Predictable implementation of avionics applications on MPPA

Journée thématique Verimag : Many-core Kalray MPPA, implementation and verification

April 29th 2019
Outline

- Conclusion – future works
Temporal Isolation of Hard Real-Time Applications on Many-core Processors

Airbus Cifre PhD (2014 – 2017): Quentin Perret

Supervisors: Pascal Maurère, Eric Noulard, Claire Pagetti, Pascal Sainrat, Benoît Triquet


Quentin Perret, Pascal Maurère, Éric Noulard, Claire Pagetti, Pascal Sainrat, Benoît Triquet: Mapping hard real-time applications on many-core processors. RTNS 2016: 235-244

Example Kalray: accesses to the external memory
Example Kalray: accesses to the external memory
Example Kalray: accesses to the external memory
Example Kalray: accesses to the external memory

I/O cluster

Compute cluster A

Local SRAM

Compute cluster B

Local SRAM

Destination

DDR Controller

DMA 1

Core 1

Core 2

DMA 2

Bank 1

Bank 2

Bank 3

Bank 4

Bank 5

Bank 6

Bank 7

Bank 8

DDRx-SDRAM

Data 1

Bank 1

Bank 2

Bank 3

Bank 4
Example Kalray: accesses to the external memory
Example Kalray: accesses to the external memory
Example Kalray: accesses to the external memory
Execution models

- **Definition**: set of rules to be followed by the designer in order to avoid or at least reduce the non predictable behaviours

- **Origin**: performance prediction models:
  Bulk synchronous parallel model

- **Principle**: enforce and segregate the accesses to the shared resources to master them

Works realised within several industrial projects (Airbus, Thales), a PhD with Airbus, internal projects (SCC) and regional project (2 post-doc TOAST/TORRENTS)
Workflow

Partitions → Budget → Describe → Allocate → Compile → Executable

Resource budget → Architecture → Exec. model → HW config.

Budgeting

Mastering execution
Mastering execution

Rule 1: In a compute cluster, each local SRAM bank and each core can be used by at most one partition

Rule 2: NoC transactions are performed during strictly periodic slots defined off-line and do not overlap (TDMA)
Mastering execution

Rule 3: All the buffers sent during the strictly periodic NoC slots must be defined off-line.

Rule 4: A bank of the DDR-SDRAM can be used by several partitions if and only if they never access it simultaneously.
**Budgeting: Architecture description**

**Partition Node (or PN): processing resources**
- Number of PEs
- Number of SRAM banks

**I/O Node (or ION): I/O transactions**
- Core on I/O cluster

**Partition Communication (or PC):**
- Communications between PN / ION
- Source and destination PN / ION
- Period, duration and offset of strictly periodic slot
Budgeting: Application’s budget

\[ \tau_i = \langle S_i, P_i, T_i \rangle \]

\[ \tau_i^j = \langle C_i^j, M_i^j, I_i^j, O_i^j \rangle \]

\[ \delta_k = \langle m_k, \text{prod, cons} \rangle \]
OPL IBM constraint programming modelling with Conditional Time-Intervals

- Very efficient for non-preemptive task

A task $t_1$ is associated with:
- an interval $i_1$ of length $t.wcet$
- $n$ conditional intervals $i_{1,1}, i_{1,2}, \ldots, i_{1,n}$

Only 1 interval present in the solution: $\text{alternative}(i_1, \text{all}(x \in 1, \ldots, n)i_{1,x})$

Solver produces $\text{presenceOf}(i_1, x)$ and $\text{startOf}(i_1)$. 
Conditional time-intervals

To express that two tasks must not overlap on a core, notion of cumulative function

constraint for all core $k$, $\sum \text{pulse}(i_{x,k}, 1) \leq 1$

replace

for all pair of tasks, $c_1 = c_2 \Rightarrow (s_1 + t_1.wcet \leq s_2) \lor (s_2 + t_2.wcet \leq s_1)$
Experimental results

1 PE per PN
- Limit interference at SRAM level
- Focus on NoC-level parallelization

Symmetric PCs
- Each PN can send data to all PNs
- All PCs with same period and duration
- PC duration close to Hypervisor’s WCET

Avionic application has been successfully mapped
Outline

  - The MPPA NoC
  - Experiments
- Conclusion – future works
Bounding MPPA network-on-chip traversal time: a network calculus approach

Contributors: Marc Boyer, Amaury Graillat, Benoît Dupont de Dinechin, Jörn Migge

- 16 switches (1 per compute cluster)
- 16 switches (dedicated to I/O and memory)
- connected by a network-on-chip (NoC)

Computing Routes and Delay Bounds for the Network-on-Chip of the Kalray MPPA2 Processor Marc Boyer, Benoît Dupont de Dinechin, Amaury Graillat, Lionel Havet ERTS 2018
DMA Engine

- 8 queues
- a traffic limiter per queue
  - token-bucket
  - send only full packet
- Round-Robin arbiter
Switch

- 5 input ports: East, West, North, South, Local
- 5 output ports
- each output port has one queue per input port
- round-robin arbiter
A real-time network

- Traffic limiter + “enough” memory + guaranteed throughput
  ➔ bounded latency
- Remarks: MPPA NoC apply “wormhole switching”
  - when a queue is full, previous switch must stop emission
  - ... then its own queue fills up
  - known as “back pressure”
    - complex analysis
    - can lead to deadlock
  ➔ limit traffic to never fill queues

- Challenge: computing a latency bound
- Method: network calculus

Outline

  - The MPPA NoC
  - Experiments
- Conclusion – future works
Network calculus: a toolbox

Several approaches used:
- Explicit Linear: closes expression formula
- LP: use of MILP solver
- Separated Flow Analysis (SFA): end-to-end network calculus
- Total Flow Analysis (TFA): per switch network calculus
First case study
Second case study

- 4 flows out of each switch local port ⇒ 128 flows
- Random destination (unicast)
- figure: average latency bound (grouped per flow length)
Conclusion on the NoC

- MPPA NoC supports real-time guarantees
- Network calculus permits to compute latency bounds
- Delay bounds: $\approx 1\mu s$ on large case study
Outline

- Conclusion – future works
Porting of applications on the MPPA

Contributors: Youcef Bouchebaba, Jean-Loup Farges
1 or 2 clusters involved
Outline

- Conclusion – future works
Future work

Coolidge (CAVIAR project 2021 – 2022)

- WCTT with network calculus
- Interference calculus (cf PHYLOG / MCP CRI / CAST 32A)
- Safety analysis