

FTT CAN ERROR CONFINEMENT

Joaquim Ferreira, Paulo Pedreiras, Luís Almeida, José A. Fonseca


DET-IEETA

University of Aveiro

P-3810-193 Aveiro, Portugal

Presented by Joaquim Ferreira at FeT 2001, Nancy-France, 15-16 November 2001

Presentation outline:

- 
- 1. Objectives**
 - 2. FTT CAN brief presentation**
 - 3. Error confinement**
 - 4. Bus traffic monitoring**
 - 5. Conclusions**

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

- **Present a first study on the influence of transmission errors in FTT CAN**
- **Develop methods to limit the impact of transmission errors in FTT CAN protocol**

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

FTT CAN: Flexible Time Triggered Protocol on CAN

- Time-triggered communication (synchronous traffic) with:
 - ✓ On-line admission control
 - ✓ Phase control, leading to
 - Composability
 - Improved jitter control
 - ✓ Low communication & processing overhead
- Centralized scheduling of dynamic synchronous traffic with timeliness guarantees
- Communication overhead is kept low by using the native distributed arbitration of CAN
- Both “Time-Triggered” and “Event-Triggered” paradigms are supported, with enforced temporal isolation

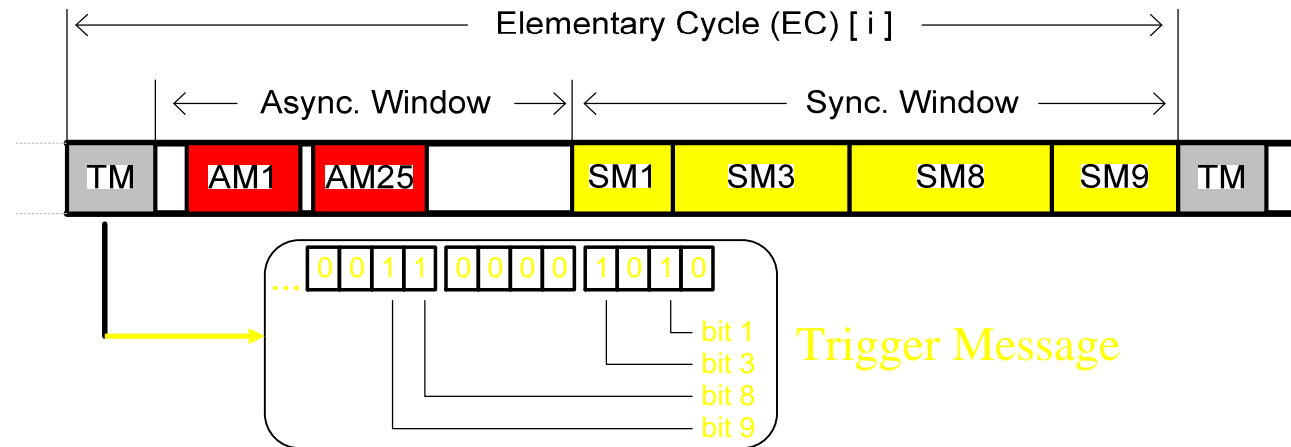
1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions



- The master node sends a message at the beginning of each EC
 - ✓ That carries the identification of the synchronous messages to be produced on that EC
 - ✓ Can act as a time mark for clock synchronization
- Each Elementary Cycle has 2 phases:
 - ✓ Asynchronous window (Event triggered traffic)
 - ✓ Synchronous window (Time triggered traffic)

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Error confinement: assumptions

- As is, the protocol enforces temporal isolation between the asynchronous and the synchronous windows
 - ✓ The domino effect between consecutive windows, due to transmission errors, is prevented
- The temporal resolution for the synchronous messages is the EC duration
 - ✓ Within the same EC, a message can be retransmitted and still maintain its timeliness.
- Error model
 - ✓ Deterministic (N errors inside each EC phase)
 - ✓ No error occurrence in the trigger message
 - ✓ Permanent transmitter or receiver failures are not considered

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Passive error confinement: soft real-time

- Soft real-time operation of FTT CAN
 - ✓ Loosing messages implies reduction on system QoS
- What synchronous message should be discarded?
 - ✓ The one affected by the error?
 - ✓ The lower priority one?

1. Objectives

2. FTT CAN brief presentation

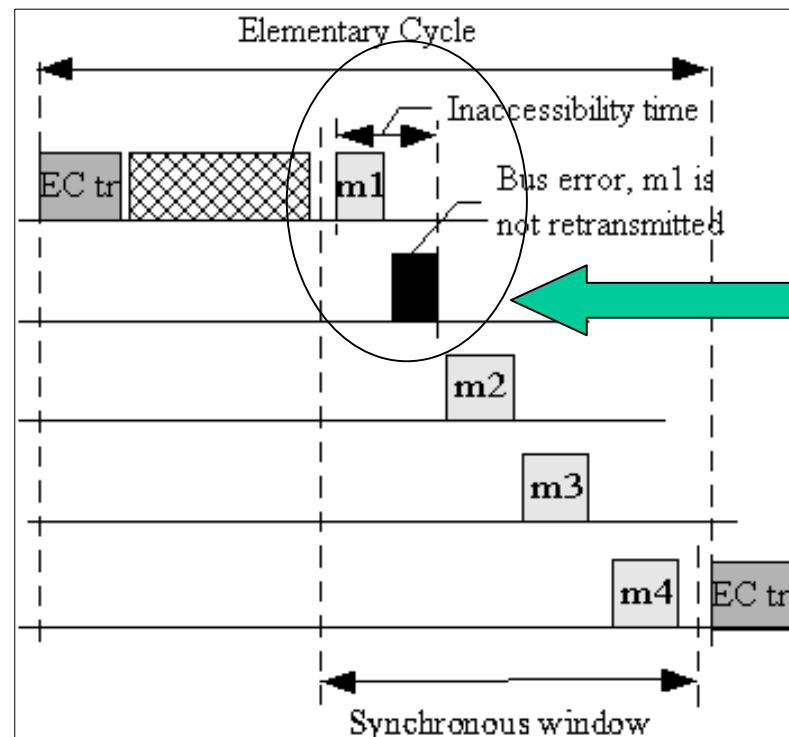
3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Passive error confinement: soft real-time

- Discarding the message disrupted by a transmission error



The message where the error occurs is not retransmitted. Single shot transmission mode required (no automatic retransmission in case of error or arbitration loss)

1. Objectives

2. FTT CAN brief presentation

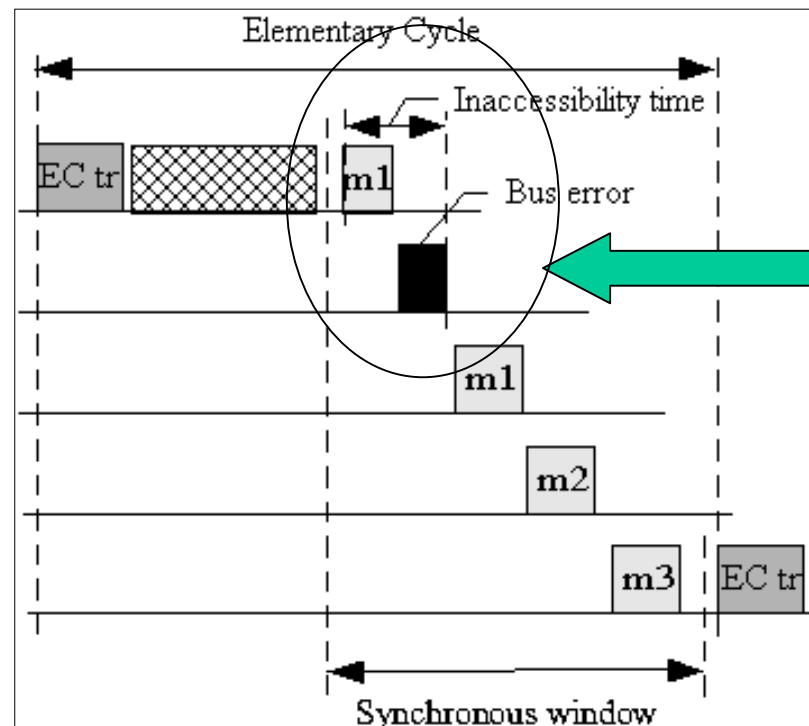
3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Passive error confinement: soft real-time

- Discarding the lowest priority synchronous message



The message where the error occurs is retransmitted and the one with the lowest priority is lost.

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Passive error confinement: soft real-time

- Retransmit the message where the error occurred (discarding the lower priority one) is the best option, because:
 - ✓ It is based on CAN normal operation mode
 - ✓ Priority of the messages is preserved (in contrast with other protocols, e.g. TT CAN)
- Overhead to accommodate each error (time to abort transmission) is, in both cases, $61t_{bit} (194 - 133)$

Maximum inaccessibility time
(Rufino and Veríssimo, 1997)

Maximum CAN 2.0A frame

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Passive error confinement: hard real-time

- Messages **must** be delivered before their deadline
- Both synchronous and asynchronous traffic is considered

A slack time of $194t_{bit}$ must be inserted in both EC phases, so that a single error can be confined

The bus utilization is reduced by $2 \times 194t_{bit}$ every EC, for each predicted transmission error

- This corresponds, assuming an EC duration of 10 ms and maximum size messages, to:
 - ✓ 37,5% for CAN 2.0A @ 125 Kbit/s
 - ✓ 4% for CAN 2.0A @ 1 Mbit/s

1. Objectives

2. FTT CAN brief presentation

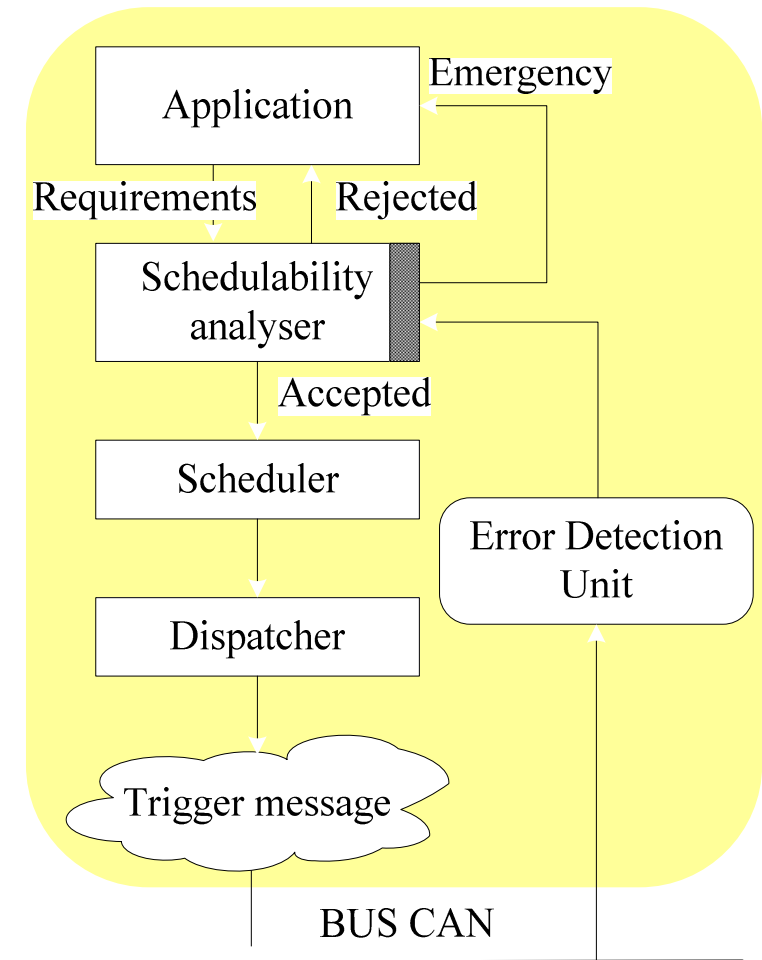
3. Error confinement

4. Bus traffic monitoring

5. Conclusions

- The bus transactions are monitored and that data is fed back into the scheduler
- Appropriate measures can then be taken:
 - ✓ Reschedule the faulty message for next ECs
 - ✓ Discard message
 - ✓ Emergency stop

Master node



1. Objectives

2. FTT CAN brief presentation

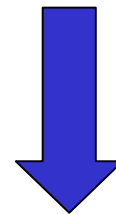
3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Active error confinement

- It is assumed that every message sent to the network without errors, is also correctly received by all nodes (atomic broadcast)
- Execution time of scheduler and acceptance test must be less than the EC time, so that corrective measures can be taken as soon as possible



Master node cannot be based on a low performance microcontroller !!

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Master node architectural options

- Use a more powerful microcontroller
 - ✓ Easier to customize
 - ✓ Smaller design time
 - ✓ Standard tools/components availableHowever ...
 - ✓ Worst case execution time: possibly longer and difficult to quantify
- Use specialized hardware
 - ✓ Wide range of synthesizable VHDL blocks available
 - ✓ Worst case execution time: reduced and easier to quantify,However ...
 - ✓ Harder to design and higher design time
 - ✓ The components/tools available are not standard and of widespread knowledge
 - ✓ Final system with more components (lower reliability)

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Use of specialized hardware

- The availability of intellectual property (IP) cores enables a rapid development of new architectural solutions through the addition of specialized functionalities on top of general purpose building blocks
 - ✓ European Space Agency (ESA) distributes, free of charge, a synthesizable VHDL CAN core (HurriCANE)
 - ✓ There are several synthesizable VHDL models of processors (8051, SPARC, MIPS, ARM, etc) available.

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Use of specialized hardware

- Besides the online dynamic scheduling and error detection and confinement, the specialized hardware can also possibly include some extra features:
 - ✓ Automatic filtering of the trigger message
 - ✓ Automatic control of synchronous and asynchronous windows timing
 - ✓ Bank of timers to assist synchronous message production
 - ✓ Automatic promptness/refreshness management of messages

1. Objectives

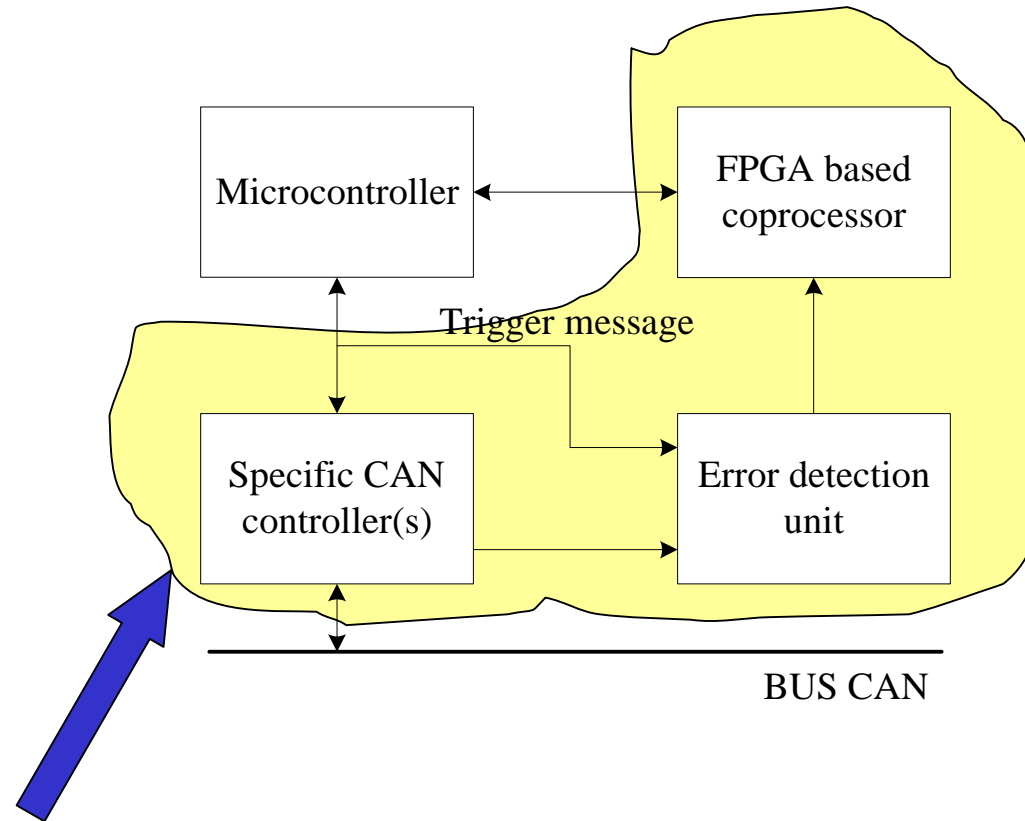
2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

Master node architecture proposal



All these blocks can be physically located in the same FPGA

1. Objectives

2. FTT CAN brief presentation

3. Error confinement

4. Bus traffic monitoring

5. Conclusions

- A first approach of the influence of transmission errors in FTT CAN was presented
- Two methods were developed to limit the impact of transmission errors on FTT CAN protocol
 1. Passive (slack time insertion), supporting hard real time traffic
 2. Active (error detection and retransmission, if possible)
- A draft architecture for the master node was proposed
- This solution is possible due to FTT CAN intrinsic properties of temporal isolation between synchronous and asynchronous phases and to the speed of of recently developed FPGA based scheduling coprocessors