Use of USB interfaces in Space Programs

Use of USB interfaces in Space Programs

Author: Mr. Giorgio Magistrati
Carlo Gavazzi Space S.p.A., Piacenza, Italy, gmagistrati@cgspace.it
Dr. Massimiliano Pastena
Carlo Gavazzi Space S.p.A., Benevento, Italy, mpastena@cgspace.it

1. Abstract

Carlo Gavazzi Space (CGS) is currently involved in the study and analysis of new technologies for space avionics with the aim to simplify and reduce the cost of development and production of space systems to be exploited by an enlarged number of potential users (commercial customers, research institutes, ….).
Clearly the adoption and the portability into the space environment of already developed and diffused commercial technologies has been considered and, among the available digital interfaces with already proven characteristics in term of robustness, speed, reliability and available support for the HW and SW parts provided by manufacturers and committees , CGS has identified the USB as the best candidates for a fast digital interface to be used as one of the satellite digital buses.

2. Introduction

The USB has several characteristics that make it advantageous and gainful as satellite communication bus among the Spacecraft OBDH and the P/L Units, Mass Memory or AOCS sensors:

1) Master-slave architecture that foresees a single host controller (that could be the Spacecraft OBDH) and several slave devices (connected into a daisy chain): up to127 devices may be connected to a single host controller

2) Hot swapping that allows the connection and disconnection of slave devices without powering down or rebooting the computer/single host controller

3) High Data Rate up to 480 Mbits/sec in the USB2.0 version (USB 2.0 High Speed, that seems ideal for Mass Memory Unit, CCD Cameras. . .) but also the version USB Full Speed with a speed of 12 Mbit/sec (USB version 1.1) or USB Low Speed with its 1,5 Mbit/sec (USB 1.0) seems attractive for certain space applications (Reaction Wheel, . . .)

4)Reduced number of needed pins: the USB 2.0 due to is serial transmission characteristics requires only four signal pins plus a shield signal that is carried by the connector shell itself, this means a reduction in the satellite harness weight and complexity.

5)The USB physical layer foresees that the slave devices can sink up to 500 mA from the host controller : this feature seems attractive in scenarios where for example the reaction wheel power electronics is supplied directly by the satellite power bus and the reaction wheel control electronics could be directly powered by the USB I/F avoiding the use of internal to the reaction wheel DC/DC converters supplied by the satellite power bus.

6)The large diffusion of USB devices will dramatically simplify the activity of manufacturing simulation/emulation HW tools for debugging or qualification test sessions.

Beside that the usage of a plug and play and smart electrical I/F as the USB seems ideal for space applications for other two different reasons:

a)Redundancies at satellite S/s equipment level: the hot swapping capability will allow the implementation of hot/cold redundancies with a very limited increase of complexity in terms of weight or H/W I/Fs

b)Possibility to upgrade some S/s equipment in a Satellite family without changes in the satellite electrical interfaces

It has to be underlined that several applications of USB in space have been already started especially in the area of microgravity: here we want to mention the European Physiology Modules facility (EPM) hosted into Columbus module and to be installed on the International Space station in December 2007 with a flight of the Shuttle Atlantis. EPM is a multi user facility supporting research in the area of human physiological adaptation to weightlessness. The EPM has been developed by OHB-system AG (Bremen Germany). USB is used in the EPM Science Module Support Computer (SMSC, based on the VMEbus system) and in the EPM Video Unit (VU, based on PC104 boards) see Ref2

3. Brief overview on USB

The USB is a digital serial bus implemented on copper cable that supports data exchange between a host computer and a wide range of simultaneously accessible peripherals. The attached and powered on peripherals share USB bandwidth through a host-scheduled, token-based protocol. The bus allows peripherals to be attached, configured, used, and detached while the host and other peripherals are in operation.

Composition of a USB system

A USB system is described by three definitional areas:

- USB host
- USB devices
- USB interconnect

USB host

There is only one host in any USB system; it is usually named as the Host Controller. The Host Controller may be implemented in a combination of hardware, firmware, or software. Usually in the commercial laptop or desktop a root hub is integrated within the host system to provide one or more attachment points

USB devices

USB devices can be one of the two following types:

- Functions, which provide capabilities to the system, such as an ISDN connection, an USB Pen Driver, a Digital Camera, a Storage Device.
- Hubs, which provide additional attachment points to the USB.

USB interconnect

The USB interconnect is the manner in which USB devices are connected to and communicate with the host. This interconnection includes the following:

- Bus Topology: Connection model between USB devices (Functions and Hubs) and the host.
- Inter-layer Relationships: In terms of a capability stack, the USB tasks that are performed at each layer in the system.
- Data Flow Models: The manner in which data moves in the system over the USB between services producers and consumers.
- USB Schedule: The USB provides a shared interconnect. Access to the interconnect is scheduled in order to support isochronous data transfers and to eliminate arbitration overhead.

The USB connects USB devices with the USB host. The USB physical interconnect is a tiered star

topology. A hub is at the centre of each star. Each wire segment is a point-to-point connection between the host and a hub or function, or a hub connected to another hub or function. Figure 1 illustrates the topology of the USB. Up to a maximum number of 127 devices can be implemented in a USB system.

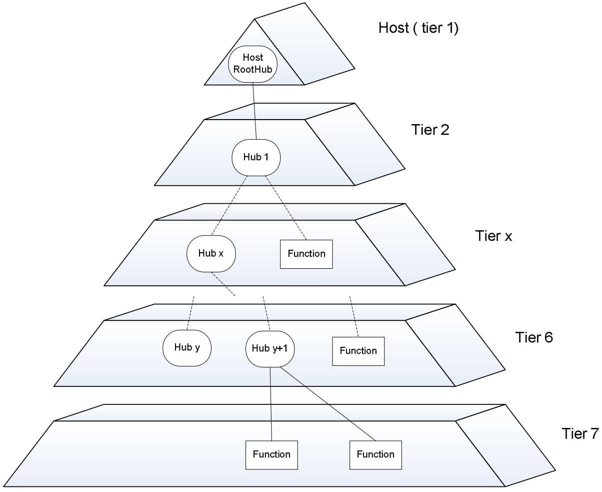

Figure 1 USB Bus Topology

Timing constraints for hub and cable propagation times limit the maximum number of tiers to seven (including the root tier). Note that in seven tiers, five non-root hubs maximum can be supported in a communication path between the host and any device. therefore, it cannot be enabled if attached at tier level seven. Only functions can be enabled in tier seven.

The USB transfers signal and power over a four-wire cable. The signaling occurs over two wires on each point-to-point segment. The shield signal is carried by the connectors shell and by the drain wire of the cable.

Signal Name	Function
Vbus	Power supply, 5 Vdc
D+	Data +, bidirectional Signal,LVDS
D-	Data - , bidirectional Signal,LVDS
Ground	Return of Power supply
Shield	Carried by connector shell and drain wire in the cable

table 1USB signals

Data Rates

There are three data rates:

• The USB high-speed signaling bit rate is 480 Mb/s (USB 2.0).

• The USB full-speed signaling bit rate is 12 Mb/s (USB 1.1).

• The USB low-speed signaling bit rate is 1.5 Mb/s (USB 1.0).

USB 2.0 host controllers and hubs provide capabilities so that full-speed and low-speed data can be transmitted at high-speed between the host controller and the hub, but transmitted between the hub and the device at full-speed or low-speed. This capability minimizes the impact that full-speed and low-speed devices have upon the bandwidth available for high-speed devices.

The low-speed mode has been defined to support a limited number of low-bandwidth devices, such as mouse, and in a space application it could be used to control devices with a limited data bandwidth such as reaction wheels.

The clock is transmitted, encoded along with the differential data. The clock encoding scheme is NRZI with bit stuffing to ensure adequate transitions. A SYNC field precedes each packet to allow the receiver(s) to synchronize their bit recovery clocks.

The USB cable also carries VBUS and GND wires on each segment to deliver power to devices. VBUS is nominally +5 V at the source. And a generic Host is able to provide up to 500mA at the load.

Each USB segment provides a limited amount of power over the cable. The host supplies power for use by USB devices that are directly connected. In addition, any USB device may have its own power supply. In a space application the power from the USB cable should be in principle used only to supply the electronics of the USB device interface, while for example in a reaction wheel the motors will be powered directly by the satellite power bus (Isolation technique should be implemented to separate from a EMC point of view the control and the driving section of the reaction wheel)

USB devices that rely totally on power from the cable are called bus-powered devices. In contrast, those that have an alternate source of power are called self-powered devices. A hub also supplies power for its connected USB devices. The USB allows cable segments of variable lengths, up to several meters, by choosing the appropriate conductor gauge to match the specified voltage drop. In order to provide guaranteed input voltage levels and proper termination impedance, biased terminations are used at each end of the cable. The terminations also permit the detection of attach and detach at each port and differentiate between high/full-speed and low-speed devices.

USB connectors

The USB Specification defines also the types of connectors to be used: two different types have been defined for commercial uses: Series A and Series B see Figure 2.

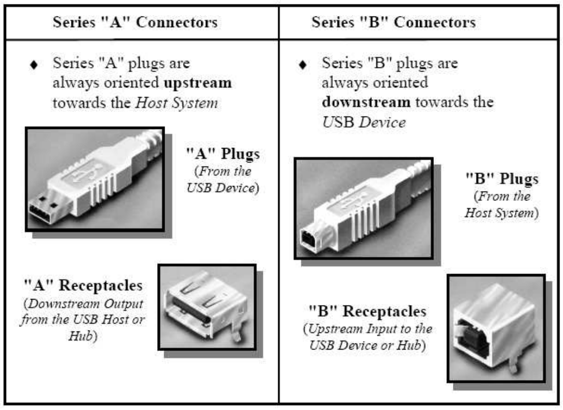
Series "A" Connectors
Series "B" Connectors
Series "A" plugs are always oriented **upstream** towards the *Host System*
Series "B" plugs are always oriented **downstream** towards the *USB Device*
"A" Plugs
(*From the USB Device*)
"B" Plugs
(*From the Host System*)
"A" Receptacles
(*Downstream Output from the USB Host or Hub*)
"B" Receptacles
(*Upstream Input to the USB Device or Hub*)

Figure 2 Series A and B of USB commercial connectors

USB Operations

The USB is a polled bus. The Host Controller initiates all data transfers. Most bus transactions involve the transmission of up to three packets. Each transaction begins when the Host Controller, on a scheduled basis, sends a USB packet describing the type and direction of transaction, the USB device address, and endpoint number. This packet is referred to as the “token packet.” The USB device that is addressed selects itself by decoding the appropriate address fields. In a given transaction, data is transferred either from the host to a device or from a device to the host. The direction of data transfer is specified in the token packet. The source of the transaction then sends a data packet or indicates it has no data to transfer. The destination, in general, responds with a handshake packet indicating whether the transfer was successful.

Some bus transactions between host controllers and hubs involve the transmission of four packets. These types of transactions are used to manage the data transfers between the host and full-/low- speed devices.

The USB data transfer model between a source or destination on the host and an endpoint on a device is referred to as a pipe. There are two types of pipes: stream and message. Stream data has no USB-defined structure, while message data does. Additionally, pipes have associations of data bandwidth, transfer service type, and endpoint characteristics like directionality and buffer sizes. Most pipes are initiated when a USB device is configured. One message pipe, the Default Control Pipe, always exists once a device is powered, in order to provide access to the device’s configuration, status, and control information.

The transaction schedule allows flow control for some stream pipes. At the hardware level, this prevents buffers from underrun or overrun situations by using a NAK handshake to throttle the data rate. When NAKed, a transaction is retried when bus time is available. The flow control mechanism permits the construction of flexible schedules that accommodate concurrent servicing of a heterogeneous mix of stream pipes. Thus, multiple stream pipes can be serviced at different intervals and with packets of different sizes.

To provide protection against such transients, each packet includes error protection fields. When data integrity is required, such as with lossless data devices, an error recovery procedure may be invoked in hardware or software.

The protocol includes separate CRCs for control and data fields of each packet. A failed CRC is considered a notification of a corrupted packet. The CRC gives 100% coverage on single- and double-bit errors. This feature of the USB seems quite attractive for its usage in space projects.

All USB devices attach to the USB through ports on specialized USB devices known as hubs. Hubs have status bits that are used to report the attachment or removal of a USB device on one of its ports. Here it has to be underlined that a hot insertion or removal of USB device (Function and HUB) from a powered USB system seems not to be a required feature in a space application.

The host assigns a unique USB address to the device and then determines if the newly attached USB device is a hub or a function.

When a USB device has been removed from one of a hub's ports, the hub disables the port and provides an indication of device removal to the host.

Bus enumeration is the activity that identifies and assigns unique addresses to devices attached to a bus.

Because the USB allows USB devices to attach to or detach from the USB at any time, bus enumeration is an on-going activity for the USB System Software. Additionally, bus enumeration for the USB also includes the detection and processing of removals.

The USB supports functional data and control exchange between the USB host and a USB device as a set of either uni-directional or bi-directional pipes. USB data transfers take place between host software and a particular endpoint on a USB device. Such associations between the host software and a USB device endpoint are called pipes. In general, data movement though one pipe is independent from the data flow in any other pipe. A given USB device may have many pipes. As an example, a given USB device could have an endpoint that supports a pipe for transporting data to the USB device and another endpoint that supports a pipe for transporting data from the USB device.

Data Transfer

The USB architecture comprehends four basic types of data transfers:

- . Control Transfers: Used to configure a device at attach time and can be used for other device-specific purposes, including control of other pipes on the device.
- . Bulk Data Transfers: Generated or consumed in relatively large and bursty quantities and have wide dynamic latitude in transmission constraints.
- Interrupt Data Transfers: Used for timely but reliable delivery of data, for example, characters or coordinates with human-perceptible echo or feedback response characteristics.
- . Isochronous Data Transfers: Occupy a prenegotiated amount of USB bandwidth with a prenegotiated delivery latency. (Also called streaming real time transfers).

A pipe supports only one of the types of transfers described above for any given device configuration.

Control Transfers

Control data is used by the USB System Software to configure devices when they are first attached. Other driver software can choose to use control transfers in implementation-specific ways (this could be case for Reaction Wheel or to start the operative phase of a CCD imager) Data delivery is lossless.

Bulk Transfers

Bulk data typically consists of larger amounts of data, such as that used for printers or scanners. In a space application the use of Bulk Transfers could be limited to Mass Memory Unit or CCD Imager (representing an alternative solution to Spacewire or Packetwire) . Bulk data is sequential. Reliable exchange of data is ensured at the hardware level by using error detection in hardware and invoking a limited number of retries in hardware. Also, the bandwidth taken up by bulk data can vary, depending on other bus activities.

Interrupt Transfers

A limited-latency transfer to or from a device is referred to as interrupt data. Such data may be presented for transfer by a device at any time and is delivered by the USB at a rate no slower than is specified by the device.

Interrupt data typically consists of event notification, characters, or coordinates that are organized as one or more bytes. An example of interrupt data is the coordinates from a pointing device. In a space application the information that a failure has occurred into a device could be transferred by means of a interrupt transfer.

Isochronous Transfers

Isochronous data is continuous and real-time in creation, delivery, and consumption. Timing-related information is implied by the steady rate at which isochronous data is received and transferred. Isochronous data must be delivered at the rate received to maintain its timing. In addition to delivery rate, isochronous data may also be sensitive to delivery delays. For isochronous pipes, the bandwidth required is typically based upon the sampling characteristics of the associated function. The latency required is related to the buffering available at each endpoint.

A typical example of isochronous data is voice. If the delivery rate of these data streams is not maintained, drop-outs in the data stream will occur due to buffer or frame underruns or overruns. Even if data is delivered at the appropriate rate by USB hardware

In a space application the isochronous transfer seems the ideal transfer type to distribute clock data info among the various Spacecraft S/s (as usually performed by the MIL-STD-1553B on a Spacecraft)

USB Bandwidth

USB bandwidth is allocated among pipes. The USB allocates bandwidth for some pipes when a pipe is established. USB devices are required to provide some buffering of data. It is assumed that USB devices requiring more bandwidth are capable of providing larger buffers. The goal for the USB architecture is to ensure that buffering-induced hardware delay is bounded to within a few milliseconds.

The USB's bandwidth capacity can be allocated among many different data streams. This allows a wide range of devices to be attached to the USB. Further, different device bit rates, with a wide dynamic range, can be concurrently supported.

The USB Specification defines the rules for how each transfer type is allowed to access to the bus.

USB Host: Hardware and Software

The USB host interacts with USB devices through the Host Controller. The host is responsible for the following:

- ➢ Detecting the attachment and removal of USB devices (Not required in a space application tbc)
- ➢ Managing control flow between the host and USB devices
- ➢ Managing data flow between the host and USB devices
- ➢ Collecting status and activity statistics
- ➢ Providing power to attached USB devices

The USB System Software on the host manages interactions between USB devices and host-based device software. There are five areas of interactions between the USB System Software and device software:

- ➢ Device enumeration and configuration
- ➢ Isochronous data transfers
- ➢ Asynchronous data transfers
- ➢ Power management
- ➢ Device and bus management information

4. Generic USB Architecture on a Spacecraft

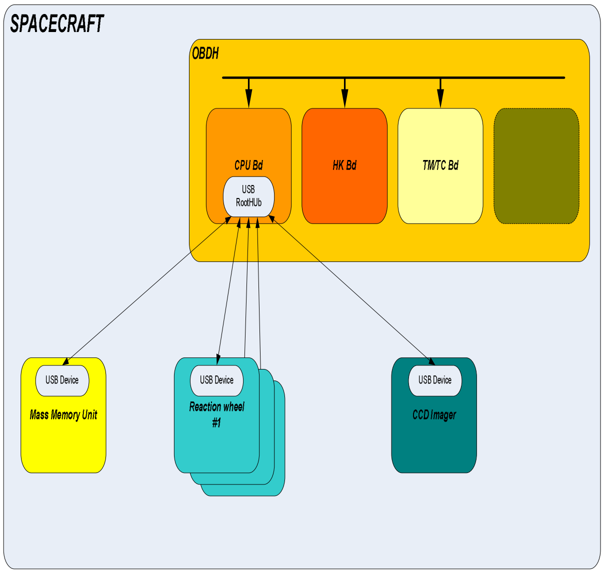

Figure 3 Generic USB Architecture on a spacecraft

Figure 3 shows a generic avionics architecture for a spacecraft using the USB. The CPU Board of the OBDH is the Host Controller of the implemented USB communication system. All the USB devices are Function type and directly seen by the Host Controller, therefore, with reference to Figure 1, an implementation of only two tiers is conceivable on a spacecraft.

The shown USB devices are:
#1 Mass Memory Unit (to be considered also a sort of P/L Memory Unit)
#3 Reaction Wheels
#1 CCD Imager (or CCD camera to be considered part of the P/L)

As a Host controller a few solutions are already available:

- A Host Controller IP Core from Gaisler Research (www.gaisler.com): The USB 2.0 Host Controller core (GRUSBHC) from Gaisler Research provides a link between the AMBA on-chip bus and the Universal Serial Bus (USB). The host controller supports High-, Full- and Low-Speed USB traffic. USB 2.0 High-Speed functionality is supplied by an enhanced host controller implementing the Enhanced Host Controller Interface (EHCI). The developed IP core is able to provide up to 15 downstream ports with up to 127 devices each , is available as VHDL RTL source code or FPGA/ASIC netlist and is sinthetizable or in a Xilinx Virtex or in a Asic. CGS has already an experience in using IP-core developed by Gaisler Research and implemented in ASIC/FPGA: in the project ARGO CDMU the core processors is a Leon3-FT implemented into a FPGA AX2000 from Actel Inc. (www.actel.com)
- COTS solution: the last years have seen an increasing use of COTS solutions (S/s and boards) in the space market (in particular International Space Station and commercial satellites). Several boards and ICs manufacturers have in their catalogues products that include a USB Host Controller in their functionalities.

Therefore CGS has concentrated its efforts in the study of the USB device (Function type) that could be implemented in several Spacecraft S/s (part of Platform and/or Payload section). The idea is to develop a USB device interface that could be considered a sort of portable building block to be mounted in different Spacecraft S/s as a sort of mezzanine or small board. The USB device interface shall support High-, Full- and Low-Speed USB traffic.

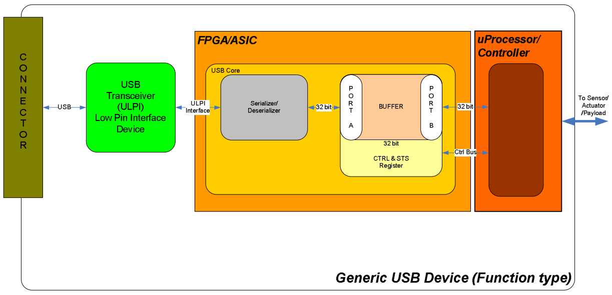

Figure 4 Generic USB Device (Function Type)

The Figure 4 shows the elements that compose the generic USB device (function type) that CGS is considering:

The USB connector,
The USB Transceiver (ULPI UTMI+ Low Pin Interface),
The USB core to be implemented into an ASIC/FPGA,

Also a uProcessor/Controller is shown in Figure 4: it represents the processing core of the device unit and the control centre of the Sensor/Actuator/Payload element. Different solutions for the uProcessor/Controller could be implemented depending on the type of the Spacecraft Subsystem to be developed: in some cases it’s also conceivable that a IP-Core, integrated in the same FPGA or ASIC where the USB core is present , will be used with an enormous advantage in term of PCB/Mezzanine size reduction.

5. USB connector for a space application

The USB connector shall be a micro miniature 9-pin D-type connector (the same type already used for the Spacewire Interface/IEEE 1355). On the USB Device side a male type connector shall be used. The micro miniature D-Type has been selected for its proven features of signal integrity and robustness. The speed reachable on the Spacewire interface/ IEEE 1355 (up to 200MBit/s) guarantees that the proposed connector shall work fine also for the USB interface (Full- and High- Speed).Cable with controlled impedance shall be used. The maximum allowable cable length shall be determined taking into account the signal pair attenuation and propagation delay, nevertheless the dimensions of a conceivable spacecraft where the USB interface could be used lead us to consider that the maximum allowable cable length is not a critical issue.

Pin Function Connector Micro-D 9 pin

1	VBUS
2	Not connected
3	Ground
4	Spare
5	Spare
6	Spare
7	Spare
8	D+
9	D-

table 2 USB pin function for Micro-D connector

6. USB Transceiver

The selected USB Transceiver (ULPI UTMI+ Low Pin Interface) is the ISP1505 from NXP (www.nxp.com, a company founded by Philips) The selected device is a Universal Serial Bus (USB) high-speed host and peripheral transceiver that is fully compliant with Universal Serial Bus Specification Rev. 2.0 and UTMI+ Low Pin Interface (ULPI) Specification Rev. 1.1.

The ISP1505 can transmit and receive USB data at high-speed (480 Mbit/s), full-speed (12 Mbit/s) and low-speed (1.5 Mbit/s). It allows USB Application-Specific Integrated Circuits (ASICs), Programmable Logic Devices (PLDs) and any system chip set to interface with the physical layer of the USB through a 12-pin interface.

Ihe ISP1505 internally Integrates 45 ohm (10 %) high-speed termination resistors and 1.5 kohm (5 %) full-speed device pull-up resistor.

The ISP1505 is available in HVQFN24 package. That is a plastic thermal enhanced very thin quad flat package;no leads; 24 terminals; body 4 x 4 x 0.85 mm.

7. Possible Implementations of a USB Core

The USB core can be implemented in a FPGA or ASIC technology solution with proven characteristic of radiation hardness.

For the implementation into a space environment tolerant FPGA two candidates are available:

RTAX Radiation Tolerant FPGA Family from ACTEL

Hereafter the principal characteristics of the RTAX family are listed.

- ➢ SEU-Hardened Registers that eliminate the need for Triple-Module Redundancy (TMR)
    - o Immune to Single-Event Upsets (SEU) to LETTH > 37 MeV-cm2/mg
    - o SEU Rate < 10-10 Errors/Bit-Day in Worst-Case Geosynchronous Orbit
    - o Expected SRAM Upset Rate of <10-10 Errors/Bit-Day with

- Use of Error Detection and Correction (EDAC) IP with Integrated SRAM Scrubber
    - Single-Bit Correction, Double-Bit Detection
    - Variable-Rate Background Refreshing
- Total Ionizing Dose Up to 300 krad (Si, Functional)
- Single-Event Latch-Up Immunity (SEL) to LET Threshold > 117 MeVcm2/mg
- Up to 4 Million Equivalent System Gates or 500 k Equivalent ASIC Gates
- Up to 20,160 SEU-Hardened Flip-Flops
- Up to 840 I/Os
- Up to 540 kbits Embedded SRAM
- Quality levels: also Class V Equivalent Flow Processing Consistent with MIL-PRF 38535
- Up to 4 Million Equivalent System Gates or 500 k
- Ceramic Quad Flat Pack and Column Grid Array packages

QPRO Virtex II Family from Xilinx

Hereafter the principal characteristics of the QPRO Virtex II family are summarized

- Guaranteed total ionizing dose to 200K Rad(Si)
- Latch-up immune to LET > 160 MeV-cm2/mg
- SEU in GEO upsets < 1.5E-6 per device day achievable with the proprietary redundancy algos implemented
- Certified to MIL-PRF-38535
- Guaranteed over the full military temperature range (–55°C to +125°C)
- Ceramic and Plastic Wire-Bond and Flip-Chip Grid Array Packages
- Architecture designed for high complex and fast digital designs:
    - Densities from 1M to 6M system gates
    - 300+ MHz internal clock speed (Advance Data)
    - 622+ Mb/s I/O (Advance Data)
- Up tp 2.5 Mb of internal dual-port RAM in 18 Kbit block
- High-Performance Interfaces to External Memory (SDRAM, DRAM and SRAM types)

Some IP-Cores (Intellectual Property-Core design) are available for Actel and Xilinx, they are written in VHDL or Verilog language and are available together with test-benches and complete code documentation.

Device	**System Gates**	**Asic Gate**	**Flip-Flops**	**Embedded RAM/Fifo blocks**		**MAX I/O Pads**
				Blocks	Max Size (Kbits)	
RTAX250S/SL	250k	30k	2816	12	54	198
RTAX1000S/SL	1M	125k	12096	36	162	418
RTAX2000S/SL	2M	250k	21504	64	288	684
RTAX4000S/SL	4M	500k	40320	120	540	840

table 3 ACTEL’s RTAX characteristics

Device	**System Gates**	**Slices**	**Max Distributed RAM Kbits**	**SelectRAM blocks**		**MAX I/O Pads**
				18Kbit Blocks	Max RAM (Kbits)	
XQR2V1000	1M	5120	160	40	720	432
XQR2V3000	3M	14336	448	96	1728	720
XQR2V6000	6M	33792	1056	144	2592	1104

table 4 Xilinx’s QPRO Virtex-II characteristics

For example for Xilinx the "Opb_usb2_device " core developed by Xilinx itself and available as part of the Software tool EDK is available : it is a USB 2.0 High Speed Device with On-chip Peripheral Bus (OPB, @ 32 bits) for the connection with the uProcessor/controller, it has the following characteristics:

- Compliant with the USB 2.0 Specification.
- Supports High Speed, Full Speed and Low Speed.
- Has a 32-bit OPB Slave Interface.
- Has a ULPI interface to external USB PHY (in out project ISP1505 from NXP)
- Supports positive and negative ULPI clocks.
- Has eight endpoints, including one control endpoint 0. Endpoints 1 - 7 may be bulk, interrupt,
- or isochronous. Endpoints are individuallyconfigurable.
- Uses Block RAM for endpoint buffers.
- Each endpoint has two ping-pong buffers.
- FPGA occupancy : Slices = 1235, Block RAMs = 4 (four)

As indicated by the presented FPGA occupancy datum, a lot of spare cells are left for the implementation of other building blocks (uProcessor ,…) or glue logic.

For an ASIC implementation ATMEL (www.atmel.com , France) is considered the preferred solution substantially for two reasons:

- It's one of the wordwide leader in the semiconductor devices for the space market with ASIC,FPGA and specific IC products (Leon2, Spacewire Controller, CAN Bus Controller, …)
- It's an European firm so no ITAR restrictions or limitations are present on their products.

It has to be underlined than an ASIC solution is worth from a economic point of view only in the case a production of more than 20-25 pieces will take place, in any case independently from the number of pieces that are conceived to be developed a preliminary breadboarding activity using a FPGA that implements the USB core should start.

Atmel has started the activity of conversion of projects originally implemented in FPGA into ASIC in 1985 so it can declare more than 20 years' experience in this specialized activity area. The default target device for an FPGA conversion of the USB core is the AT40KEL040 that covers up to 40K ASIC gates equivalent projects, but bigger ASICs are also available.

8. Radiation Tests

From a radiation hardness point of view the only potential critical item of the Generic USB device architecture presented in Figure 4 is the ULPI Transceiver ISP1505 from NXP. The idea is to characterize from a radiation point of view (TID, SEE) this device. Contacts have been already established with the MAPRad s.r.l .(Perugia, Italy www.maprad.com) that is company working in the field of space and high-reliability electronics characterization and qualification. The company has a direct access to the following facilities:

- INFN-LNS (Catania, Italy) at superconducting Cyclotron for SEE test
- INFN-SIRAD (Legnaro, Italy) SEE test
- ENEA-Calliope (Rome, Italy) Co60 gamma ray source for total dose irradiation

Furthermore a preliminary test using a pulsed nanosecond IR laser system available at MAPRad premises could be performed in order to evaluate the detailed mapping of the device sensitivity and SEE error rate versus particle energy see Ref 3.

9. Demonstration Platform and On Going activities

A study phase on the USB core to be implemented into a FPGA has already started: the development and the suitability of a VHDL design to be implemented inside a FPGA available also in a ruggedized and radiation tolerant technology is the key point of the here presented R&D activity.

Two different boards are available in CGS premises hosting on board Virtex II FPGA from Xilinx:

ADM-XRC-II from Alpha Data Inc (www.alpha-data.com) that hosts a 6M of Gates Virtex II from Xilinx (XC2V6000) equipped with a flexible front and rear panel I/O interface options, able to provide a bandwidth up to 5GBytes/sec, PCI format

GR-PCI-XC2V from PENDER ELECTRONIC DESIGN GmbH equipped with a 3M of Gates Virtex II from Xilinx (XC2V3000).This board has been developed in cooperation with Gaisler Research (www.gaisler.com) especially to support the early development and fast prototyping of LEON systems, nevertheless the incorporation of on-board volatile and non-volatile memories, together with serial , ethernet and fast debugging interfaces makes this board ideal for implementing high speed designs.The board form factor is PCI but the board is capable of operating also in a stand-alone configuration.

Figure 5 GR-PCI-XC2V board from PENDER ELECTRONIC DESIGN GmbH

10. Conclusions

CGS has initiated an activity to develop a USB device interface to be used in space applications. The rationale behind the choice of the USB interface as cheap, powerful and theoretically plug and play fast digital interface is presented. A brief overview on the features of the USB communication system is summarized. A generic architecture for a USB Device is described showing its major building blocks. A connector type and a USB transceiver device are proposed as baseline. The selection process of FPGA types suitable for the implementation of the USB core is briefly described. The on going activities foresees the design of a USB core to be synthetizable in a Xilinx FPGA. The developed VHDL code shall be verified and tested using a commercial development breadboard.

11. References

Ref1 *Universal Serial Bus Specification Revision 2.0* April 27, 2000

Ref2 *Qualification of COTS computer systems for use on the International Space Station* A.Winkler, U.Bruns, L.Gunther, I.Gerhard, Proceedings of the DASIA 2006 (ESA SP-630, July 2006)

Ref3 *The radiation sensitivity mapping of ICs using an IR pulsed laser system* B. Alpat and others Microelectronics Reliability 43 (2003) 981–984

Ref4 *USB 2.0 Host Controller IP Core* , user's manual Jan Andersson, Jonas Ekergarn, Gaisler Research May 2007 www.gaisler.com