Using a Hardware Coprocessor for Message Scheduling in Fieldbus-based Distributed Systems

José A. Fonseca
Ernesto V. Martins

University of Aveiro

Paulo Neves

Escola Superior de Tecnologia
Polytechnic Institute of Castelo Branco

Presentation organization

- Framework
- Objectives
- Arbiter architecture in industrial networks
- The planning scheduler
- PSCoP – the planning scheduler coprocessor
- Implementation details and PSCoP performance
- Conclusions and future work

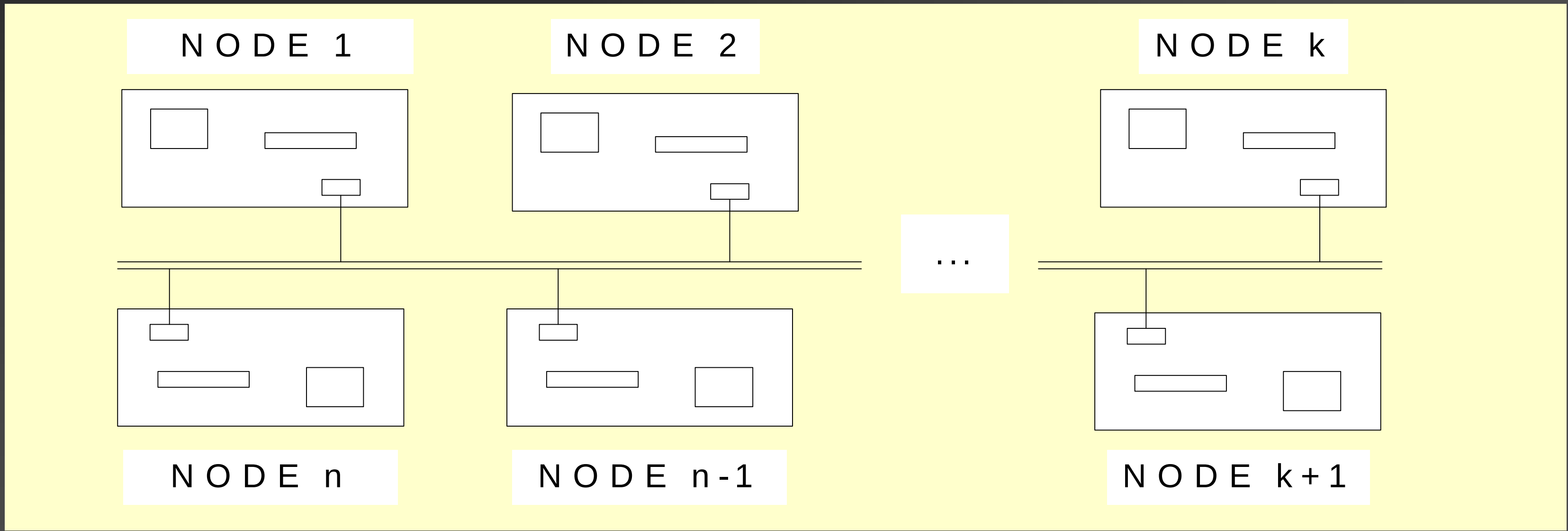
Framework

- Distributed Embedded Systems (Industrial, Automotive)
- Industrial Networks (fieldbuses)
  - CAN – Controller Area Network
- Real-time constraints
- FTT-CAN protocol (flexible time-triggered)
  - Implementation of the planning scheduler over CAN

Objectives

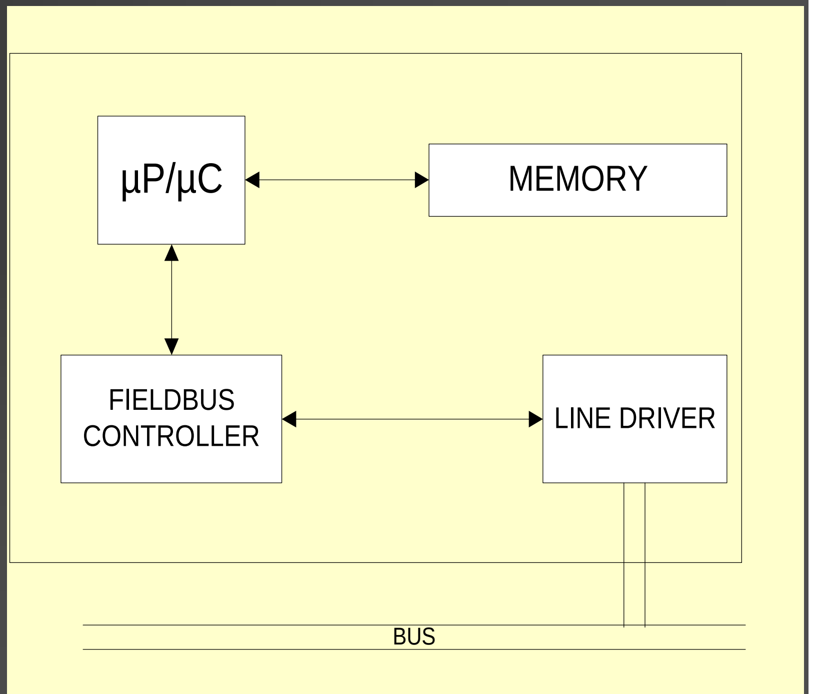
- To improve the response time of systems based in the planning scheduler paradigm
- To allow the use of the typical μC/μP for the target applications (which are low-processing power, e.g., in cars)
- Minimal hardware changes in the system nodes

Typical architecture of a fieldbus-based distributed system

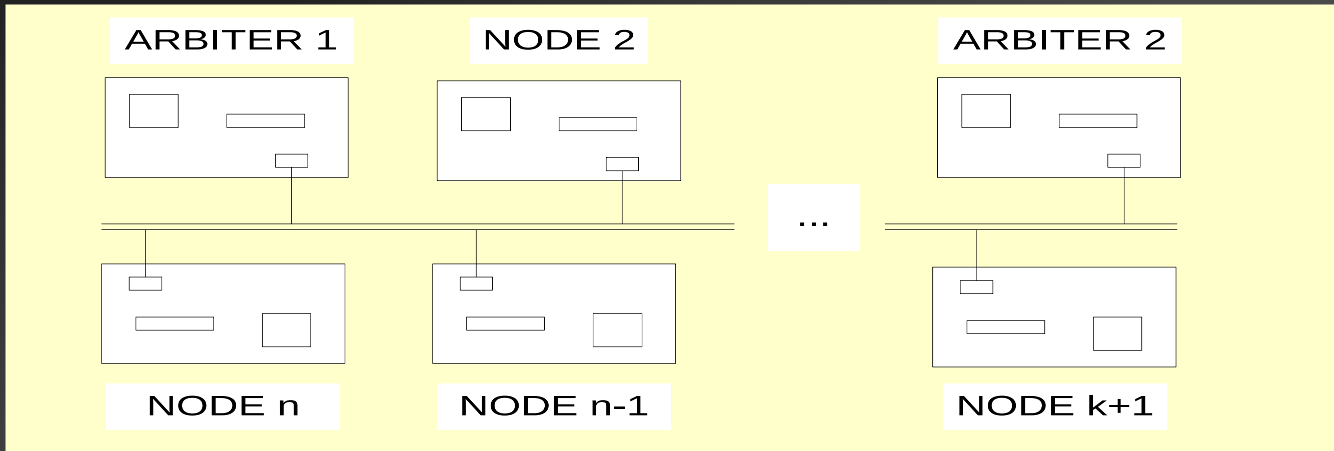


Typical node of a fieldbus system

- μC/μP
- Program/data memory
- Fieldbus controller
- Transceiver



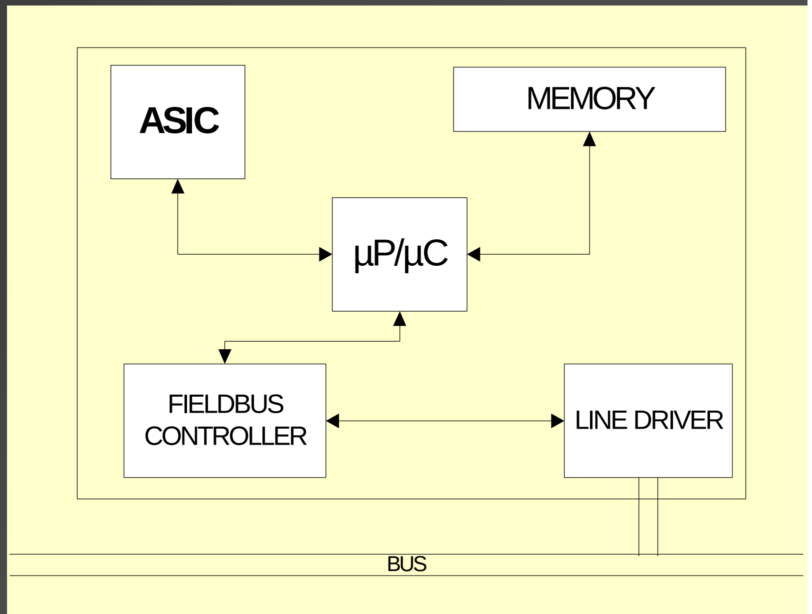
Centralised Medium Access Control (MAC)



- Centralised MAC – arbiter node
  - FIP, FTT-CAN
  - Dispatching table – macro cycle

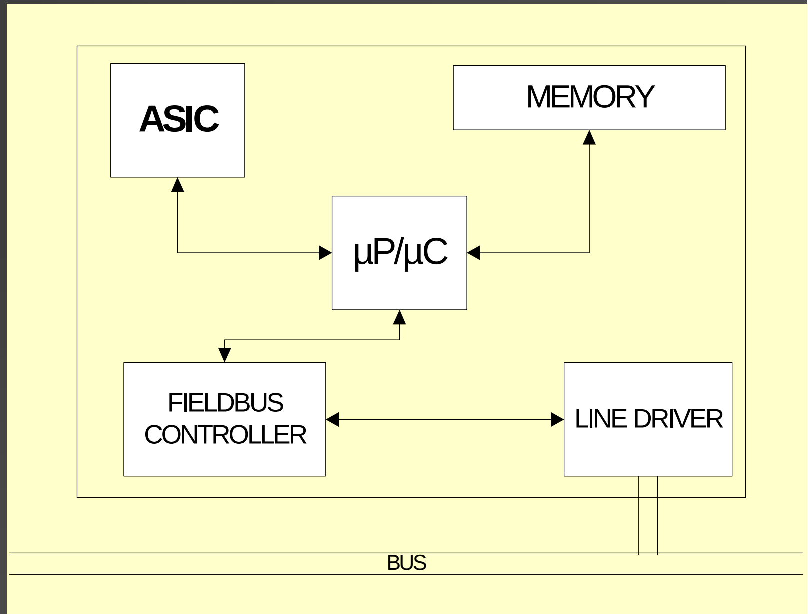
Arbiter node architecture

- Scheduling function
  - ASIC or another µC/µP?
- ASIC can be faster
- µC/µP may present the need for extra support chips

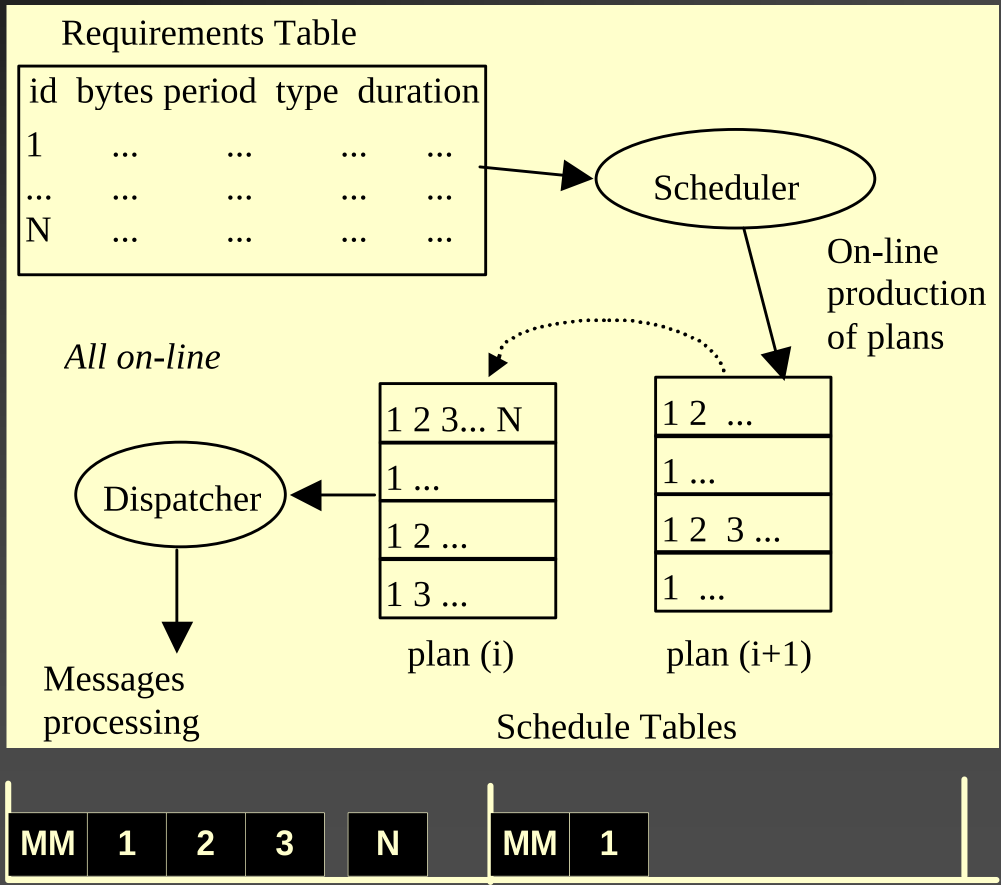


Arbiter node architecture

- FPGA-based coprocessor
  - Fast programming
  - Software programming tools
  - Availability in the research group



The planning scheduler

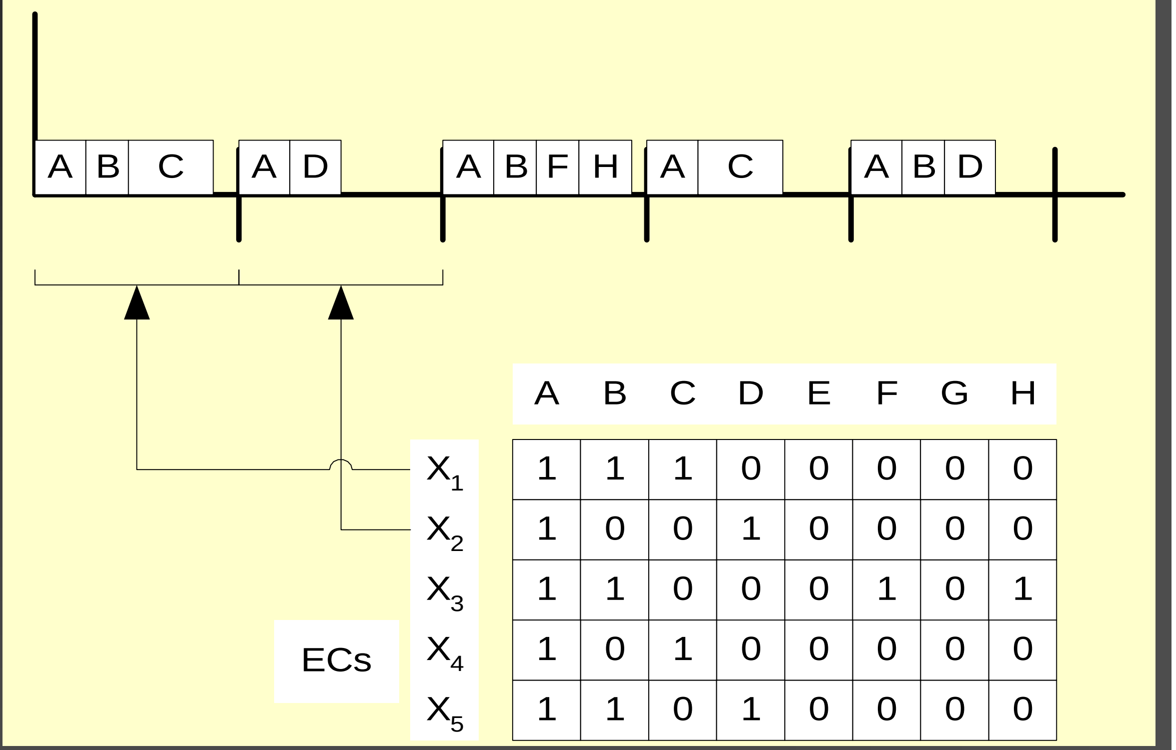


The Planning Scheduler CoProcessor (PSCoP)

- Direct mapping on the node µC/µP for configuration
  - Handles 8 messages in current prototype
  - 3 8-bit registers for each message: period, initial phase and transmission time
- Static parameters – static message priorities
- Results passed to the node µC/µP

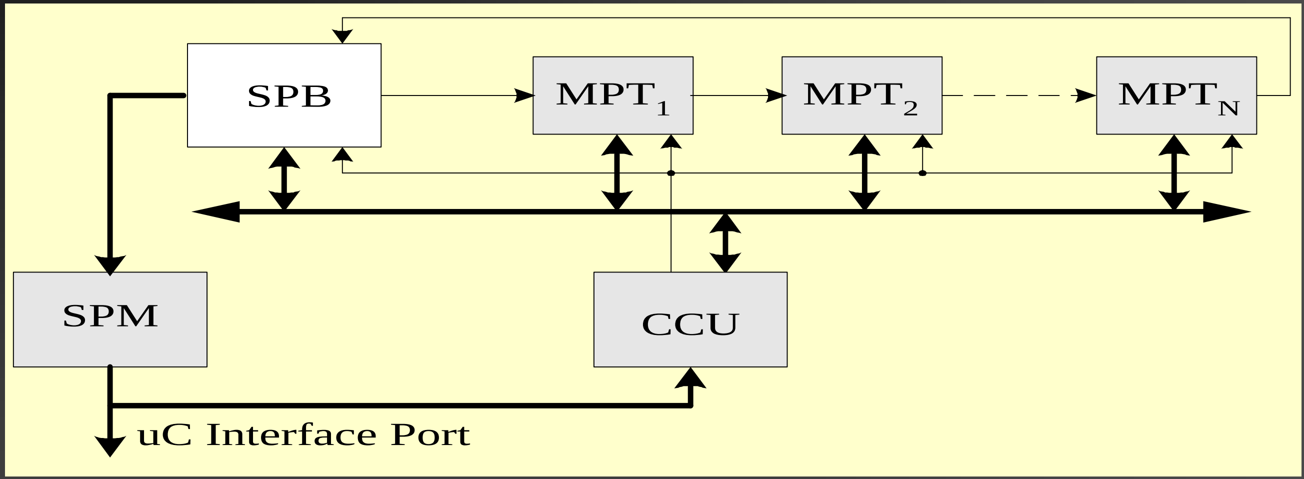
PSCoP output

- 8 bits per EC
  (EC - Elementary cycle)
- 16 ECs per plan

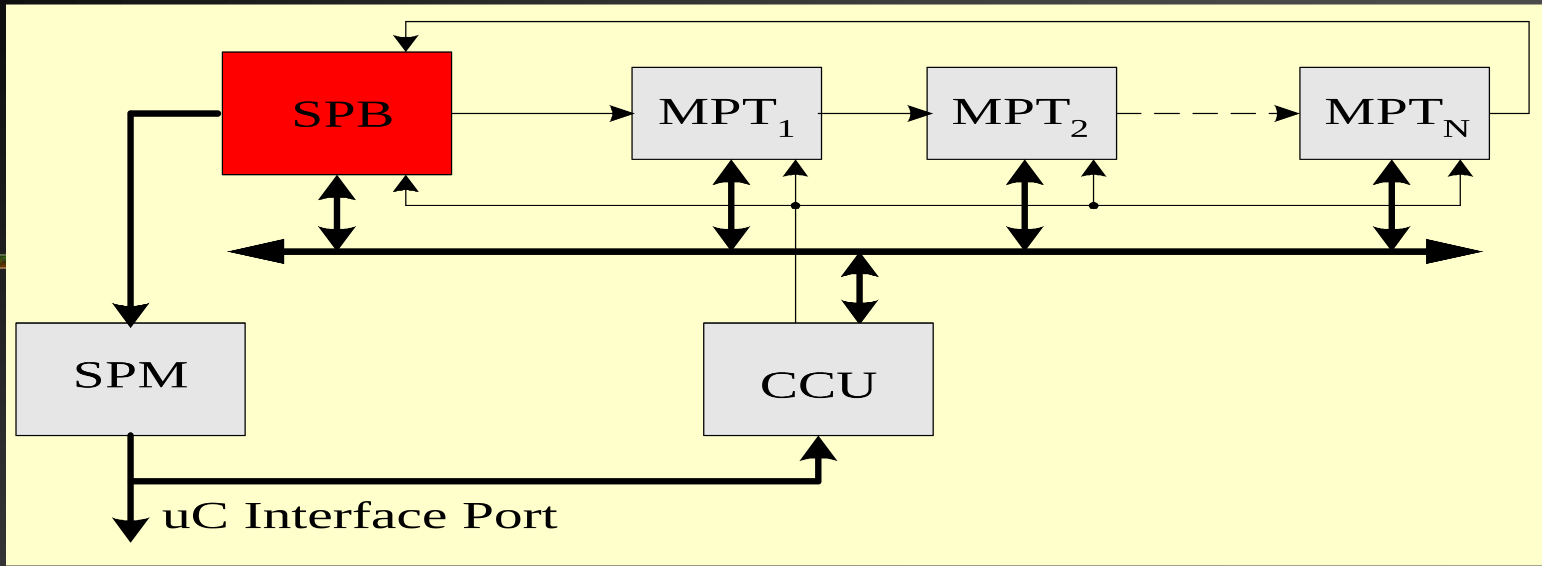


	A	B	C	D	E	F	G	H
X_1	1	1	1	0	0	0	0	0
X_2	1	0	0	1	0	0	0	0
X_3	1	1	0	0	0	1	0	1
X_4	1	0	1	0	0	0	0	0
X_5	1	1	0	1	0	0	0	0

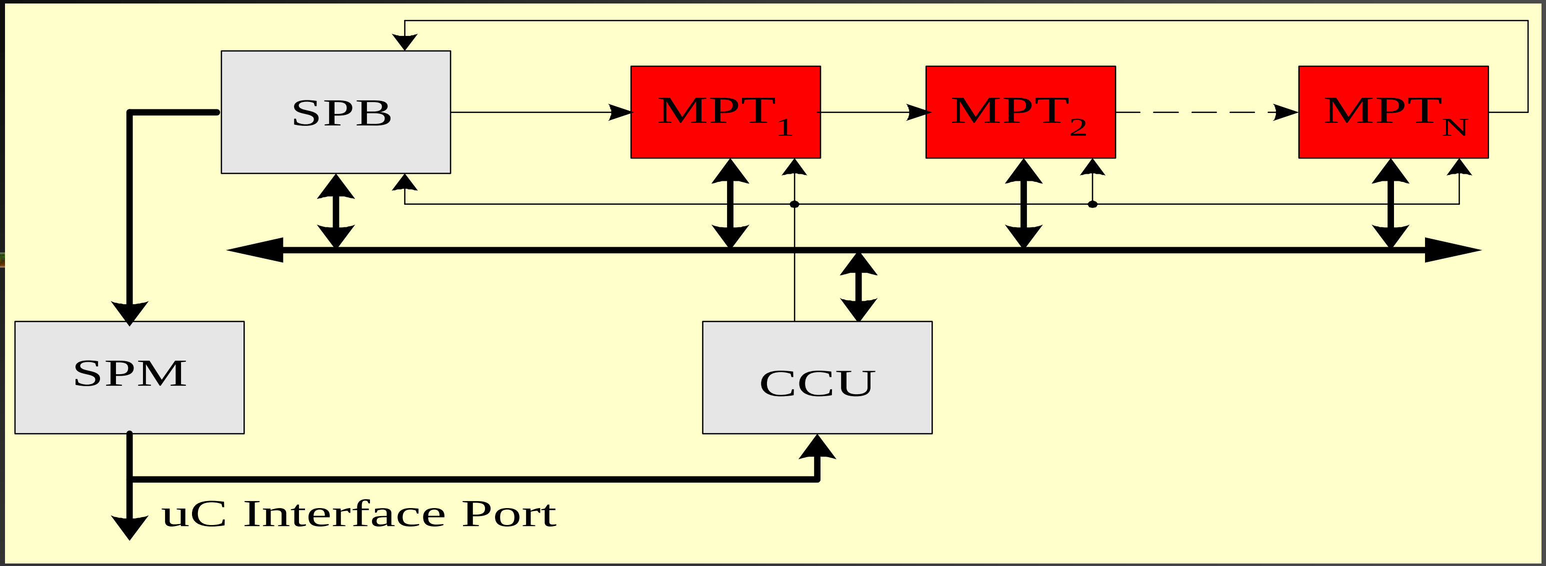
PSCoP architecture



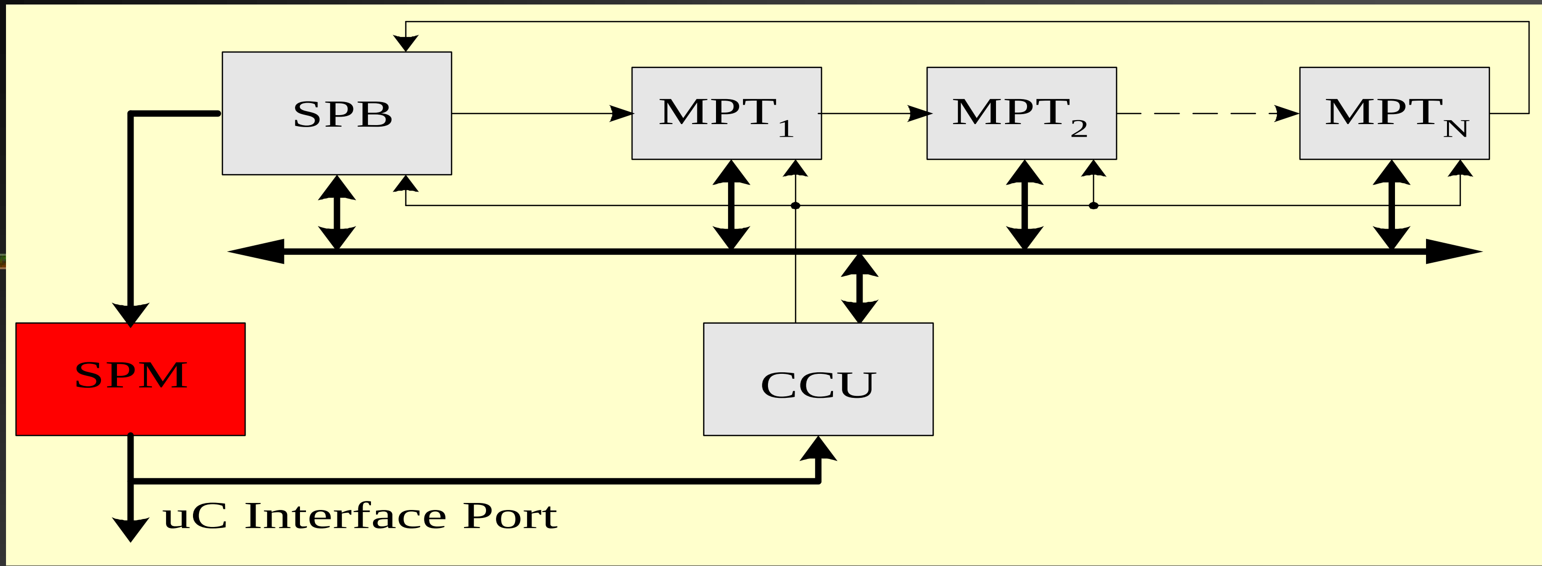
- SPB – Schedule Plan Builder
- MPT – Message Production Timer
- SPM – Schedule Plan Memory
- CCU – Configuration Control Unit



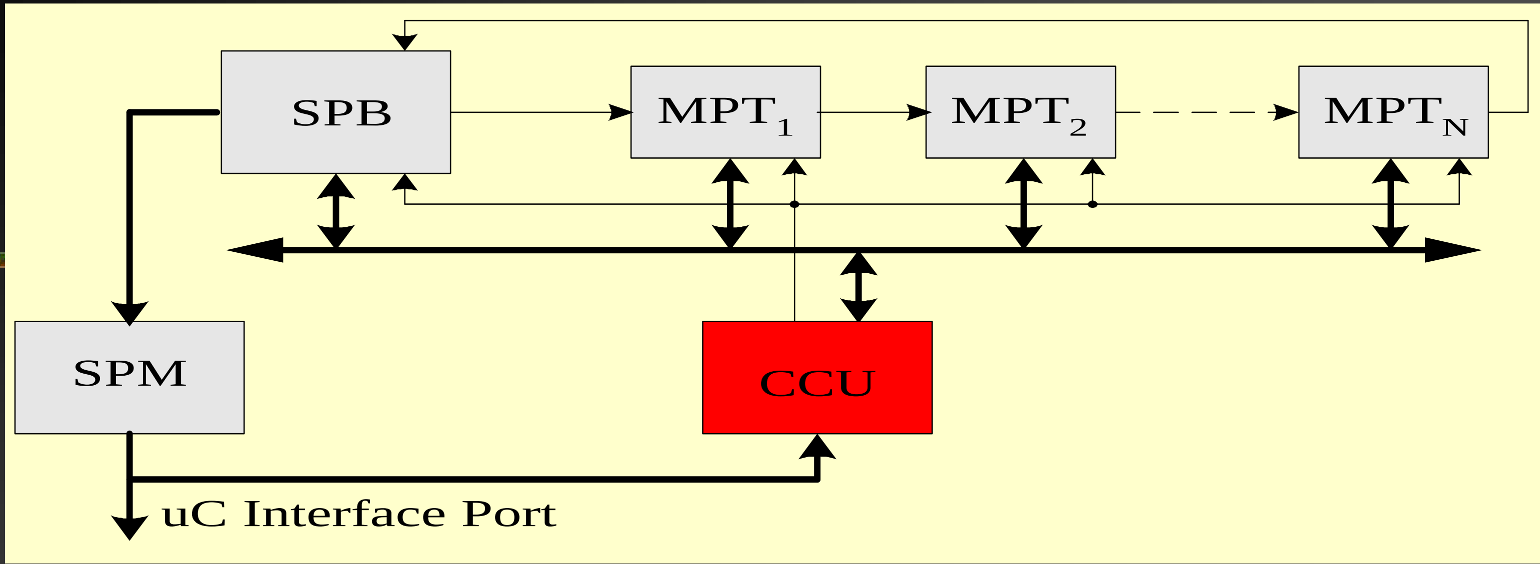
- Schedule Plan Builder
  - Control over the current EC time
  - Control over transaction's acceptance
  - Placing of transactions in the respective EC



- Message Production Timers
  - Each one holds the message parameters (except transaction time)
  - Control over message production
  - Missed deadline detection
  - Daisy chain architecture
    - Static priorities



- Schedule Plan Memory
  - Stores the plans
  - Prepares data for the node µC/µP



- Configuration Control Unit
  - Initialisation of the PSCoP parameters
  - Stores the parameters in the appropriate registers based on the address

Implementation details

- Xilinx XC4010XL FPGA
- 8031 series μC
- 400 CLBs (100% of the FPGA die)
- 44 IOBs and 2 clock IOBs
- Equivalent gate count of 8533 gates

PSCoP performance

- Measurements taken at 12MHz
- 9 message sets, with all the messages produced in all ECs
- Execution time of a complete plan

# Messages	0	1	2	3	4	5	6	7	8
Exec. Time (μsec)	8	16	22	30	36	41	50	56	63

Conclusions

- Device suitable for DES applications
- Compatible with the FTT-CAN protocol
- PSCoP results may lead to an almost dynamic approach
  - 1 CAN message @ 1Mbps: 53μsec to 130μsec
  - PSCoP worst performance: 63μsec

Future work

- Increasing the number of messages
- Change of parameters during operation
- Different scheduling policies
  - Rate Monotonic
  - Deadline Monotonic
  - On-line switching between policies

Using a Hardware Coprocessor for Message Scheduling in Fieldbus-based Distributed Systems

José A. Fonseca
Ernesto V. Martins
Paulo Neves