CANopy: A VHDL implementation of CANopen Protocol for CAN Bus On Board Spacecraft

IP Cores Workshop

F. Tortosa López Vitrociset/ESTEC/ESA
P. Roos ESTEC/ESA
CAN in Space

- CAN Working Group Draft Recommendation
  - Higher Layer Protocol (CANopen based)
  - Physical Layer, Redundancy Architecture & Algorithm
- CAN Building blocks available
  - Transceivers, CAN Controllers … What for the HLP?
- Why a VHDL implementation of CANopen?
- CANopen traditionally implemented in software
  - Good approach for terrestrial application
  - Mass production/Low-cost microcontrollers widely available/Simplifies system evolution
- For space, hardware-only implementation advantages
  - Availability of microcontrollers/processors is limited
  - Possibility of having CPU-less implementations
  - For use in simple remote terminals
  - Suitable for System-on-Chip solutions
  - Reduced complexity of the software in CPU-based nodes
CANopy interfaces

- General I/O Interface
- AMBA Interface
- CAN Controller Interface
CANopy architecture

Input Buffer → Selector → SDO Handler

Rx_msg

SYNC Consumer

RPDO Handler

SYNC Signal

I/O Data Interface or AMBA

Object Dictionary

Heartbeat producer

TPDO

NMT State Machine

Node State Signal

Output MUX → Output Buffer

Tx_msg
Preliminary synthesis results

- Version 1.3 synthesised using Synplify Pro for Actel FPGA RT54SX72S-1
  - Object Dictionary of 32 objects
  - SDO Server
  - NMT State machine
  - 2 TPDOs, no RPDOs implemented
  - 5 messages Priority ordered output buffer
  - HurriCANe CAN Controller integrated

- CANopy: 36%
- HurriCANe: 23%
- CANopy+HurriCANe: 59%

- Version 1.4 including RPDOs and 60 objects in the object dictionary is currently under simulation tests