

TMS320C6000™ DSP Multiprocessing With a Crossbar Switch

Wireless ASP Products

ABSTRACT

Current generations of telecommunications infrastructure require intensive real-time computation. To meet the demands of next generation equipment, such as 3G wireless base stations, designers must seek methods to improve hardware and software efficiency.

These real-time systems are typically distributed systems consisting of one master processor and one or more slave processors. Software flexibility and architecture features enable the TMS320C6000™ digital signal processor (DSP) family from Texas Instruments to match most requirements of both the master and slave subsystems. The TMS320C6000 DSP is used here to illustrate the requirements and implementation of a multiprocessor system based on cross bar switching.

A wireless base station implementation requires an architecture with multiprocessor connections. This application report focuses on the data-passing technique provided by a bidirectional first-in first-out (BiFIFO) interface to a crossbar switch designed and implemented by Blue Wave Systems on a ComStruct™ platform.

Contents

1	Introduction	2
1.1	A Wireless Base Station Architecture	2
1.2	Basic Crossbar Architecture	5
1.3	Example of 2G/3G CDMA Base Station Using Crossbar Architecture	6
2	A Description of a Crossbar Implementation	7
2.1	System Characteristics	7
2.2	A Description of a Crossbar Data Transfer	7
2.2.1	Data Transfer Between DSP and BiFIFO	8
2.2.2	Packet Routing	8
2.2.3	Simple Packet Format	9
2.2.4	Extended Packet Format	9
3	Additional Considerations	10
3.1	Alternative Data Exchange Architectures	11
3.1.1	Master-Slave Type Architectures	11
3.1.2	Peer-to-Peer Type Architectures	11
4	Acknowledgement	11
5	References	12
6	Trademarks	12

List of Figures

1	2G/3G CDMA Base Station Architecture	3
2	UMTS Uplink Processing at the BTS	4
3	Simplified Example of Crossbar Architecture	5
4	Transmit Data Path (dashed) Through Distributed System	6
5	Receive Data Path (dashed) Through Distributed System	6
6	Principal Sequence of Data Transfer	7
7	Bit Alignment in the BiFIFO	8

List of Tables

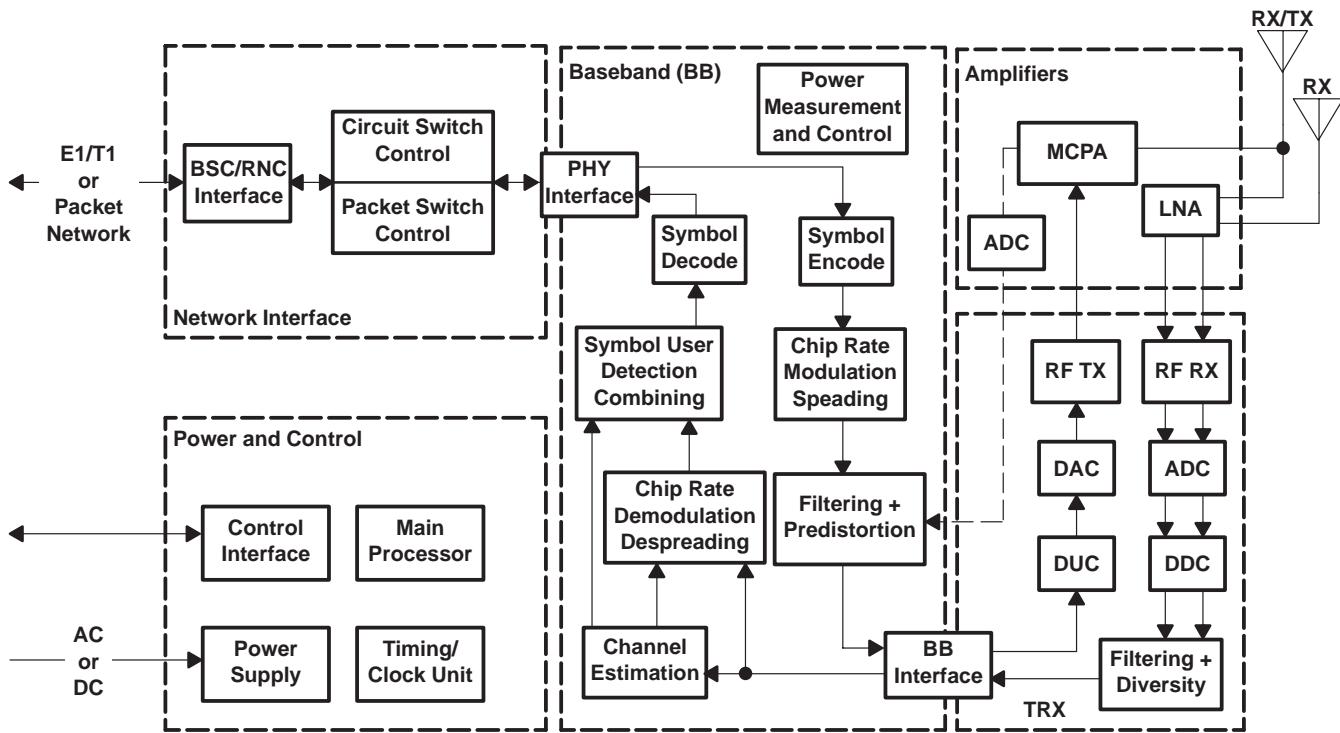
1	Bit Alignment for Simple Package Transfers	9
2	Bit Allocation Example for Routing Bits	9
3	Header Structure for Extended Packet Transmission	10
4	Extended Packet Header	10

1 Introduction

With the advent of applications such as third generation (3G) wireless base stations that require intensive real-time computation, distributed processing systems consisting of several interconnected DSPs are increasingly in demand. These systems combine the advantage of high computational power with the inherent flexibility of DSP-based systems. The number of 3G air interfaces that must be supported, and the variety of channel coding and modulation schemes defined for each standard make flexibility highly desirable. Wideband CDMA (W-CDMA), multicarrier CDMA (cdma2000), time division duplex CDMA (TD-CDMA), and TDMA frequency division multiple access (TDMA/FDMA) methods are examples of such air interfaces that are defined under the IMT2000 family of systems.

1.1 A Wireless Base Station Architecture

The computational and data flow characteristics of the 3G baseband processing unit for a universal mobile telecommunications system (UMTS) or W-CDMA base station receiver is shown in Figure 1.



RF = Radio frequency

RX = Receive

TX = Transmit

BSC = Base station controller

RNC = Radio network controller

MCPA = Multichannel power amplifier

ADC = Analog-to-digital converter

DAC = Digital-to-analog converter

LNA = Low noise amplifier

DUC = Digital upconverter

DDC = Digital downconverter

Figure 1. 2G/3G CDMA Base Station Architecture

The signal bandwidth between the baseband block and the TRX is approximately 4 MHz, depending on the chip rate selected: 3.68 Mchips/s (cdma2000) or 3.84 Mchips/s (W-CDMA). Within this spread signal there can be multiple user channels, the number of which is dependent on the sum of the individual bit rates of each channel. Various algorithms are then used to process each channel according to its coding and modulation characteristics. Figure 2 is an example of the algorithm flow for the receive path. The algorithm processing loads are not discussed here; however, this information is available from Texas Instruments.

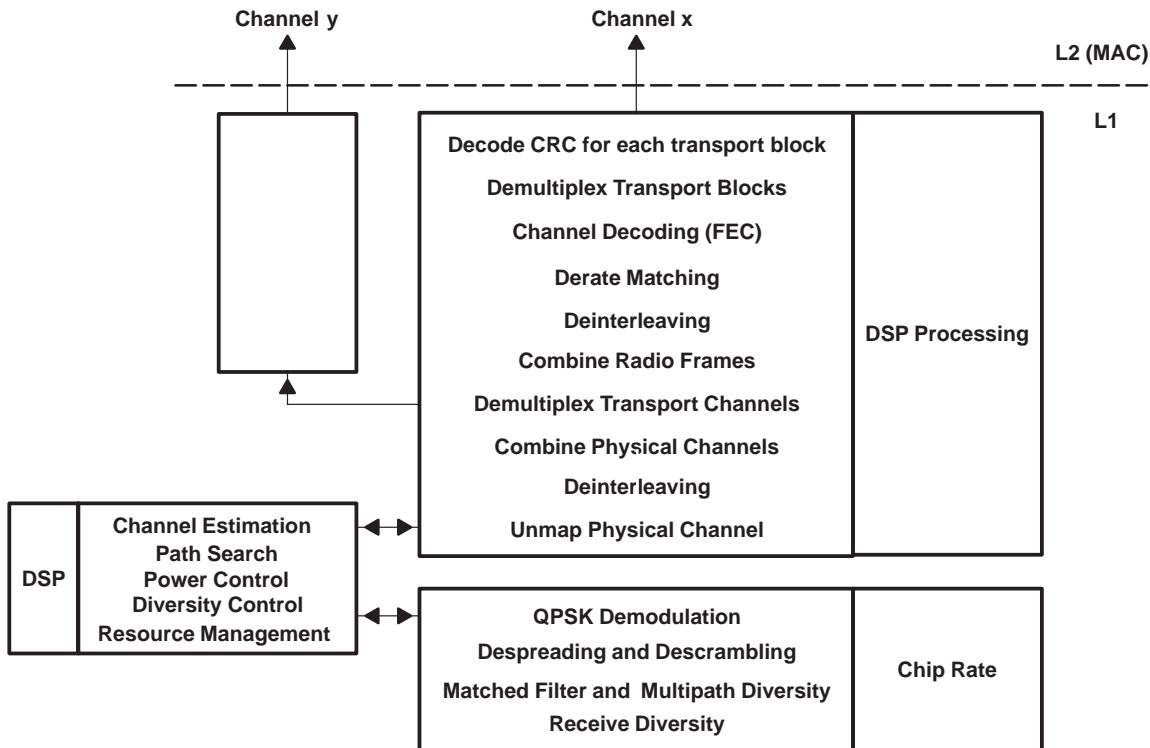


Figure 2. UMTS Uplink Processing at the BTS

All of these tasks place high demands on the computational power of the system. As a result, an efficient data transfer method is required to prevent data transfer overhead drain on CPU resources.

Data transfer rates on the periphery of the baseband block are high. For example, a W-CDMA system with a chip rate of 3.84 Mchips/s has a data rate of about 64 Mbytes/s at the baseband interface. This assumes 16-bit in-phase/quadrature (I/Q) data with 4x over-sampling. If we consider the uplink, this 64 Mbytes/s rate can consist of 32 to 64 voice channels destined for a single baseband processing block. After despreading, the incoming data rate to the DSP subsystem can still be approximately several Mbytes/s, depending on the data formats use.

Within the DSP processing block, there are other data paths that must be handled. These paths consist of DSP-to-DSP data passing, peripheral supply/demand, and control information.

This description shows that, you can see that the design of the baseband block must be optimized for the following characteristics:

- Multichannel to handle multiple users
- Multiprocessor to handle computation intensity
- High-bandwidth links to handle over-sampled processing and high-bit-rate channels

A common approach to meeting these requirements uses a distributed system that consists of several dedicated DSPs, each performing a set of specific tasks. Specific tasks are allocated so that deterministic DSP loading can be performed under dynamic channel-loading conditions. High bandwidth links and low processor transfer overhead are necessary for inter-processor/peripheral communication. A system design suitable to address these issues is presented in the next section.

1.2 Basic Crossbar Architecture

Using a crossbar architecture for inter-processor and processor-to-peripheral communication is one approach that addresses the baseband block requirements. This architecture typically consists of some I/O buffering, a crossbar, and a controller, as shown in Figure 3.

The I/O buffers are bidirectional FIFOs (BiFIFOs) that must fulfill the speed requirements imposed by the data rate for both rate matching and buffering. An example of such a device is the SN74ACT3632 from Texas Instruments. It features a 36-bit word size, depth of 512 words, and a maximum clock speed of 67 MHz.

The crossbar switch fabric connects the different ports and must be selected for minimum switching latency and lowest propagation delay.

A dedicated controller, also called a transfer agent (TA), performs all control functions when the TMS320C62x™ initiates the data transfer. The TA can either be a dedicated controller or another C62x™.

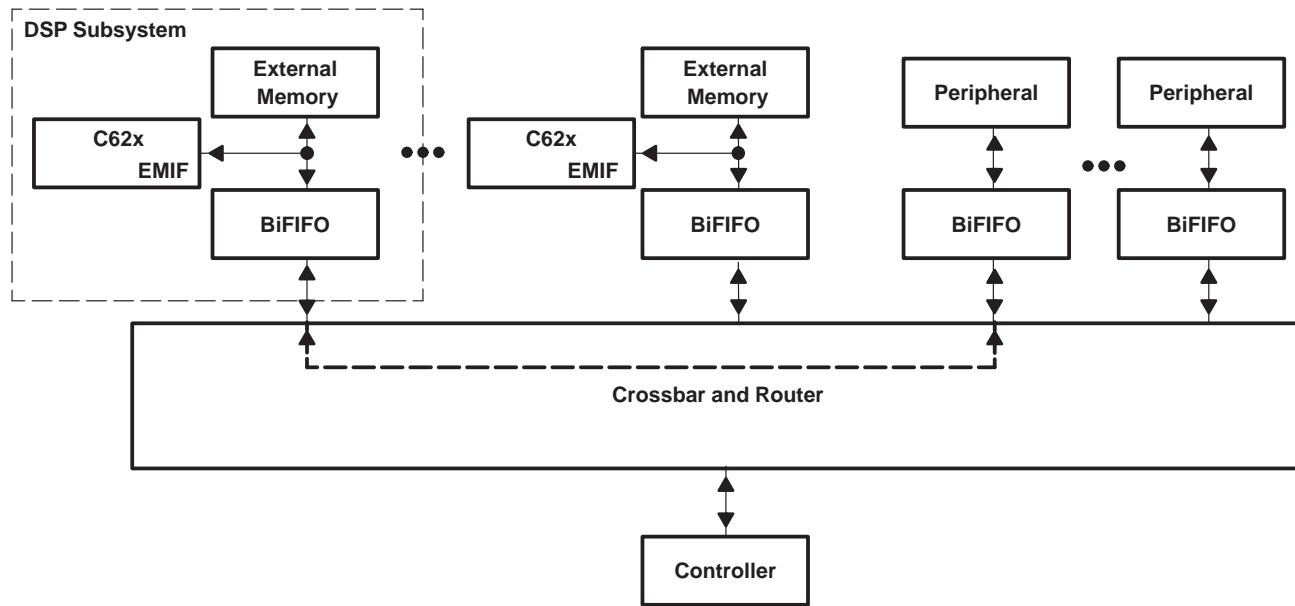


Figure 3. Simplified Example of Crossbar Architecture

This crossbar approach yields several advantages over shared memory schemes, most notably increased flexibility and low overhead for the C62x processors. In this approach, it is not necessary to reserve a designated memory area for data exchange, as is required with shared-memory schemes. When using the crossbar approach, the receiving processor can dynamically allocate free memory space for the data on demand, thus using memory more efficiently. The communication overhead for the processors is lower than in other schemes. Each DSP need only read/write the BiFIFOs while the TA handles all other switch control. Transfer overhead is absorbed by the DMA. DMA transfer time then depends on the FIFO speed and the C62x interface used.

1.3 Example of 2G/3G CDMA Base Station Using Crossbar Architecture

A system may be partitioned in a variety of ways. Figure 4 shows that the chip rate, symbol rate, encode, and decode functions are all performed on a single board. Other architectures may separate the encode and decode, or the chip rate and symbol rate; in either configuration the crossbar can be used as an interboard connection.

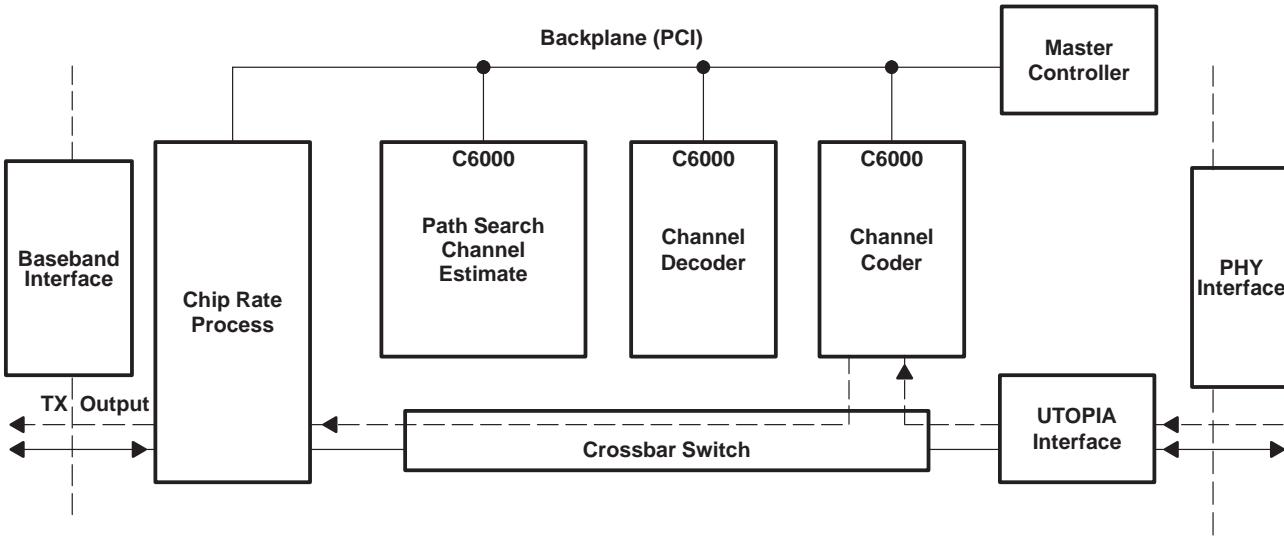


Figure 4. Transmit Data Path (dashed) Through Distributed System

The transmit path is shown in Figure 4. The transmit path flows from the network interface, through the channel coder, and out to the chip rate processor. The channel coder increases the bit rate by 3-4x, typically. So if the input over UTOPIA has an information rate of 32×8 kbps for 32 channels of speech, then the output is likely to be at least 1 Mbps. The information rate over UTOPIA seems relatively small, however the rate that the crossbar must deal with is likely to be around 400 Mbps, which is the physical burst rate.

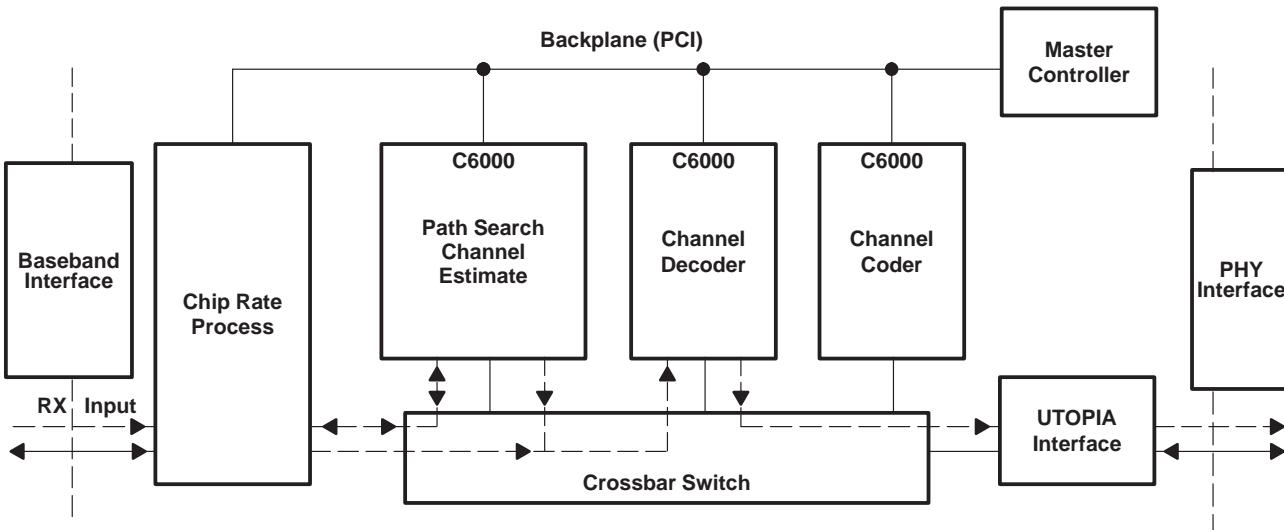


Figure 5. Receive Data Path (dashed) Through Distributed System

The receive data path is more complex than the transmit, resulting in higher demands for processing power and data exchange. Input is the baseband output of the RF stage. As discussed previously, the input rate can be 50–100 Mbytes/s. The path search and channel estimation tasks are closely linked to the chip rate processing block. The crossbar allows the chip rate peripheral to be shared between those tasks and the receive and transmit paths.

2 A Description of a Crossbar Implementation

2.1 System Characteristics

A Blue Wave Systems implementation of a suitable crossbar switch illustrates the type of data transfer methods possible, as shown in Figure 3. The interconnections can be routed between four C62x DSPs and five additional peripherals (such as VersaModule Eurocard—VME, peripheral component interconnect—PCI, universal test and operations interface for ATM (UTOPIA), and chip rate processor).

With this solution, up to four concurrent transfers of up to 200 Mbytes/s each (32-bit @ 50 MHz) can be achieved with low switching overhead (< 1 μ s). Board-to-board transfers are also possible. The transmit overhead for the DSP consists of formatting the packet, adding the header, and moving it via the DMA to the BiFIFO. The subsequent tasks are handled by the TA, which supervises the buffers for any pending data and dynamically allocates a data path if necessary. In receive mode, the DSP must retrieve the packet from the BiFIFO using the same method.

2.2 A Description of a Crossbar Data Transfer

The crossbar switch, as shown in Figure 3, allows packets to be sent from any source to any destination, based on routing information. Additionally, data may be broadcast to single, designated multiple, and all available destinations.

Note that all transfers are packets. It is the responsibility of the sender—either in hardware or software—to generate a correctly formatted packet, even if the data is a continuous stream. At a minimum, a packet is a block of data with a single header routing word and, more significantly, an end-of-packet (EOP) marker. The EOP marker is significant because a connection between a source and destination is maintained until an EOP marker is detected.

Figure 6 shows the principal steps involved in the data transfer.

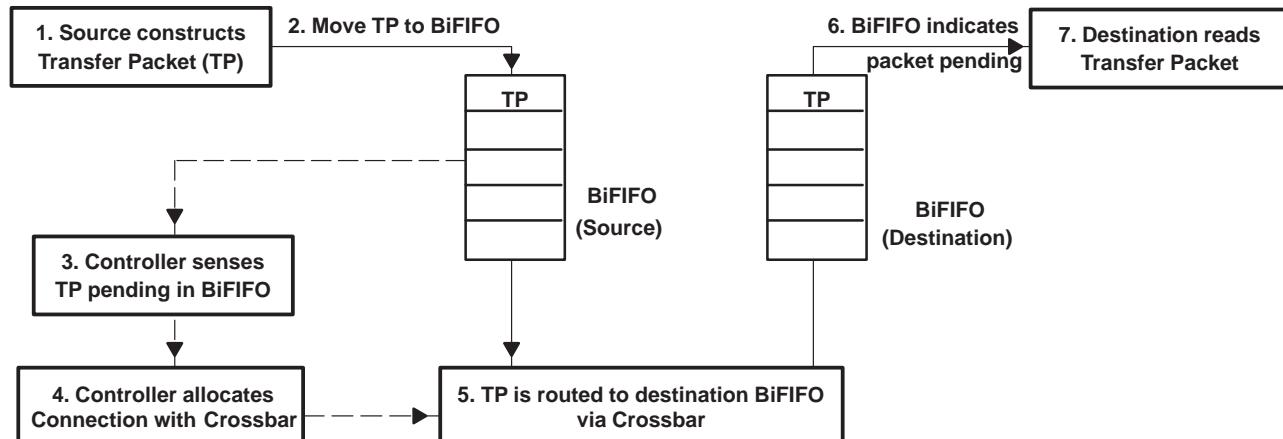


Figure 6. Principal Sequence of Data Transfer

2.2.1 Data Transfer Between DSP and BiFIFO

The main task of the DSP in the data transfer is to move the transmit data to and retrieve the received data from the BiFIFO. This is done using the DMA with either the external memory interface (EMIF) or expansion bus (XBUS). For this function, bidirectional synchronous FIFOs (BiFIFOs) must be employed. In addition to data inputs and outputs, a set of control flags is provided, including almost full and almost empty flags with programmable levels. The flags can be used in several ways to initiate data transfers depending on the system demands.

If high data throughput is desired, you can monitor the almost full flag or the full flag and burst transfer the whole content of the buffer. Alternatively, if minimum latency is a key issue, the empty flag can be monitored and data transferred whenever the buffer has data. Flag supervision can occur either by polling (using a redundant serial input line) or by connecting the flag output to an external interrupt line (interrupt driven) [1].

2.2.2 Packet Routing

As mentioned in Section 1.2, *Basic Crossbar Architecture*, the SN74ACT3632 BiFIFO has a data width of 36 bits. Because only 32 bits are needed for the actual data, you can easily define a transport protocol using the remaining 4 bits, as shown in Figure 7.

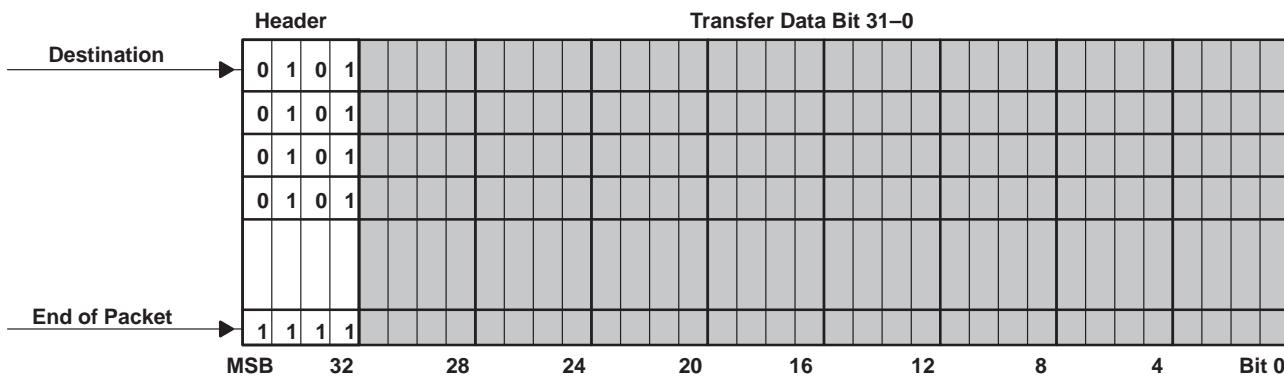


Figure 7. Bit Alignment in the BiFIFO

In this crossbar architecture, each crossbar port BiFIFO has four most significant bits (MSBs), which contain routing information hardwired to the controller. In simple protocol, this allows direct data transfer between port BiFIFOs without the need to read and extract routing information from the data packet. This works for a maximum of 15 addresses because the combination [1,1,1,1] is reserved as an EOP indication.

The latency required to establish a transfer sequence from packet pending (empty flag high) to first write is estimated at ~640 nanoseconds for a particular TA implementation. This type of data transfer is called a simple packet transfer. A simple packet allows transfers to a single destination or to all available on-board destinations.

Alternatively, an extended packet with a complex header structure allows designation of multiple destinations and also allows board-to-board transfers. For extended packet transfers, the TA must extract the 32-bit header from the packet and interpret the routing information when servicing the packet request. This increases the latency to establish a transfer to approximately 1.3 μ s. A complete description of the packet types and format is provided in Section 2.2.3, *Simple Packet Format*.

Transfer requests are placed in a circular queue. After each transfer setup has been completed, the oldest request in the queue with available resources is executed first. A round-robin arbitration is used for multiple packets, pending requests to the TA.

2.2.3 *Simple Packet Format*

Table 1 defines a simple packet.

Table 1. Bit Alignment for Simple Package Transfers

Control Bits (D35–D32)	Data Bits (D31–D00)
	Packet header
	[Extended header]
	[Extended header]
	Data word
	Data word
	Data word
:	:
<EOP> [1111]	Don't care
Extended header: optional based on header control bits	

The significance of this packet type is that the TA can route the packet without the need to read any data within the packet apart from the specific route code in the 4 MSBs. This code specifies the on-board destination, as shown in Table 2. The actual packet header within the first three words of the packet has no impact on the transfer agent. The packet can be of any length, subject to the recognition that the circuit is not broken until the EOP marker <EOP> is recognized. Thus, the specified resources are locked or unavailable until the packet transfer is complete. This emphasizes the necessity to correctly format the packet by applying the EOP marker.

Table 2. Bit Allocation Example for Routing Bits

Control Bits (D35–D32)	Stream Destination
0000	C62x 1
0001	C62x 2
0010	C62x 3
0011	C62x 4
0100	Peripheral 1
0100	Peripheral 2
0110	Peripheral 3
0111	Peripheral 4
1000	Peripheral 5
1010	All DSPs on-board (broadcast)

2.2.4 *Extended Packet Format*

The simple packet format does not support selective broadcasts or off-board transactions. A more complex packet construction exists in which the transfer agent must interpret bits within the header word to complete the routing of the packet. This format is distinguished from the simple packet format by the route bits [1100] in the first word of the packet and [0000] thereafter through the EOP marker, which is the same as for the simple packet. Table 3 shows the format of these extended packets.

Table 3. Header Structure for Extended Packet Transmission

Description	Control Bits (D35–D32)	Data Bits (D31–D0)
Extended packet	[1100]	Extended packet header
[Bit29 = 1] Destination address	[0000]	Valid address
[Bit29 = 1] Number of data words	[0000]	Length in words (N)
Data word 0	[0000]	Data
Data word 1	[0000]	Data
Data word 2	[0000]	Data
	:	
Data word N–1	[0000]	Data
Data word N	[0000]	Data
End of packet	[1111]	Don't care

Table 4 defines the extended packet header is defined from the TA point of view.

Table 4. Extended Packet Header

Bit Position	Bit Description	
31,30, 28–13	Reserved, protocol (board specific) defined	
29	Bit 29 = 0	Bit 29 = 1
12	Off-board packet	Address and length follow in the next 2 words.
11–9	Destination board ID	
8–0	9 bit fields. Each bit corresponds to a potential destination:	
Bit 8	Peripheral 5	
Bit 7	Peripheral 4 (TA)	
Bit 6	Peripheral 3 (Chip rate processor)	
Bit 5	Peripheral 2 (UTOPIA Interface)	
Bit 4	Peripheral 1 (PCI)	
Bit 3	C62x 4	
Bit 2	C62x 3	
Bit 1	C62x 2	
Bit 0	C62x 1	

By using bits 8–0 as Boolean type destination indicators, it is possible to send a message to each combination of these destinations on board. Setting bit 29 = 1 one can further enhance addressing abilities; for example, to enable other boards to use a whole data word for the address.

3 Additional Considerations

The crossbar architecture, as previously described, is well suited for distributed processing systems with sequential data passing from one processor to the next. This approach is optimized specifically for write transactions, such as a 2G/3G base station application. Other examples include all systems with dedicated processors that receive data, process, and pass to the next stage.

For applications geared to other processing methods, especially for distinct master-slave type systems with virtually no communication between the slave processors, an alternative architecture may be better suited.

3.1 Alternative Data Exchange Architectures

Several alternatives to the crossbar architecture exist. Suitability depends on the system hierarchy, data throughput, and interface options.

3.1.1 *Master-Slave Type Architectures*

Master-slave architecture typically involves one master processor that communicates with several slave processors. The master transfers data to one of the slave processors and receives the processed data when it is ready. In off-the-shelf solutions, the master processor often communicates via the PCI bus with a dedicated controller. The controller then translates the PCI commands to the host port interface (HPI) of the slave processor.

One example of such a controller is the PCI 2040 device from Texas Instruments [2]. This controller communicates with the master via the PCI bus and with up to four C62x (or C5000™) DSP processors using the HPIS.

Another controller example is the Tsi920 device from Tundra Semiconductor Corporation [3]. The master processor communicates with the Tsi920 via the PCI bus. The Tsi920 controller then uses the HPI interface to communicate with up to four C5000 family DSP devices, which can be single or multi-core. A key difference is the presence of FIFOs on-chip, which enable efficient packet transfer, as seen in the crossbar architecture.

The inherent advantage of master-slave architectures is the relatively low part count. The disadvantage is the limited peer-to-peer communication ability.

3.1.2 *Peer-to-Peer Type Architectures*

A common approach to peer-to-peer data exchange is the use of shared memory schemes. If only a small amount of data is being shared, you can dedicate a section of internal DSP memory, which is accessible via the HPI. This enables quick and efficient exchange of small amounts of data. If larger data transfers are necessary, the shared memory buffers must be moved externally. For both of these examples, bus arbitration and buffering are necessary when dealing with multiple point-to-point communications.

A combination of shared memory for the exchange of commands and the crossbar architecture for the actual data exchange is a commonly used architecture. It allows high-bandwidth low-overhead data transfer with peer-to-peer functionality and the ability to send requests to the other processors.

4 Acknowledgement

Blue Wave Systems Inc

2410 Luna Road

Carrollton, TX 75006

(972) 277-4600

FAX: (972) 277-4666

[http://ussales@bluews.com](mailto:ussales@bluews.com)

Loughborough Park

Ashby Road

Loughborough

Leicestershire, LE11 3NE

+44 (0) 1509 634444

+44 (0) 1509 634450

[http://uksales@bluews.com](mailto:uksales@bluews.com)

5 References

1. *TMS320C6000™ EMIF to External FIFO Interface Application Report (SPRA543)*
2. *PCI 2040 PCI1-DSP Bridge Controller Data Manual (SCPS048)*
3. *Tundra Semiconductor Corporation Homepage: <http://www.tundra.com>*

6 Trademarks

TMS320C6000, TMS320C62x, C62x, and C5000 are trademarks of Texas Instruments.

ComStruct is a trademark of Blue Wave Systems Inc.

Other trademarks are the property of their respective companies.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, license, warranty or endorsement thereof.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations and notices. Representation or reproduction of this information with alteration voids all warranties provided for an associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Resale of TI's products or services with *statements different from or beyond the parameters* stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Also see: [Standard Terms and Conditions of Sale for Semiconductor Products](http://www.ti.com/sc/docs/stdterms.htm). www.ti.com/sc/docs/stdterms.htm

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265