Message scheduling to reduce AFDX jitter in a mixed NoC/AFDX architecture

Jérôme Ermont, Sandrine Mouysset, Jean-Luc Scharbarg and Christian Fraboul

Université de Toulouse - IRIT - INPT/ENSEEIHT

October 12, 2018
**Context: Avionics architecture of modern planes**

- **Avionics computers:**
  - Mono-core processors: execute avionics functions following IMA (Integrated Modular Avionics)
  - End Systems: interface between CPU and AFDX

- **AFDX network:**
  - Interconnection of several avionics computers
  - VL: unidirectional flow between one source ES to one or more destination ES
  - BAG: minimum interval time between 2 frames of a VL
Transmission of VLs by an ES

- VLs from different partitions share the same ES
- Scheduler between VLs into the ES
- Introduces a jitter: delay between the beginning of the BAG and the effective transmission of the frame
- AFDX constraint: jitter $< 500 \mu s$
To replace mono-core processing unit by many-cores
Different applications can be executed in parallel
2 different communications:
  - Intra-NoC communication
  - Inter-NoC communication
Problem Statement

The jitter depends on the WCTT of flows from other applications. WCTT depends on the blocking mechanism of the NoC.

Problem

How to reduce the jitter induced by the transmission on the NoC?
Problem Statement

The jitter depends on the WCTT of flows from other applications. WCTT depends on the blocking mechanism of the NoC.

Problem
How to reduce the jitter induced by the transmission on the NoC?
Problem Statement

The jitter depends on the WCTT of flows from other applications and the WCTT depends on the blocking mechanism of the NoC. The problem is how to reduce the jitter induced by the transmission on the NoC.

J. Ermont, S. Mouysset, J.-L. Scharbarg, C. Fraboul

RTNS 2018 - 10-12 october 2018
The jitter depends on the WCTT of flows from other applications.

WCTT depends on the blocking mechanism of the NoC.

Problem

How to reduce the jitter induced by the transmission on the NoC?
Problem Statement

The jitter depends on the WCTT of flows from other applications. WCTT depends on the blocking mechanism of the NoC.

Problem

How to reduce the jitter induced by the transmission on the NoC?
Problem Statement

The jitter depends on the WCTT of flows from other applications. WCTT depends on the blocking mechanism of the NoC.

Problem

How to reduce the jitter induced by the transmission on the NoC?
The jitter depends on the WCTT of flows from other applications
- WCTT depends on the blocking mechanism of the NoC
The jitter depends on the WCTT of flows from other applications
WCTT depends on the blocking mechanism of the NoC

Problem
How to reduce the jitter induced by the transmission on the NoC?
A first solution: minimizing the contention for other applications

- A new application mapping: Extended Map\textsubscript{IO} [1]

Minimizing contentions reduces the maximum jitter

But not for all configurations

[1] Towards a mixed NoC/AFDX architecture for avionics applications, Laure Abdallah and al., WFCS 2017
Our proposition

- One node dedicated to schedule the VLs on the ES
- Use of a TDMA table
Our proposition

- One node dedicated to schedule the VLs on the ES
- Use of a TDMA table
Our proposition

- One node dedicated to schedule the VLs on the ES
- Use of a TDMA table

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>APP4</td>
<td>APP1</td>
<td>APP3</td>
<td>APP2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>APP1</td>
<td>APP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>APP2</td>
<td>APP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>APP1</td>
<td></td>
<td>APP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>APP2</td>
<td>APP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>APP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>APP2</td>
<td>APP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>APP1</td>
<td>APP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>APP2</td>
<td>APP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>APP1</td>
<td>APP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>APP2</td>
<td>APP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>APP1</td>
<td>APP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>APP2</td>
<td>APP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>APP1</td>
<td>APP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>APP1</td>
<td>APP1</td>
<td>APP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WCTT of VL1

Bag of VL4 (from App4): 128 ms

VL4 is ready when line 6 of the table is executed

VL4 will wait line 0
Our proposition

- One node dedicated to schedule the VLs on the ES
- Use of a TDMA table

Bag of VL4 (from App4): 128 ms

VL4 is ready when line 6 of the table is executed
Our proposition

- One node dedicated to schedule the VLs on the ES
- Use of a TDMA table

![Diagram of TDMA table]

- Bag of VL4 (from App4): 128 ms
- VL4 is ready when line 6 of the table is executed
- VL4 will wait line 0
Our proposition

- One node dedicated to schedule the VLs on the ES
- Use of a TDMA table

![Diagram of TDMA table]

- Bag of VL4 (from App4): 128 ms
- VL4 is ready when line 6 of the table is executed
- VL4 will wait line 0

How to reduce this waiting delay?

Our solution

To give more slots for the VLs → Oversampling of slots
Our proposition

- One node dedicated to schedule the VLs on the ES
- Use of a TDMA table

- Bag of VL4 (from App4): 128 ms
- VL4 is ready when line 6 of the table is executed
- VL4 will wait line 0

How to reduce this waiting delay?

Our solution

To give more slots for the VLs → Oversampling of slots
How to map the slots to the VLs in the table?

**Constraint**

The VLs should respect their BAGs

- VLs with $\text{BAG} = 1\text{ms}$ allocated to all lines
- Allocation by considering the minimum BAG value ($>1\text{ms}$)
How to map the slots to the VLs in the table?

**Constraint**

The VLs should respect their BAGs

- VLs with BAG = 1ms allocated to all lines
- Allocation by considering the minimum BAG value (> 1ms)
How to map the slots to the VLs in the table?

Constraint

The VLs should respect their BAGs

- VLs with \( \text{BAG} = 1\text{ms} \) allocated to all lines
- Allocation by considering the minimum BAG value (\( > 1\text{ms} \))

<table>
<thead>
<tr>
<th>APP1</th>
<th>APP2</th>
<th>APP3</th>
<th>APP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
</tbody>
</table>
How to map the slots to the VLs in the table?

**Constraint**

The VLs should respect their BAGs

- VLs with $\text{BAG} = 1\text{ms}$ allocated to all lines
- Allocation by considering the minimum BAG value ($>1\text{ms}$)
How to map the slots to the VLs in the table?

**Constraint**

The VLs should respect their BAGs

- VLs with BAG = 1ms allocated to all lines
- Allocation by considering the minimum BAG value (> 1ms)
Formulation by a Bin Packing Problem

Objective

To allocate VL transmissions into a minimum number of lines

- Number of lines in which VLs are allocated
  \[ N = \min_{j=1\ldots m, BAG_j \neq 1} \ BAG_j \]

- Objective function

\[
\begin{align*}
\min & \quad \sum_{i=1}^{N} y_i \\
\text{s.t} & \quad \sum_{j=1}^{m} \omega_j x_{ij} \leq C y_i, \forall i = 1, \ldots, N \\
& \quad \sum_{i=1}^{N} x_{ij} = 1, \forall j = 1, \ldots, m \\
& \quad y_i \in \{0, 1\}, \forall i = 1, \ldots, N \\
& \quad x_{ij} \in \{0, 1\}, \forall i = 1, \ldots, N, \forall j = 1, \ldots, m.
\end{align*}
\]
Evaluation case study

- A 10x10 Tilera-like NoC
- 2 types of applications:
  - FADEC: engine control
    - critical application
    - little amount of exchanged data: 1500 bytes
    - full transmission between task
  - HM: HouseKeeping
    - non-critical application
    - lots of data exchanged: > 130 Koctets
    - data are stored in the memory
9 (critical or non-critical) considered applications: FADEC\textsubscript{7} (4), FADEC\textsubscript{11} (8), FADEC\textsubscript{13} (16), HM\textsubscript{7} (4), HM\textsubscript{9} (2), HM\textsubscript{10} (16), HM\textsubscript{11} (32), HM\textsubscript{12} (16), HM\textsubscript{16} (32)

2 system configurations:
- 8 applications: HM\textsubscript{7} is removed
- 9 applications

3 mapping strategies:
- SHiC [1]: mapping by considering the core-to-core communications
- Map\textsubscript{IO} [2]: mapping by considering core-to-IO communications
- exMap\textsubscript{IO} [3]: mapping by considering both core-to-IO and IO-to-core communications

[1] Smart hill climbing for agile dynamic mapping in many-core systems, Mohammad Fattah and al
[3] Towards a mixed NoC/AFDX architecture for avionics applications, Laure Abdallah and al., WFCS 2017
### VLs transmissions packed into the table

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | FADEC7 | FADEC11 | FADEC13 |
| 1 | HM9 | HM10 | HM11 | HM12 | HM13 |

### SHiC

### MapIO with 8 applications

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | FADEC7 | FADEC11 | FADEC13 |
| 1 | HM9 | HM10 | HM11 | HM12 | HM13 |

### Ex_MapIO with 8 applications

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | FADEC7 | FADEC11 | FADEC13 |
| 1 | HM9 | HM10 | HM11 | HM12 | HM13 |

### MapIO with 9 applications

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | FADEC7 | FADEC13 | HM9 | HM10 | HM11 |
| 1 | FADEC11 | HM7 | HM12 | HM13 |

### Ex_MapIO with 9 applications

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | FADEC7 | FADEC13 | HM9 | HM10 | HM11 |
| 1 | FADEC11 | HM7 | HM12 | HM13 |
Results

- TDMA table guarantees the transmission every BAG
- Jitter constraint is respected when using a dedicated node
To replace mono-core processors by NoC based many-cores architecture

Sharing the same output port could lead to an execution for which the jitter constraint is exceeded

Mapping strategy Extended Map\textsubscript{IO} minimizes the jitter by reducing the contention
  - But jitter constraint can be exceeded

Our proposition: one dedicated node schedules the outgoing flows using a TDMA table
  - The jitter only depends on the contentions for the outgoing flow
  - The jitter is then significantly reduced

Construction of a scheduling table
  - Guarantee of the BAG constraint
  - Over allocation of slots in order to reduce waiting delays
Further works

- SHiC mapping example

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | FADEC7 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1 |   | HM9 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

- What happens if HM$_{13}$ needs 10 slots?
- Different possible solutions
  - Reduce more the contentions on outgoing flows
  - Relax the constraint of the minimum number of lines for larger BAG value → Variable capacity size bin packing or cutting stock problem

- Global transmission delay from one manycore to another via AFDX
- Implementation of the solution in a real manycore system such as Tilera or Kalray