which is primarily used for short latency signaling and multicast services. The analysis will show that a bus enhanced NoC is the optimal solution for the proposed control architecture, too. The bus system not only provides low latency random read accesses but also simplifies the design exploration by splitting the communication services of the pre-configuration and the reconfiguration phase. Furthermore, the bus system shows a very simple architecture since there are only two master nodes, the CPU subsystem and the interface unit.

Figure 3 shows an example block diagram of the Bus enhanced MinRoot NoC with Root Injection connecting 13 unit controllers, the CPU subsystem, the interface unit and the global timer.

The bus system of the application-specific NoC is compliant with WISHBONE Revision B4 [11]. The implementation includes two layers which support pipelined transfer of data and addresses. The bus system itself does not imply any register stages, therefore, data and addresses are transferred immediately. By using these features, simultaneous slave accesses from two master nodes are possible. In the event of two master nodes accessing the same slave, a priority arbitration scheme will be applied.

As stated before, the router microarchitecture of the tree-based NoC, which is depicted in Figure 4, provides dedicated communication channels for the TAS distribution service, the time distribution service and the messaging service. Since the TAS and time distribution services are only used for broadcasting, the channels do not require any routing hardware. Only a
The following analysis is based on the setup shown in figure 3 using an operation frequency of 104 MHz. The router interconnection of the messaging interface and the TAS interface, and the data bit width of the bus system is set to 16 Bit. The interconnection of the time interface is only one bit wide. The package size of the messaging service, the TAS distribution service and time distribution service is 6 flit, 2 flit and 20 flit, accordingly.

The throughput capability of the TAS and time distribution services is constant and can be calculated by

\[ \text{Throughput} = f \cdot W_{\text{Interface}} \]  

(4)

where \( f \) is the operation frequency and \( W_{\text{Interface}} \) represents the bit width of the router interconnection. The calculated values are approx. 1.6 GBit/s for the TAS distribution service and 104 MBit/s for the time distribution service. The constant package latency of the services can be expressed by

\[ \text{Package Latency} = P_{\text{Size}} + N_{\text{Hop}} \]  

(5)

where \( P_{\text{Size}} \) specifies the number of flits per package and \( N_{\text{Hop}} \) represents the hop count. By applying root injection to these services, the hop count can be reduced to 2 resulting in a package latency of 4 clock cycles and 22 clock cycles for the TAS distribution service and the time distribution service, respectively.

The package latency as well as the throughput capability of the messaging service heavily depends on the hop count and therefore on the Manhattan distance between two communicating clients. In the given setup, 1 or 3 hops are possible. According to (5), this results in a package latency of 7 clock cycles or 9 clock cycles. Of course, this is only true if the packages are routed without any wait state caused by collisions. The minimum throughput capability of this service is given if all packages have to be routed over the root router of the NoC. In this case, (4) results in a throughput of approx. 1.6 GBit/s. The maximum throughput is achieved if the communication is limited to the subnet. This means that no packages are routed over the higher level messaging interface of the routers. Therefore, the throughput capability of a single router can be multiplied by the number of subnets. For the given setup, this results in a maximum throughput of approx. 6.7 GBit/s.

Since the bus system of the application-specific NoC forwards data and addresses without any wait states, the access latency to a synchronous memory is very low. In case of a registered interface at the master node, the write and read access latencies are 1 clock cycle and 2 clock cycles, accordingly. The throughput capability of a single master node, which is given by (4), is approx. 1.6 GBit/s. In theory, the bus system is able to concurrently transfer accesses from two master nodes resulting in a throughput of approx. 3.3 GBit/s. Of course, this is only true if there are no access collisions.

III. BENCHMARKING

A. Overview

As stated in Section II-B, the default solution for providing the required communication services would be implementing a QoS based NoC. Therefore, a reference implementation, which features this QoS approach, is used to benchmark the previously described application-specific NoC.

Since the reference NoC makes use of the same MinRoot architecture like the application-specific NoC, the number of routers in the system is the same. The basic principle of the QoS based NoC router microarchitecture, which is depicted in Figure 5, is equal to the part of application-specific router microarchitecture which provides the messaging service. The main difference is the introduction of virtual channels which are used to support the different priorities of the communication services. As stated before, this leads to a multiplication of the FIFO for the lower level as well as for the higher level messaging interface. Since four different communication services with different priorities are required in the system,
The number of FIFOs is quadrupled. The priority round robin scheduler in the router analyzes the priority number, which is located in the first flit of each package, and forwards the package according to its priority to the respective FIFO. Since the scheduler provides a dedicated layer for each lower level messaging interface, several packages with different priorities can be forwarded simultaneously. The higher and lower level decoders process one package at a time, packages with lower priority number are routed first. Similar to the priority round robin scheduler, the priority multiplexer analyzes the first flit of each package received on the higher level messaging interface and distributes the package according to its priority to the respective FIFO. Unlike the dispatcher in the application-specific NoC router microarchitecture, the dispatcher in the QoS based NoC router microarchitecture forwards packages with lower priority numbers first. In case of two packages having the same priority, the package from the higher level messaging interface is forwarded first.

The specifications for the part of the application-specific NoC, which provides the messaging service, and for the QoS based NoC are equal. The router interconnections are 16 Bit wide, and the FIFO sizes are set to 2 flit and 6 flit for the lower level FIFOs and the higher level FIFOs, respectively. For the application-specific NoC, the interconnection of the TAS interface is 16 Bit wide and a serial wire (1 Bit) is used for the time interface. Furthermore, the bus system features an address port width of 32 Bit and a data port width of 16 Bit. The operating frequency is set to 104 MHz which is also the target frequency for synthesizing both systems. The C65LP CMOS technology library is used for the synthesis.

The simulation environment injects packages with a constant size of 6 flit (96 Bit) to the messaging interface of both systems. The size of the packages injected to the TAS interface and the time interface of the application-specific NoC is 2 flit (32 Bit) and 20 flit (20 Bit), accordingly. Each package, injected to the QoS based NoC, encapsulates a priority number in the first flit which assigns the package to one of the four different communication services. The estimation in Table II shows the distribution of communication services for the targeted application. This distribution is used by the simulation environment for generating the priorities of the injected packages.

The probability for the injection of a broadcast package to the messaging interface of both systems is set to 5%. The destination addresses of packages injected to the messaging interface of both systems, and destination addresses of random read/write accesses injected to the bus system of the application-specific NoC are uniformly distributed. This means that the probability of addressing a client is the same for all of them.

### B. Throughput

Figure 6 demonstrates the system throughput capabilities of the application-specific NoC. Thereby, the horizontal axis depicts the bit rate, which the simulation environment tries to inject, and the vertical axis shows the bit rate which is accepted by the NoC. It can be seen that the bit rate of the time distribution service has a very low saturation point of only approx. 100 MBit/s. This is due to the fact that the time distribution service is distributed over a single serial wire. The TAS distribution service, the random read/write access service and the messaging service provide a bit rate of approx. 1.7 GBit/s, 1.8 GBit/s and 2.1 GBit/s, respectively. The overall system throughput, which equals the sum of throughputs...
The system throughput capabilities of the QoS based NoC are depicted in Figure 7. The overall system throughput has a saturation point of approx. 2.1 GBit/s which is quite similar to the throughput of the messaging service provided by the application-specific NoC. This is due to the fact that the QoS based NoC has a router microarchitecture which is similar to the part of application-specific NoC router microarchitecture which provides the messaging service. Since the application-specific NoC implements dedicated physical channels and structures for each communication service, the overall system throughput is approx. 3.4 GBit/s higher than the overall system throughput of the QoS based NoC.

Because of the fact that the physical channels and structures of the QoS based NoC are shared among all communication services, the distribution used by the simulation environment for generating the package priorities can be seen in the distribution of the throughput, too. 50% of the overall system throughput belongs to the messaging service, 40% to the random read/write access service, 8% to the TAS distribution service and 2% to the time distribution service.

As stated above, these throughputs are provided by the systems if the destination addresses for the packages and the random read/write accesses are uniformly distributed. But tree-based topologies of NoCs provide an interesting feature when using a normal distribution for the destination address generation with a mean value which is centered at the subnet address of the particular client. This means that more packages are transferred within the subnet and therefore a higher system throughput can be achieved by the NoC. Figure 8 shows the relative frequency of destination addresses, depending on different standard deviations, for packages injected at client zero. Similar, the mean value of the normal distribution in other subnets is centered at their subnet address. Figure 9 demonstrates the throughput provided by the messaging service of the application-specific NoC in case of using normally distributed destination addresses. Compared to the system throughput of 2.1 GBit/s which is achieved when using uniformly distributed destination addresses, the system throughput can be approx. 2.6 times higher when using a normal distribution with a standard deviation of one.

C. Latency

Another important key point of an on-chip communication system is the latency of a package transmission or a bus access. The average package and access latency of the application-specific NoC, depending on the accepted bit rate, is depicted in Figure 10. Two important requirements, which have been stated in Section II-A, are met by the application-specific NoC.

- The TAS and time distribution services generate a constant package latency of 4 clock cycles and 22 clock cycles, accordingly. This is due to the fact that the NoC provides dedicated physical channels and structures for the root injection of these services. Hence, no additional hardware structures are required to enable a constant latency distribution of the TAS and the time.
Since the random read/write access service is provided by the bus enhancement of the NoC, the average access latency of this service is very low. Assuming that the slave provides a synchronous interface and the master interface is registered, the average access latency is only between 1 clock cycle and 2 clock cycles.

The average package latency of the messaging service shows a typical NoC behavior. With increasing bit rate, the average package latency increases until it reaches its saturation point at approx. 5.5 GBit/s. This point equals the maximum throughput capabilities of the application-specific NoC.

Unlike the application-specific NoC, the average package latency of all services provided by the QoS based NoC depends on the accepted bit rate. This behavior, which is caused by sharing the physical channels and structures among all services, can be seen in Figure 11. It has to be kept in mind that additional hardware structures are needed to provide a constant latency for the distribution of the TAS and the time. Since only a small percentage of all injected packages belong to the TAS and time distribution services, and their priorities are the highest, these services provide a low average package latency of approx. 13 clock cycles. The messaging service, occupying most of the throughput capabilities of the NoC, reaches an average package latency of approx. 15 clock cycles. Since the random read/write access service has the lowest priority, its average package latency saturates at the maximum throughput capability of 2.1 GBit/s.

D. Silicon Area

The silicon area effort of an on-chip communication system is a further important key point, especially for low power RF designs. Table III demonstrates the figures of the application-specific NoC and the QoS based NoC measured in gate equivalents. It appears that the effort for implementing the TAS and time distribution services of the application-specific NoC is very low. This is due to the fact that these services are injected at the root, and dedicated physical channels and structures are used to distribute the packages to the clients of the NoC. As predicted in Section II-B, the gate count of the bus system, which enables the random read/write access service of the application-specific NoC, has a very low number of approx. 4 kGate. This number is achieved because only two master nodes are accessing the bus system. The overall gate count of the application-specific NoC is approx. 13 kGate.

The QoS based NoC, on the other hand, has a gate count of approx. 43 kGate. This is more than three times the gate count of the application-specific NoC. As stated before, the main reason for this increase are the multiple FIFOs at the lower level as well as at the higher level messaging interface within the router microarchitecture. These FIFOs are necessary to enable the virtual channels which are used to support the different priorities of the communication services.

Furthermore, the gate count of the QoS based NoC does not include the additional hardware structures which are needed to provide a constant package latency of the TAS and time distribution services.

IV. CONCLUSION

In this paper we presented an application-specific NoC for a new generation of control architectures used in RF transceivers. Based on these architectures, this work includes an elaboration of required communication services as well as the definition of requirements and restrictions for the NoC. The resulting design targets the optimal solution for
each communication service in terms of wait states, latency and throughput. But also low power consumption and low silicon area effort are considered in the design decisions. Furthermore, a benchmarking of the application-specific NoC and a reference NoC showed the advantages of the application-specific NoC design over traditional approaches.

REFERENCES


