~ Sli.do Q&A ~
Payload and Platform Sensor / Actuator Bus Nodes: Micronode - Athena mission

What are the lesson learned from using the GR716

Pay close attention to the datasheet and recommendations. In particular to the decoupling arrangement and return currents on the analogue power rails.

Defining your pinout is a key task. Use the spreadsheet tool provided by Gaisler and take the time to understand it in detail. Micronodes had a complex set of I/O constraints as it supported 1553, SpW, 2x CANbus and 2xSPI and several GPIO. You need to be sure you can achieve your desired I/O functionality before baselining your requirements.

External memory uses a lot of pins, so practically code must run in internal RAM. Careful management of the code memory footprint is essential, particularly for command/control message encoding/decoding.

Consider your reset strategy carefully. Independent external watchdog timer is typically preferable for Space applications rather than being solely reliant on the internal watchdog.

Make use of internal 1.8V LDO to simplify power design. Ensure your application meets power dissipation constraints to enable you to make use of this feature.

Procure the CPCI Mother board dev. Kit that is now available as it comes with the necessary I/O panel to give access to the SPI, UART, CAN, 1553 interfaces which will allow most functions to be tested in advance. If this had been available for our development, it would have made software integration easier as we'd have caught bugs and raised them with Gaisler earlier on in the design process.

If you think carefully about frequency of operation of the various interfaces, it should be possible to generate all your required frequencies from a single clock source. Ideally you want to do this analysis before baselining requirements.

We had a good experience implementing the GR716 on our own PCB. The design worked first time and there was only a single wire mod to correct the power rail of a pull up resistor. The main issues we experienced were in software debug rather than hardware. In particular the second SPI interface did not perform as expected.

The GR716 is fairly well provisioned with memory for a microcontroller, but the SPARC instruction set does not generate particularly compact code, although REX may help. Prototyping at the requirements stage is essential.

MBSE approach is very applicable to GR716. A high level of auto-generation (85%) is possible.

Bare-metal programming is straightforward on the GR716 and is suitable for most microcontroller applications, which tend to have regular and synchronous execution patterns.

Technical support from Gaisler was excellent and very responsive.

So clearly for most sensors CANopen is superior to SpaceWire?

CANopen is a protocol that runs on top of the CANbus link layer and is ideal for microcontrollers that are providing the sensing function.

The closest SpaceWire protocol is RMAP that runs on top of the SpaceWire link layer. RMAP is ideal for FPGA or microcontrollers that are providing the sensing function.

MIL-STD-1553 is often also used and is ideal for FPGA or Microcontrollers.

The GR716 is a mixed signal LEON3 microcontroller and can support MIL-STD-1553, SpaceWire or CANbus.

The choice of protocol also comes down to a trade-off between the properties of the link layer and physical layer.

1) Master/Slave Protocol - MIL-STD-1553
    a. Easy to implement protocol on microcontrollers such as the GR716 that provide the MIL-STD-1553 interface
    b. Easy to implement slave protocol in FPGA.
    c. 1 Mbps maximum data rate.
    d. Excellent electrical isolation (1000V)
    e. Excellent fault propagation protection
    f. A+B Redundancy built into the standard
    g. High size, mass and power due to magnetics and PHY.
    h. Low cable mass as it is multi-drop. Good for when several sub-systems have to be commanded by a central unit.

2) CANOpen – CANbus
    a. Easy to implement protocol on microcontrollers such as the GR716 that provide the CANbus interface.
    b. Difficult to implement protocol in FPGA.
    c. 1 Mbps
    d. Reasonable electrical isolation (5V) compared to SpaceWire (1V), but not close to being as good as MIL-STD-1553 (1000V).
    e. Reasonable fault propagation protection if transceivers fail open circuit.
    f. Similar redundancy to MIL-STD-1553 with separate A and B CANbus links.
    g. Low size, mass and power as CANbus transceiver is a small IC.
    h. Low cable mass as it is multi-drop. Good for when several sub-systems have to be commanded by a central unit.

3) SpaceWire RMAP – SpaceWire:
    a. Easy to implement protocol on microcontrollers such as the GR716 that provide the interface
    b. Easy to implement protocol on FPGAs.
    c. Data rates up to 100 Mbps on GR716. 100x CANbus and MIL-STD-1553 data rate.
    d. Poor electrical isolation (1V).
    e. Care has to be taken to prevent fault propagation.
    f. Point to point link so requires two links to support N+R redundancy.
    g. Low size, mass and power as LVDS transceiver built into GR716 / FPGA
    h. Cable mass is high if the requirement is multi-drop and each unit requires a dedicated N+R SpaceWire interface.