sRIO SERIAL BUFFER FLOW-CONTROL DEVICE

Product Brief
80KSBR201

Device Overview

The IDT80KSBR201 is a high speed Serial Buffer (SerB) that can connect up to two high-speed Serial RapidIO interfaces. This device is built to work with any sRIO device and especially with the IDT Pre-Processing Switch (PPS), IDT70K2000. The SerB performs buffering and off-loading of data as well as buffer-delay of data samples in various applications. This device can act as either a slave, waiting for other devices to read from it, or as a master in which the SerB writes data to a programmed location once some criteria have been meet. This combination of storage and flexibility make it the perfect buffering solution for sRIO systems.

Features

- **Two Independent Serial RapidIO Ports**
- **Partial Bridging Functions**
  - *sRIO to sRIO*
  - *sRIO to Parallel*
  - *Parallel to sRIO*
- **Configurable Queues and Sizes**
- **Single/Dual Port Buffering**
- **Optional External QDR SRAM Available**
  - *Up to 288 Mbit external QDR SRAM*
  - *200 MHz; 18M, 36M, 72M, 144M or 288 M*
- **Seamless Integration of External and Internal Memory**
  - *Internal and external memory functions as a single buffer*
- **Provides Status Flags for Combined Internal/External Memories**
  - *Full, Empty, Partially Empty, Partially Full*
- **Direct or polled operation of flag status bus**
- **Optional Water mark**
  - *Serial Buffer can Either Send a Flag or Transmit Data at a Specific Packet Count or Byte Count*
- **Interface - Serial Rapid IO (sRIO)**
  - *One four-bit (x4) link, configurable to one-bit (x1) link*
  - *Port Speeds selectable: 3.125 Gbps, 2.5 Gbps, or 1.25 Gbps*
  - *Short haul or long haul reach for each PHY speed*
  - *Error management supports standard and enhanced port operations*
  - *sRIO version 1.3*
  - *Class 1+ End Point Device*
- **Interface - Parallel Port**
  - *Support for an optional external microprocessor or FPGA*
  - *Supports QDRII Burst of 2 Interface*
  - *Supports Packet or Raw-data format*
- **Interface - I^2C Interface Port**
  - *One I^2C port for maintenance and error reporting*
- **Interface - JTAG**
  - *JTAG Functionality for boundary scan and programming*
- **10 Gbps Throughput**
- **High-Speed CMOS Technology**
  - *1.2V Core operation with 3.3/2.5V JTAG interface*
- **Package: 484-pin Plastic Ball Grid Array**
  - *23mm x 23mm, 1.0mm ball pitch*

Block Diagram

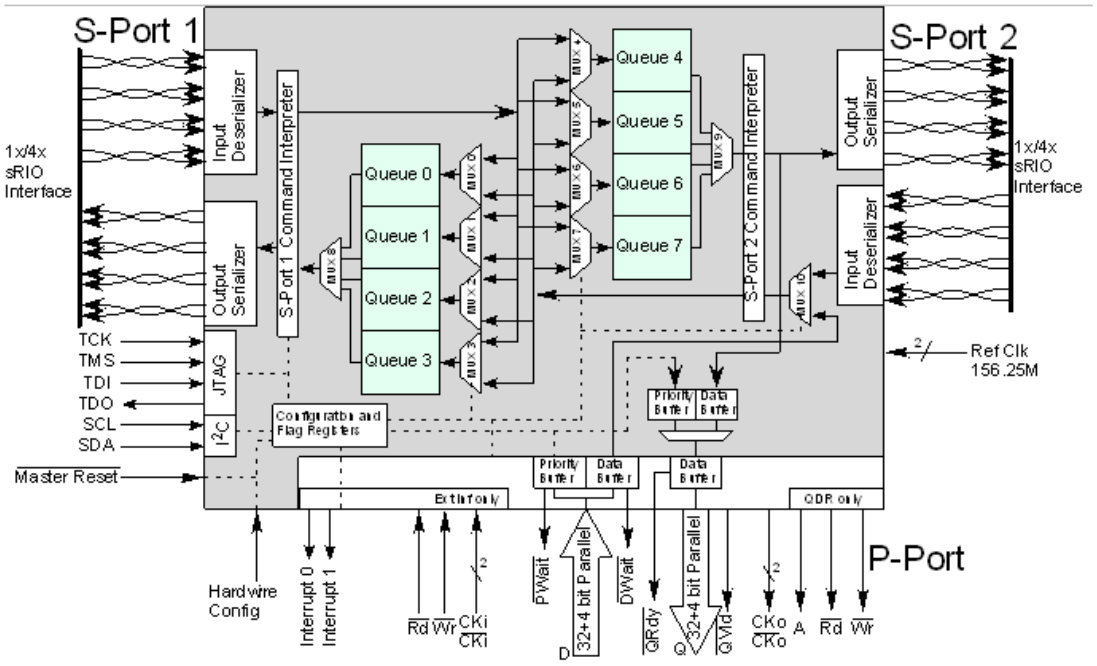


Figure 1. SerB Block Diagram

Notes

1.0 Functional Description

The IDT80KSBR201 is a Serial RapidIO™ sequential buffer (SerB) flow-control device consisting of up to 18Mbits of on-chip memory with expansion of one QDRII SRAM externally bringing the total buffering capacity to 90Mbits of storage. This device is built to work with any sRIO device and especially with the IDT Pre-Processing Switch (PPS) number IDT70K2000.

In this configuration where multiple DSPs are used with the PPS, the SerB can function as an over-flow port to handle traffic that is on any given port or, as a delay buffer to store data and present it at a later time. This is important in DPS applications where time samples are compared with the previous sample such as Cellular Base Stations. Please refer to the application note “Serial Buffer and Pre-Processing Switch”.

The 80KSBR201 fully complies to sRIO specification version 1.3 and is implemented to a class 1+ end-point device.

This device can operate as a master or a slave. In the sRIO environment, a master is defined as a device that originates data transfers, either to or from that device. A slave is one that responds to commands from other devices to move data. As a master, the SerB can receive data and at a pre-programmed water level (either number of packets or bytes) the device will form and transmit either packets or status (e.g., doorbells) to a programmed location. As a slave, the device will produce the data requested by other devices.

For applications requiring larger buffers, an additional 72Mbits of QDR SRAM can be attached via the Parallel Port. The two memories are seamlessly connected by the Serial Buffer to form a large, 90 Mbit buffer memory. The QDR SRAM interface runs at speeds of only 155MHz allowing lower cost memories to be used as well as easier board layout. Data rates still support up to 10Gbits/s (OC-192) throughput in the device to maintain full sRIO four-lane compliance.

The device provides Full flag and Empty flag status for the queue selected for write and read operations respectively, and a Programmable Almost Full and Almost Empty flag for the queue is also provided.

The device is configured into a single queue comprising the full internal memory and potentially the external memory if attached. The device treats the full amount of memory, internal or a combination of internal and external, as a single memory block. Status flags from that queue, either referring to the writes (full flags) or the reads (empty flags) to or from that queue represent the total amount of memory. Flags can be read from the serial port or from the I^2C or JTAG port. Proactive flags can be configured to send a doorbell and/or change the interrupt pin once a flag is set. Partial full and empty flags can be programmed to provide reaction time for writes and reads respectively. Flags associated with reaching water marks are available in addition to the full and empty flags.

The SerB is capable of translating between the selected protocols when more than one port is active.

A JTAG test port is provided running at 3.3V, here the multi-queue flow-control device has a fully functional Boundary Scan feature, compliant with IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture. The SerB can also be programmed via the JTAG port. There is also an I^2C processor port for programming and retrieving information from the configuration registers.

In all applications, the SerB is a low pin count device, compared with equivalent FIFO storage devices that utilize parallel interfaces at an equivalent total bandwidth. The high-speed serial interfaces allow reduced pin count over parallel interfaces.

2.0 Applications

2.1 Pre-Processing Switch Data Storage

The SerB’s primary application is for a Basestation using the IDT’s Pre-Processing Switch (PPS). The SerB will be a storage device, holding large amounts of data passed to it by the PPS. In this application, the S-Port 1 on the SerB will connect to one of the 4x ports of the PPS. The PPS will pass approximately 10ms of data to the SerB at which time the SerB will start to pass it back to the PPS as a multicast. It is expected that the data flow will remain constant with 10ms (or other designated quantity) worth of data always in storage.

The Basestation uses the data for sample scattering (noise reduction) and alignment of control and data packets.

Notes

The following are items of note concerning the PPS application:

1. The SerB has the ability to act as a simple master.
   - The SerB's primary application with the PPS will be to broadcast data. It must be a master to perform a broadcast, even if the data is requested.
   - The SerB has the ability to initiate writes. Mainly to prevent overflow and to perform broadcasts when waterlevel is reached (timed event). This avoids requiring the DSP to increase congestion by requesting data and controlling the SerB.
2. The SerB will typically perform SWRITEs.
   - The target address(s) generated by the SerB is programmable.
   - The packets are stored in the format they come in and are broadcast with simple changes to the headers
3. The DSPs have the ability to read the SerB through the PPS.
   - The DSP may send a doorbell to the SerB requesting a packet in which SerB will respond with a single packet.
   - The DSP may read the same data from the SerB more than once for multiple antennas. To accomplish this, the same data must have been loaded to more than one queue on S-Port 1. Each queue may read the same data only once, but as many as four queues may be used to hold the same data.

2.1.1 PPS Configuration 1

In the first configuration for the PPS, the SerB shall connect to the PPS on S-Port 1. P-Port shall connect to one of the designated QDR SRAMs. S-Port 2 is disabled. All queues except queue 0 is disabled. Queue 0 is configured to both receive and transmit data on S-Port 1. The PPS shall feed data into the SerB where it will be stored. Upon reaching the appropriate waterlevel, designated by the programmed partial full flag, the SerB shall send a multicast packet back to the PPS, and wait until the waterlevel is again exceeded to send another packet.

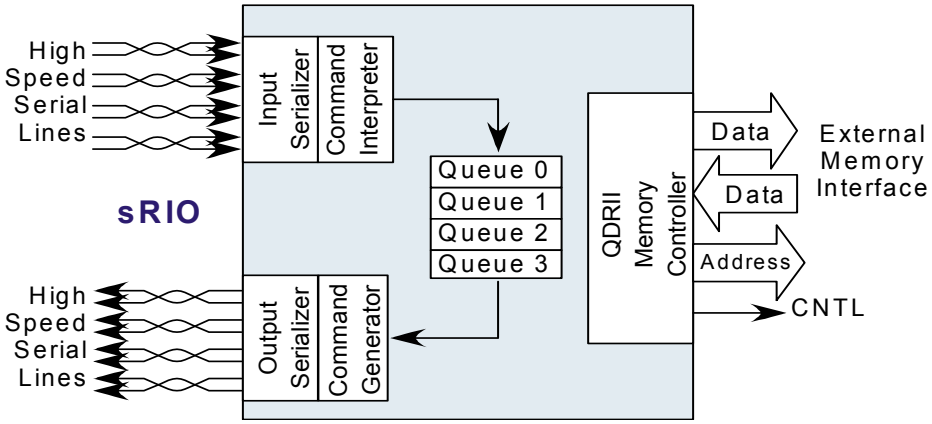


Figure 3. PPS Offload Application

2.1.2 PPS Configuration 2

In some PPS applications, it may be desirable to connect an FPGA to S-Port 2 on the SerB, allowing the FPGA to communicate to the PPS and the rest of the system beyond the PPS. Figure 4 is an example of this application. As shown, Queue 5 is connected between S-Port 1 and S-Port 2. Queue 1 is connected to S-Port 1 on both input and output. Queue 2 is connected with S-Port 2 as input and S-Port 1 as output. Queue 4 is connected to S-Port 2 on both input and output. All other queues are disabled. In this configuration, the PPS or other sRIO port device may write and read data on S-Port 1. The data may be stored for later retrieval on S-Port 1, sent to S-Port 2, or sent to both places. The FPGA or other Lite protocol device on S-Port 2 would have the same ability to send data in on S-Port 2, sending it on to S-Port 1 or holding it for retrieval on S-Port 2, or both.

Notes

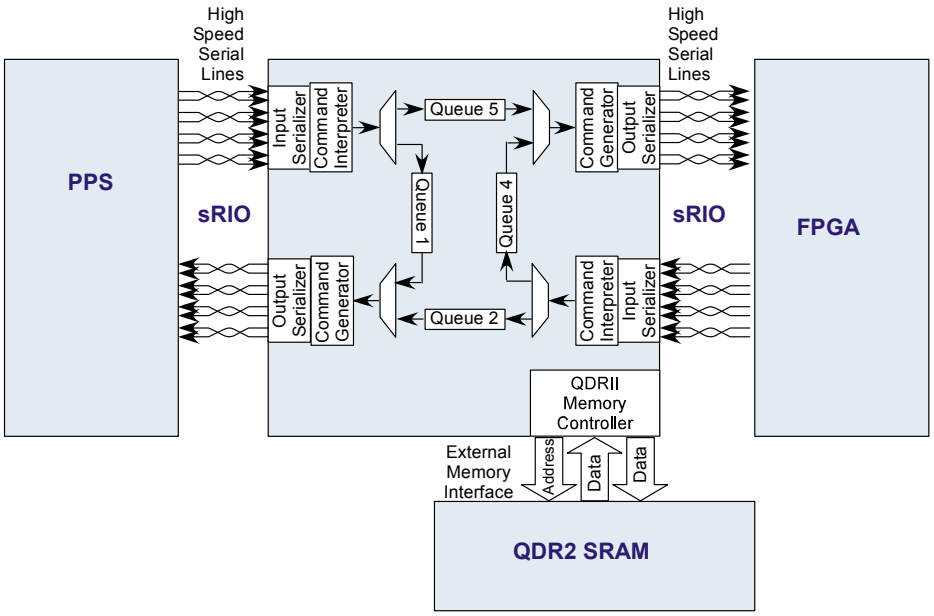


Figure 4. PPS Combined with an FPGA (Distributed)

2.2 FPGA Offload Device

In this application, the SerB will connect directly to an FPGA and act as a FIFO. This application may or may not use additional external memory. Since most FPGAs will avoid unnecessary intelligence, the SerB must be able to accept simple commands with little overhead. Figure 5 shows a diagram of this application.

2.3 Buffer between Two Busses

All of the following applications involve the SerB sitting between two busses. In this configuration, data may be passed directly from one port to the other, or data that enters one port may be fed back to the same port.

For each of these configurations, the SerB may be programmed to be active or passive device. As an active device, whenever data is received by the SerB, the SerB shall attempt to send the data to its final destination as soon as a complete packet has been received. As a passive device, the SerB shall receive data and respond to any requests on any port, but all received data shall be stored until transmission is requested by the destination port.

2.3.1 FPGA to FPGA Buffer

In this application, the SerB will sit between two FPGAs and offer buffering capability between the two devices. This application may or may not use the external memory. The SerB may act as either the initiator or receiver on both ports.

Notes

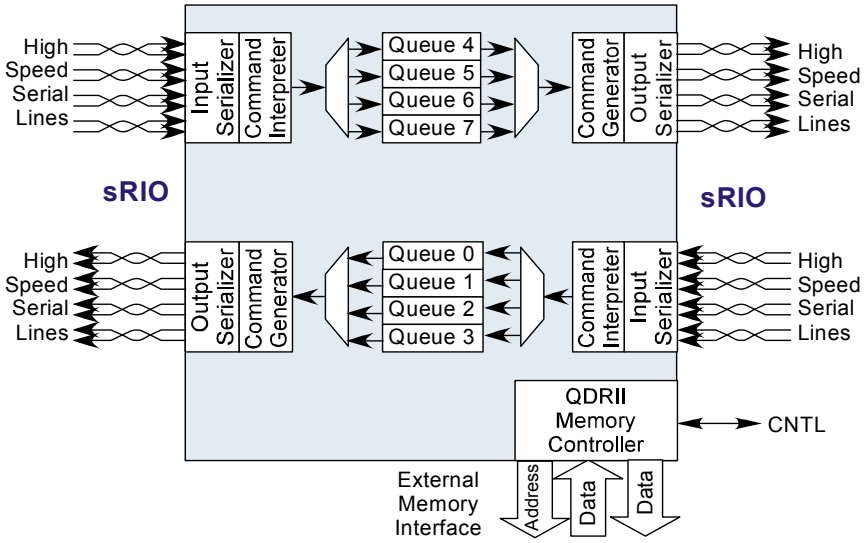


Figure 5. SerB as a FIFO between two FPGAs

In Figure 5 above, the SerB acts as a simple FIFO buffer between the two devices with no ability for FPGA to offload data for its own use. Figure 6 below is similar, except each FPGA has a path within the SerB back to itself allowing the offloading of data.

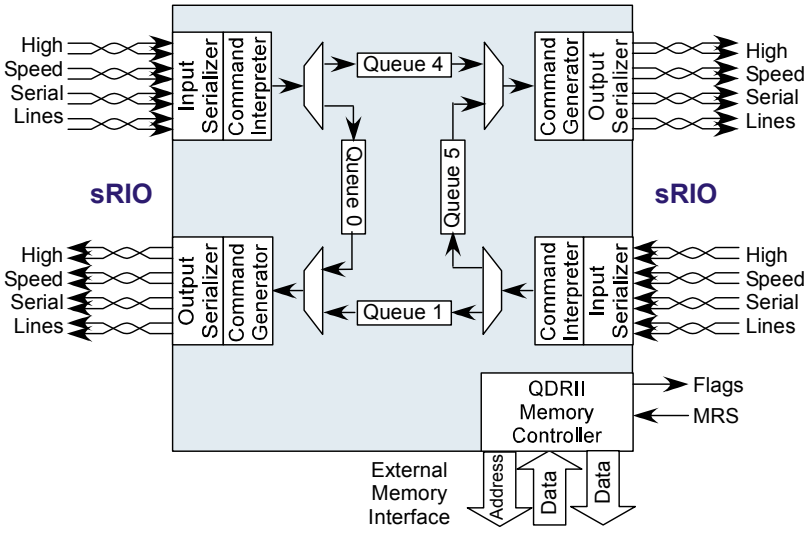


Figure 6. SerB as a combined FIFO and Offload Device for an FPGA

2.3.2 sRIO to sRIO Buffer

The simplest form of translation is where both ports regard the SerB as an end-point memory. Each port may write data to the SerB or read data from the SerB in the active protocol. No command translation is performed or required. Each port will handle all link negotiations, responses, and transfers needed for the active protocol. Packet ready flags may be used to indicate the presence of data within the buffer.

This is similar to the sRIO translation configuration except the SerB isolates the two ports. The device on S-Port 1 is not aware of the device on S-Port 2 (or P-Port).

Notes

2.4 sRIO Translation

The translation capabilities of the SerB are primarily through the ability to read and write memory in both protocols. When going from sRIO to a lite protocol, the sRIO port will pass data into memory, which then may be passed out the lite port. In the reverse direction, the lite port passes data into the SerB. An sRIO command can then be formed to issue a command with the data. The commands are limited to ones that can be developed by the "case scenarios" within the SerB.

An example of translation is shown in Figure 7. The sRIO device may initiate a command. That command will be picked up by the SerB, interpreted by the SerB, and passed directly to the FPGA. Meanwhile, the SerB shall return a receipt to the sRIO device indicating the packet was received. The FPGA may respond with a symbol, which will be received by the SerB, acknowledging receipt. The SerB would connect directly to the sRIO port as an end-point, or as a bus master. All link negotiations, retransmission requests, and status requests shall be maintained at the sRIO interface without requiring intervention by S-Port 2.

The FPGA could also be the initiator, sending a packet to the SerB, which interprets the command and forwards the appropriate command/data to the sRIO Device. The response packets will be handled locally at the interfaces, along with any retransmissions needed. The FPGA only needs to understand the codes for the commands it will be passing.

2.4.1 Parallel to Serial Translation

The SerB has the ability to translate from sRIO or the lite protocol to a parallel port. This allows the device to be connected to an inexpensive FPGA and providing a serial port access or PPS access.

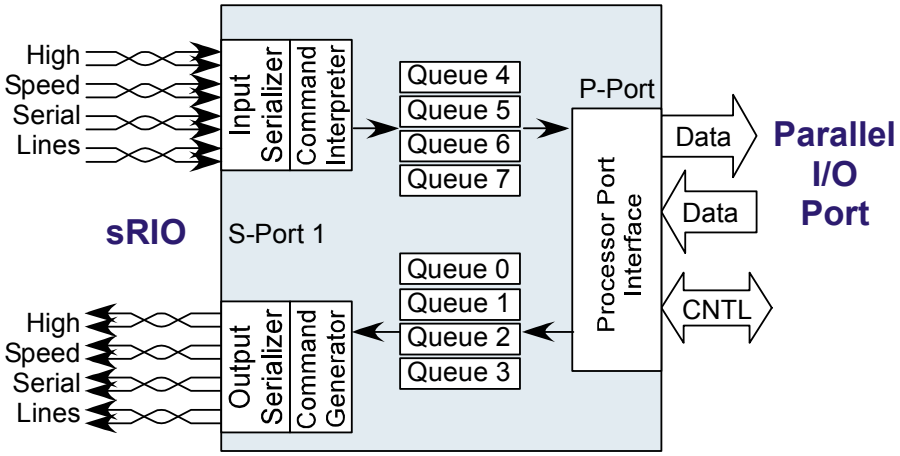


Figure 7. Parallel to Serial Translator

In this application, the P-Port activities are identical to the S-Port 2 activities of the previously discussed translations. The address bits on P-Port is used to select sRIO commands and destination IDs, along with the queue selections.

In addition, the SerB may be used as a passive memory between S-Port 1 and P-Port, the same as described earlier between S-Port 1 and S-Port 2.

3.0 Revision History

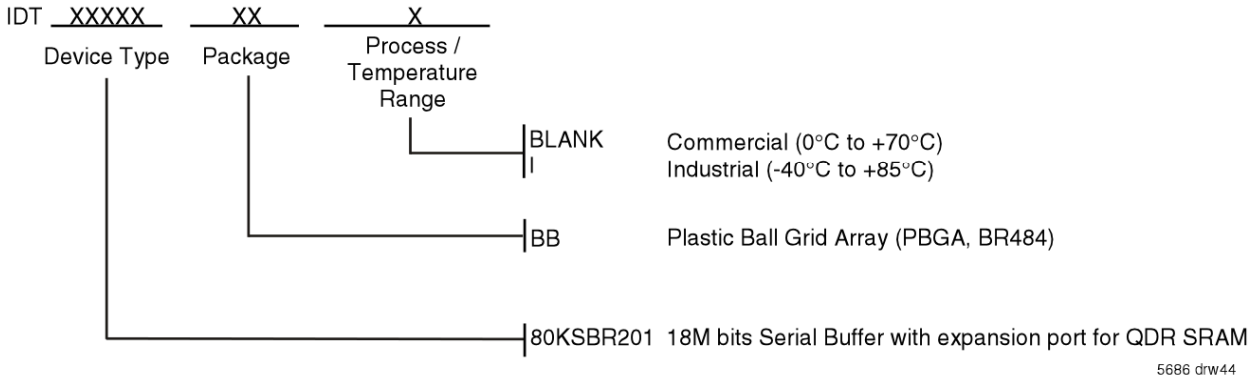
11/26/2007: Product Brief (Rev A)

4.0 Product Brief Datasheet: (Definition)

"Product Brief datasheets are informational only" and are subject to change without notice.

5.0 Ordering Information

For specific speeds, packages and powers, contact your sales office



CORPORATE HEADQUARTERS
6024 Silver Creek Valley Road
San Jose, CA 95138

for SALES:
800-345-7015 or 408-284-8200
fax: 408-284-2775
www.idt.com

for Tech Support:
408-360-1533
email: Flow-Controlhelp@idt.com